PANORAMIC IMAGING – PART 1

Take Home Message

Q: What is a Panorama?

A: Wider-angle image than a normal camera can capture

Compensate for shortcomings of traditional cameras by capturing and fusing many images

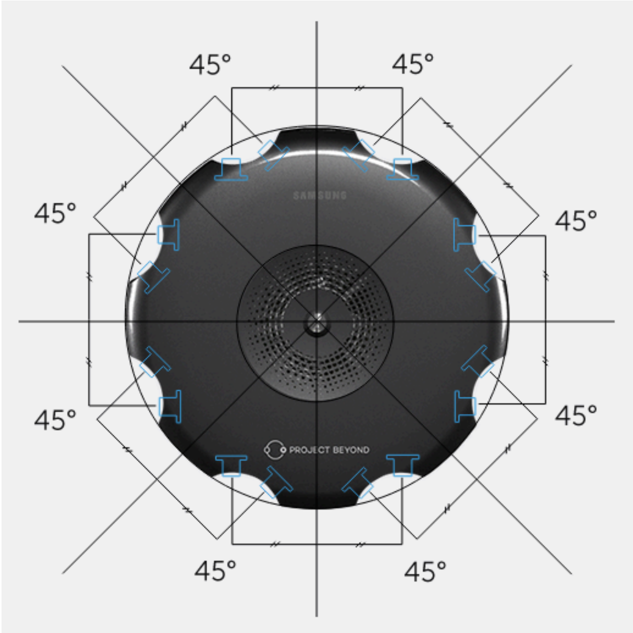
Technique:

- Take N images at different directions
- Deduce the image that would have been taken by a wide angle lens

Application: Virtual Reality

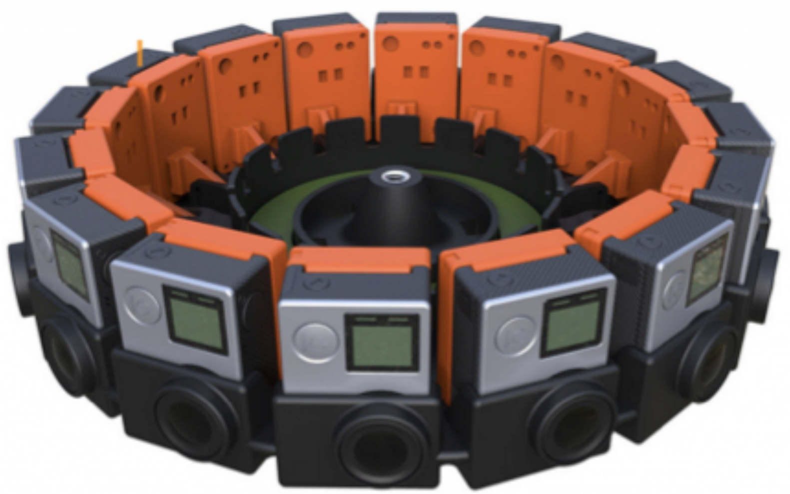
Real World Capture Setups

Jaunt NEO



Samsung Beyond

Google Jump



Nokia OZO

Panorama: Virtual Wide Angle

http://people.csail.mit.edu/fredo/Panos/

Two lectures:

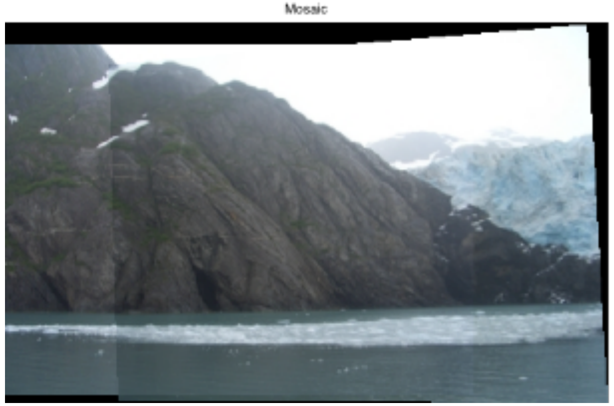
Part 1: Monoscopic Panoramas
(today's lecture)

Part 2: Stereoscopic Panoramas
(next lecture)

Panoramic Imaging Pipeline

Estimating Correspondences

Warping



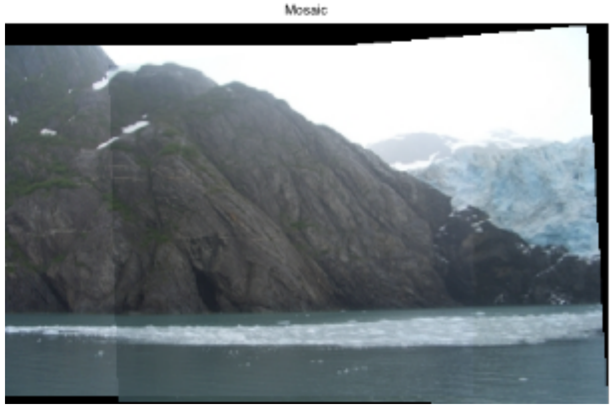
Blending

Panoramic Imaging Pipeline

Estimating Correspondences

1. Detect keypoints
2. Find out which keypoints in im1 match which keypoints in im2

Warping



3. Compute transformation between im2 and im1
4. Transform im2 to overlap with im1

Blending

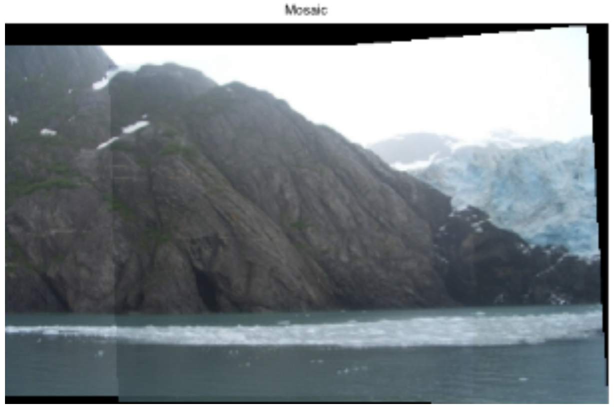
5. Blend the images to create a smooth transition
6. Repeat for other images

Panoramic Imaging Pipeline

Estimating Correspondences

Covered in upcoming lectures
For now: assume correspondences are known

Warping



3. Compute transformation between im2 and im1

4. Transform im2 to overlap with im1

Blending

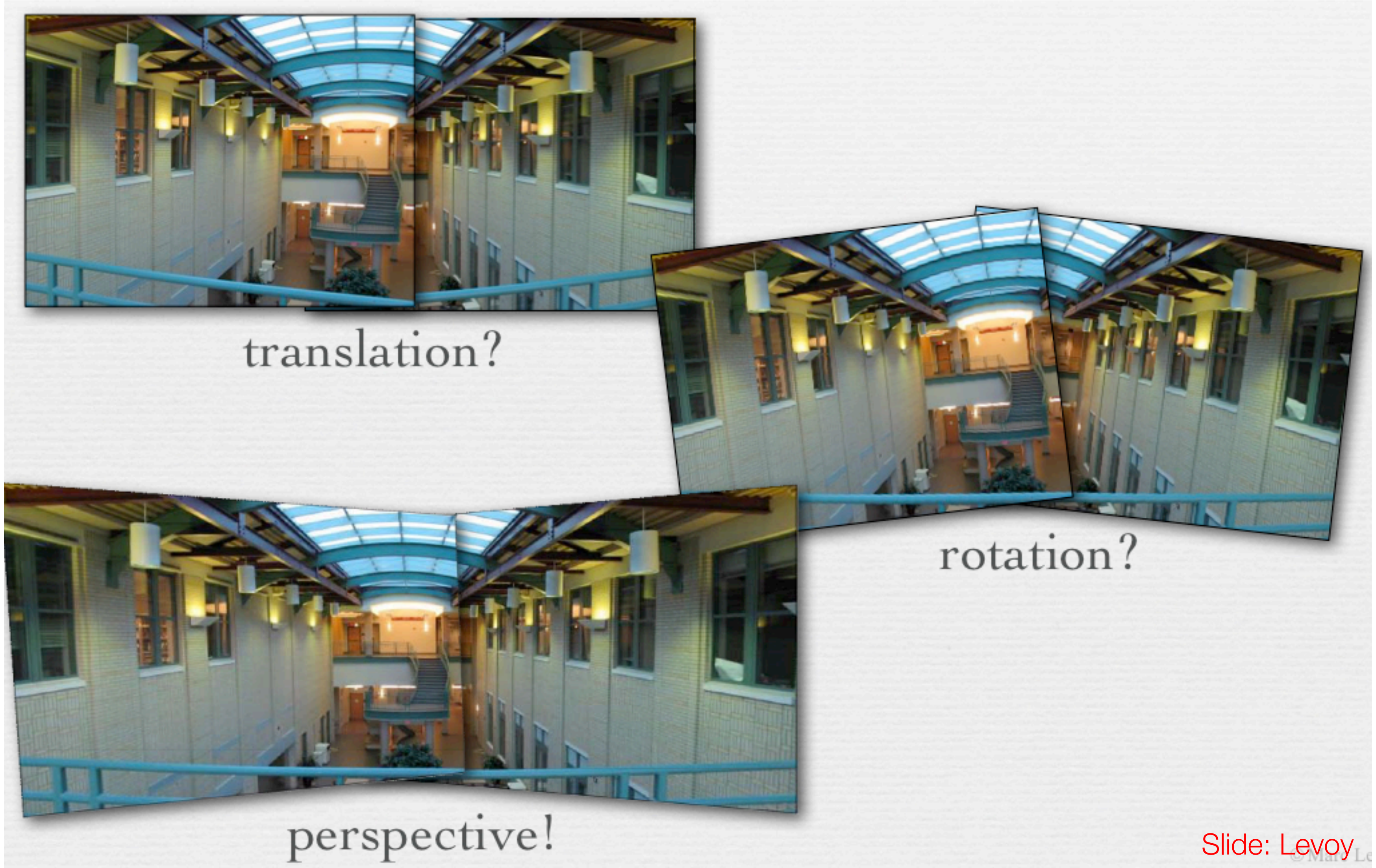
5. Blend the images to create a smooth transition

6. Repeat for other images

Warp to Align Images

Images: Levoy

How to Warp?



Slide: Levoy

Image Warping: Mathematical Representation

image filtering: change ***range*** of image

$$g(x) = T(f(x))$$

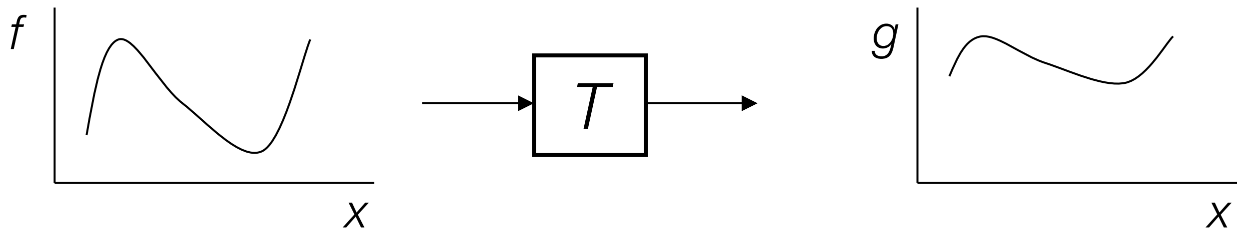
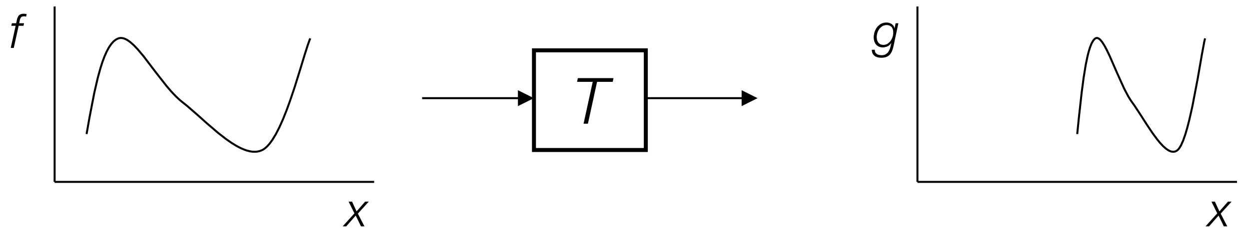


image warping: change ***domain*** of image

$$g(x) = f(T(x))$$



Slide: Efros

Image Warping: Mathematical Representation

image filtering: change ***range*** of image

$$g(x) = T(f(x))$$

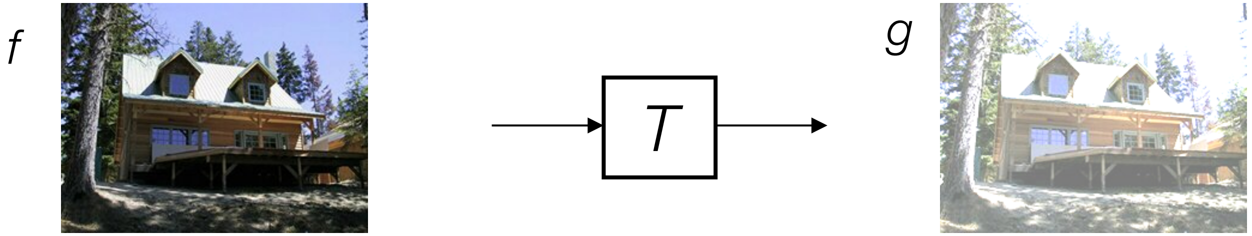
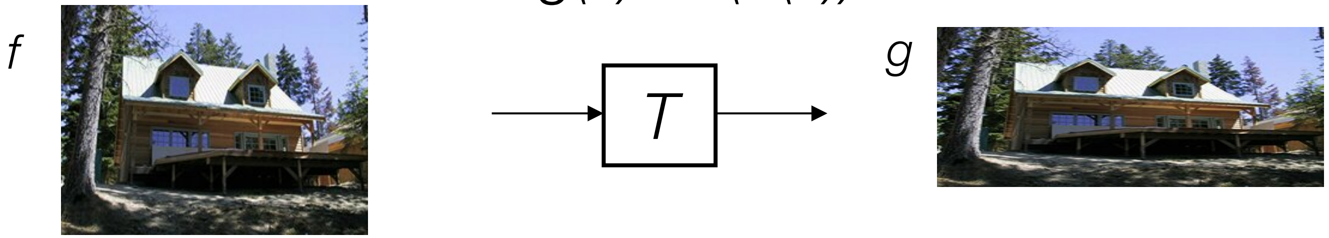


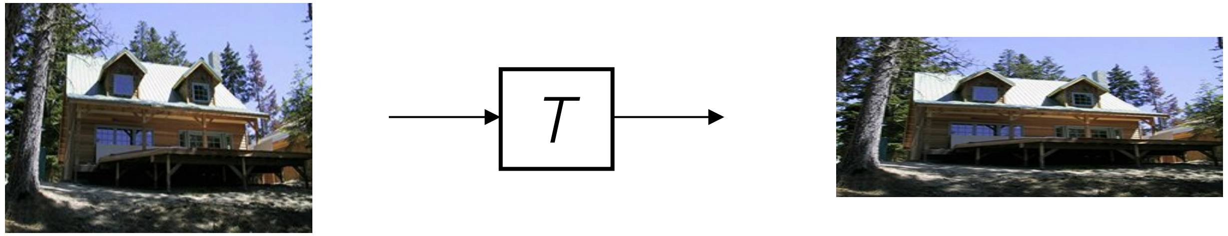
image warping: change ***domain*** of image

$$g(x) = f(T(x))$$



Slide: Efros

Image Warping: Mathematical Representation



$p = (u, v)$ $\qquad$ $p' = (u', v')$

Generic model

$$p' = T(p)$$

Can we model the transformation as a matrix?

2D Transformations

Scaling

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

Rotation

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

What about translation?

$$u' = u + t_u$$

$$v' = v + t_v$$

Homogeneous Coordinates

- Notation trick
- Represent 2D points with 3 numbers
- Homogeneous coordinates (u, v, w) → image coordinates (u/w, v/w)

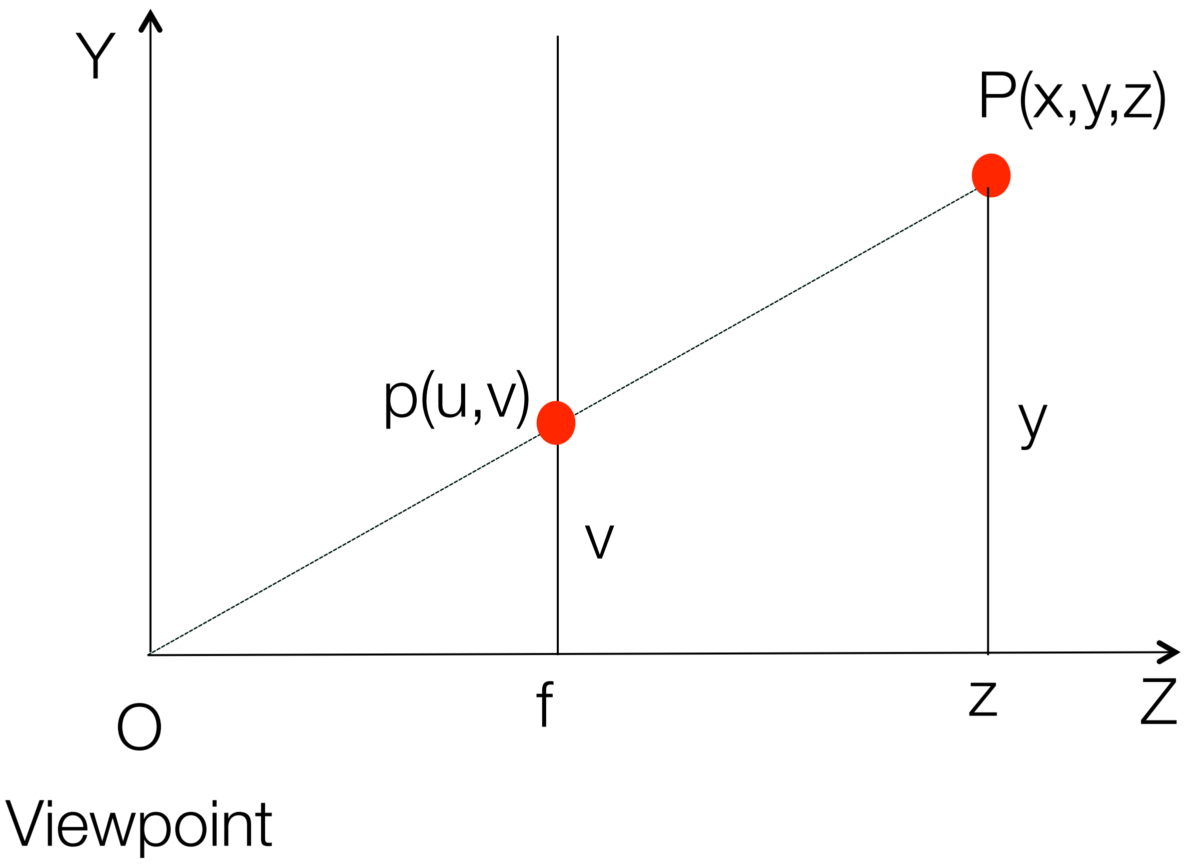
$$u' = u + t_u$$
$$v' = v + t_v$$

→

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_u \\ 0 & 1 & t_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Simple Perspective Projection

Project all points to z = f plane



$$u = f.x / z$$

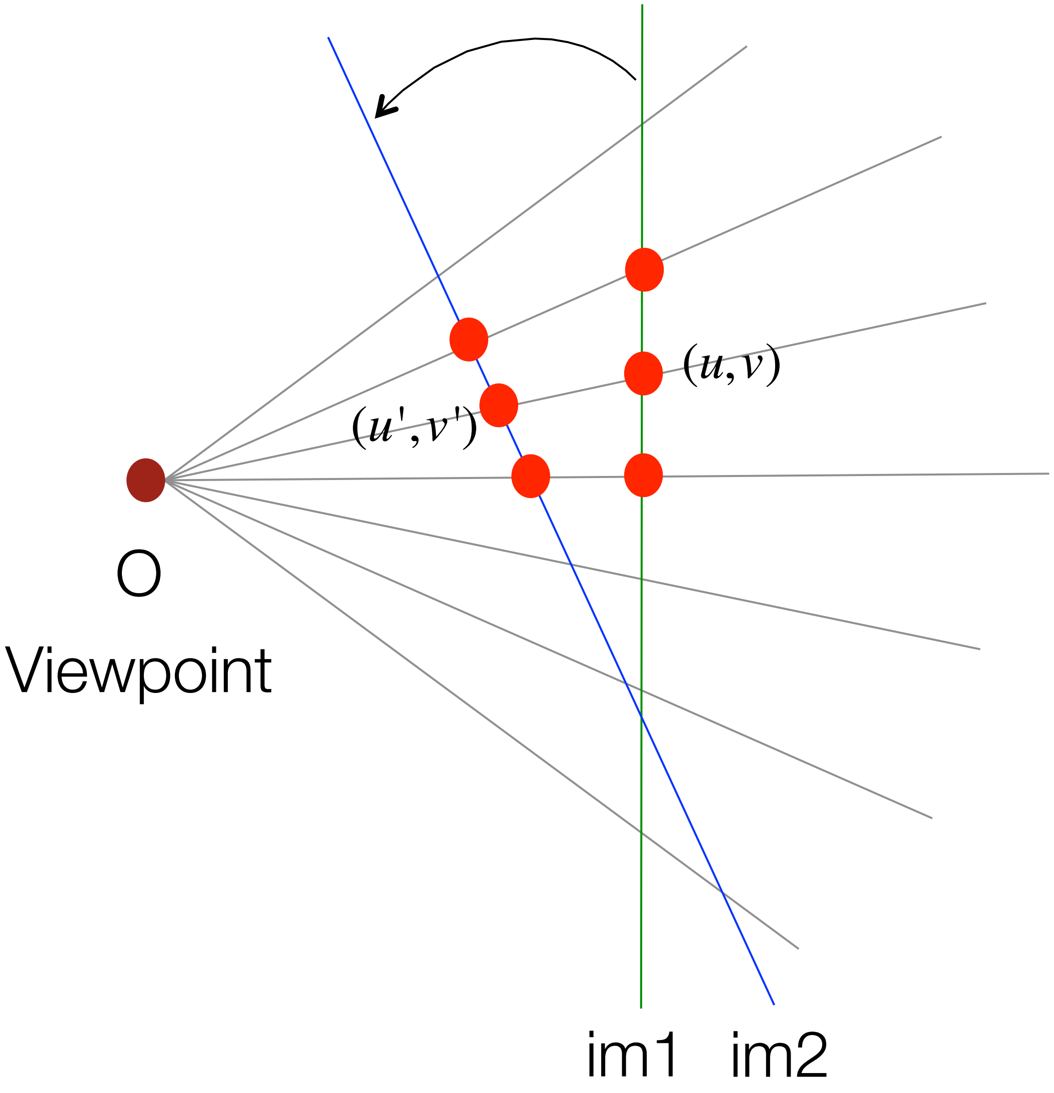
$$v = f.y / z$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Image coordinates

Homogeneous coordinates

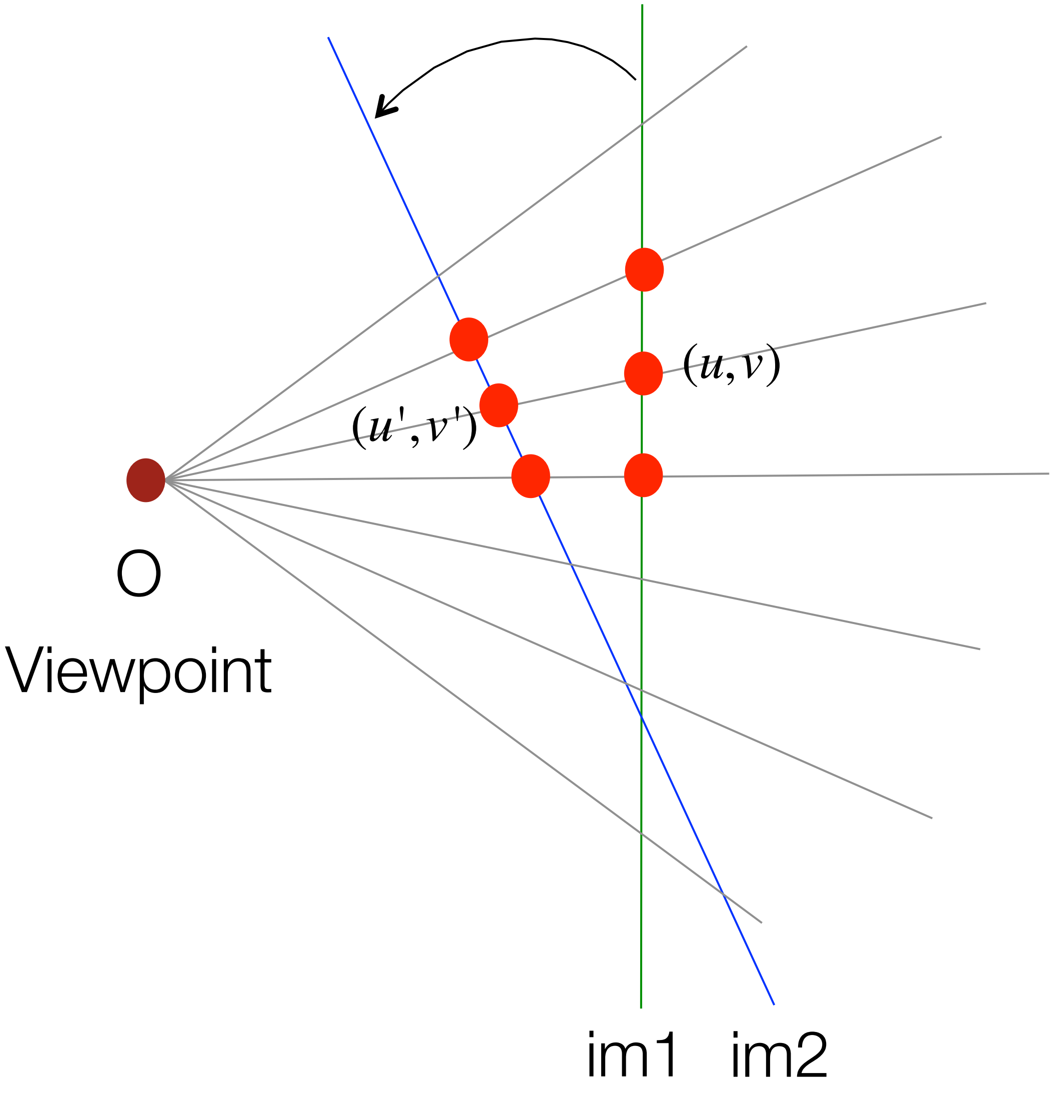
Central Projection with Rotating Camera



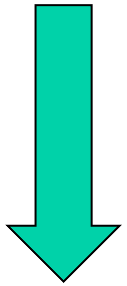
Transform between two images
with the same camera center

Which transform links (u, v) to (u’, v’) ?

Central Projection and Homography



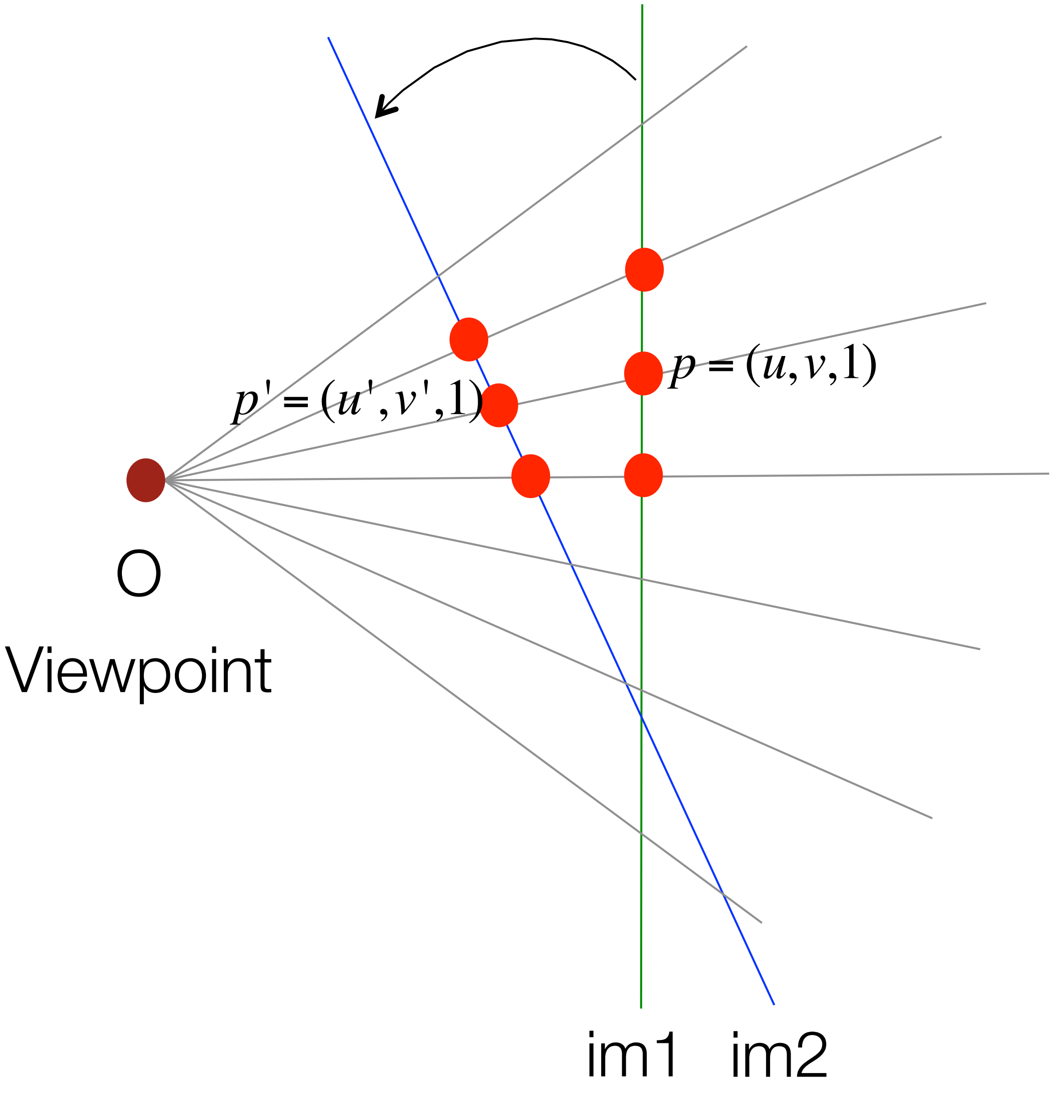
Observation:
Lines in im1 map onto lines in im2



Any transformation of a plane onto a plane that preserves straight lines must be a homography

Hartley & Zisserman, Theorem 2.10

Central Projection and Homography



H: non-singular 3x3 matrix

$$p' = H.p$$

$$\begin{bmatrix} wu' \\ wv' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Valid up to scale

So can set $i = 1$

Estimating Homography

$$u'_k = \frac{a.u_k + b.v_k + c}{g.u_k + h.v_k + 1} \Rightarrow u_k a + v_k b + c - u_k u'_k g - v_k v'_k h = u'_k$$

$$v'_k = \frac{d.u_k + e.v_k + f}{g.u_k + h.v_k + 1} \Rightarrow u_k d + v_k e + f - u_k u'_k g - v_k v'_k h = v'_k$$

$$\vdots$$

Set up a system of linear equations

$$Ax = b$$

$$x = [a,b,c,d,e,f,g,h]^T$$

Solve for Homography

Each correspondence gives 2 equations

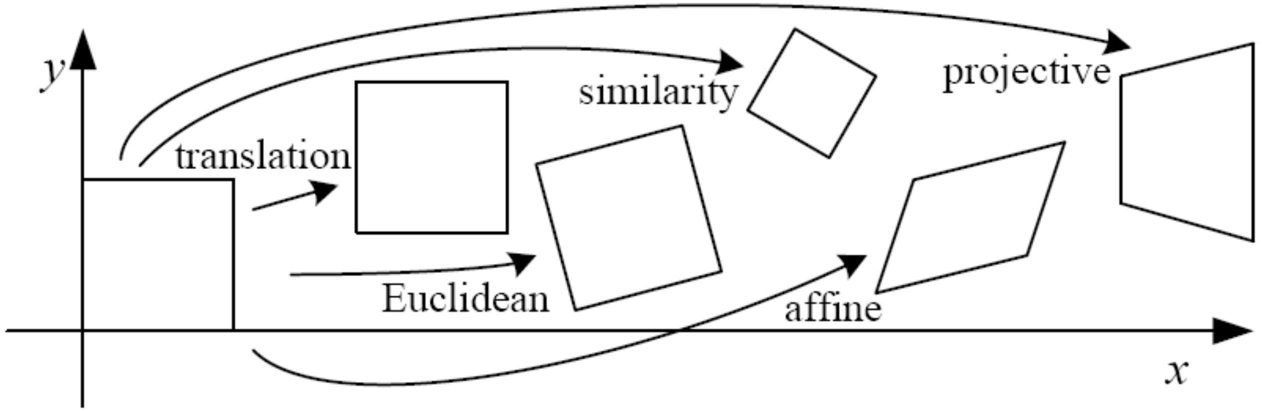
Need at least 8 equations, but the more the better…(why?)

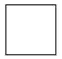
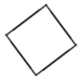
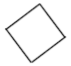
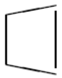
If overdetermined, solve using least squares

Minimize $\| Ax - b \|^2$

Can be done using “\” in Matlab

2D Image Transformations

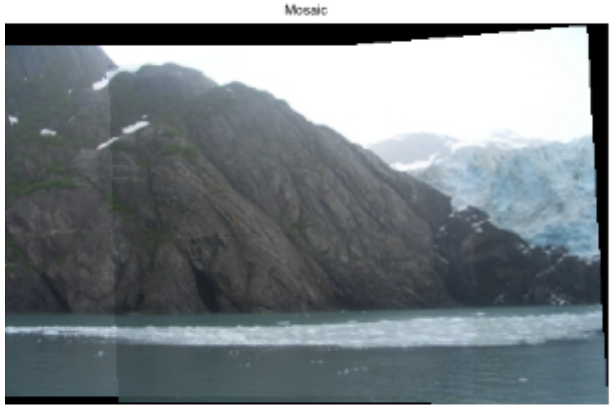


Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation $+\cdots$	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths $+\cdots$	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles $+\cdots$	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism $+\cdots$	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	straight lines	

Panoramic Imaging Pipeline

Estimating Correspondences

Warping



3. Compute transformation between im2 and im1

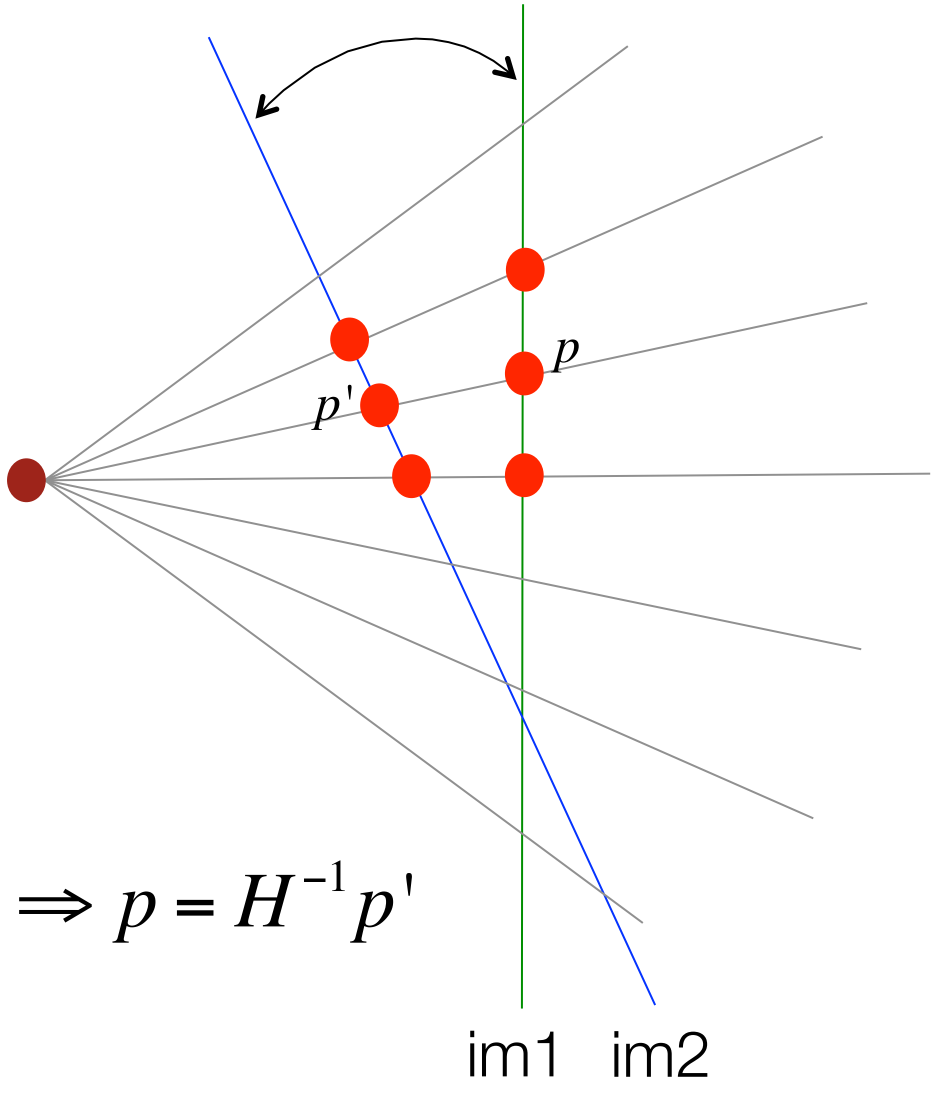
4. Transform im2 to overlap with im1

Blending

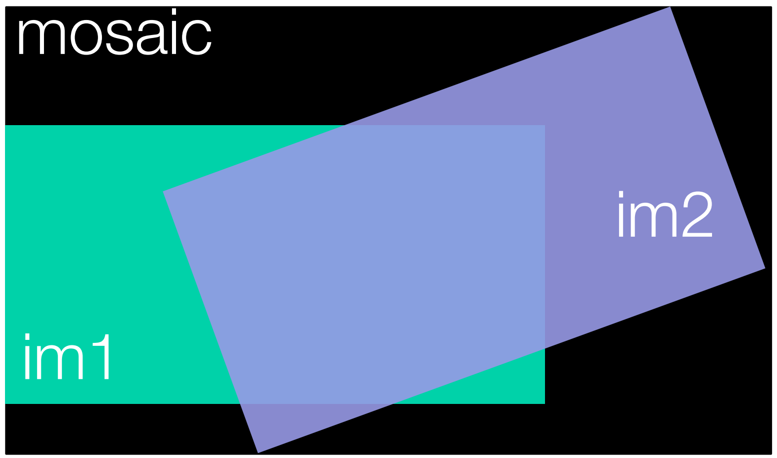
5. Blend the images to create a smooth transition

6. Repeat for other images

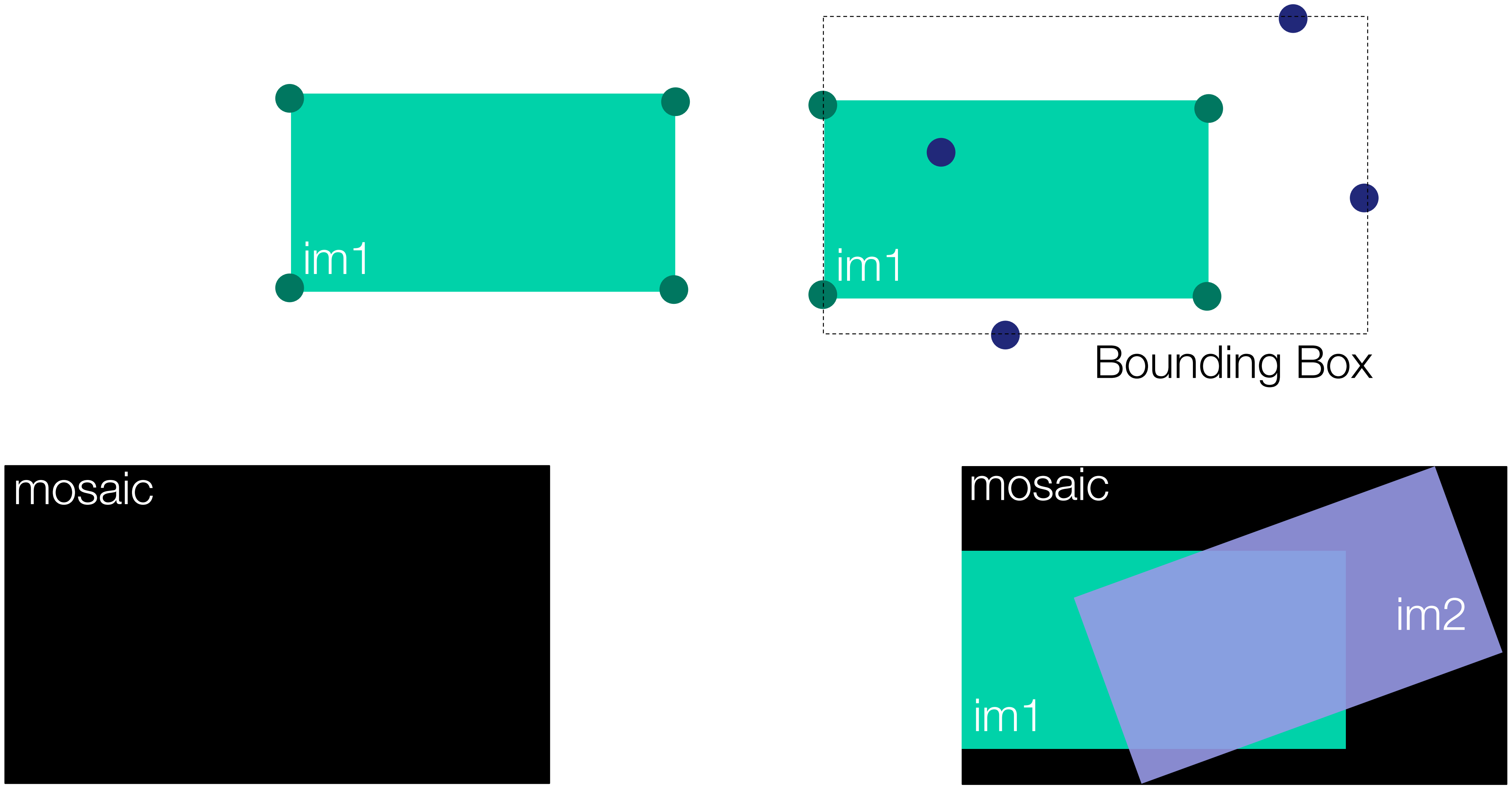
Transform between images



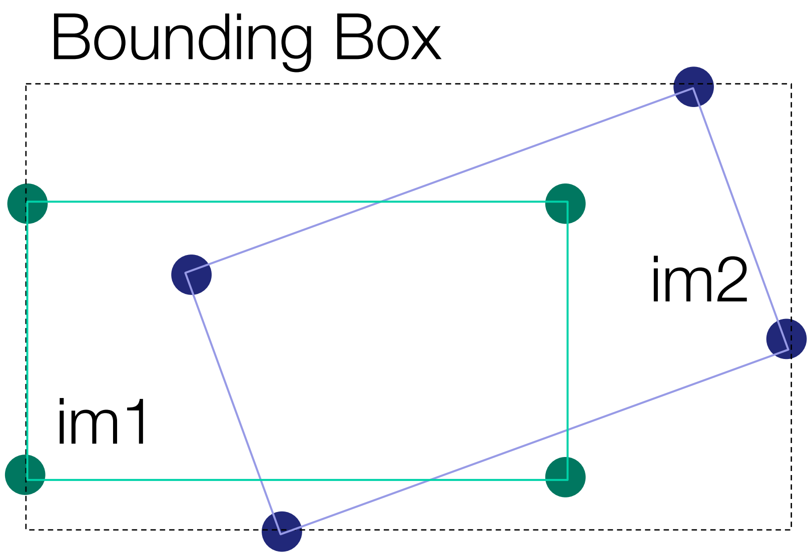
$$p' = H.p \Rightarrow p = H^{-1}p'$$



Building a Mosaic



Filling the Bounding Box



Determine coordinates of bounding box in im1 frame

Get samples from im1, possibly with interpolation

Apply homography to mosaic coordinates to go to im2 frame

Get samples from im2, possibly with interpolation

Overlay samples from im1 and im2

Can use `meshgrid` Matlab function to create mosaic coordinates
Can use `interp2` Matlab function to get sub-pixel values
Watch out for `NaN`

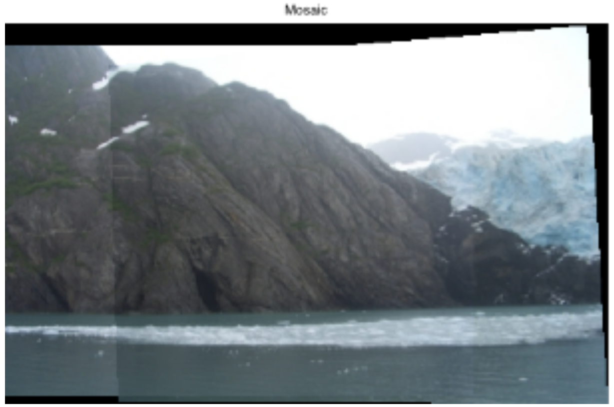
Homography Example: 3D Pavement Drawings

http://www.julianbeever.net/

Panoramic Imaging Pipeline

Estimating Correspondences

Warping



3. Compute transformation between im2 and im1

4. Transform im2 to overlap with im1

Blending

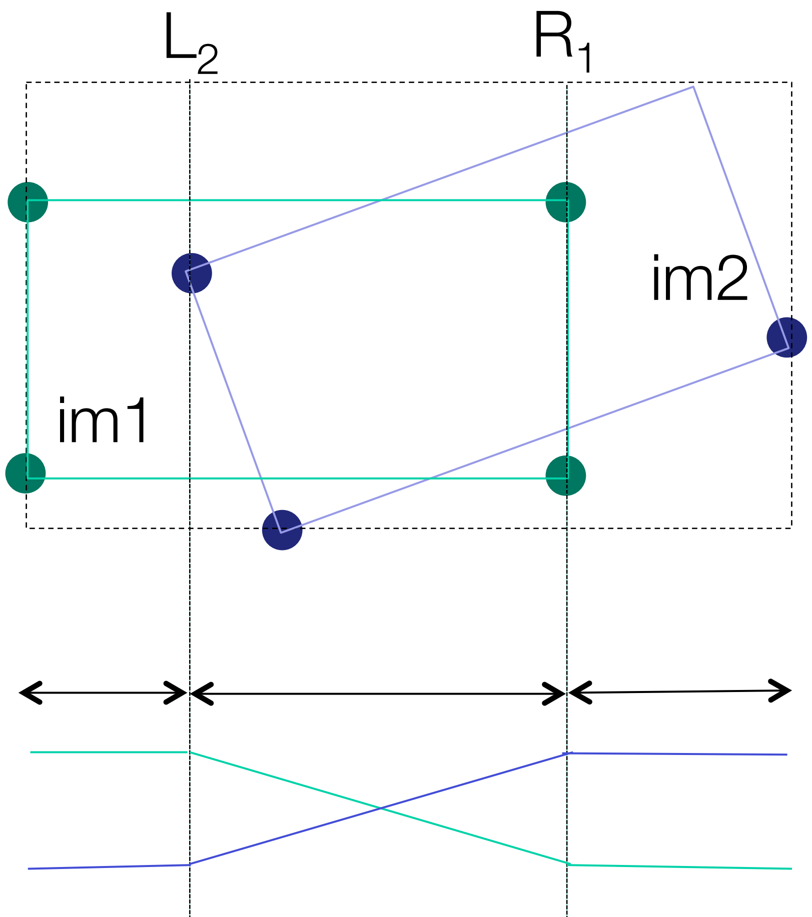
5. Blend the images to create a smooth transition

6. Repeat for other images

Seams during Stitching

Image: Durand

Simple Linear Blending



Set weight=1 if
only one im available

Potential Artifacts

Ghosting
Blurring

Image: Durand

Two-scale Blending

Idea: Combine smooth and abrupt transition

Split image into low and high frequency

Smooth blending for low frequencies: corrects exposure differences

Abrupt blending for high frequencies: avoids ghosting, preserves sharpness

Two-scale Blending Result

Other methods:

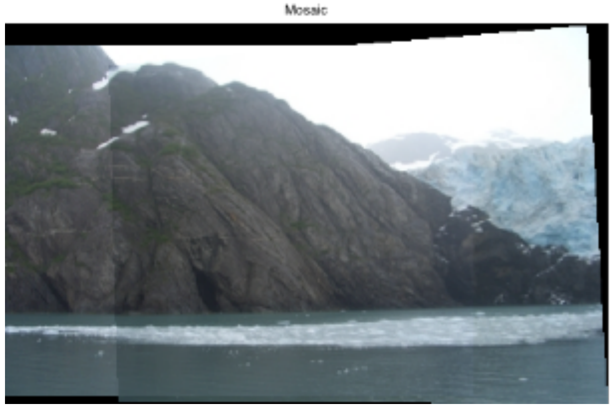
- Pyramid based
- Gradient domain

Image: Durand

Panoramic Imaging Pipeline

Estimating Correspondences

Warping



3. Compute transformation between im2 and im1
4. Transform im2 to overlap with im1

Blending

5. Blend the images to create a smooth transition
6. Repeat for other images

Example: Stitching 3 Images

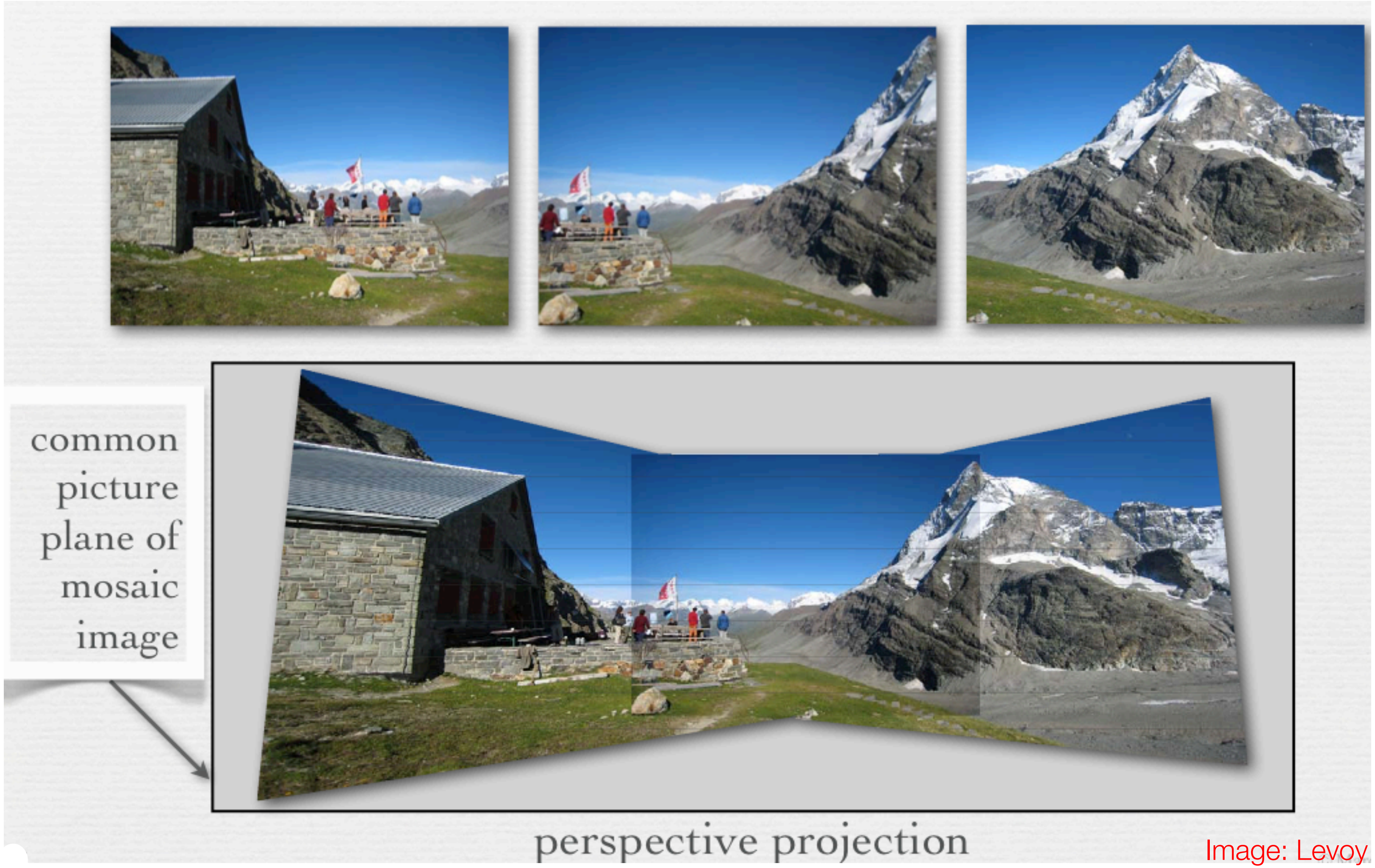


Image: Levoy

Example: Stitching 4 Images

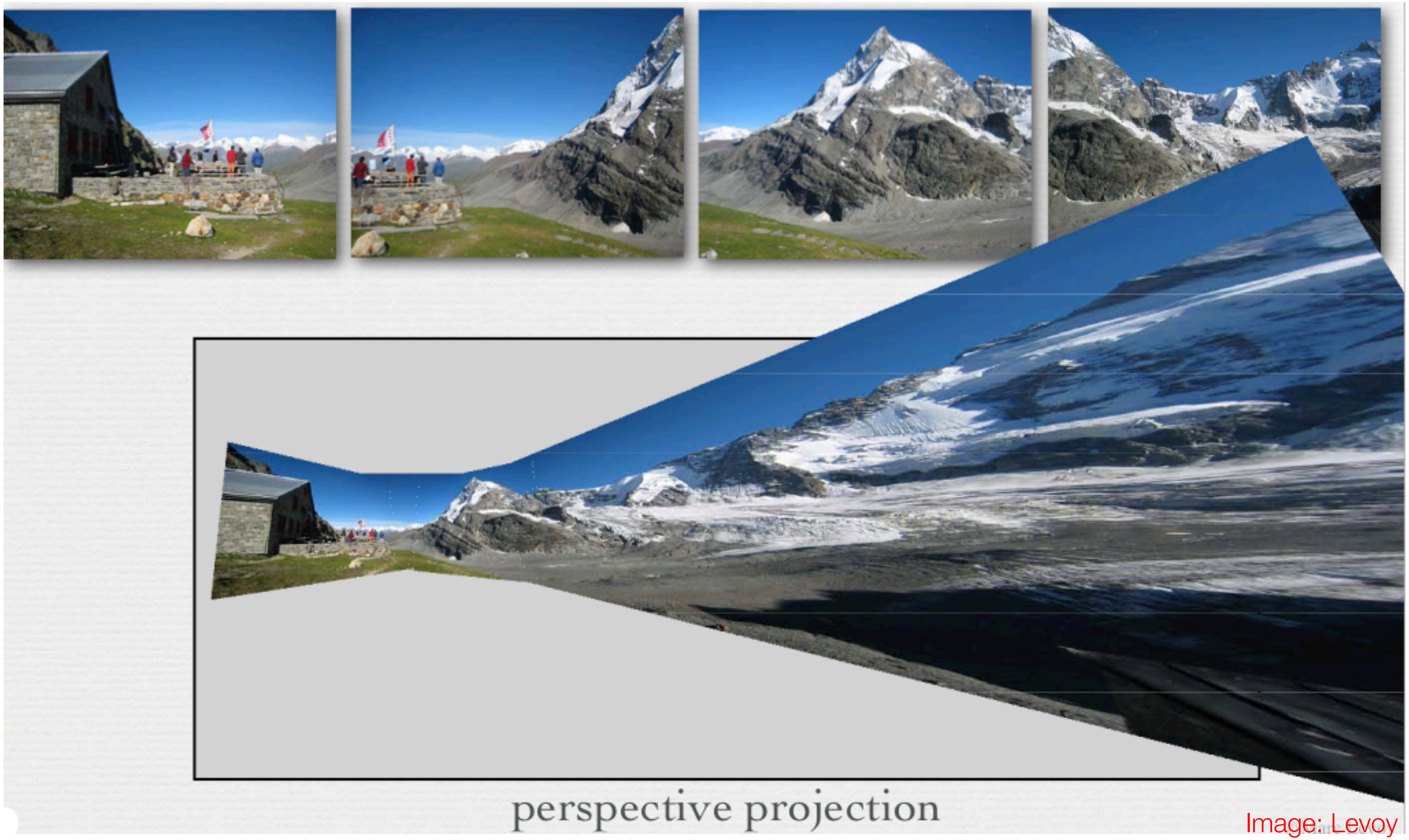


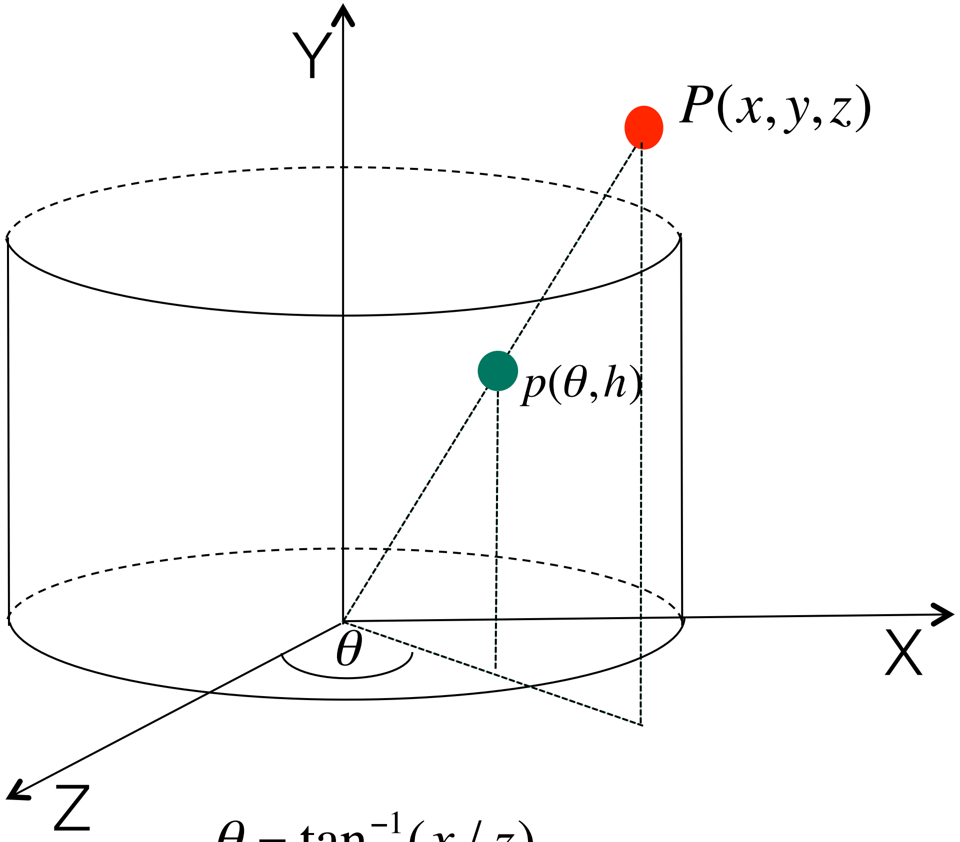
Image: Levoy

How to avoid this stretching?

Don't restrict image surface to a plane

Cylindrical Panorama

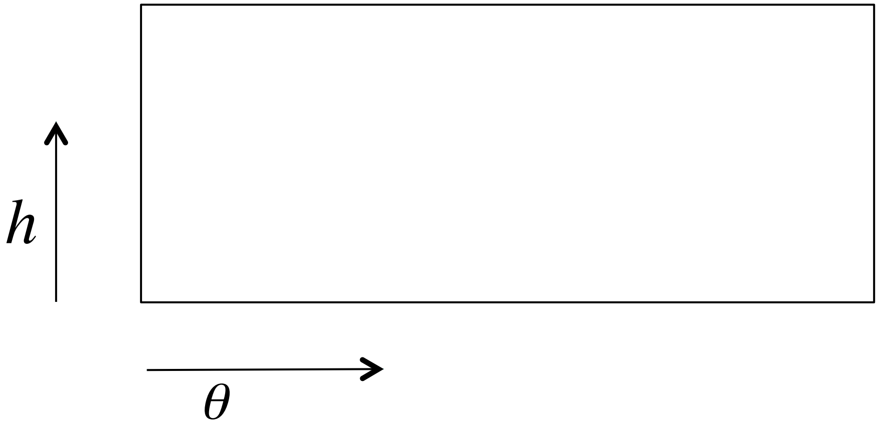
Assume cylinder of unit radius



$$\theta = \tan^{-1}(x/z)$$

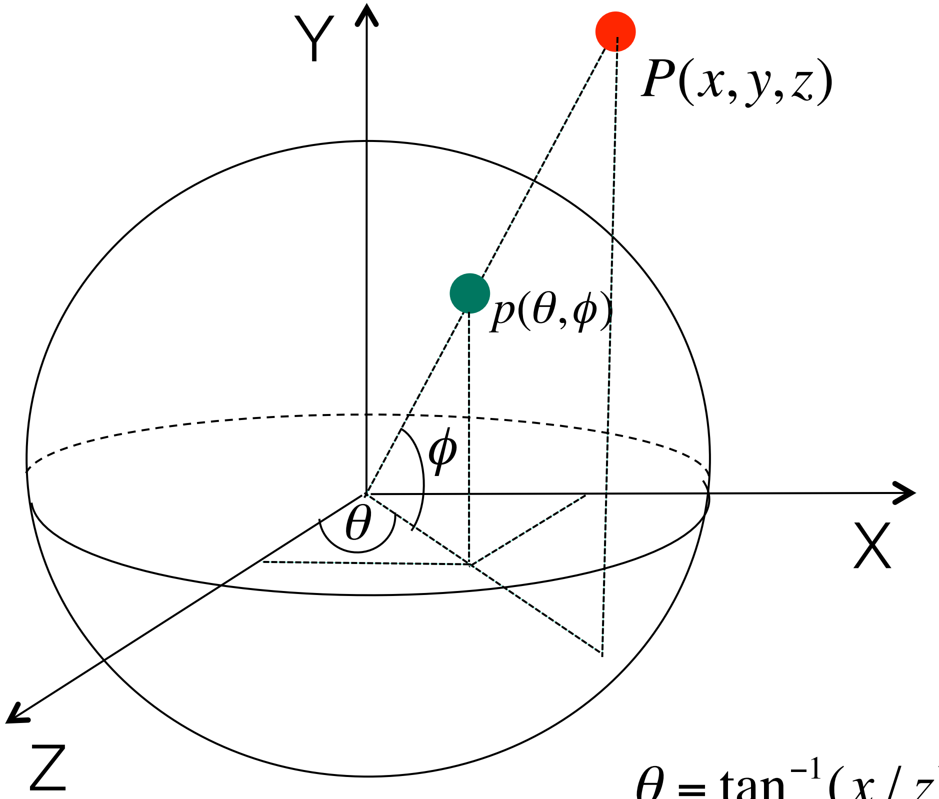
$$h = y/\sqrt{x^2 + z^2}$$

Unwraped Cylinder



Spherical Panorama

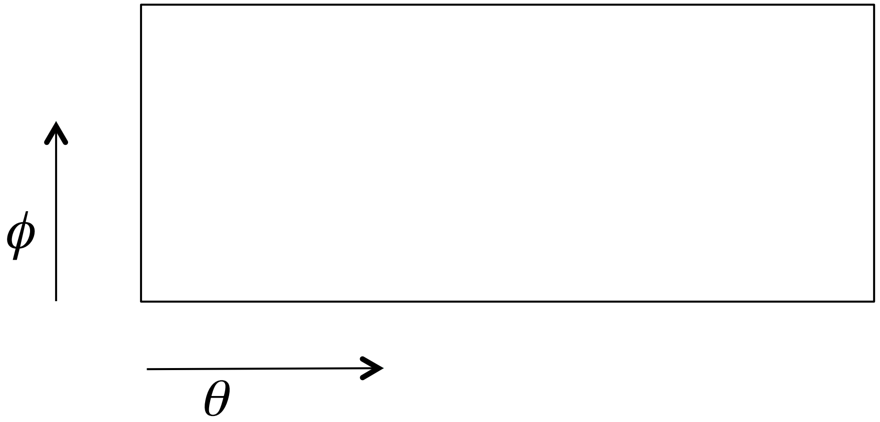
Assume sphere of unit radius



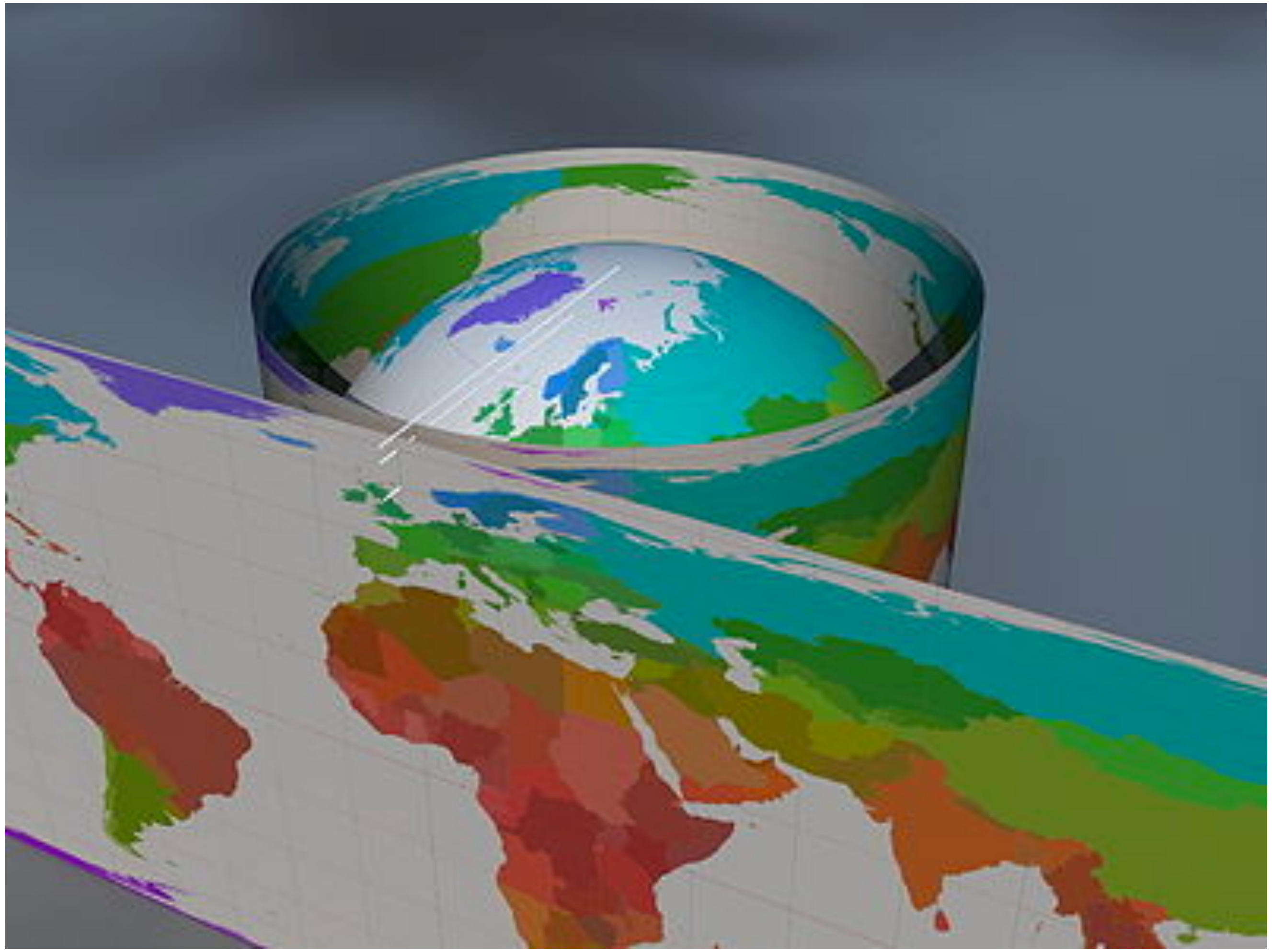
$$\theta = \tan^{-1}(x/z)$$

$$\phi = \tan^{-1}(y/\sqrt{x^2+z^2})$$

Unwraped Sphere



Sphere-to-plane mapping



Popular Mappings:

- Equirectangular
- Equal-area projection
- Mercator
- Cube maps
- …

Example of Spherical Panorama

Constructed from 54 photographs
[Image Alignment and Stitching: A Tutorial, by Szeliski, 2006]

Demo

https://graphics.stanford.edu/courses/cs178/applets/projection.html

youtube.com/360