Mosaics

CSE 455, Winter 2010

February 8, 2010

Announcements

- The Midterm went out Friday
  - See email to the class
  - Also linked off the course colander
  - Open-note, Open-book, **Non Open-Friend**
  - **Clarification questions only** in office hours, over email, and on the forum
  - You may still ask us general questions about the course material

Announcements

- Professor office hours are M 2:30-3:30
- 6 TA hours during the week (WF 4-5:30 and TTh 10-11:30).

- We’ve enjoyed interactive with those that have come
- We would love to see more of you!

- We are also available for questions over email or to meet by appointment outside of these times.

- Late Days, the weekends don’t count

Announcements

- Class forum:
  https://catalysttools.washington.edu/gopost/board/iansimon/15087/
- Course reader:

  http://www.cs.washington.edu/education/courses/455/10wi/reader.htm
- Other resources:

  Richard Szeliski's book Computer Vision: Algorithms and Applications. The book draft is currently available online.

  R. Hartley, A.Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2000.

  R. Hartley, A.Zisserman., Multiple View Geometry - Tutorial. CVPR (1999).

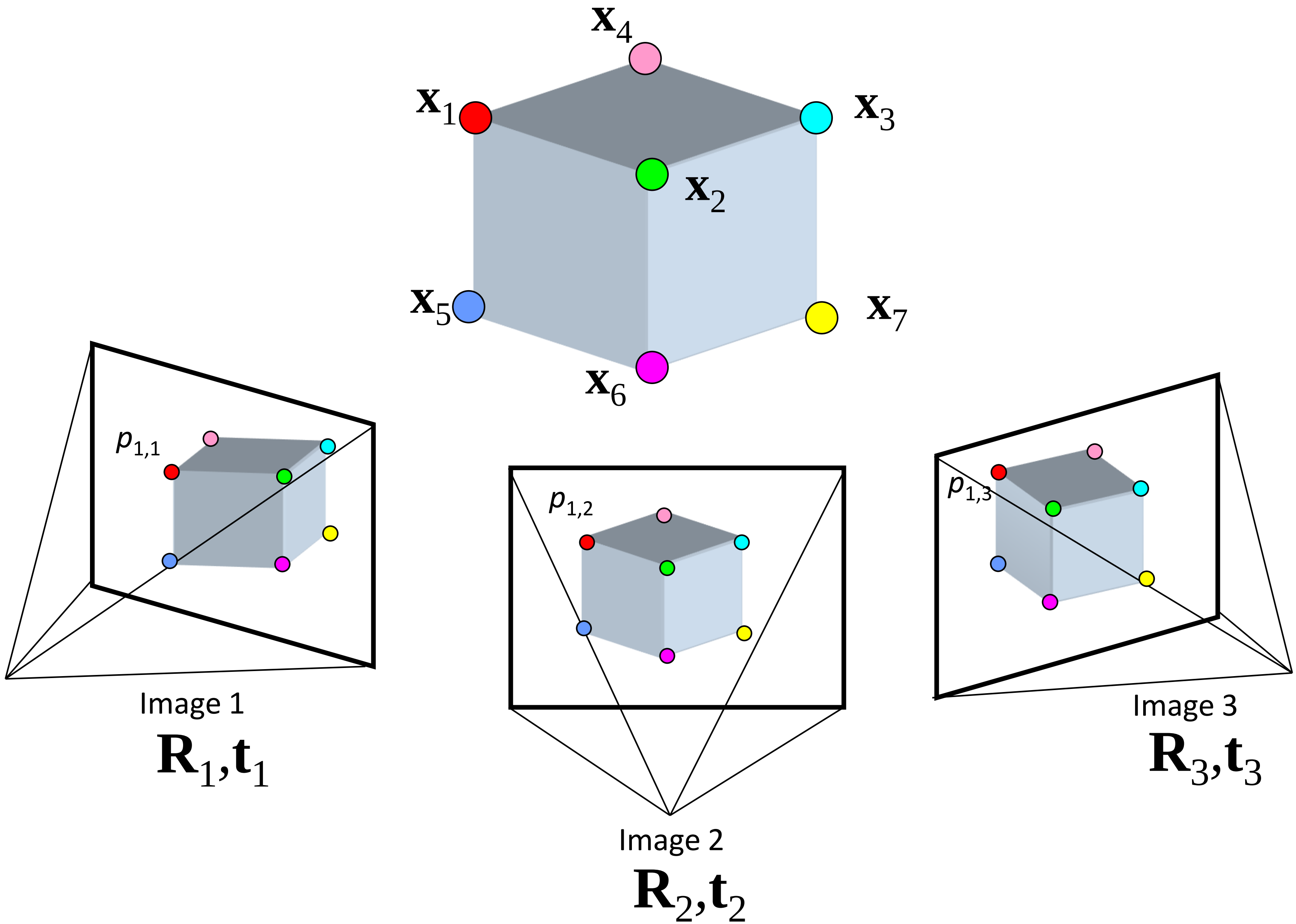
  D. A. Forsyth, J. Ponce. Computer Vision a Modern Approach. Prentice Hall, 2003.

Review From Last Time

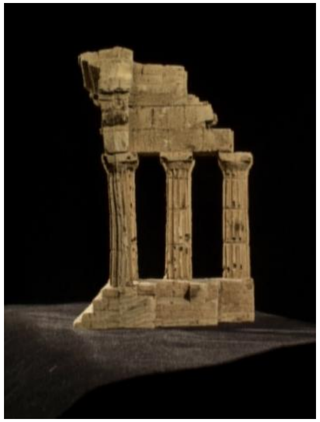
Structure from Motion

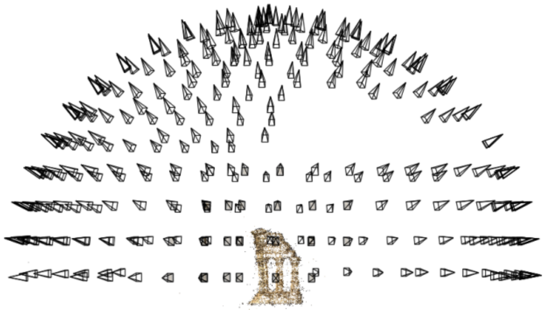
- Snavely, Seitz, Szeliski, **Photo Tourism: Exploring Photo Collections in 3D**. *SIGGRAPH 2006*.

  http://phototour.cs.washington.edu/Photo_Tourism.pdf

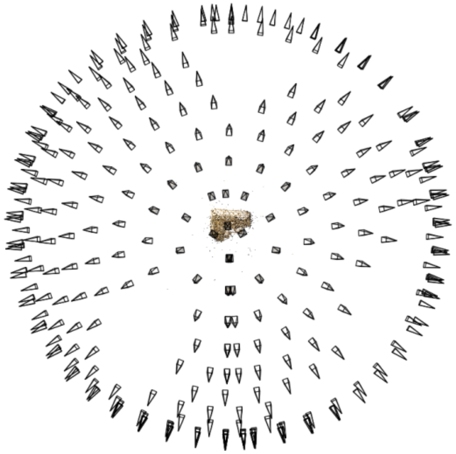

$\mathbf{x}_4$
$\mathbf{x}_1$
$\mathbf{x}_3$
$\mathbf{x}_2$
$\mathbf{x}_5$
$\mathbf{x}_7$
$\mathbf{x}_6$
$p_{1,1}$
$p_{1,2}$
$p_{1,3}$
Image 1
$\mathbf{R}_1,\mathbf{t}_1$
Image 2
$\mathbf{R}_2,\mathbf{t}_2$
Image 3
$\mathbf{R}_3,\mathbf{t}_3$

Structure from motion





Reconstruction (side)



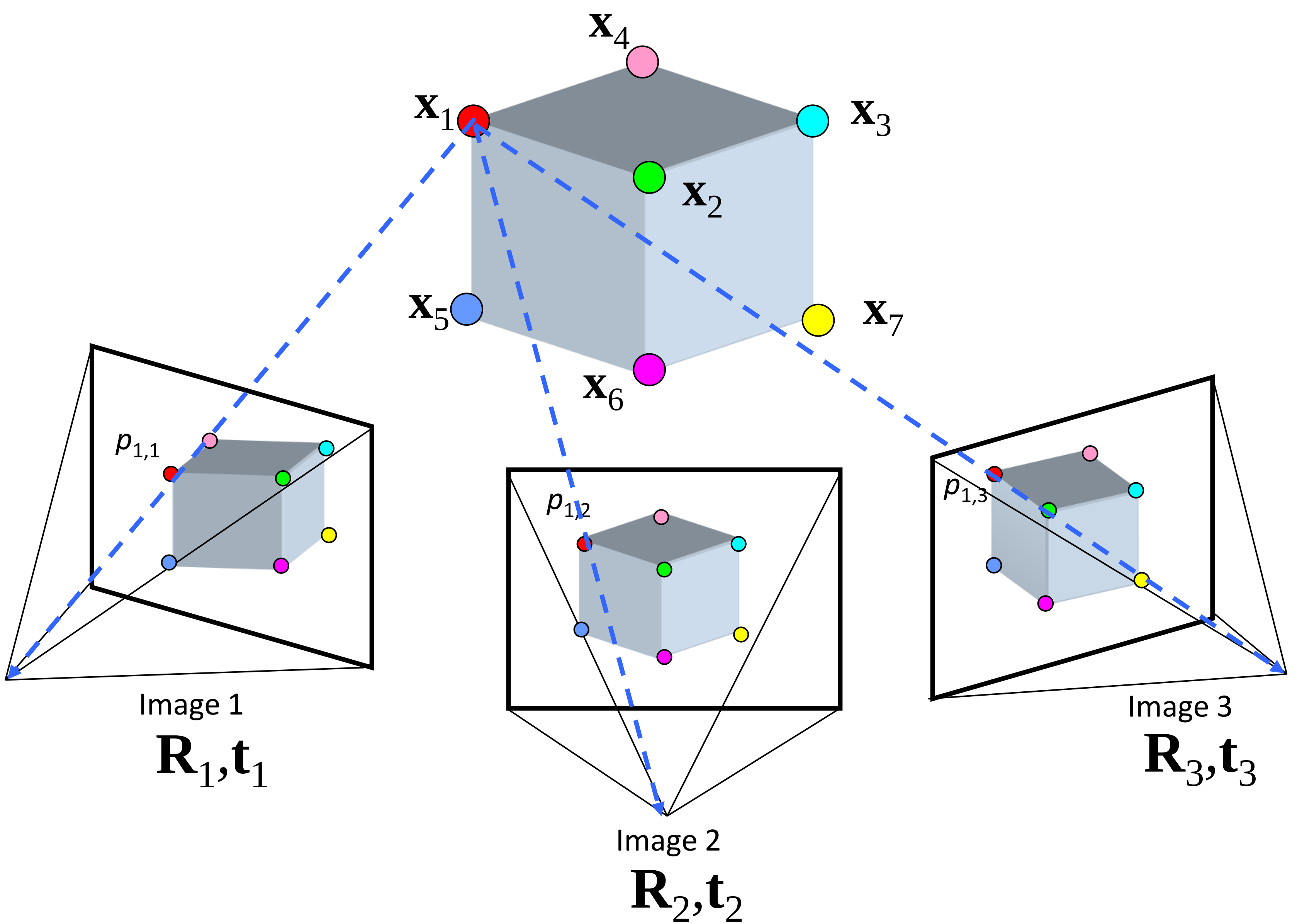
(top)

- Input: images with points in correspondence

  $p_{i,j} = (u_{i,j}, v_{i,j})$

- Output
  - structure: 3D location $\mathbf{x}_i$ for each point p_i
  - motion: camera parameters $\mathbf{R}_j$, $\mathbf{t}_j$

- Objective function: minimize *reprojection error*



$\mathbf{x}_4$
$\mathbf{x}_1$
$\mathbf{x}_3$
$\mathbf{x}_2$
$\mathbf{x}_5$
$\mathbf{x}_7$
$\mathbf{x}_6$
$p_{1,1}$
$p_{1,2}$
$p_{1,3}$
Image 1
$\mathbf{R}_1,\mathbf{t}_1$
Image 2
$\mathbf{R}_2,\mathbf{t}_2$
Image 3
$\mathbf{R}_3,\mathbf{t}_3$

SfM objective function

- Given point **x** and rotation and translation **R**, **t**

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \mathbf{R}\mathbf{x} + \mathbf{t} \qquad u' = \frac{fx'}{z'} \qquad v' = \frac{fy'}{z'} \qquad \begin{bmatrix} u' \\ v' \end{bmatrix} = \mathbf{P}(\mathbf{x}, \mathbf{R}, \mathbf{t})$$

- Minimize sum of squared reprojection errors:

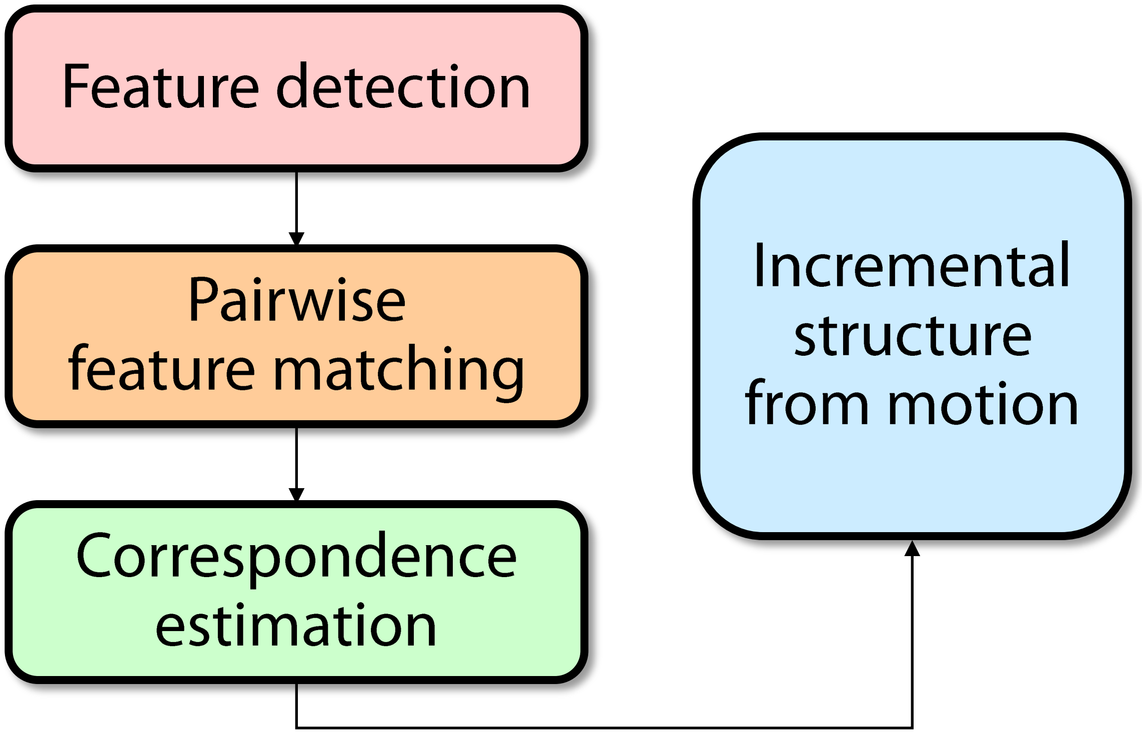
$$g(\mathbf{X}, \mathbf{R}, \mathbf{T}) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \cdot \left\| \underbrace{\mathbf{P}(\mathbf{x}_i, \mathbf{R}_j, \mathbf{t}_j)}_{\substack{\textit{predicted} \\ \text{image location}}} - \underbrace{\begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}}_{\substack{\textit{observed} \\ \text{image location}}} \right\|^2$$

Solving structure from motion

- Minimizing *g* is difficult:
  - *g* is non-linear due to rotations, perspective division
  - lots of parameters: 3 for each 3D point, 6 for each camera
  - difficult to initialize
  - gauge ambiguity: error is invariant to a similarity transform (translation, rotation, uniform scale)

- Many techniques use non-linear least-squares (NLLS) optimization (*bundle adjustment*)
  - Levenberg-Marquardt is one common algorithm for NLLS
  - Lourakis, **The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm**, http://www.ics.forth.gr/~lourakis/sba/
  - http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm

Scene reconstruction



Mosaics

VR Seattle: http://www.vrseattle.com/
Full screen panoramas (cubic): http://www.panoramas.dk/
Mars: http://www.panoramas.dk/fullscreen3/f2_mars97.html

- Today's Readings
  - Szeliski and Shum paper (sections 1 and 2, skim the rest)
    - http://www.cs.washington.edu/education/courses/455/08wi/readings/szeliskiShum97.pdf

Image Mosaics

 +  + ... +  =

Goal

- Stitch together several images into a seamless composite

How to do it?

- Easier than Structure from Motion or Harder?
  - Easier
- Let's Start on your next project right now in class
- Basic Procedure
  - Take a sequence of images from the same position
    - Rotate the camera about its optical center
  - Then what?
  - In Class Exercise

Panoramic Stitching

Input

Goal

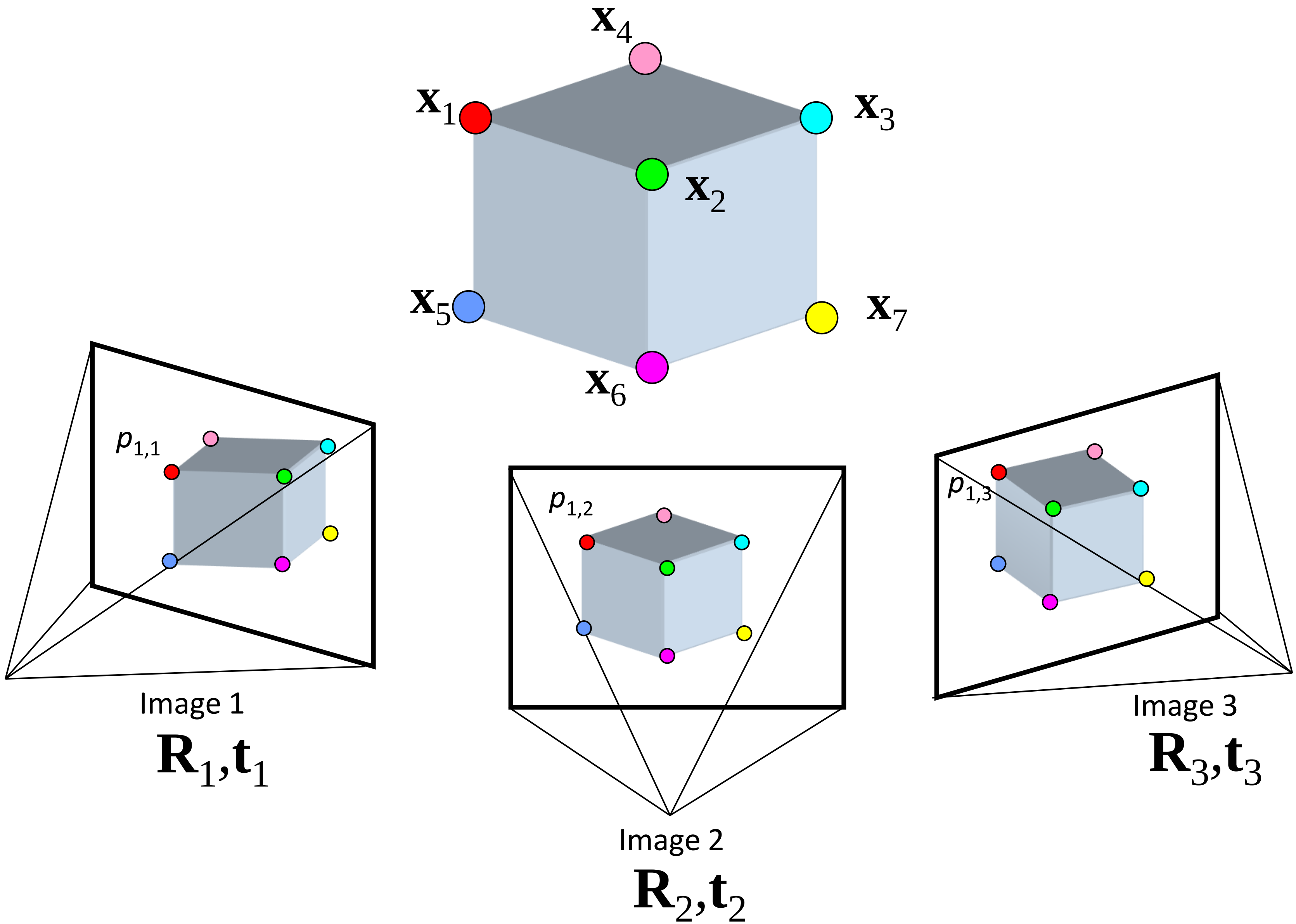
- Try doing using your brain, eyes, and hands – Can you align the images?
- Write an rough algorithm
  - What steps are there, what do you need to find, what things are tricky

How to do it?

- Basic Procedure
  - Take a sequence of images from the same position
    - Rotate the camera about its optical center
  - Compute transformation between second image and first
  - Transform the second image to overlap with the first
  - Blend the two together to create a mosaic
  - If there are more images, repeat

Aligning images

- How to account for warping?
  - Translations are not enough to align the images
  - Photoshop demo


$\mathbf{x}_4$
$\mathbf{x}_1$
$\mathbf{x}_3$
$\mathbf{x}_2$
$\mathbf{x}_5$
$\mathbf{x}_7$
$\mathbf{x}_6$
$p_{1,1}$
$p_{1,2}$
$p_{1,3}$
Image 1
$\mathbf{R}_1,\mathbf{t}_1$
Image 3
$\mathbf{R}_3,\mathbf{t}_3$
Image 2
$\mathbf{R}_2,\mathbf{t}_2$

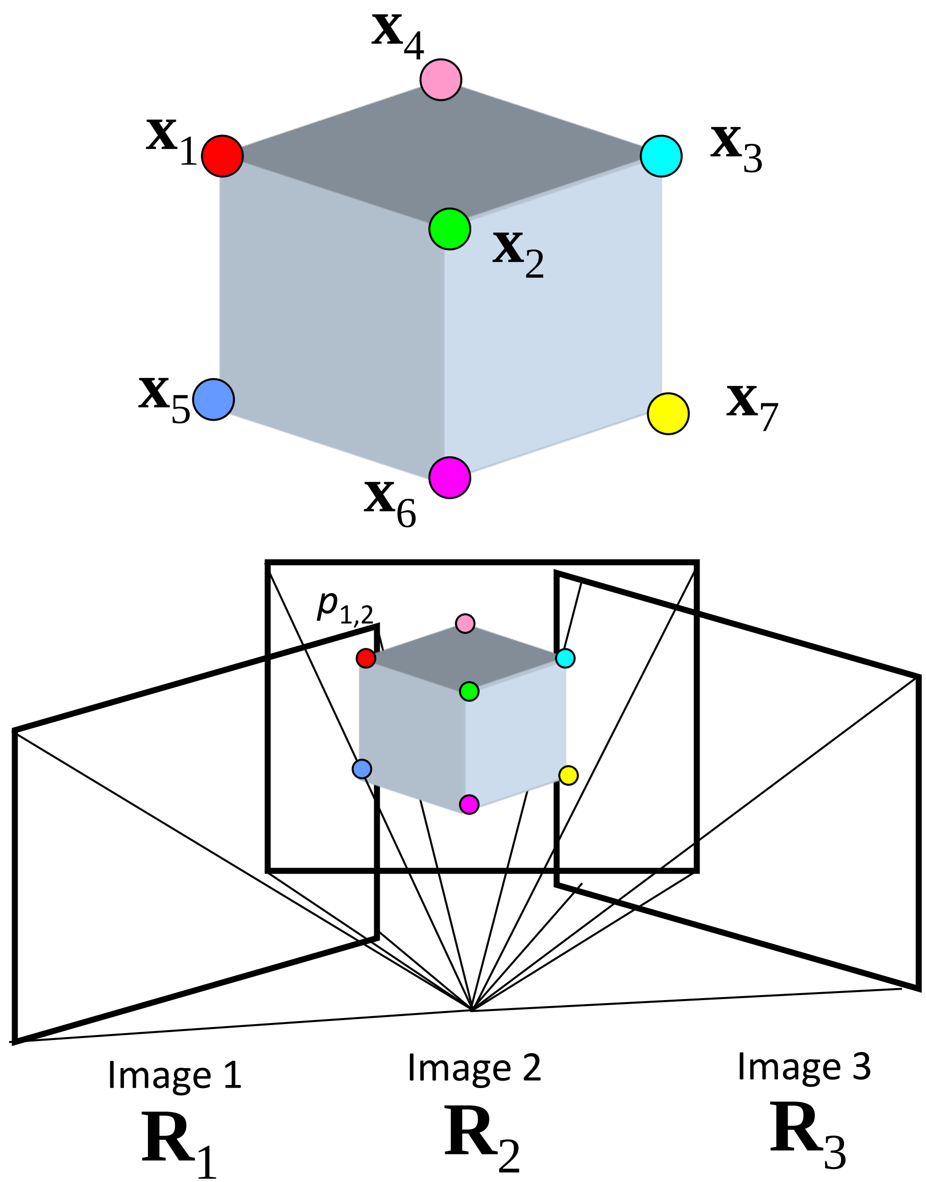
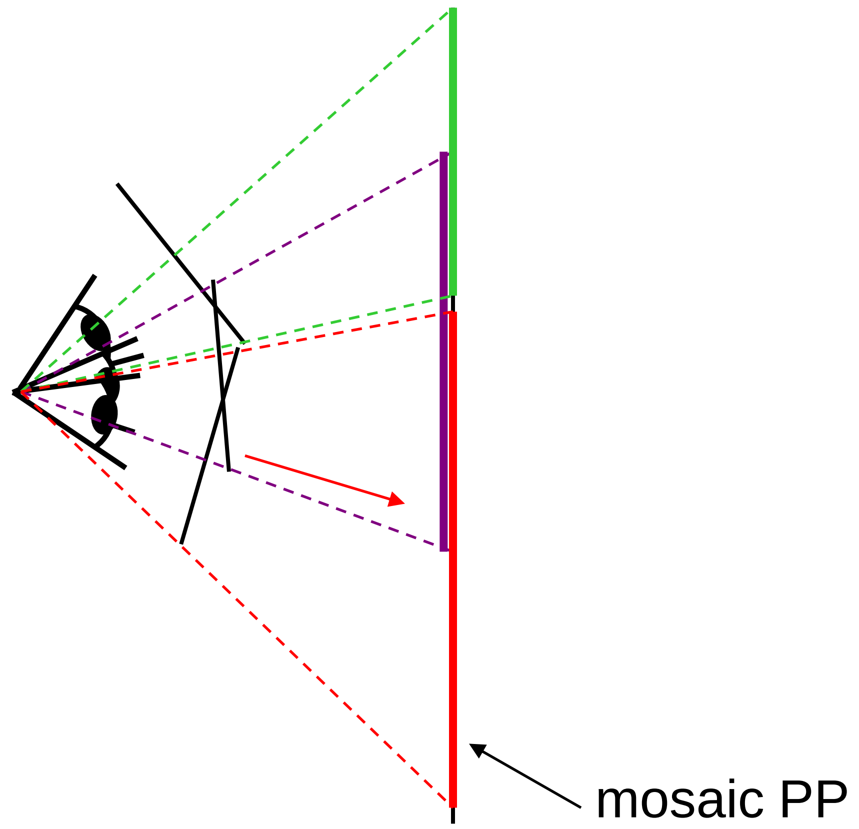

$\mathbf{x}_4$
$\mathbf{x}_1$
$\mathbf{x}_3$
$\mathbf{x}_2$
$\mathbf{x}_5$
$\mathbf{x}_7$
$\mathbf{x}_6$
$p_{1,2}$
Image 1
$\mathbf{R}_1$
Image 2
$\mathbf{R}_2$
Image 3
$\mathbf{R}_3$

Image reprojection



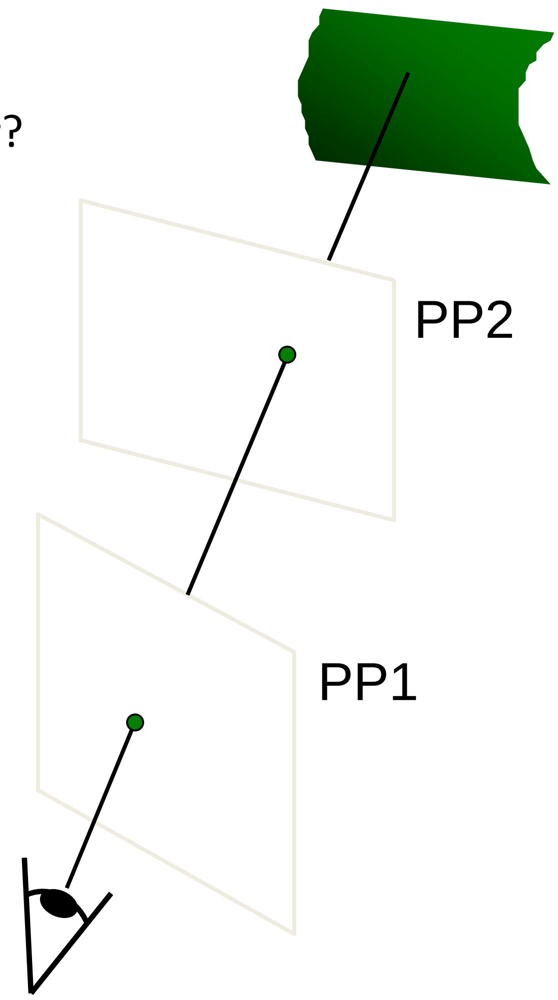
- The mosaic has a natural interpretation in 3D
  - The images are reprojected onto a common plane
  - The mosaic is formed on this plane

Image reprojection

- Basic question
  - How to relate two images from the same camera center?
    - how to map a pixel from PP1 to PP2

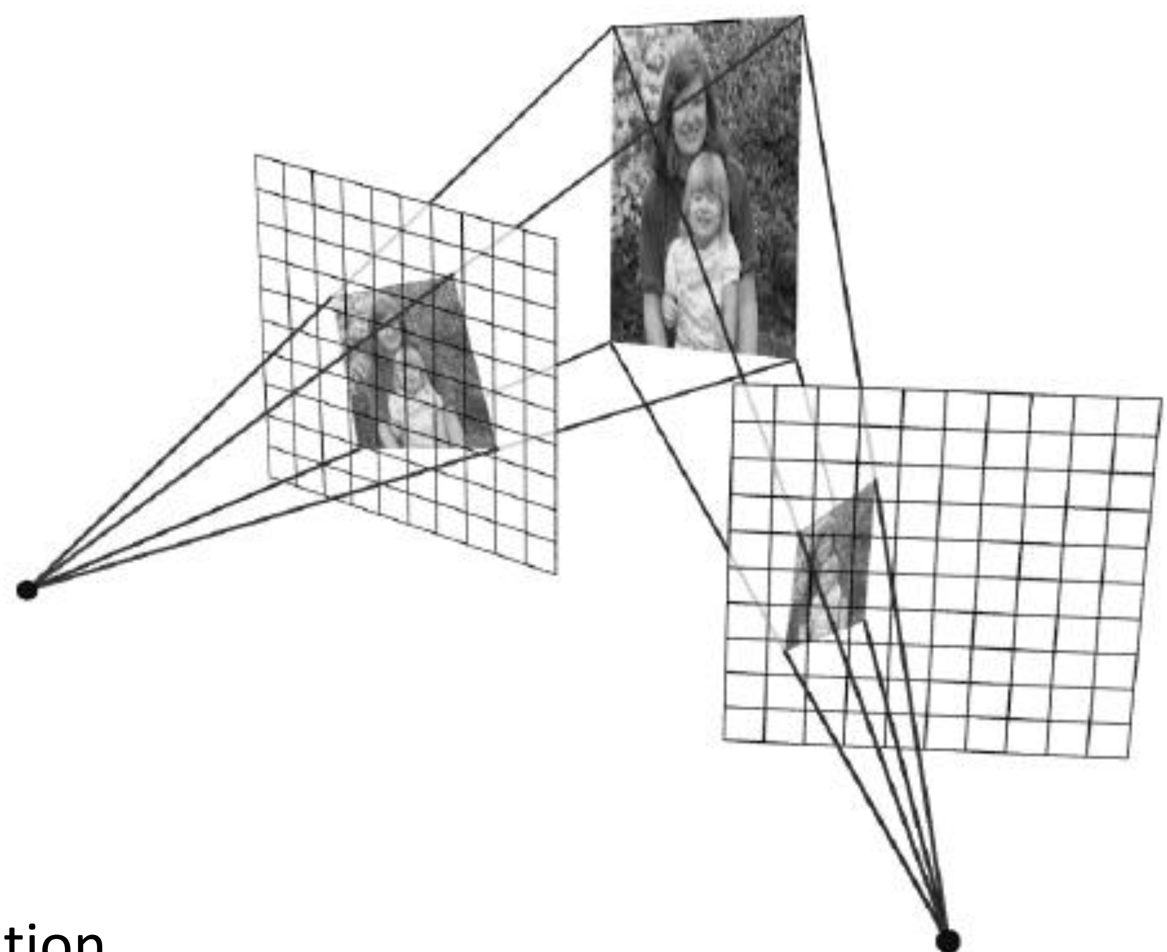
Answer

- Cast a ray through each pixel in PP1
- Draw the pixel where that ray intersects PP2



Don't need to know what's in the scene!

Image reprojection



- Observation
  - Rather than thinking of this as a 3D reprojection, think of it as a 2D image warp from one image to another

Homographies

- Perspective projection of a plane
  - Lots of names for this:
    - **homography**, texture-map, colineation, planar projective map
  - Modeled as a 2D warp using homogeneous coordinates

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

p' **H** **p**

To apply a homography **H**

- Compute **p'** = **Hp** (regular matrix multiply)
- Convert **p'** from homogeneous to image coordinates
  - – divide by w (third) coordinate

Image warping with homographies