Topic 13:

Homographies & Image Mosaics

- Introduction to image mosaicing
- Homogeneous coordinates for points & lines
- Image homographies
- Estimating homographies from point correspondences
- The autostitch algorithm

Building Panoramic Image Mosaics

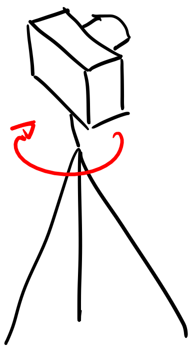
Input images

⇓ automatically created mosaic

Image Mosaicing

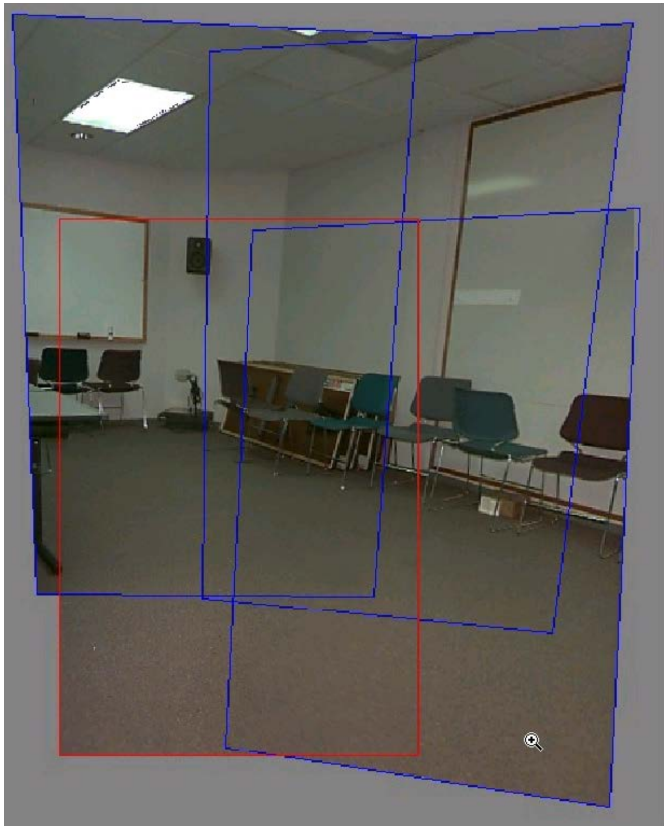
Technique:

① Take multiple photos while rotating camera on a tripod (or by hand)



② Warp & align the photos

③ Blend photos to compute final mosaic



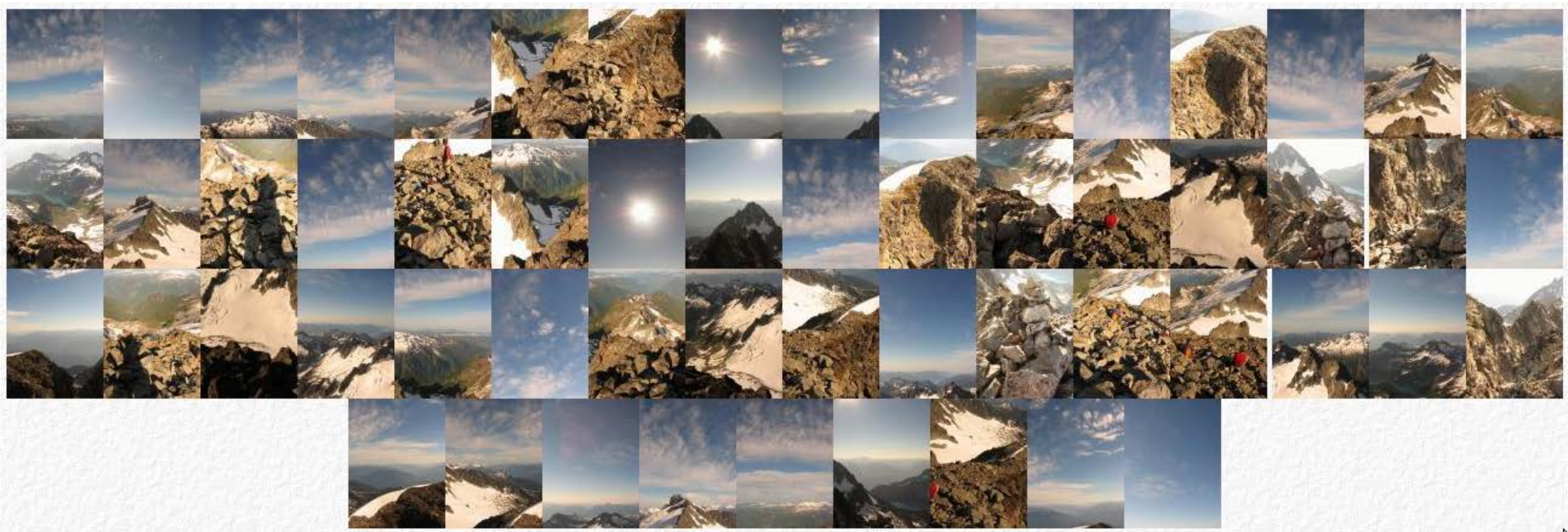
* In general, photos must be <u>warped</u> to align their contents!

Step 1: Capture

Important:

- Camera should change orientation, not position
- Keep camera settings (gain, focus, speed, aperture) fixed if possible

Step 2: Warp & Align



⇓ 28/57 images aligned

Step 2: Warp & Align (Continued)

⇑ 57/57 images aligned

Step 3: Blend

Laplacian Pyramid Blending ⇓ seams not visible anymore

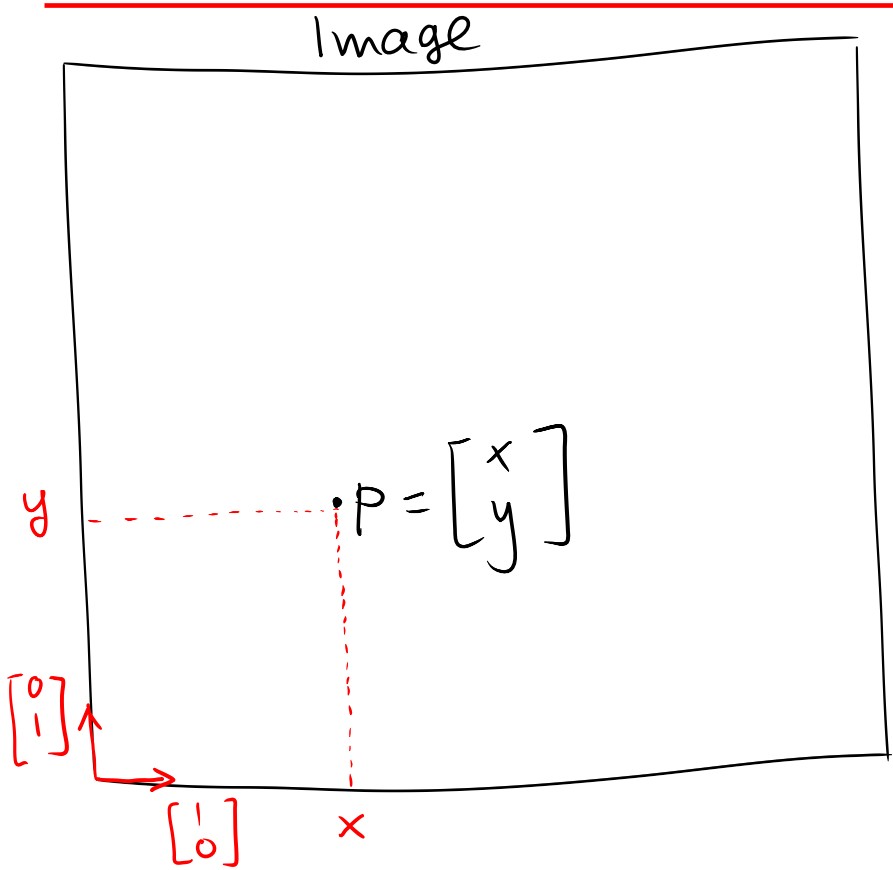
(Brown & Lowe ; ICCV 2003) google " Lowe Brown Autostitch "

Topic 12:

Homographies & Image Mosaics

- Introduction to image mosaicing
- Homogeneous coordinates for points & lines
- Image homographies
- Estimating homographies from point correspondences
- The autostitch algorithm

Representing Pixels by Euclidean 2D Coordinates



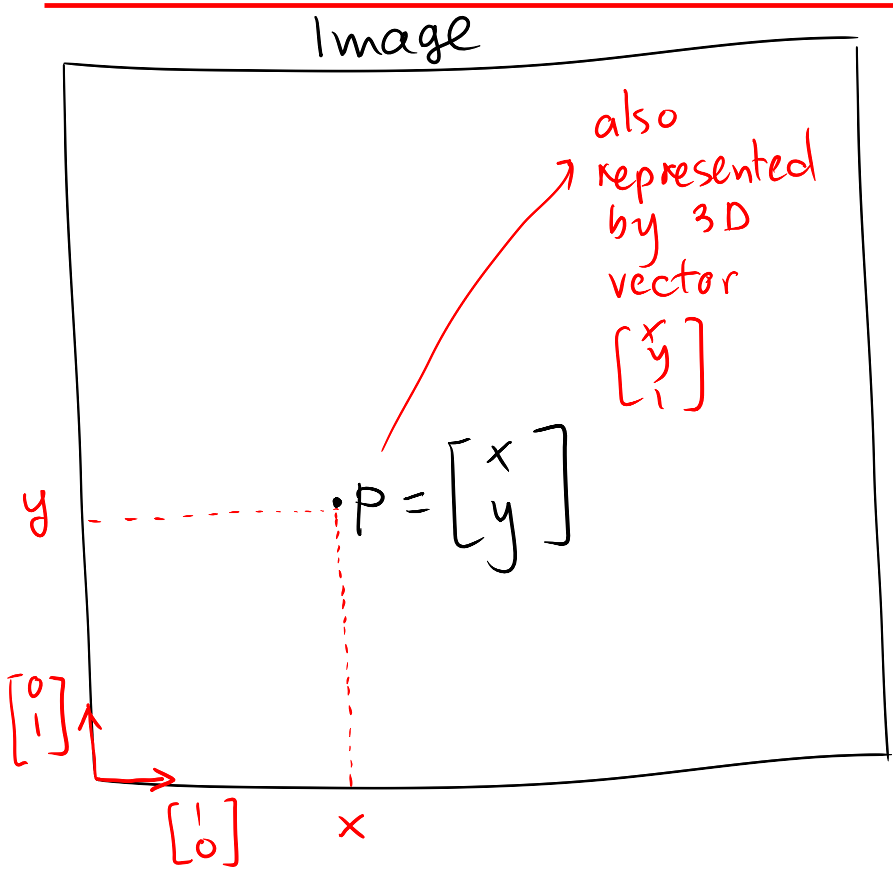
- "Standard" (Euclidean) representation of an image point p:

$$p = x \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + y \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

basis vectors

Euclidean coordinates

Euclidean Coordinates ⇒ Homogeneous Coordinates



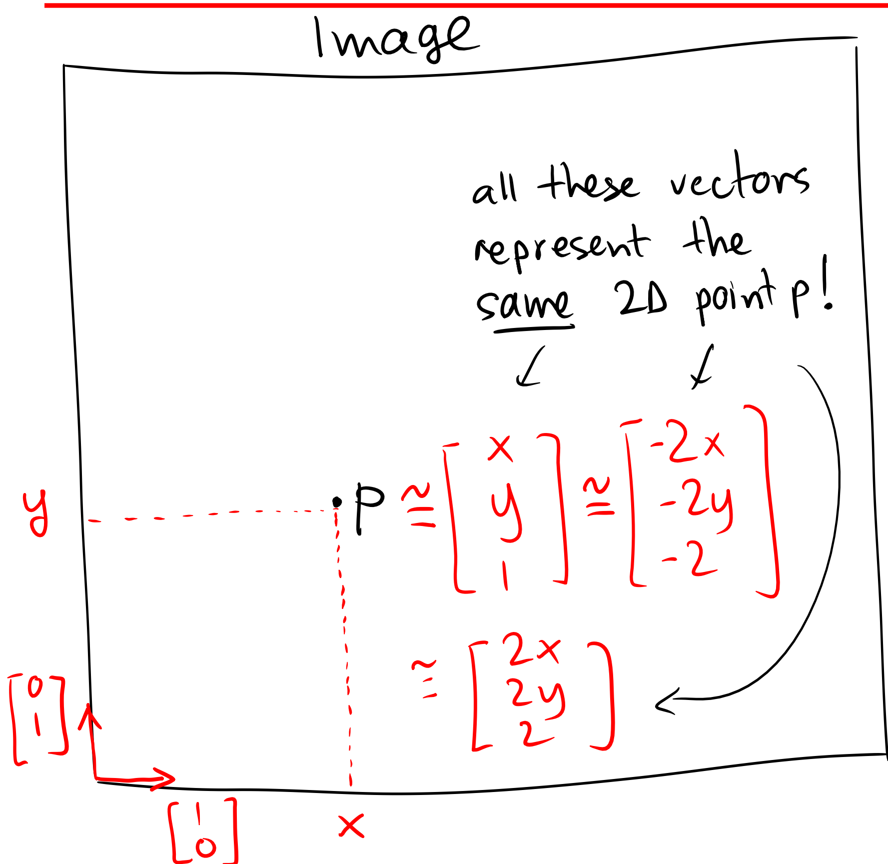
- "Standard" (Euclidean) representation of an image point p:

$$p = x \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + y \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

- Homogeneous (a.k.a. Projective) representation of p

image coordinates		homogeneous 2D coordinates
$\begin{bmatrix} x \\ y \end{bmatrix}$	$\longrightarrow$	$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

2D Homogeneous Coordinates: Definition



- For <u>any</u> $\lambda \neq 0$, the numbers $\lambda x, \lambda y, \lambda$ are called <u>homogeneous coordinates</u> of point p

Definition:

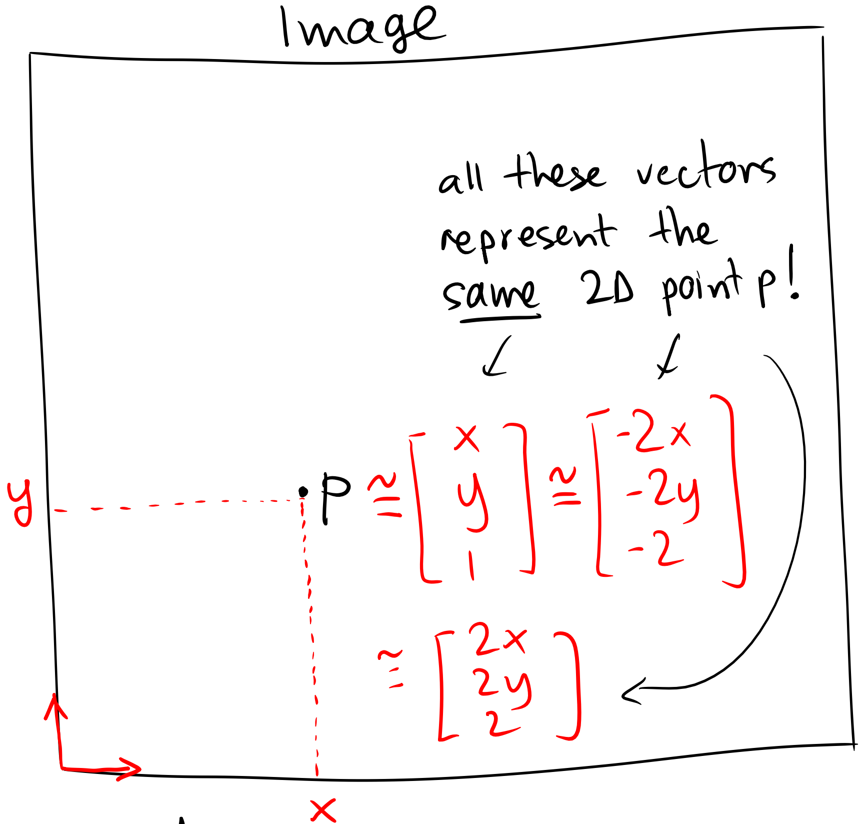
Homogeneous representation of p

$\boxed{\text{p represented by } \underline{\text{any}} \text{ 3D vector } \begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} \text{ with } \lambda \neq 0}$

- Homogeneous (a.k.a. Projective) representation of p

image coordinates		homogeneous 2D coordinates	
$\begin{bmatrix} x \\ y \end{bmatrix}$	$\longrightarrow$	$\begin{bmatrix} \lambda \cdot x \\ \lambda \cdot y \\ \lambda \cdot 1 \end{bmatrix}$	$\lambda \neq 0$

2D Homogeneous Coordinates: Equality



Definition (Homogeneous Equality)

Two vectors of homogeneous coords $V_1 = \begin{bmatrix} x \\ y \\ w \end{bmatrix}$ and $V_2 = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix}$ are called equal if they represent the same 2D point:

$$V_1 \cong V_2$$

denotes homog. equality

$$\Longleftrightarrow$$

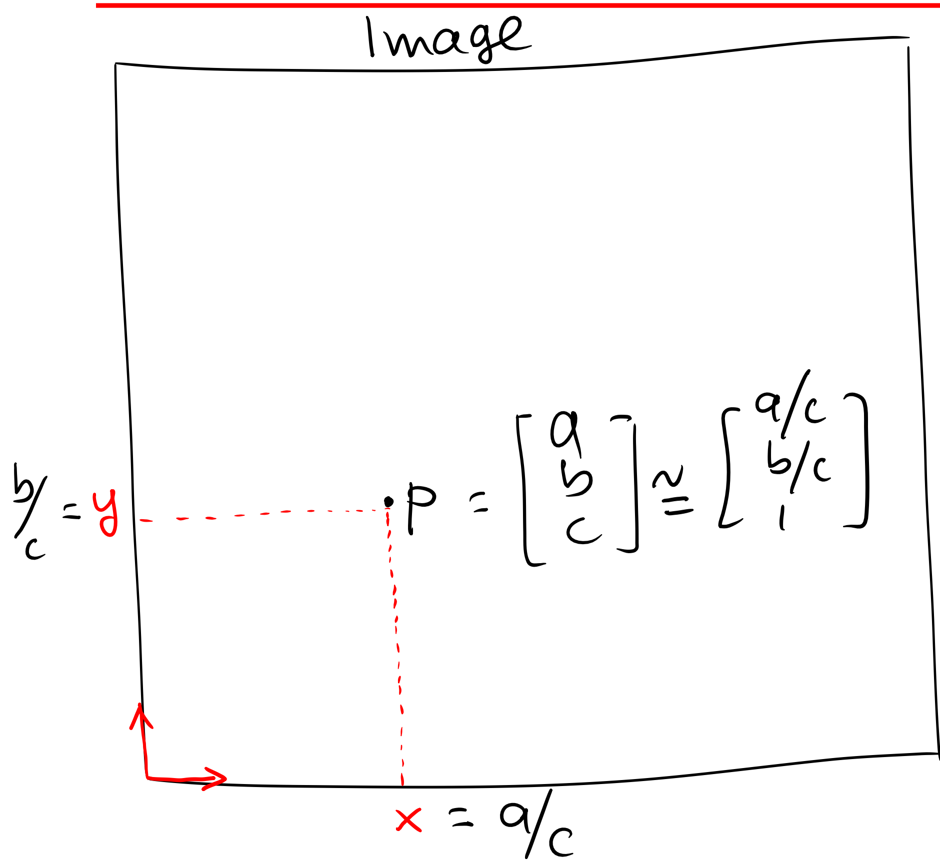
there is a $\lambda \neq 0$ such that

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \lambda \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix}$$

Examples:

- Is $\begin{bmatrix} 3 \\ 4 \\ 6 \end{bmatrix} \cong \begin{bmatrix} 6 \\ 8 \\ 12 \end{bmatrix}$? yes (take $\lambda = 2$)
- Is $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cong \begin{bmatrix} 0 \\ 0 \\ 30 \end{bmatrix}$? yes (take $\lambda = 30$)
- Is $\begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \cong \begin{bmatrix} -2 \\ 0 \\ 4 \end{bmatrix}$? no!

Homogeneous Coordinates $\Rightarrow$ Euclidean Coordinates



Converting from homogeneous to Euclidean coordinates:

$\begin{bmatrix}a\\b\\c\end{bmatrix}, \begin{bmatrix}a/c\\b/c\\1\end{bmatrix}$ represent the same 2D point

$\iff$ 2D coordinates are $\begin{bmatrix}a/c\\b/c\end{bmatrix}$

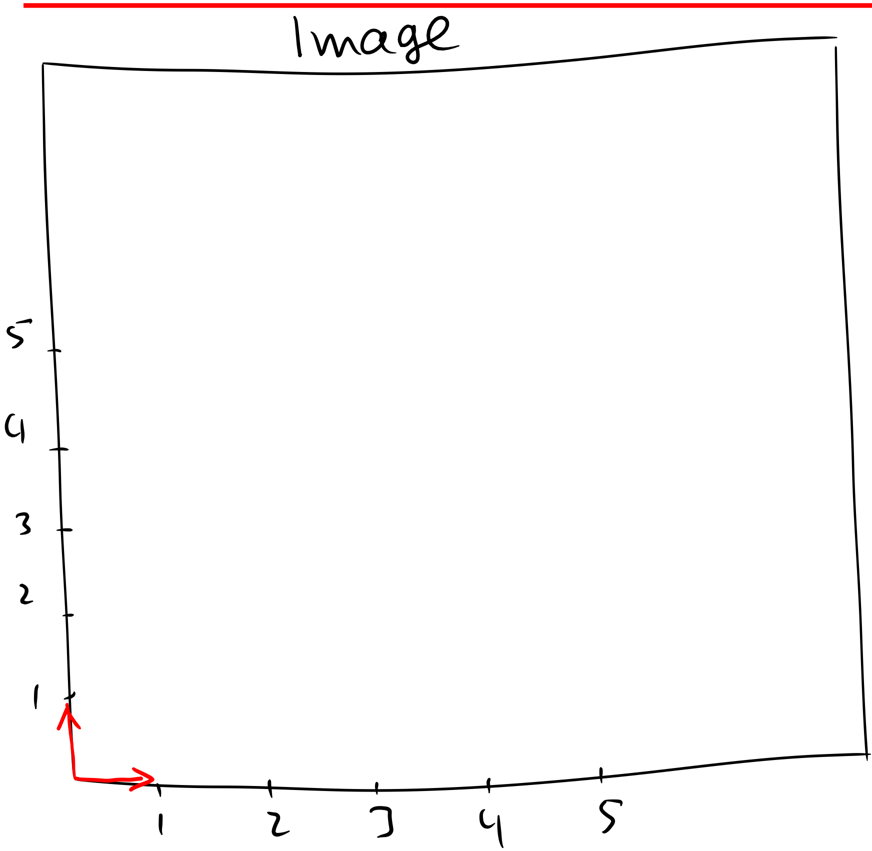
$$v_1 \cong v_2$$

$$\iff$$

there is a $\lambda \neq 0$ such that

$$\begin{bmatrix}x\\y\\w\end{bmatrix} = \lambda \begin{bmatrix}x'\\y'\\w'\end{bmatrix}$$

Homogeneous Coordinates $\Rightarrow$ Euclidean Coordinates



Converting from homogeneous to Euclidean coordinates:

$\begin{bmatrix} a \\ b \\ c \end{bmatrix}, \begin{bmatrix} a/c \\ b/c \\ 1 \end{bmatrix}$ represent the same 2D point

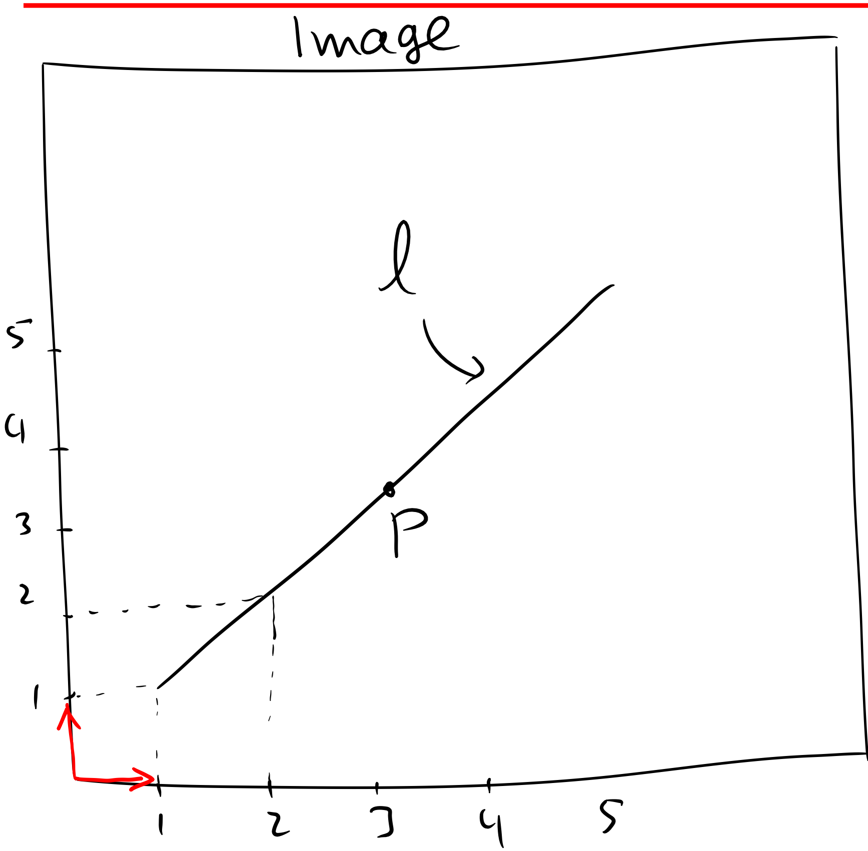
$\Leftrightarrow$ 2D coordinates are $\begin{bmatrix} a/c \\ b/c \end{bmatrix}$

Practice exercise: Plot positions of the following points

$P_6 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ $P_7 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$

$P_1 = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$ $P_2 = \begin{bmatrix} 10 \\ 0 \\ 2 \end{bmatrix}$ $P_3 = \begin{bmatrix} 0 \\ 8 \\ 4 \end{bmatrix}$ $P_4 = \begin{bmatrix} 1 \\ 0 \\ 0.0001 \end{bmatrix}$ $P_5 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

Line Equations in Homogeneous Coordinates



- The equation of a line

$$\underbrace{a}\,x+\underbrace{b}\,y+\underbrace{c}=0$$

line parameters

- In homogeneous coordinates

$$\begin{bmatrix} a & b & c \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}=0$$

or $\ell^T \cdot p = 0$

vector holding line parameters

vector holding homogeneous coordinates of a point

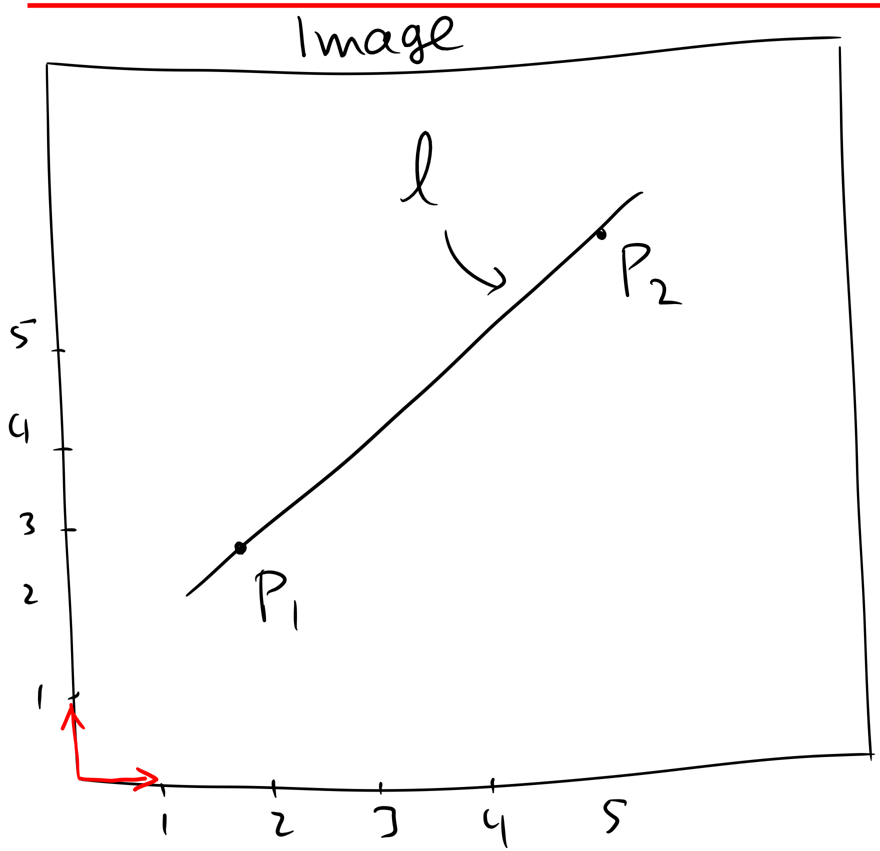
Example: line $y=x$ in homogeneous coords:

$$1 \cdot x - 1 \cdot y + 0 \cdot 1 = 0$$

line parameters of ℓ

$$\begin{bmatrix} 1 & -1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}=0$$

The Line Passing Through 2 Points



- Calculating the parameters of a line through two points with homogeneous coordinates p_1, p_2

$$\boxed{l = p_1 \times p_2}$$

cross product of two 3D vectors

- In homogeneous coordinates

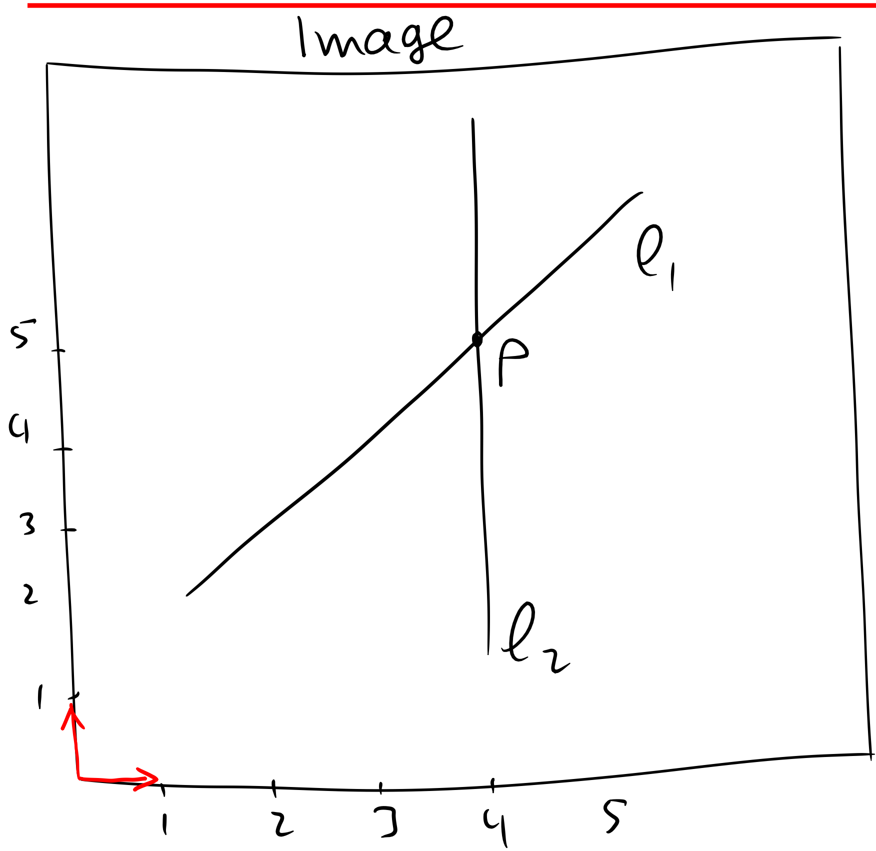
$$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

or $l^T \cdot p = 0$

- l must satisfy $l^T \cdot p_1 = 0$, $l^T \cdot p_2 = 0$
- taken as 3D vectors, l is perpendicular to both p_1 and p_2

$\Rightarrow$ it is along the cross product, $p_1 \times p_2$

The Point of Intersection of Two Lines



- Calculating the homogeneous coordinates of the intersection of two lines l_1, l_2

$$\boxed{p = l_1 \times l_2}$$

↑ cross product of two 3D vectors

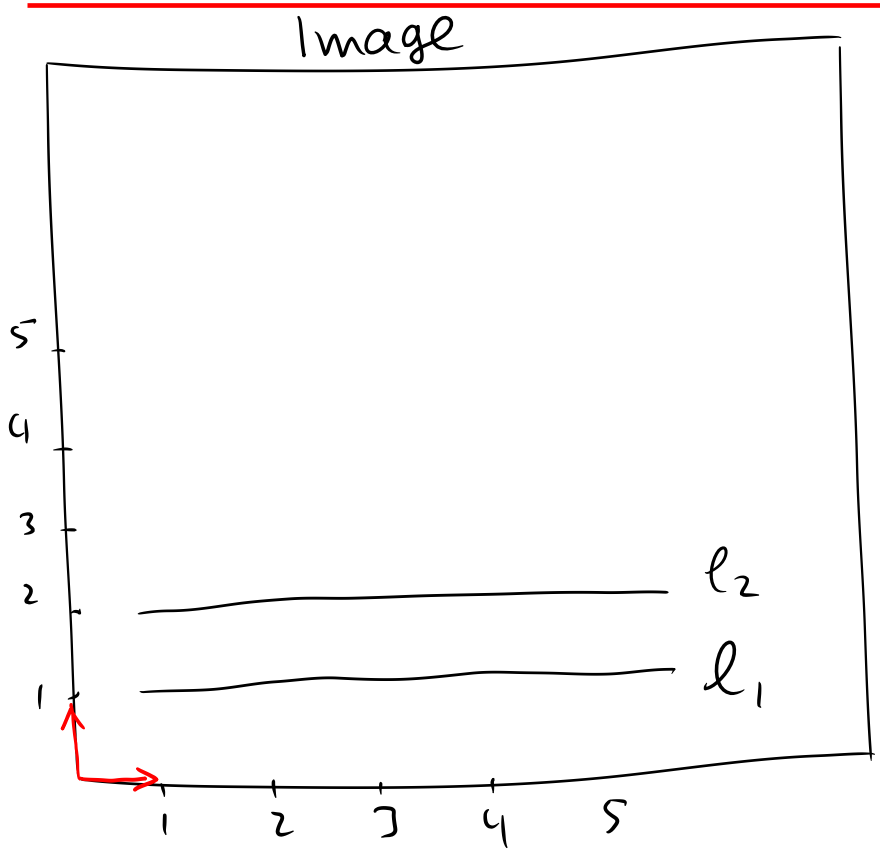
- p must satisfy $l_1^T p = 0$, $l_2^T p = 0$
- taken as 3D vectors, p is perpendicular to both l_1 and l_2
  $\Rightarrow$ it is along the cross product, $l_1 \times l_2$

- In homogeneous coordinates

$$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

or $l^T \cdot p = 0$

Computing the Intersection of Parallel Lines



Calculating the homogeneous coordinates of the intersection of two lines ℓ_1, ℓ_2

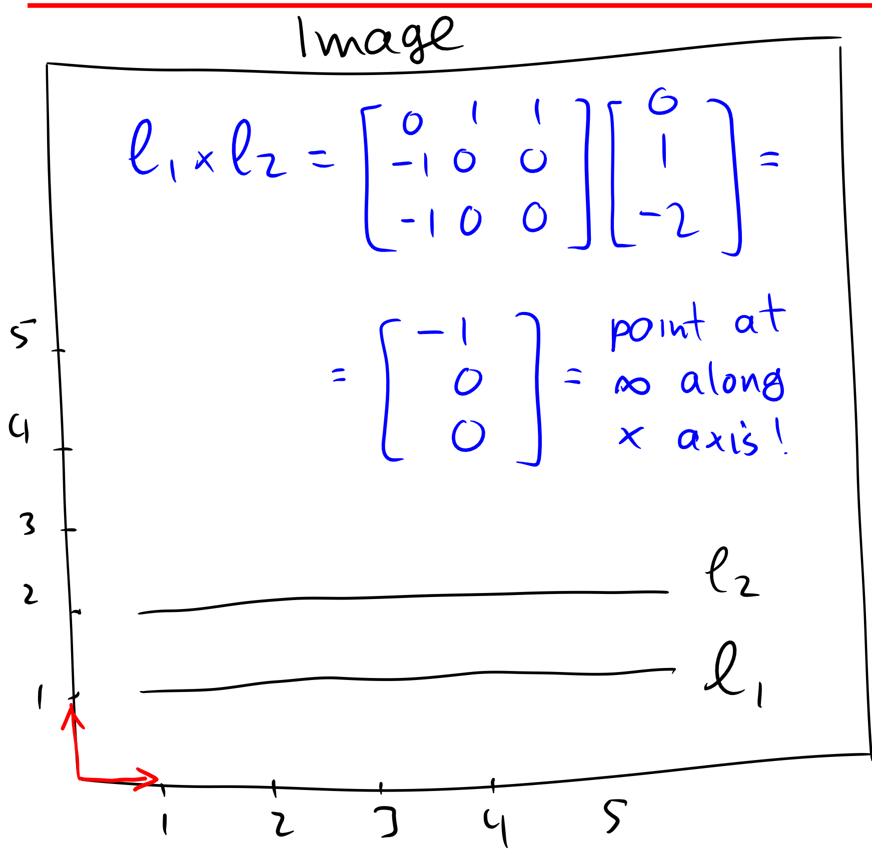
$$\boxed{p = \ell_1 \times \ell_2}$$

cross product of two 3D vectors

This calculation works even when ℓ_1, ℓ_2 are parallel!

(no floating point exceptions or divide-by-zero errors!)

Computing the Intersection of Parallel Lines



$$\ell_1 \times \ell_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ -2 \end{bmatrix} =$$

$$= \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} = \text{point at } \infty \text{ along } x \text{ axis!}$$

- Calculating the homogeneous coordinates of the intersection of two lines ℓ_1, ℓ_2

$$\boxed{p = \ell_1 \times \ell_2}$$

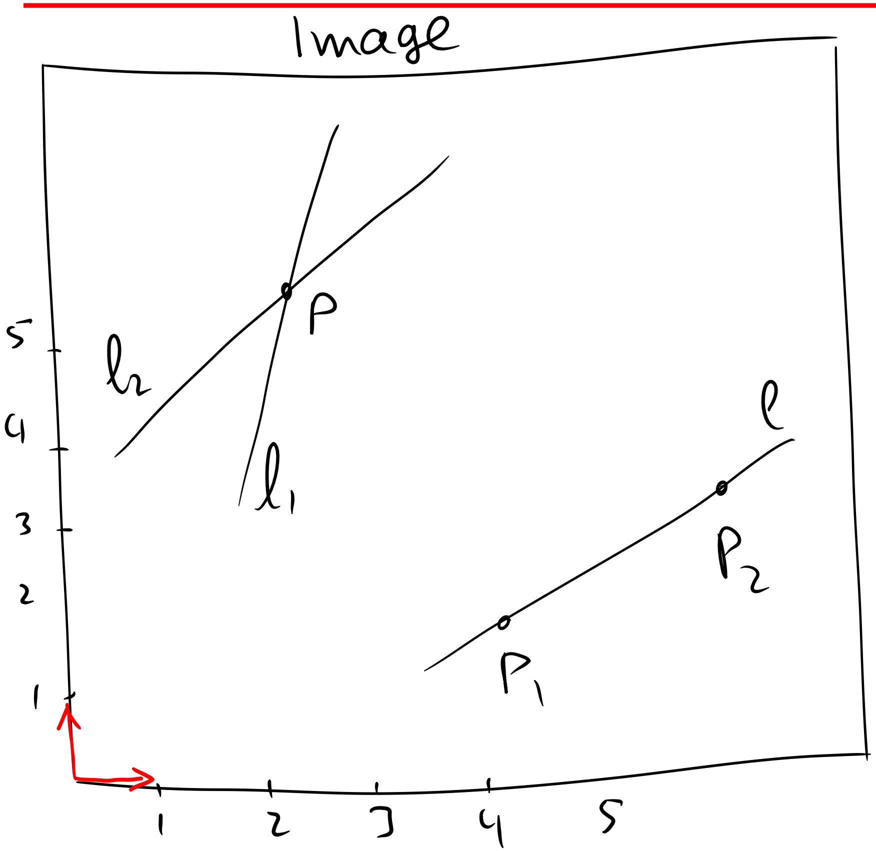
cross product of two 3D vectors

- Line eq. of ℓ_1 is $y=1$. Also written as $0 \cdot x + 1 \cdot y - 1 = 0$. So $\ell_1 = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$
- Similarly $\ell_2 = \begin{bmatrix} 0 \\ 1 \\ -2 \end{bmatrix}$

Aside (calculating cross products): If $\ell_1 = (a, b, c)$ then

$$\ell_1 \times \ell_2 = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix} \ell_2$$

Lines from Points & Points from Lines



Useful property #2

- Very simple way of computing & intersecting lines
- Numerical stability even when result is at ∞

Line through 2 points

$$l = p_1 \times p_2 = \begin{bmatrix} 0 & -p_1^z & p_1^y \\ p_1^z & 0 & -p_1^x \\ -p_1^y & p_1^x & 0 \end{bmatrix} \begin{bmatrix} p_2^x \\ p_2^y \\ p_2^z \end{bmatrix}$$

Intersection of 2 lines

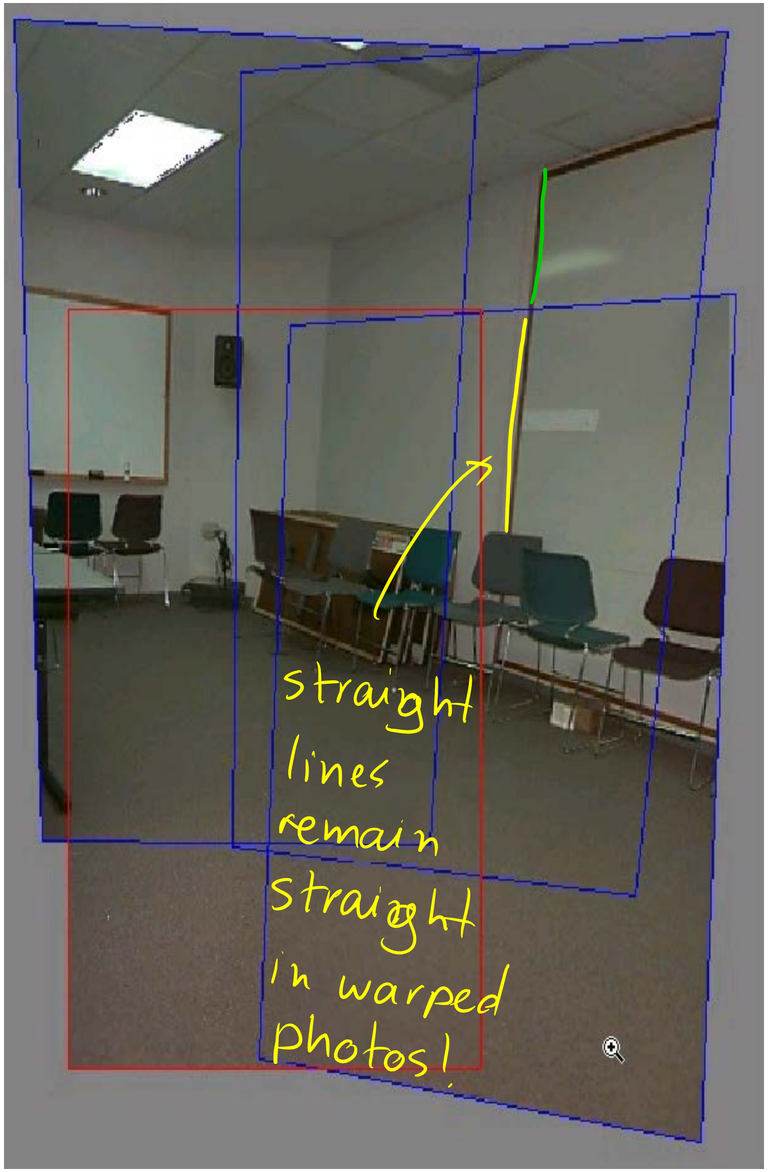
$$p = l_1 \times l_2 = \begin{bmatrix} 0 & -l_1^z & l_1^y \\ l_1^z & 0 & -l_1^x \\ -l_1^y & l_1^x & 0 \end{bmatrix} \begin{bmatrix} l_2^x \\ l_2^y \\ l_2^z \end{bmatrix}$$

Topic 12:

Homographies & Image Mosaics

- Introduction to image mosaicing
- Homogeneous coordinates for points & lines
- Image homographies
- Estimating homographies from point correspondences
- The autostitch algorithm

Linear Image Warps

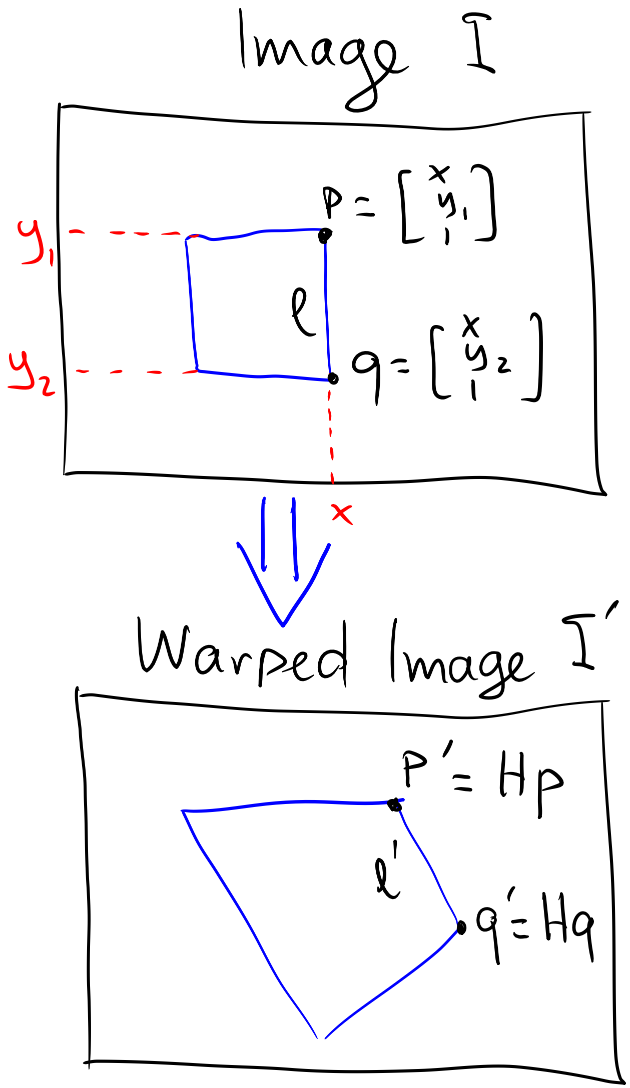


Basic insight:

To align multiple photos for mosaicing we must warp them in a way that preserves all lines

(i.e. lines before warping remain lines after warping)

Linear Image Warps & Homographies



The matrix H is called a **Homography**

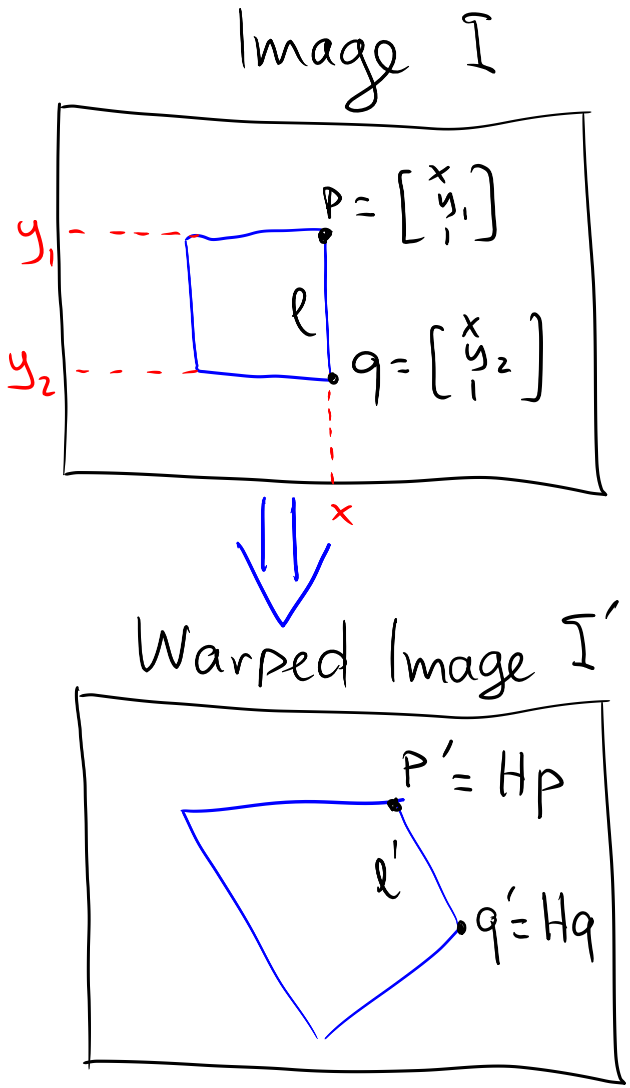
- **Definition (Linear Image Warps)**

  An image warp is called linear if every 2D line ℓ in the original image is transformed into a line ℓ' in the warped image (i.e. the warp preserves all lines in the original photo)

- **Property (w/out proof)**

  Every linear warp can be expressed as a 3×3 matrix H that transforms homogeneous image coordinates

Warping Images Using a Homography



The matrix H is called a Homography

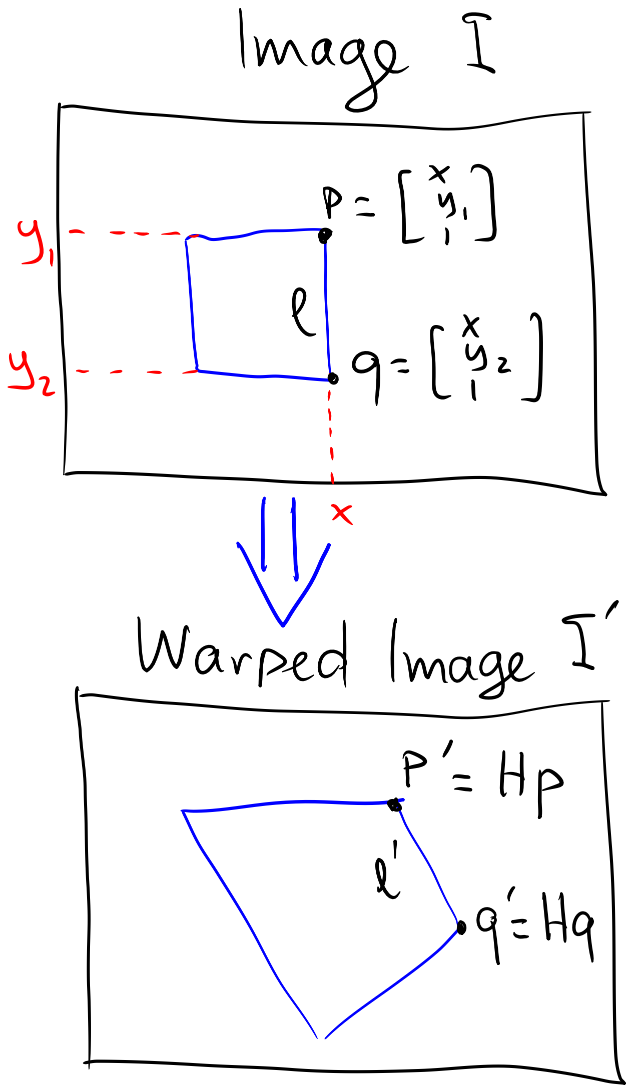
- Linear warping equation

$$\underbrace{I(p)}_{\text{intensity at pixel in source image with homogeneous coordinates } p} = \underbrace{I'(Hp)}_{\text{intensity at pixel in warped image with homogeneous coordinates } p' = Hp}$$

- Property (w/out proof)

Every linear warp can be expressed as a 3x3 matrix H that transforms homogeneous image coordinates

Warping Images Using a Homography



The matrix H is called a **Homography**

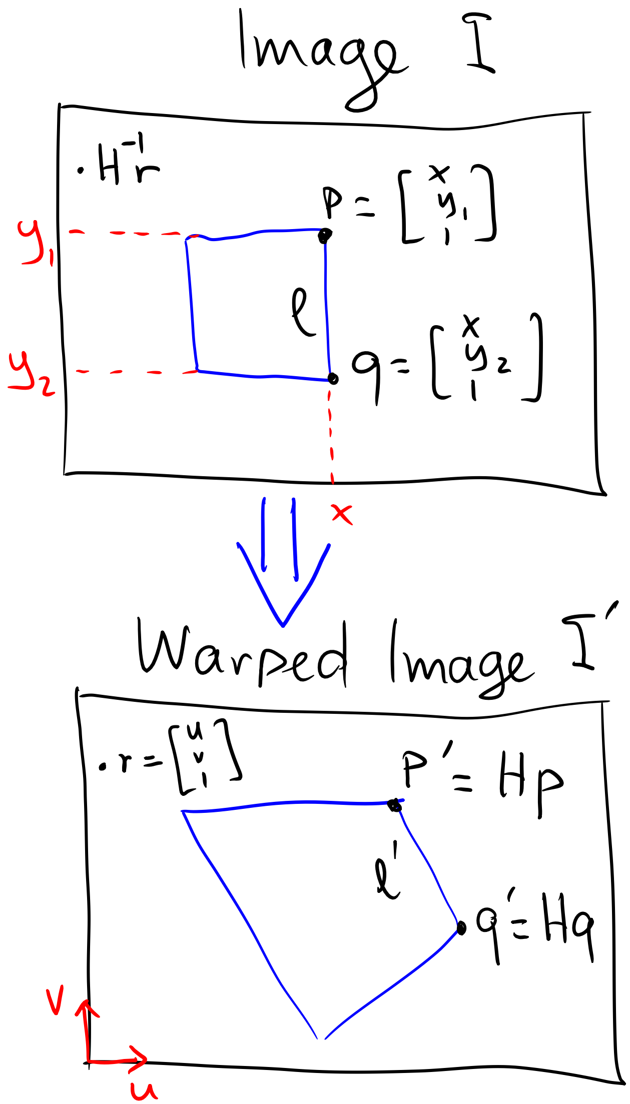
- **Linear warping equation**

$$\underbrace{I(p)}_{\substack{\text{intensity at pixel in source}\\ \text{image with homogeneous}\\ \text{coordinates } p}} = \underbrace{I'(Hp)}_{\substack{\text{intensity at pixel in warped}\\ \text{image with homogeneous}\\ \text{coordinates } p' = Hp}}$$

- Note: Scaling H by a factor $\lambda \neq 0$ does not change the homography:

$$(\lambda \cdot H)\,p = H \cdot (\lambda p) \cong Hp$$

Warping Images Using a Homography



The matrix H is called a Homography

- Linear warping equation

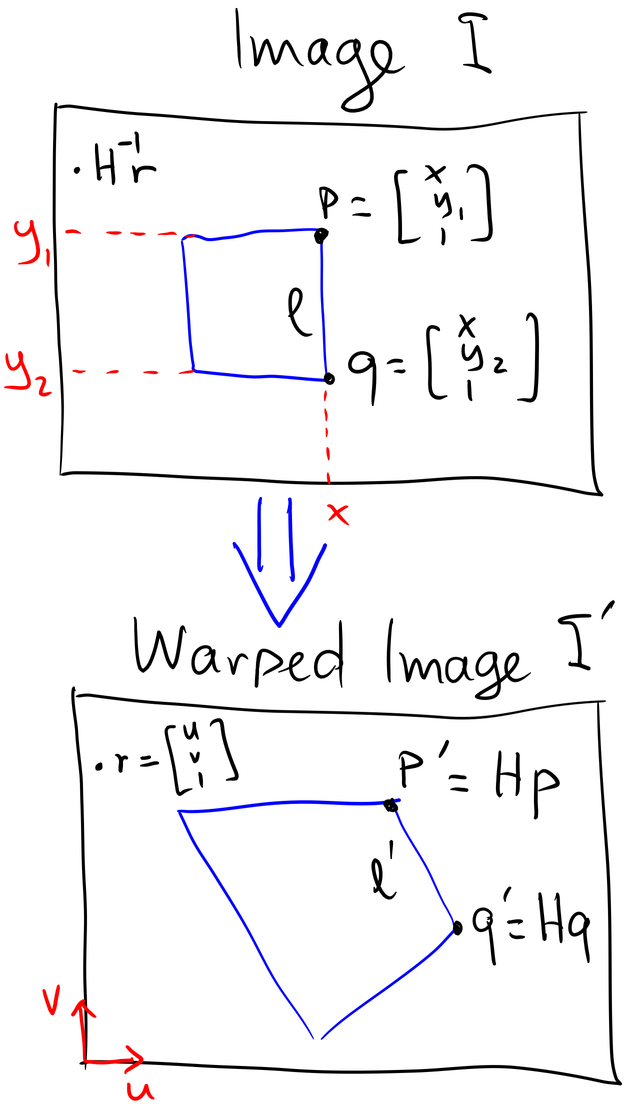
$$I(p) = I'(Hp)$$

$$I(H^{-1} r) = I'(r)$$

- Property (w/out proof)

Every linear warp can be expressed as a 3x3 matrix H that transforms homogeneous image coordinates

Warping Images Using a Homography



The matrix H is called a **Homography**

- Linear warping equation

$$I'\left(\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}\right) = I\left(H^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}\right)$$

$$I(H^{-1}r) = I'(r)$$

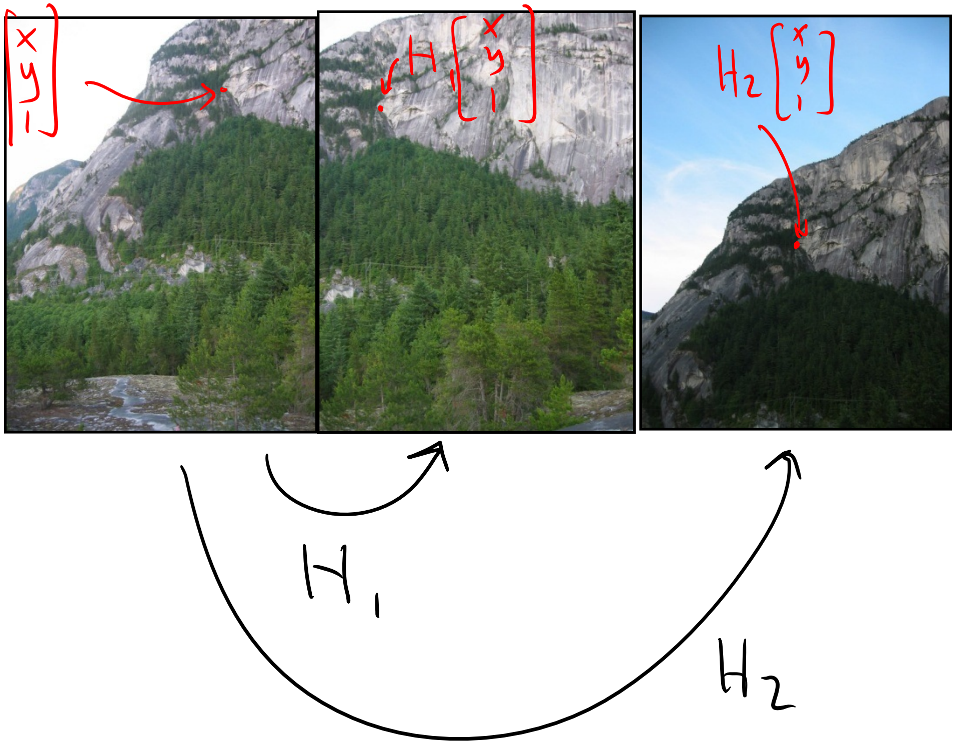
- Computing warp I' from I and H

① Compute H^{-1}

② To compute color of pixel (u,v) in warped image:

- compute $\begin{bmatrix} a \\ b \\ c \end{bmatrix} = H^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$
- copy color from $I(a/c, b/c)$

Homographies & Image Mosaicing

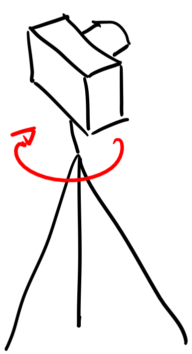


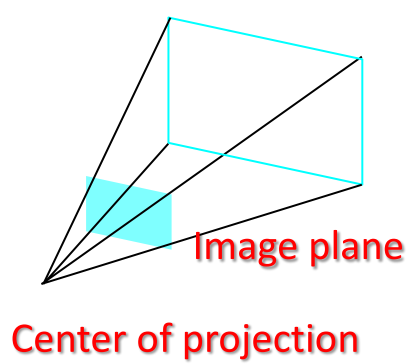
Useful property #3

- Every photo taken from a tripod-mounted camera is related by a homography

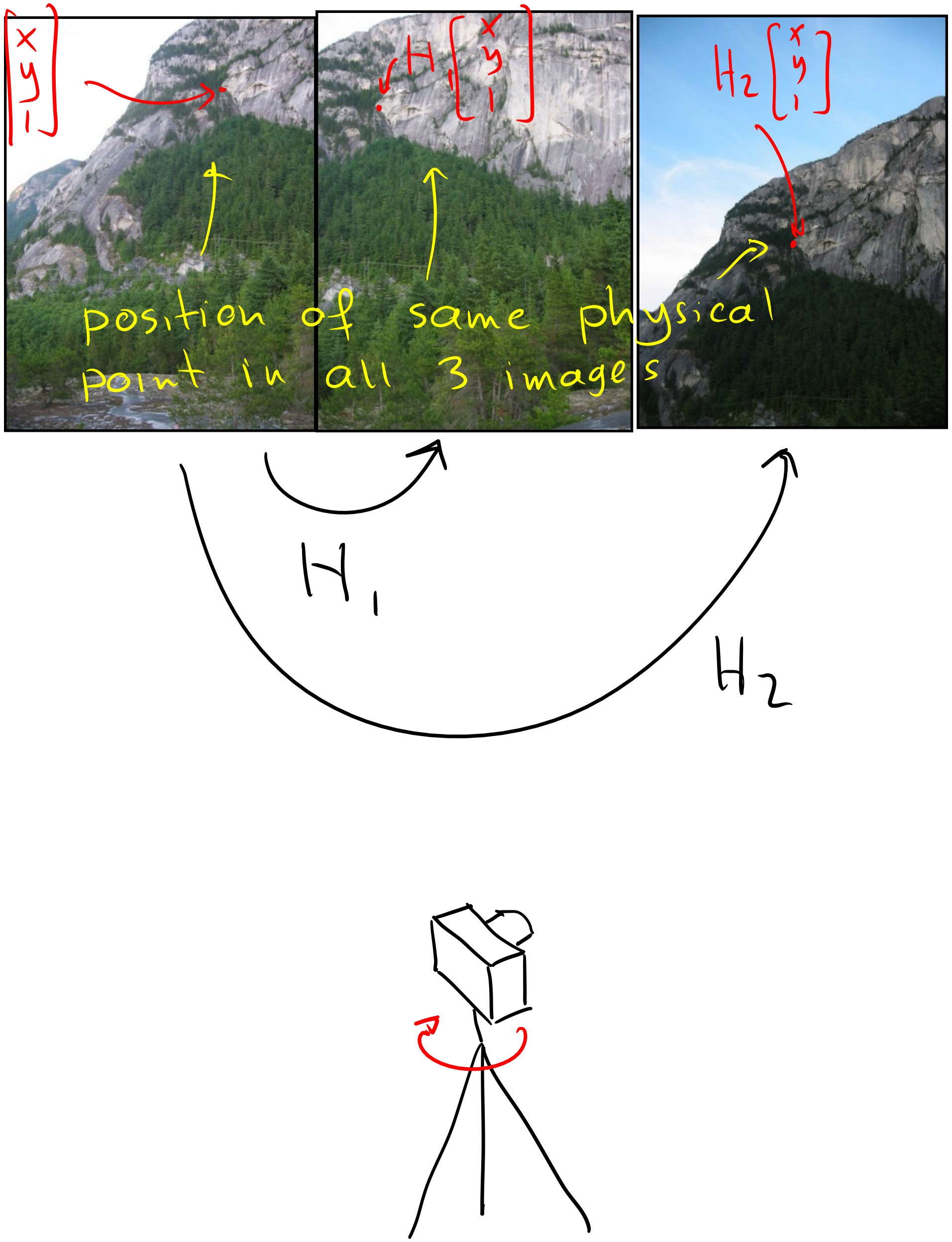
Assumptions:

- No lens distortions
- Camera's center of projection does not move while camera is mounted on tripod





Homographies & Image Mosaicing



Useful property #3

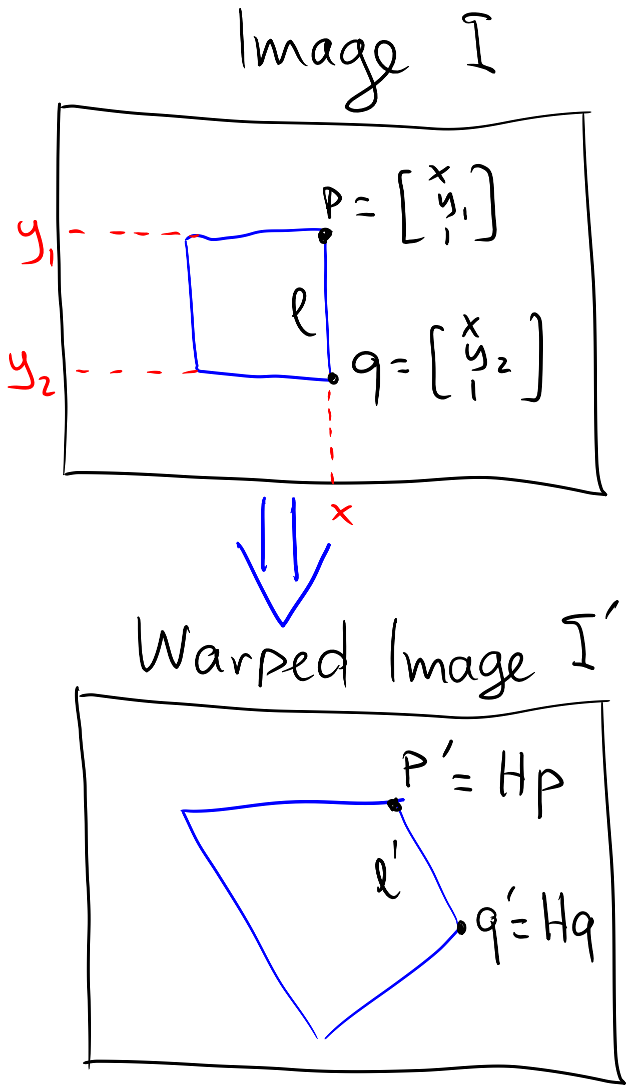
- Every photo taken from a tripod-mounted camera is related by a homography
- These homographies are <u>unknown</u>
- To align these photos for mosaicing we must estimate $H_1, H_2, \ldots$ etc

Topic 12:

Homographies & Image Mosaics

- Introduction to image mosaicing
- Homogeneous coordinates for points & lines
- Image homographies
- Estimating homographies from point correspondences
- The autostitch algorithm

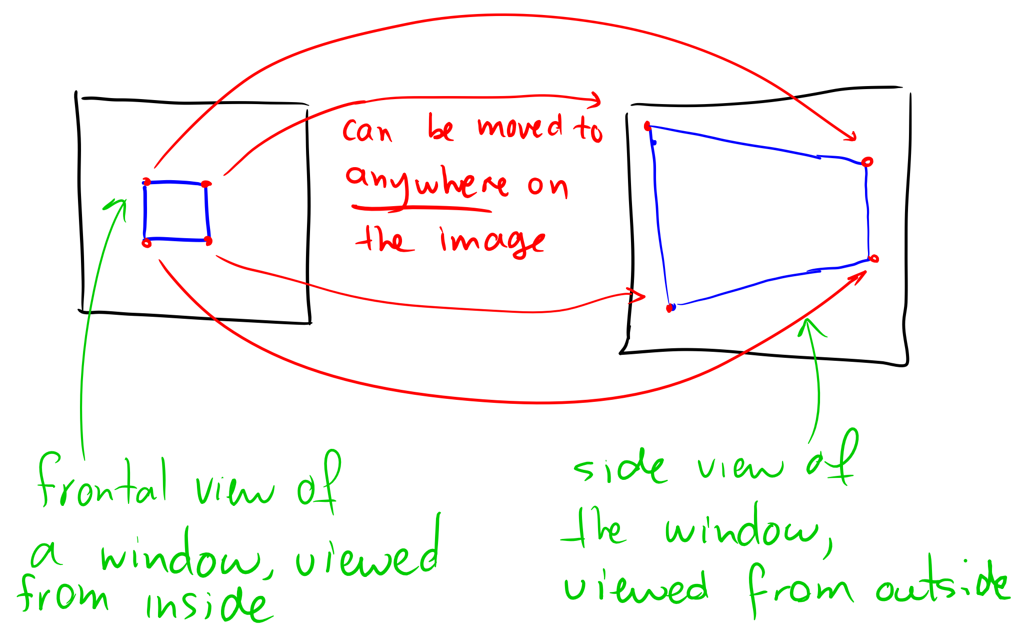
Homography Estimation: Basic Intuition



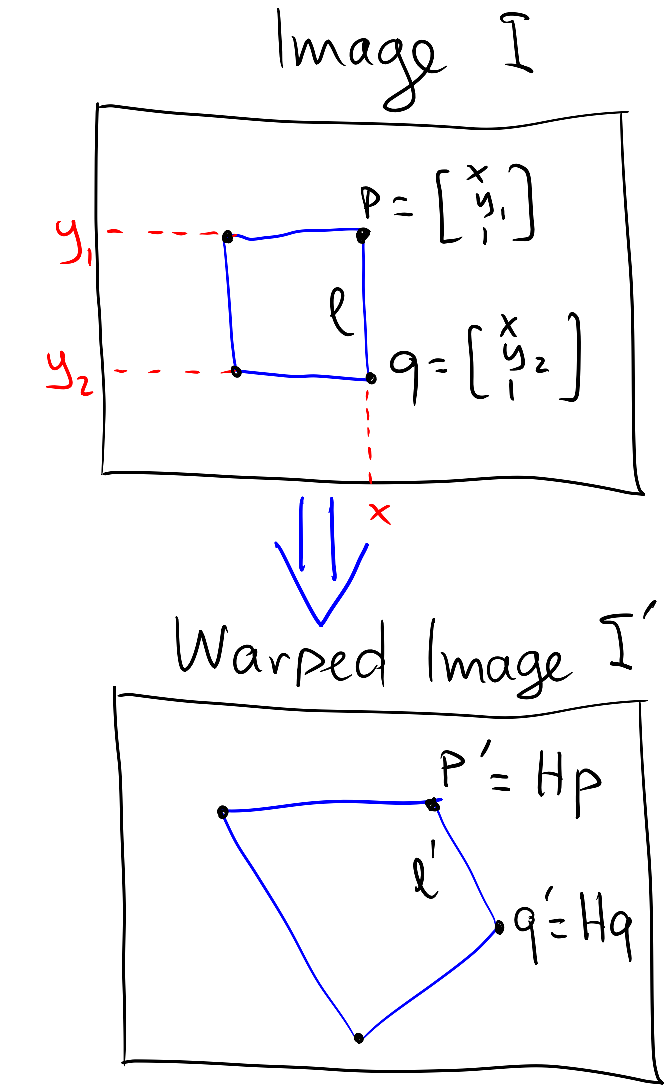
The matrix H is called a **Homography**

- **Intuition:**

Linear warps correspond to every possible distortion of a square created by moving its vertices to arbitrary locations

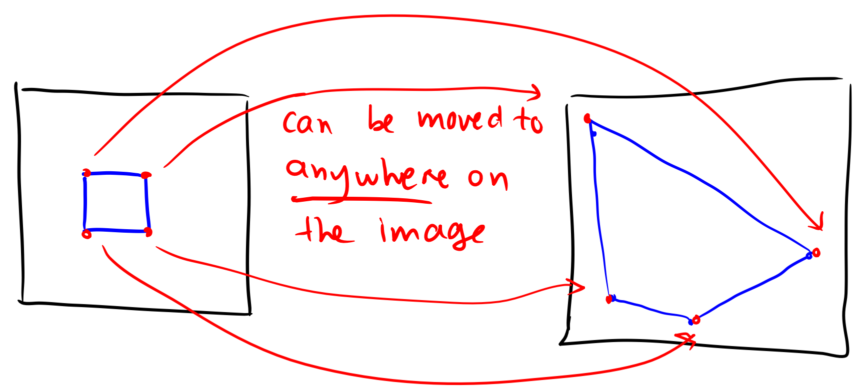


Estimating Homographies from Point Correspondences



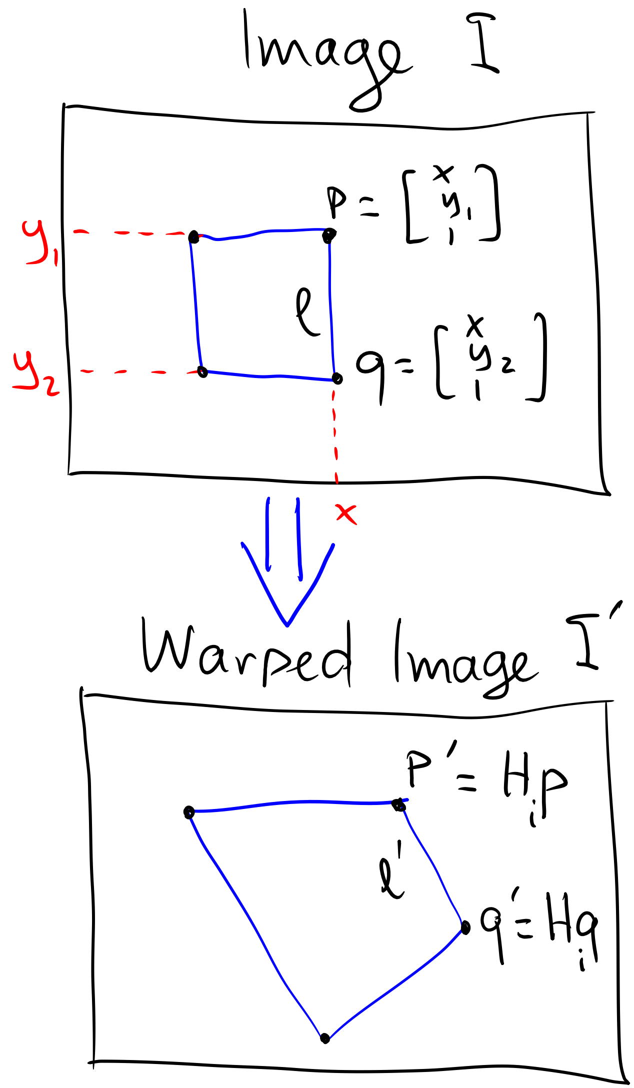
- Intuition:

If we have a correspondence between 4 points in the two images, we can compute H



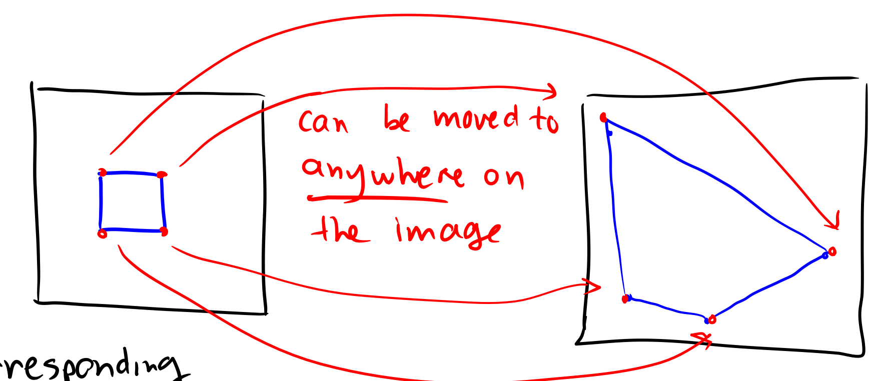
The matrix H is called a Homography

Estimating Homographies from Point Correspondences



- Intuition:

If we have a correspondence between 4 points in the two images, we can compute H_i

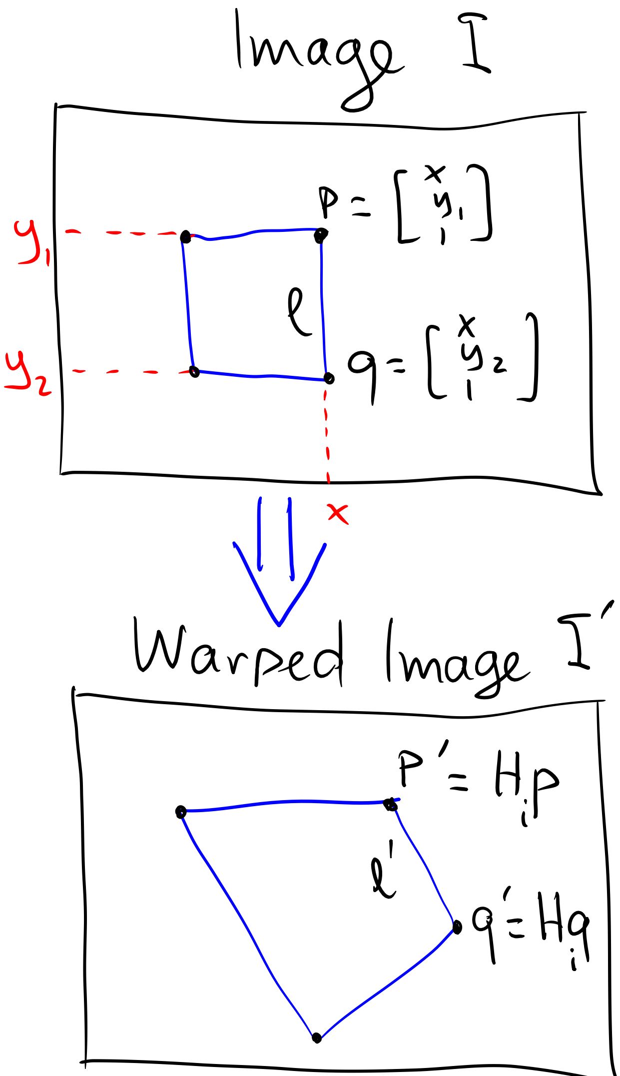


corresponding pixel in I'

pixel in I

$$\underbrace{\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix}}_{\text{known}} \simeq \overset{H}{\underbrace{\begin{bmatrix} a & b & c \\ d & e & f \\ h & i & j \end{bmatrix}}_{\text{unknown}}} \underbrace{\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}}_{\text{known}}$$

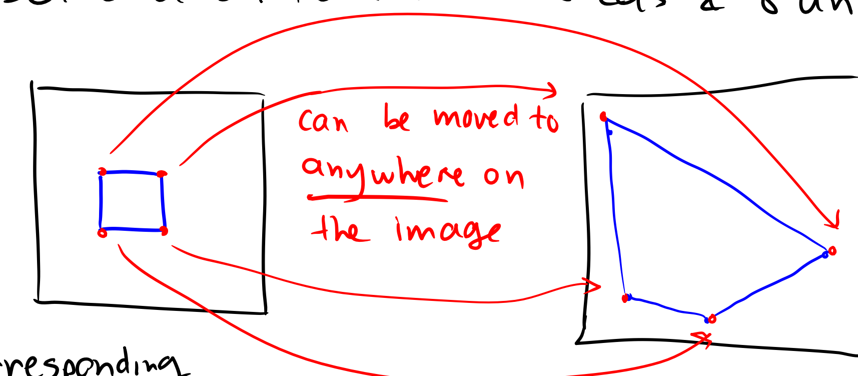
Homography Estimation by Solving Linear System



① Each correspondence gives 2 linear equations in the 9 unknowns
(so 4 correspondences ⇒ 8 eqs, 9 unknowns)

$$\boxed{\begin{aligned} a x_i + b y_i + c - x_i' \cdot (h x_i + k y_i + \cancel{\ell}^{\,1}) &= 0 \\ d x_i + e y_i + f - y_i' \cdot (h x_i + k y_i + \cancel{\ell}^{\,1}) &= 0 \end{aligned}}$$

② Since any multiple of H will do, we pick one element and set it to one (e.g. $\ell = 1$) & solve a system with 8 eqs & 8 unknowns



corresponding pixel in I'

pixel in I

$$\underbrace{\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix}}_{\text{known}} \simeq \overset{H}{\underbrace{\begin{bmatrix} a & b & c \\ d & e & f \\ h & k & \cancel{\ell}^{\,1} \end{bmatrix}}_{\text{unknown}}} \underbrace{\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}}_{\text{known}}$$

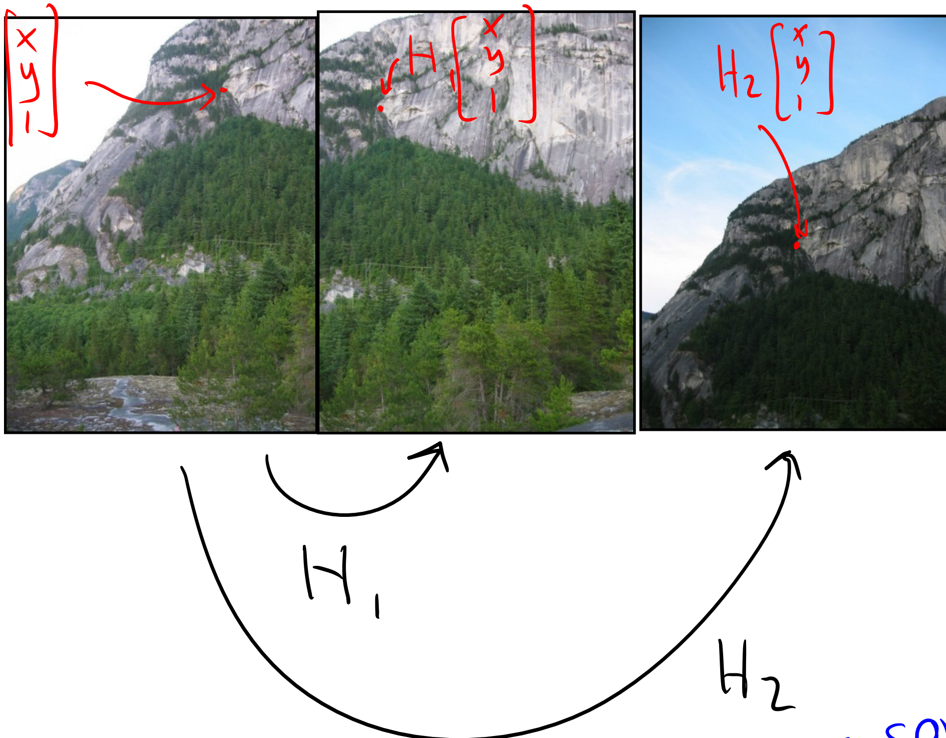
$$\boxed{\begin{aligned} x_i' &= (a x_i + b y_i + c)/(h x_i + k y_i + \ell) \Leftrightarrow \\ x_i'(h x_i + k y_i + \ell) &- (a x_i + b y_i + c) = 0 \end{aligned}}$$

Topic 11:

Homographies & Image Mosaics

- Introduction to image mosaicing
- Homogeneous coordinates for points & lines
- Image homographies
- Estimating homographies from point correspondences
- The autostitch algorithm

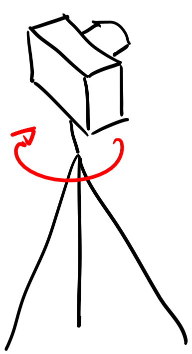
Feature-Based Image Matching



The Autostitch Algorithm

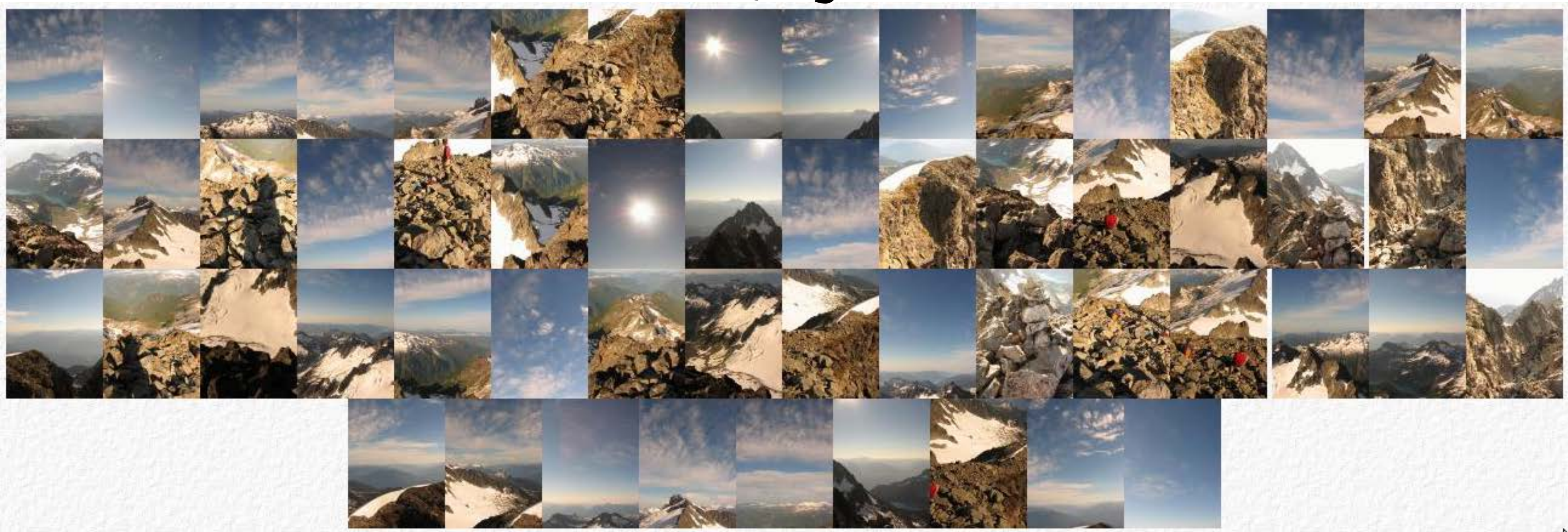
1. Run SIFT on each image
2. Match SIFT features across all photos
3. For every pair of photos:
   - Choose 4 matches & compute H
4. Warp photos according to H
5. Blend photos using pyramid blending

because some matches maybe incorrect (i.e. outliers) RANSAC is used

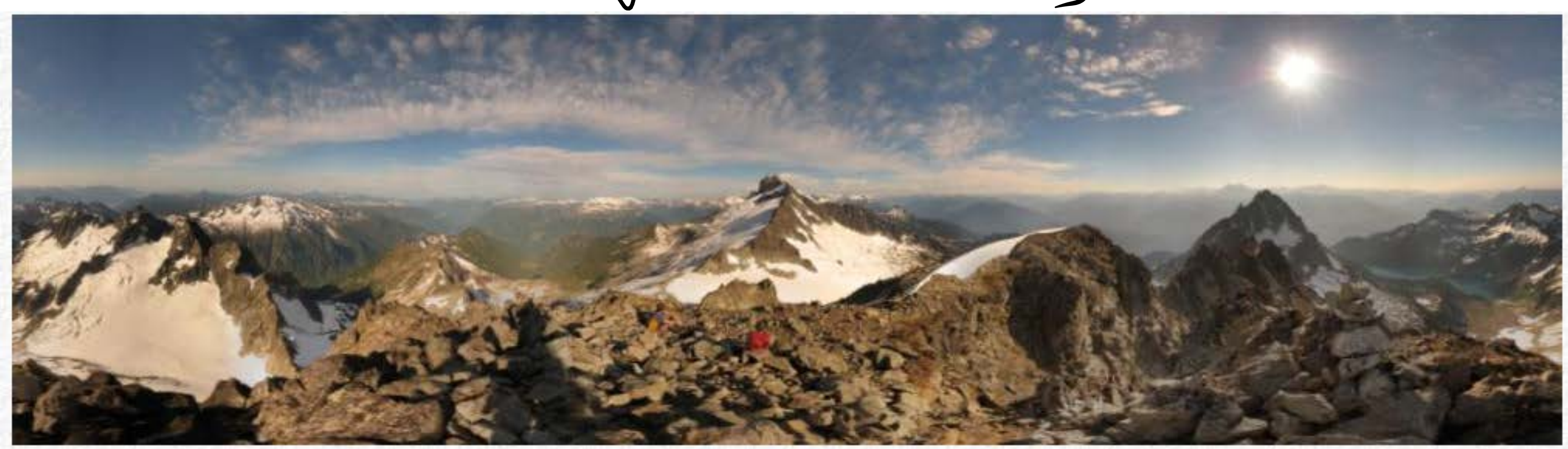


Building Panoramic Image Mosaics

Input images



automatically created mosaic



What You Will Take Away …

#1: Yes, math IS useful in CS !!

#2: How to turn math into pictures

#3: Basics of image analysis & manipulation

#4: How to read research papers

Visual Computing Principles

Imaging essentials

Understanding pixel intensity & color

Image representation & transformation

Image ⇔ 2D array of pixels

Camera response functions
Pixel representations & matting

Image ⇔ continuous 2D function

Poly fitting, WLS, RANSAC
Derivative estimation
Gradients, Laplacian
Edge detection

Image ⇔ n-dimensional vector

Correlation, conv, PCA,
Filtering ⇔ derivative computations
Smoothing
Fourier analysis

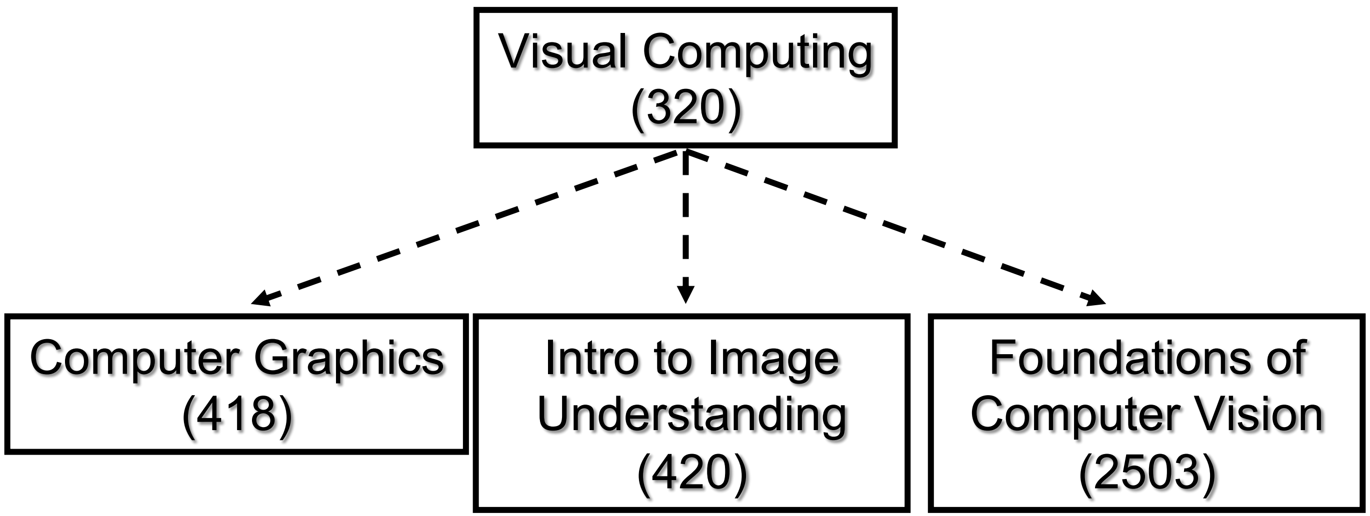
Hierarchical image representations

Pyramids, wavelets
Scale-space representations,
SIFT

Homogeneous representations

+ Applications: Alpha matting, inpainting, morphing, mosaicking, feature matching..

Where does this course fit in?



- CSC320 is not a pre-requisite for these courses
- Math foundations are the same, and will help to understand the foundations of these topics