CIS 5800

Machine Perception

Instructor: Lingjie Liu

Lec 13: March 19, 2025

Administrivia

- Some changes for my OH in these two weeks:
  - Location: Levine 570
  - This week: 4-5pm, Friday (March 21)
  - Next week: 3:30-4:30pm, Tuesday (March 25)
- Deadline of HW 3 is March 28 (next Friday).
- Midterm Exams (Next Wednesday)
  - The format is slightly different from that of the review questions, but the coverage of the key knowledge points will be the same.
  - 5 long questions (50pts in total = 10pts x 5)
  - Knowledge Points: Everything covered up to last class.
- Combined slides for your mid-term exam preparation have been uploaded to Canvas.
- Review for Knowledge Points: Next Class (Next Monday).
- We will have additional OHs from today until next Tuesday. TAs will announce the specific times asap.

2-View SfM: How we recovered $R, T, \{\lambda_i, \mu_i\}_{i=1}^{n}$

- Invent an object $E = \hat{T}R$
- Express epipolar constraint for every point correspondence $q_i^T E p_i = 0$
- Solve equations to get initial estimate E_0
- $SVD(E_0) = U\Sigma V^T$, Then set $E = U\mathrm{diag}(1,1,0)V^T$
- Then, long proof to show you can set $\pm$(last left singular vector of E) to be T, and there are two possible rotation matrices you can pair each translation vector with.
  - Solve depth by solving $\mu_i R\boldsymbol{p}_i + T = \lambda_i \boldsymbol{q}_i$ for each of these 4 solutions.
  - Settle on the solution that has all (or most) depths positive. (**Sidenote/Q: Are λ_i *and* μ_i *really* "depths?"**)

What is the meaning of these ambiguities in the solution to SfM?

Types of Ambiguity in 2-View SfM Solutions

Mirror ambiguity: If T is a solution, then $-T$ is a solution, too. There is no way to disambiguate from the epipolar constraint: $q^T(-T \times Rp) = 0$.

Just extending scale ambiguity to negative scales. Some people don't even count this as a "true" ambiguity of SfM.

$$\sigma(\widehat{UT_{-z}})\ \left(UR_{z,+\pi/2}V^T\right)$$

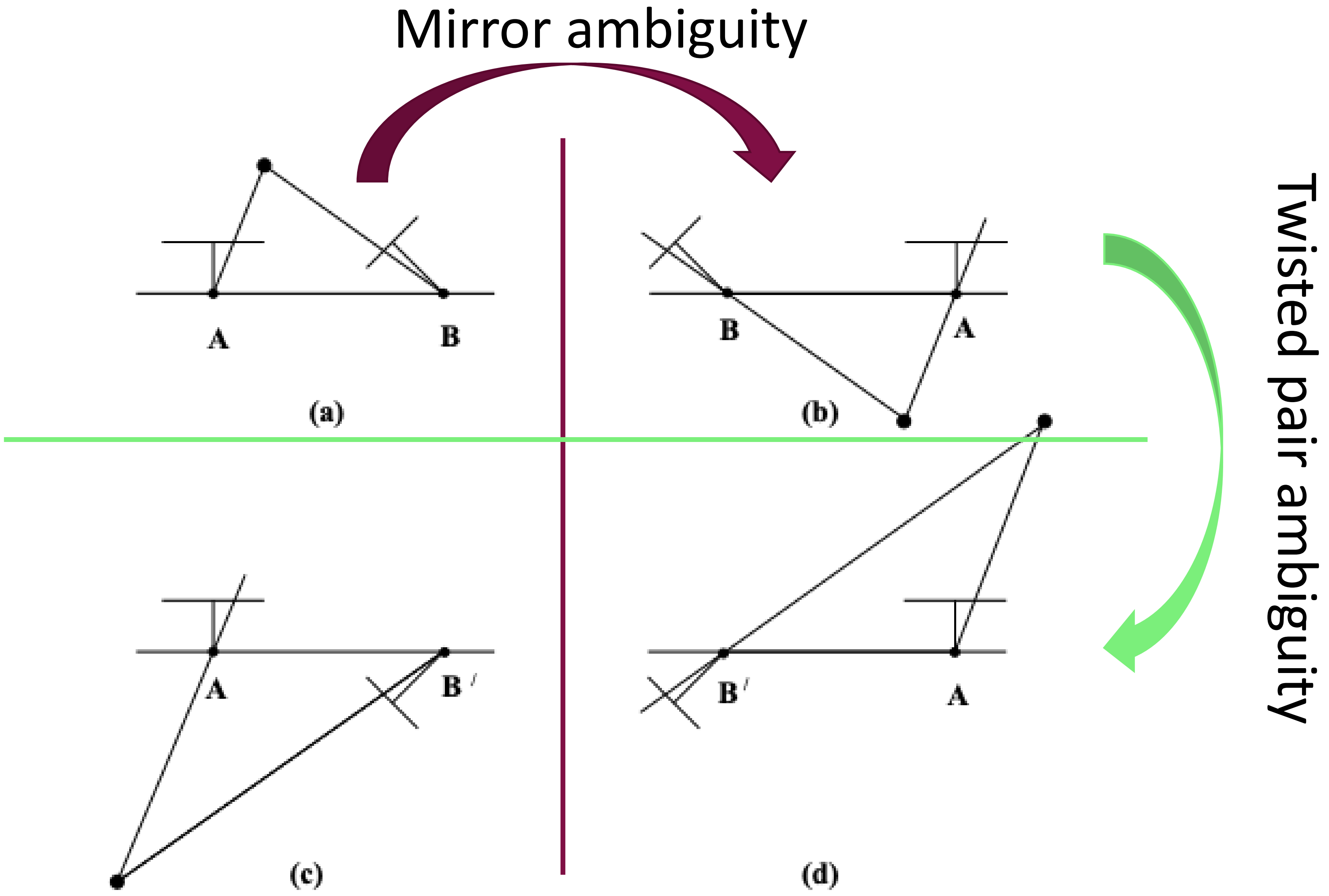
$$\sigma(\widehat{UT_{+z}})\ \left(UR_{z,+\pi/2}V^T\right)$$

$$\sigma(\widehat{UT_{+z}})\ \left(UR_{z,-\pi/2}V^T\right)$$

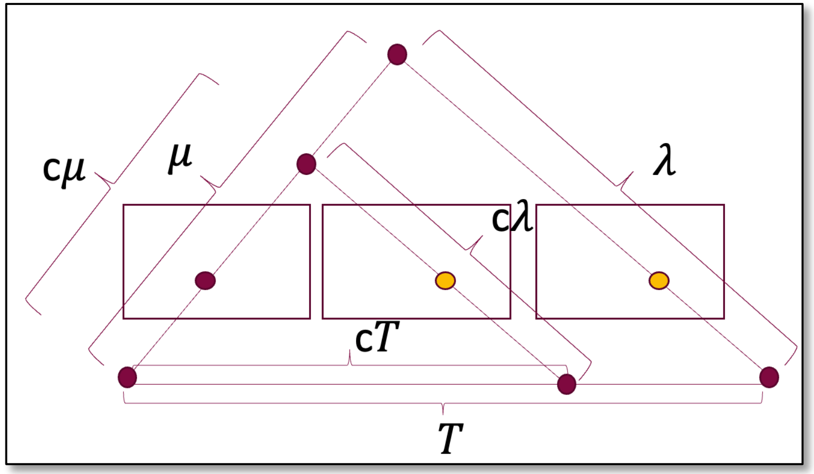
$$\sigma(\widehat{UT_{-z}})\ \left(UR_{z,-\pi/2}V^T\right)$$

Twisted pair ambiguity: If R is a solution, then also $R_{T,\pi}R$ is a solution. The first camera is "twisted" around the baseline 180 degrees.

MSKS Ch 5



Mirror ambiguity
A
B
(a)
B
A
(b)
A
B /
(c)
B /
A
(d)
Twisted pair ambiguity



cμ
μ
λ
cλ
cT
T

What *is* "solving" when solutions have "ambiguity"?

- We said we could solve for camera pose given 3 2D->3D correspondences (P3P)
  - But there was ambiguity: 4 solutions!
- We said ... ences.
  - But t ... se each E into ... roperty of E)
- So what
  - We r
  - In ot ... finite spac
  - We r ... in P3P) to pin c
  - This ... problem setu ... roblems, and ... e relaxed problem.

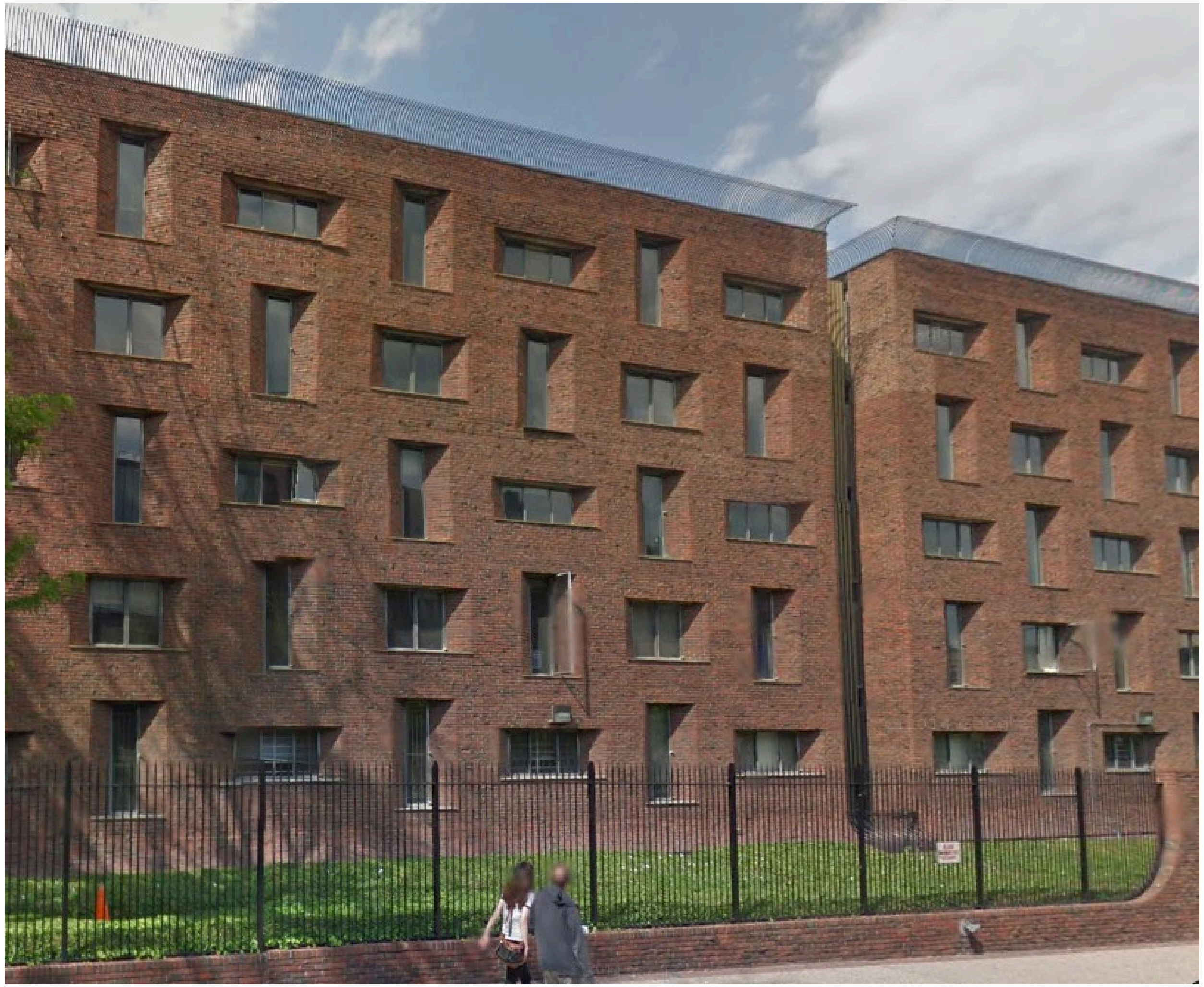
P3P Step 1: The algebraic drudgery

Reduces to two quadratic equations in u and v.

$$
\begin{aligned}
d_{13}^2(u^2 + v^2 - 2uv\cos\delta_{23}) &= d_{23}^2(1 + v^2 - 2v\cos\delta_{13}) \qquad (1)\\
d_{12}^2(1 + v^2 - 2v\cos\delta_{13}) &= d_{13}^2(u^2 + 1 - 2u\cos\delta_{12}) \qquad (2)
\end{aligned}
$$

a) Solve Eqn (1) for u^2 in terms of u, v, v^2 (and constants).
b) Plug this solution into Eqn (2), so that it has no u^2 term. Solve for u in terms of v, v^2, and constants.
c) Plug this solution for u back into Eqn (1), so that it has no more u or u^2. Instead, it is a 4th degree equation in v. *Get the 4 real solutions analytically.*
d) Then plug back into the solution found in step b) above, to get u.
e) Then get d_1 from the quadratic equations on the last page.
f) Then plug back into $d_2 = ud_1$ and $d_3 = vd_1$ from the last page to get d_2, d_3.

What is the relationship between two views of the same plane? (facade)



Answer: As we saw early in the course, homographies!

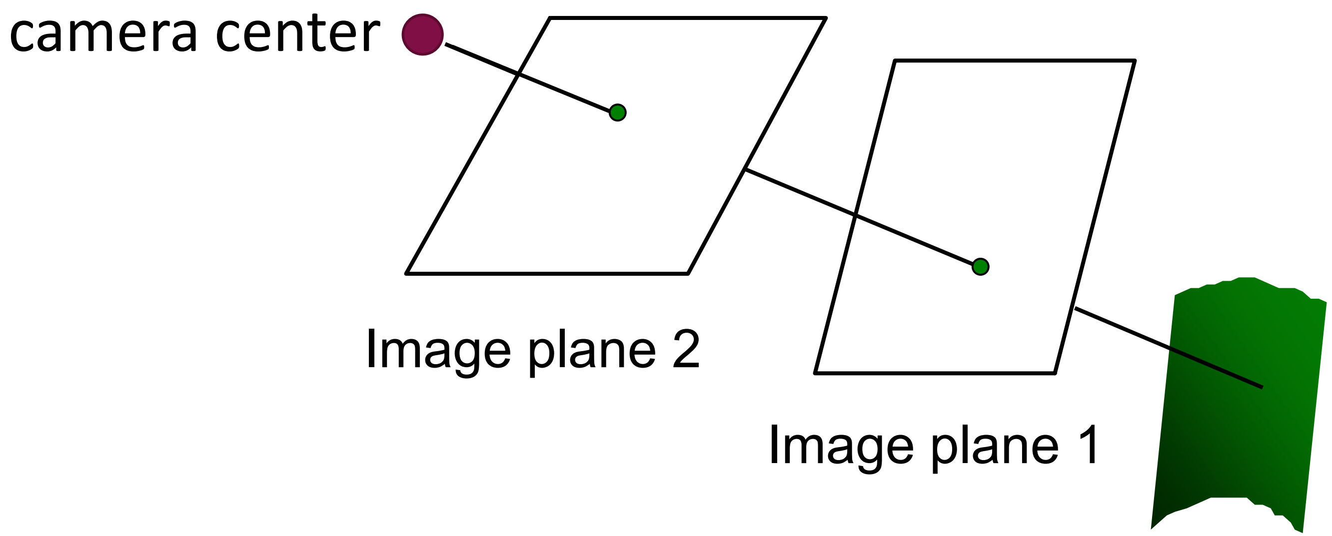
In 2-view SfM, what if there is no translation?

- Epipolar constraint becomes 0=0! So, cannot do all the SfM stuff above!
- Go back to $\lambda q = \mu R p + T (= 0) = \mu R p$
  - In other words: $q \sim Rp$
  - Going back to pixel coordinates from calibrated:
    - $K^{-1} q_{px} \sim R K^{-1} p_{px}$
    - $q_{px} \sim K R K^{-1} p_{px}$
- KRK^{-1} is invertible (determinant =/= 0)
- So this is a homography $H = KRK^{-1}$!

Two views from the same camera center and different camera orientations are related by a homography even if the world is not planar!

No-Translation Image = Image of a Plane!

Two views from the same camera center and different camera orientations are related by a homography even if the world is not planar!



The image formed of the world on plane 2, may also be thought of as:
the image of image plane 1. i.e. image of a plane i.e. homography!

Recap: Checking for no translation during SfM

- If you set up the n point correspondences as $A_{2n\times 9}\boldsymbol{h}_{9\times 1} = 0$ (as we have done before when solving for homography)
- Then, if $\mathrm{rank}(A)$ is (approximately) 8, then you can (approximately) compute H. This means that, actually, there was no (or insignificant) translation, so you can't do SfM!

Q: Is this the only case when H is computable?
A: No, also when imaging a plane, of course.

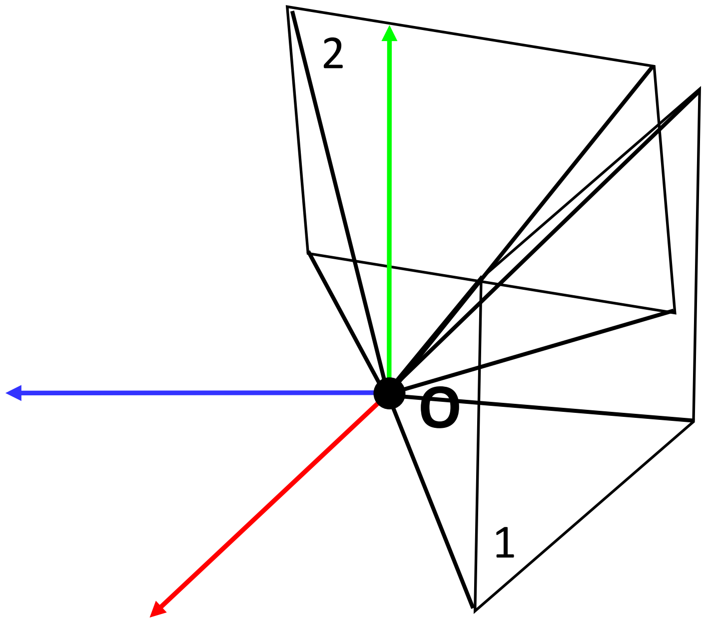
This check is a common component in SfM systems.

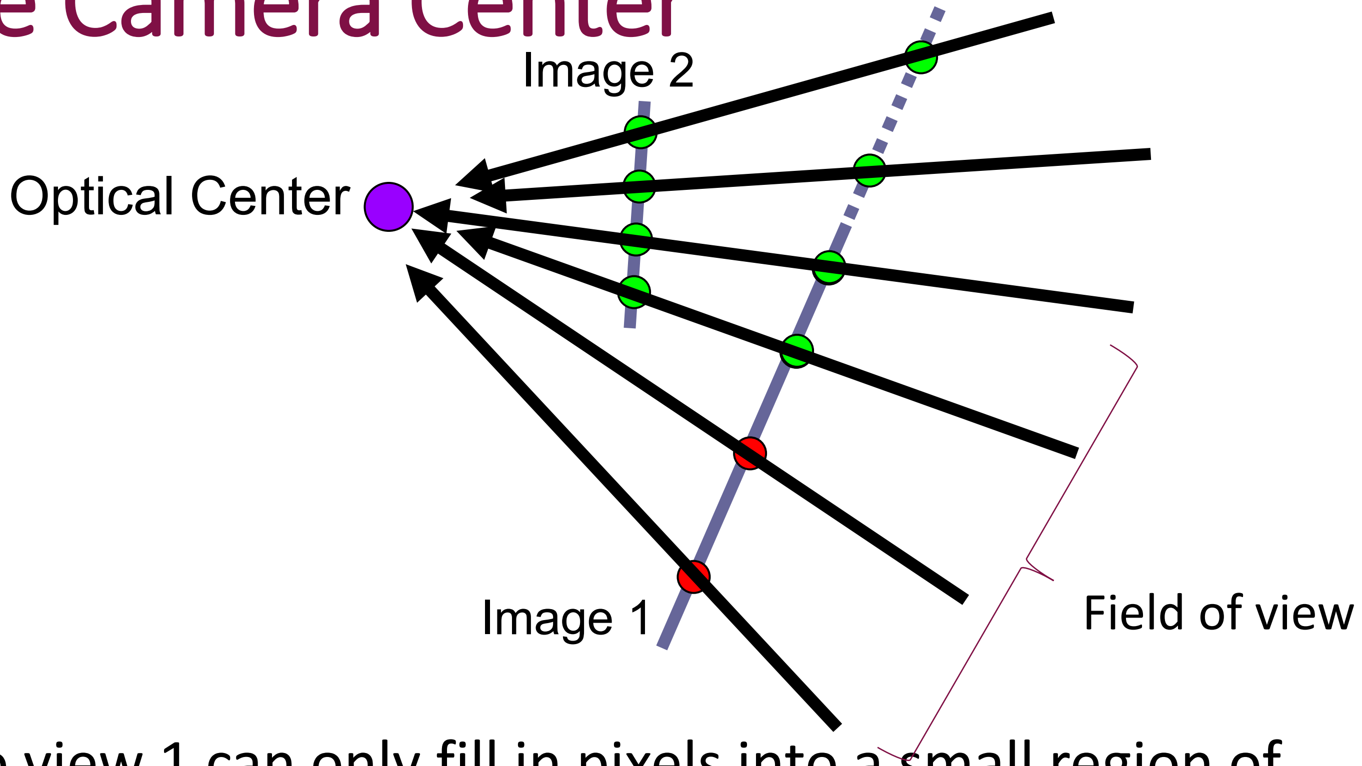
Note: While you can't do *SfM* with no translation of the camera, you can actually do other cool things, like building 360° image "panoramas"!
Coming up next!

Image Stitching / Mosaicing From Rotated Views

Based on slides by Richard Szeliski

2 Views From The Same Camera Center





- Camera field of view (FOV) is finite, so view 1 can only fill in pixels into a small region of image plane 1
- View 2 could see things outside the FOV of view 1. Could we add those pixels into view 1 to extend the image?

Yes. Project both images onto the same image plane e.g. image 1!

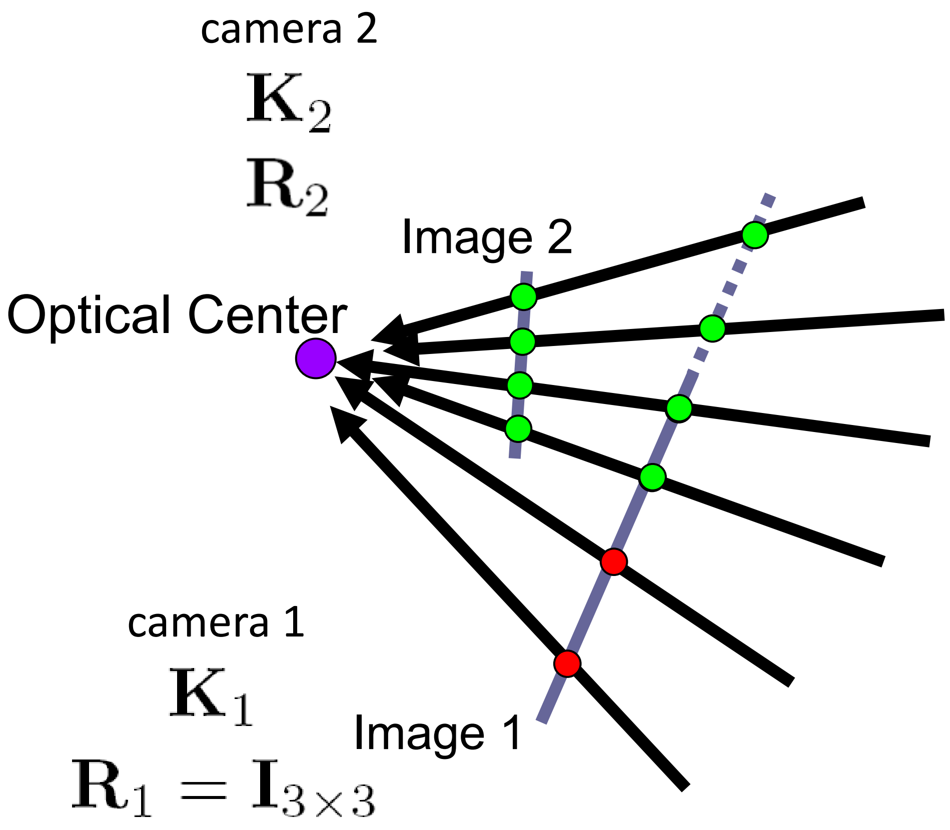
Your human vision system does something like this when your eyeball moves.

Image Stitching Results Example

https://pyimagesearch.com/2018/12/17/image-stitching-with-opencv-and-python/

What is the transformation between these views?

For every pixel in image 2, we need to find the right pixel location to place it inside image 1.



3x3 homography ($\mathbf{K}_1\mathbf{R}_1\mathbf{R}_2^{\mathrm{T}}\mathbf{K}_2^{-1}$)

$$\underbrace{\begin{bmatrix}x_1\\y_1\\1\end{bmatrix}}_{\text{image coords (in image 1)}} \sim \underbrace{\mathbf{K}_1 \underbrace{\mathbf{R}_1 \underbrace{\mathbf{R}_2^{\mathrm{T}} \underbrace{\mathbf{K}_2^{-1} \underbrace{\begin{bmatrix}x_2\\y_2\\1\end{bmatrix}}_{\text{image coords (in image 2)}}}_{\text{3D ray direction (in cam 2 coordinates)}}}_{\text{3D ray direction in world coordinates}}}_{\text{3D ray direction in camera 1 coordinates}}}_{\text{Project through camera 1!}}$$

Homography even though not restricting to imaging a plane!

In practice, the homography is *estimated*

- We now know that the two images are related by a homography.

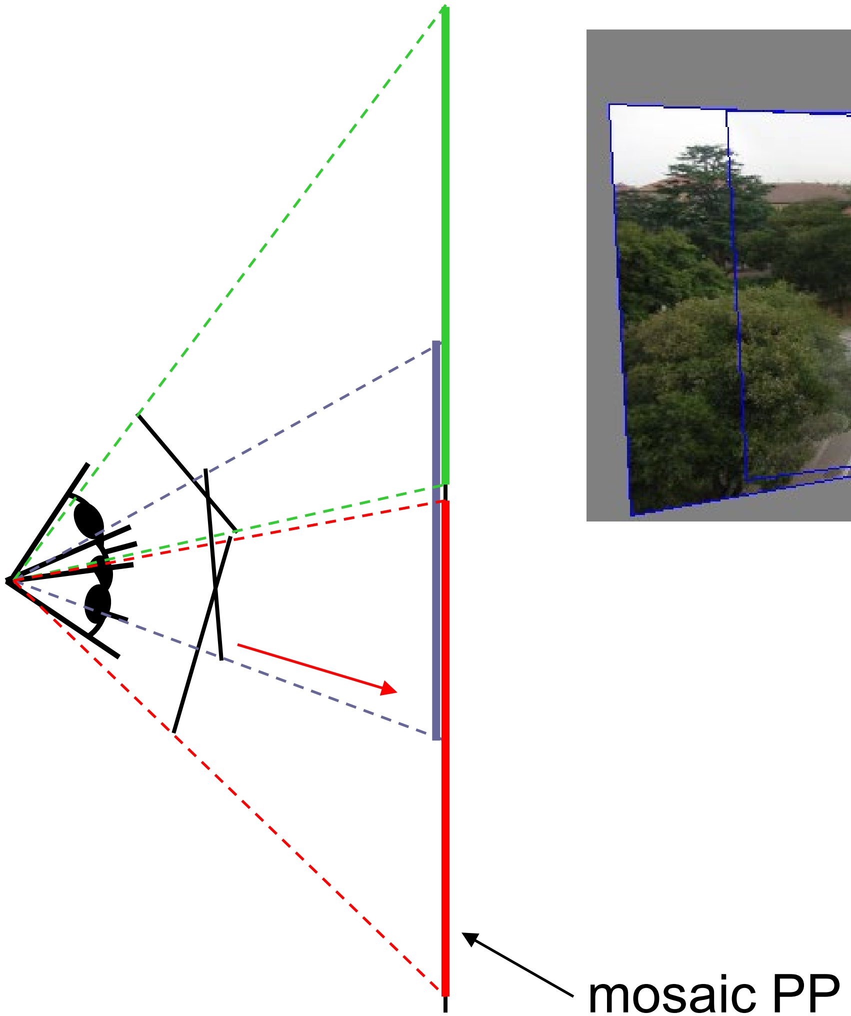
- In practice, we don't know R or even necessarily K_1 or K_2

- But fortunately, we know how to solve for homography H given point correspondences.
  - So we need the two images to overlap so that you can find correspondences and estimate the homography.

Creating a panorama from >2 images

- 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

Projecting images onto a common plane



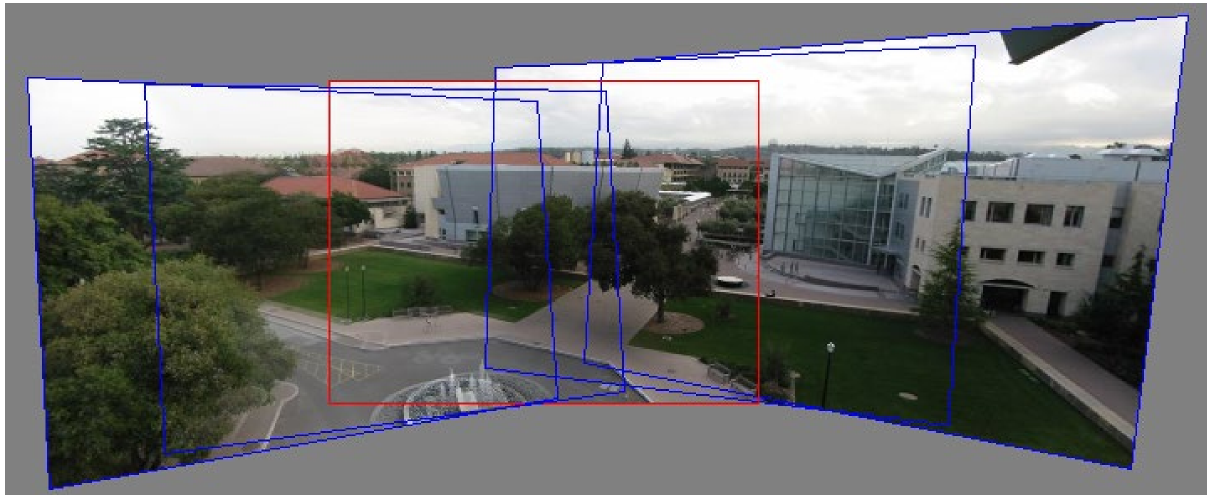
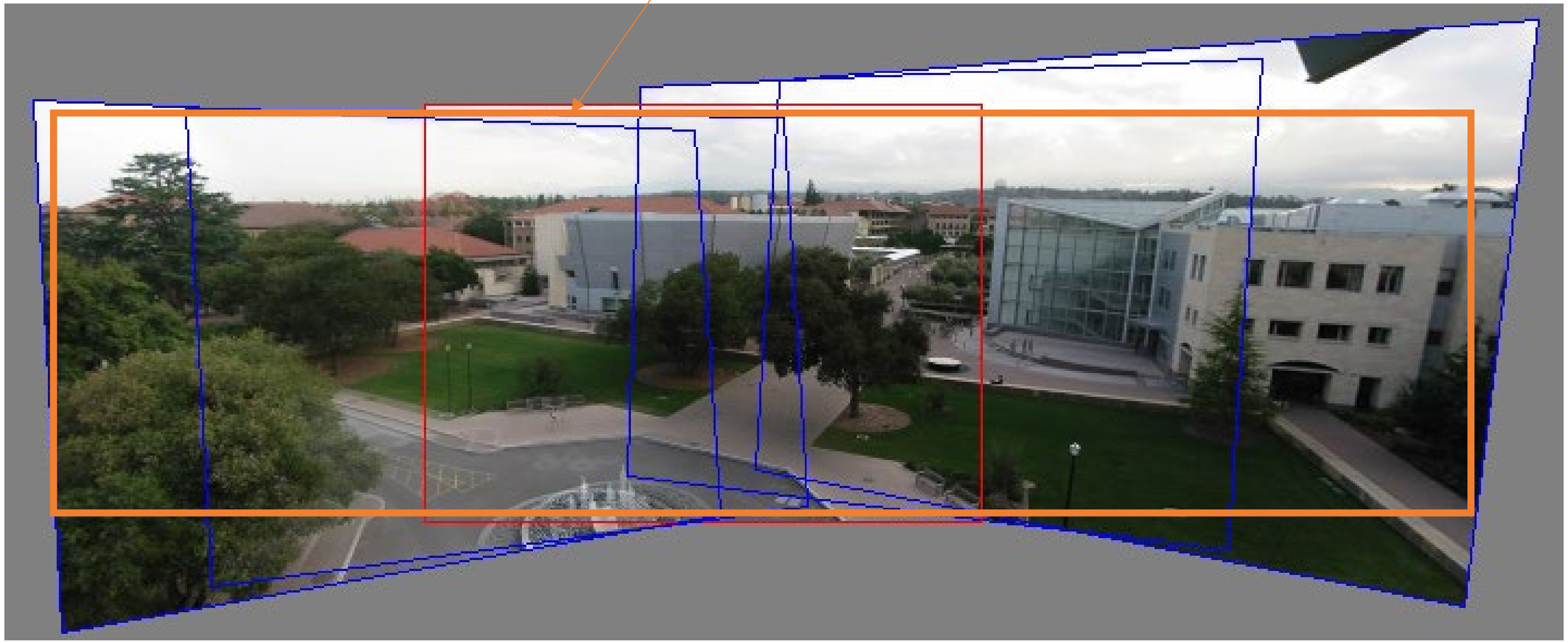


Image alignment

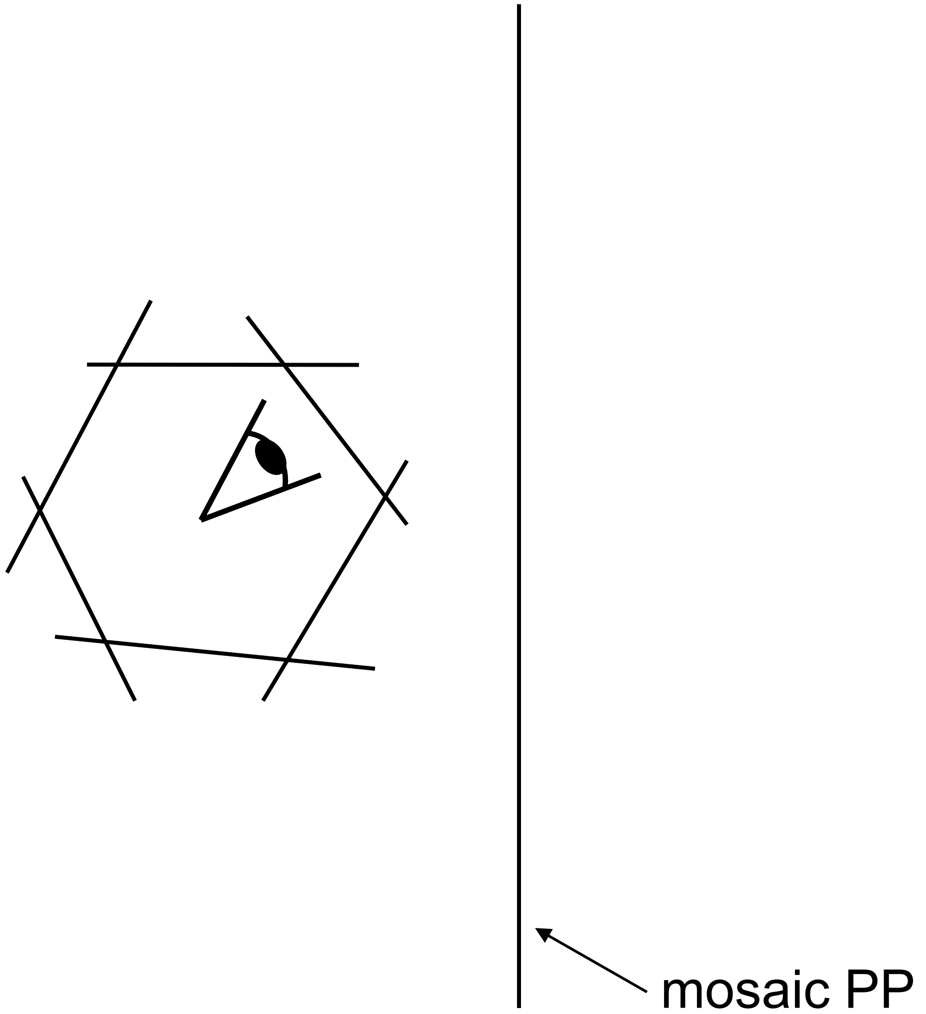
Often cropped like this to produce an image without empty regions



Can we use homography to create a 360° panorama?

Microsoft Lobby: http://www.acm.org/pubs/citations/proceedings/graph/258734/p251-szeliski

Can we use homography to create a $360°$ panorama?



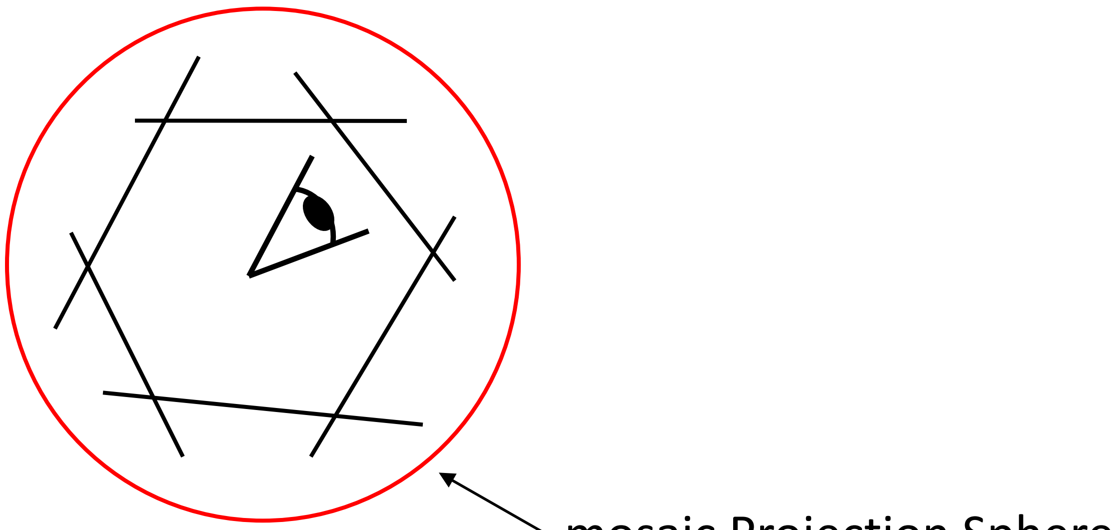
Projecting backwards-facing views onto the same image plane?

Breaks down.

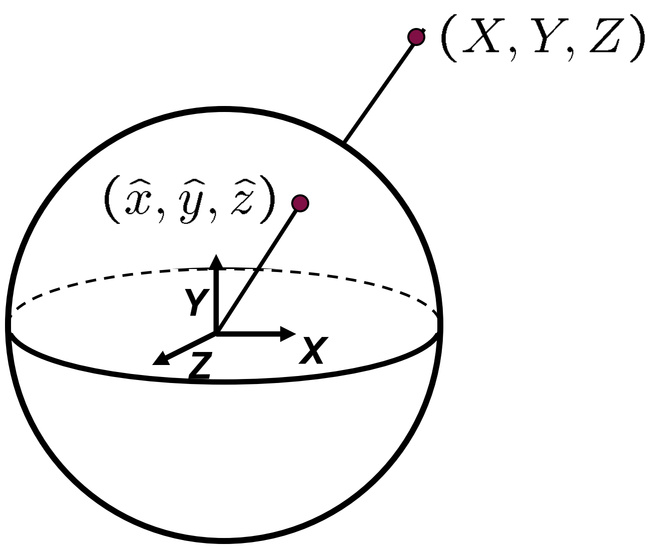
Panoramas

Instead of projecting onto a plane, you project onto a sphere!

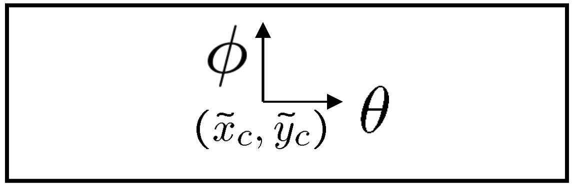
i.e. for every 3D point, draw the line connecting it to the camera center, and find its intersection with a projection sphere --- this is the point's image!



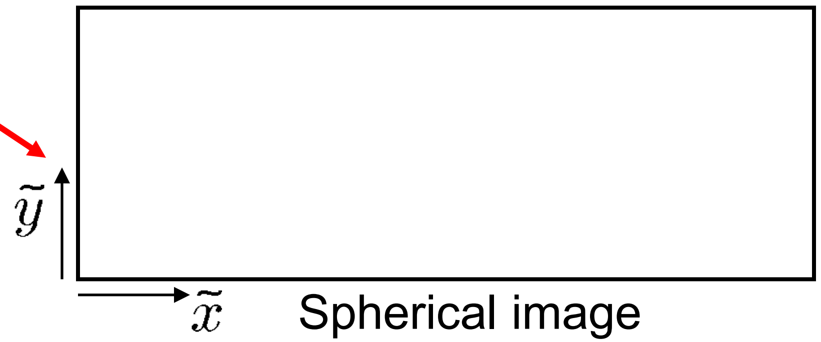
Spherical projection (overview)



unit sphere



unwrapped sphere



Spherical image

- Map 3D point (X,Y,Z) onto sphere

$$(\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2+Y^2+Z^2}}(X, Y, Z)$$

- Convert to spherical coordinates

$$(sin\theta cos\phi, sin\phi, cos\theta cos\phi) = (\hat{x}, \hat{y}, \hat{z})$$

- Convert to spherical image coordinates

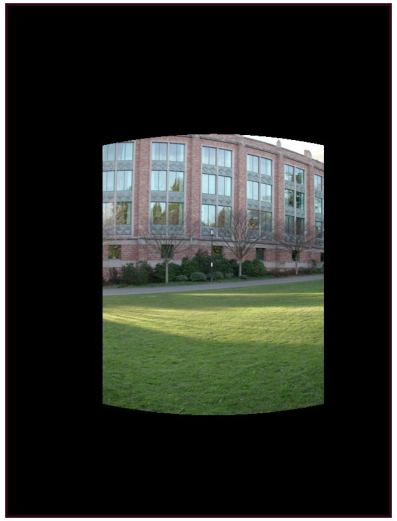
$$(\tilde{x}, \tilde{y}) = (s\theta, s\phi) + (\tilde{x}_c, \tilde{y}_c)$$

  - s defines size of the final image
    - often convenient to set s = camera focal length

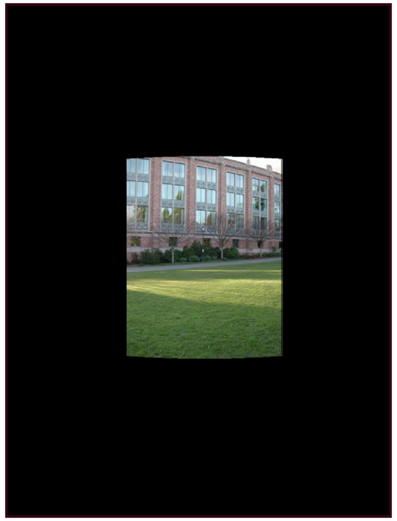
Spherical reprojection

input

f = 200 (pixels)



f = 400



f = 800

- Map image to spherical coordinates
  - need to know the focal length

Spherical Panorama Example

Note the distortions

Microsoft Lobby: http://www.acm.org/pubs/citations/proceedings/graph/258734/p251-szeliski