Robot Vision

Visual Markers & Homography

Dr. Chen Feng
cfeng@nyu.edu

ROB-UY 3203, Spring 2024

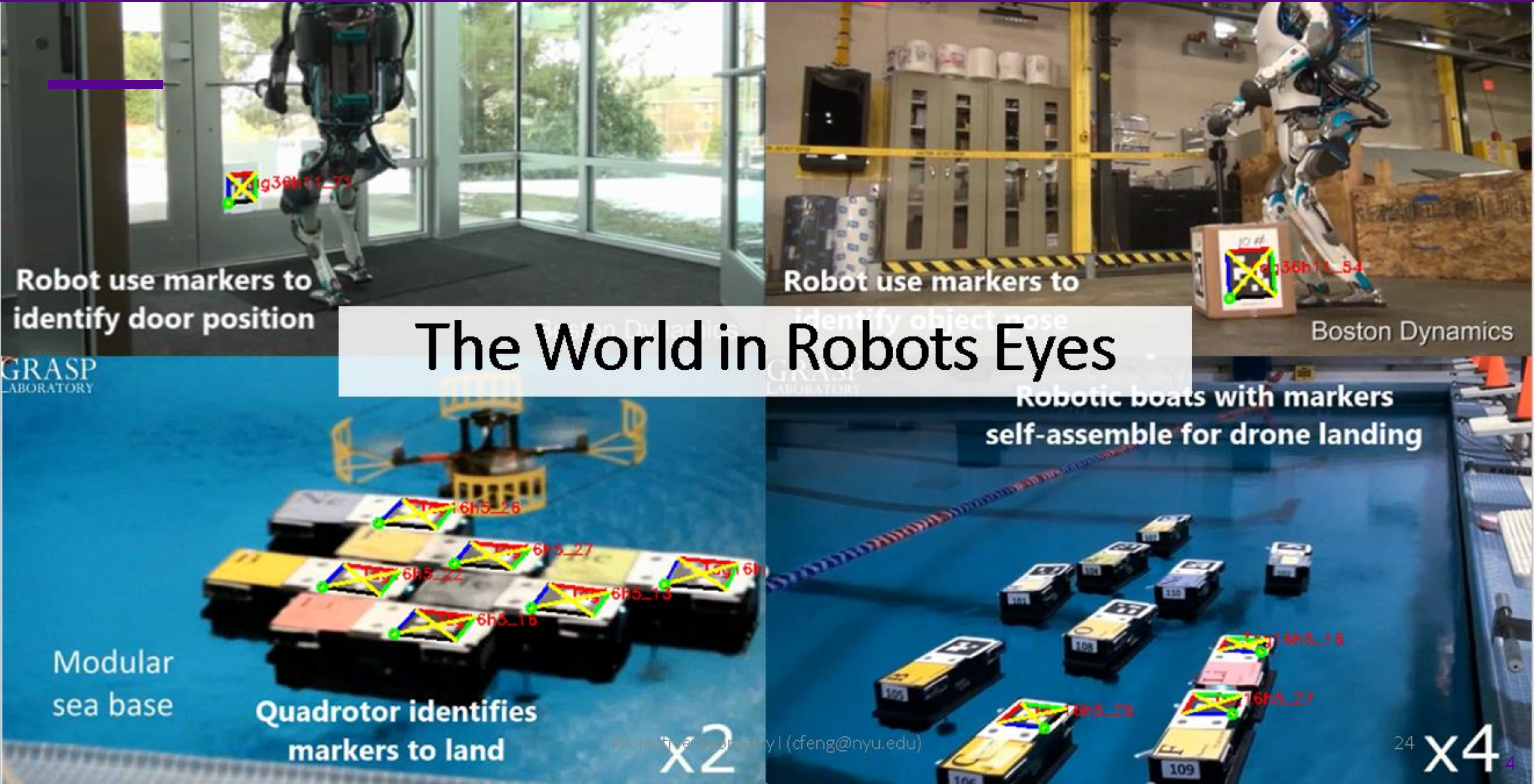
Overview

- Fiducial markers
- Single-view geometry: Homography
  - Review of equation solving
    - Ax=b
    - Ax=0
- Camera calibration: Zhang's method
- Camera pose estimation: PnP problem

References

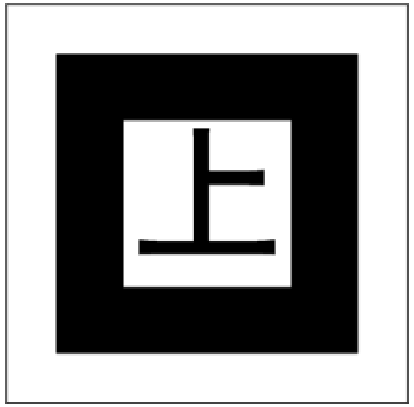
- Forsyth & Ponce 2011: Computer Vision: A Modern Approach
  - Section 1.2, 1.3, 12.1
- Szeliski 2011: Computer Vision: Algorithms And Applications
  - Section 6.3.1
- Corke 2011: Robotics, Vision, & Control
  - Section 11.2, 11.1
- Hartley & Zisserman 2003: Multiple View Geometry In Computer Vision
  - Section 2.3, 4.1, 4.4, 7.1, 7.2, 7.4
- Linear algebra:
  - Szeliski 2011: section A.1.1, A.1.2, A.1.3, A.2, A.2.1
  - Hartley & Zisserman 2003: A5.1, A5.2, A5.3

The World in Robots Eyes

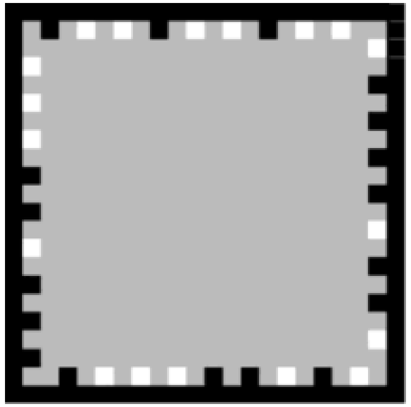


Fiducial Markers: More Than QR Codes

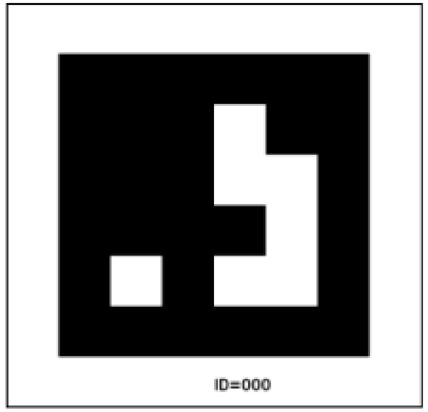
Fiducial:



(Kato and Billinghurst 1999)



(Wagner et al. 2008)



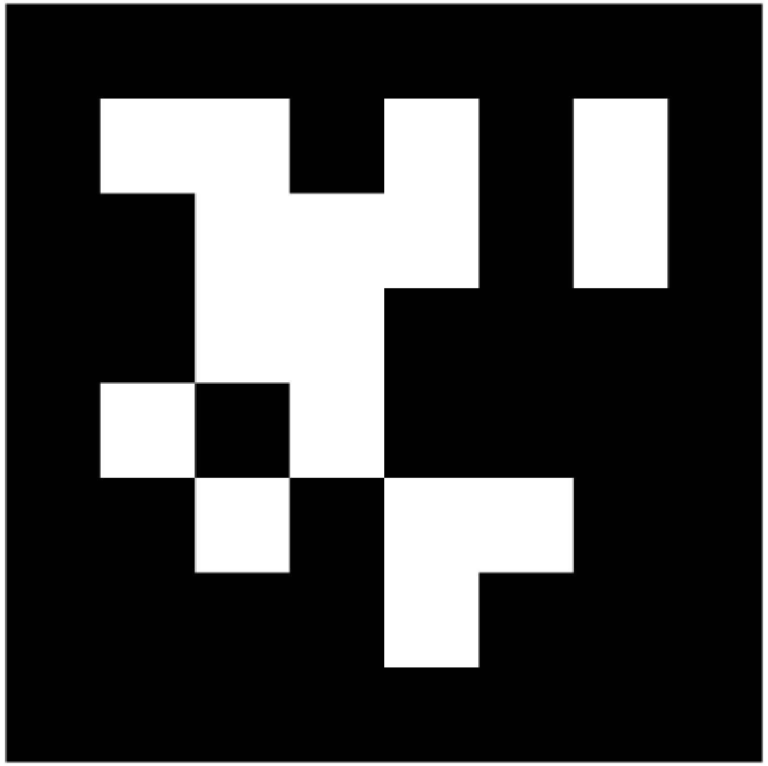
(Olson 2011)

Natural:
(Lepetit and Fua 2005)

https://developer.vuforia.com/library/articles/Solution/Natural-Features-and-Ratings

(Feng and Kamat 2013)

What Is an AprilTag?



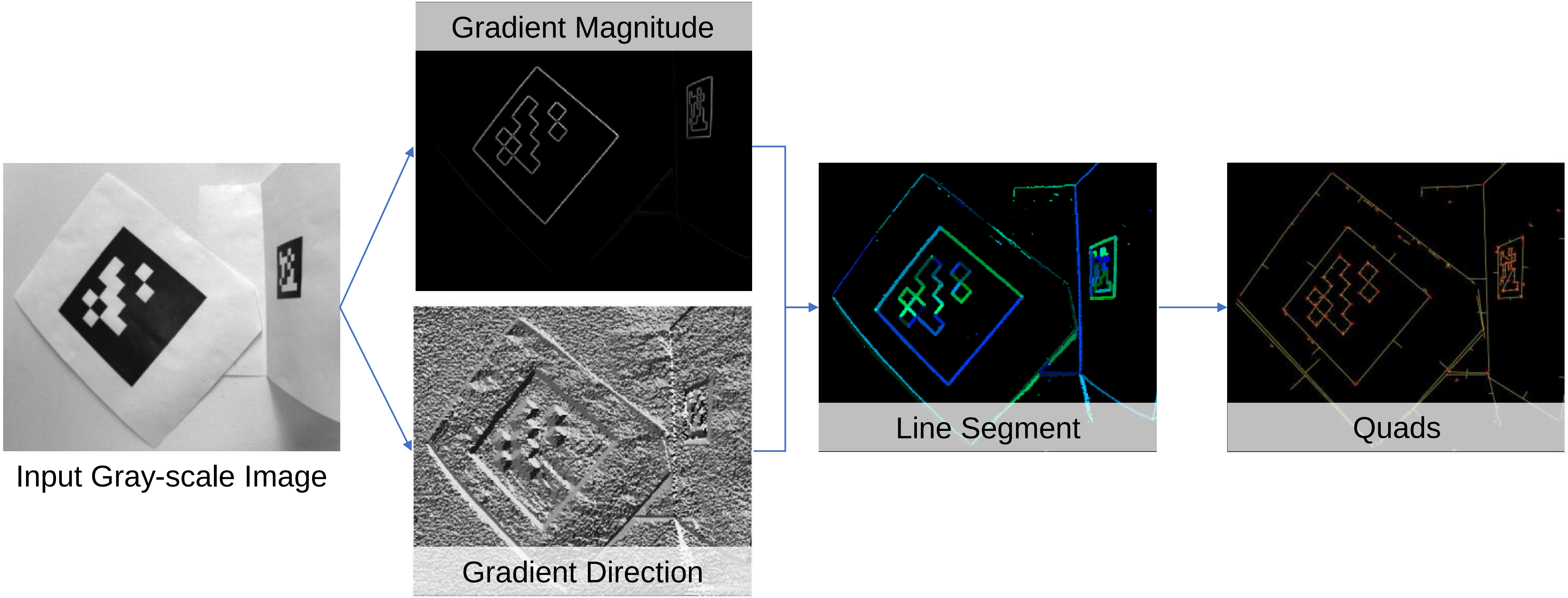
$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$[110101011101011000101000010110000100]$$

$$\begin{cases} id = 0 \\ family = Tag36h11 \end{cases}$$

Olson 2011

How Is an AprilTag Detected?



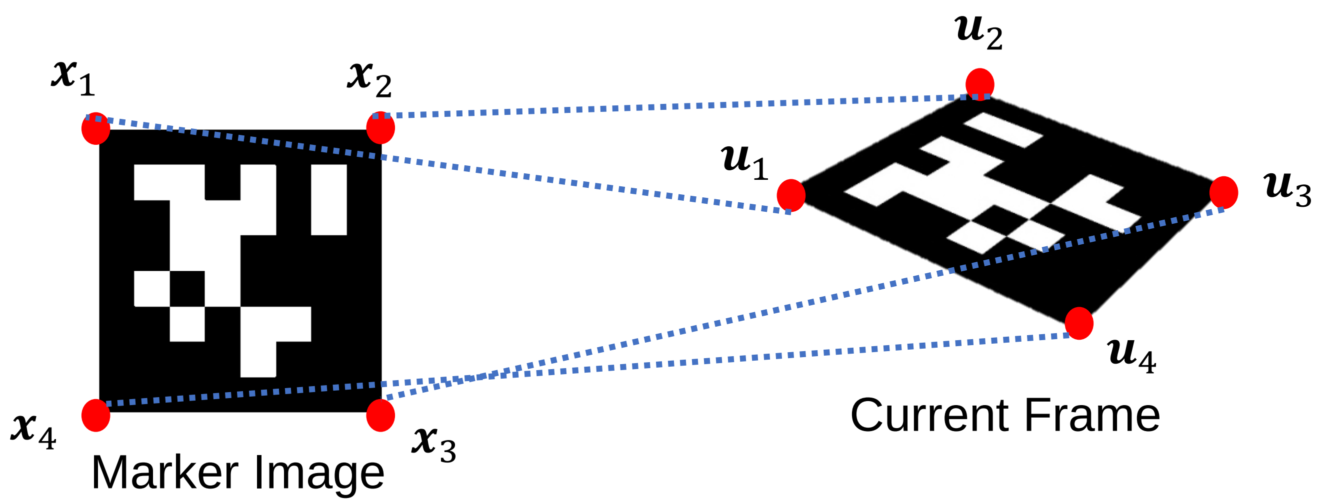
Olson 2011

Advantages of AprilTag

- Fast
  - >25Hz for 640x480 webcam Image on normal laptop
- Robust
  - Higher detection rate
  - Fewer false alarm
- Larger Range
  - Distance
  - View direction
  - Illumination

Max Detectable Distance (m)		Marker Angle (degree)			
		0	45	0	45
Marker Size (m^2)	0.2 x 0.2	6.10	4.88	11.28	8.84
	0.3 x 0.3	8.23	7.01	14.94	11.58
	0.46 x 0.46	13.41	11.28	25.91	21.64
	0.6 x 0.6	19.51	16.46	34.44	30.48
Image Resolution		640 x 480		1280 x 960	
Focal Length		850 pixels		1731 pixels	
Processing Rate		20 Hz		5 Hz	

AprilTags Provide Point Correspondences



- Useful for many projective geometry applications
- E.g., Homography Estimation

Homography == Projective Transformation

Definition 2.9. A *projectivity* is an invertible mapping h from $\mathbb{P}^2$ to itself such that three points $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ lie on the same line if and only if $h(\mathbf{x}_1)$, $h(\mathbf{x}_2)$ and $h(\mathbf{x}_3)$ do.

- They all mean the same thing:
  - Homography
  - Projectivity
  - Collineation

Theorem 2.10. *A mapping $h : \mathbb{P}^2 \rightarrow \mathbb{P}^2$ is a projectivity if and only if there exists a non-singular 3×3 matrix $\mathtt{H}$ such that for any point in $\mathbb{P}^2$ represented by a vector $\mathbf{x}$ it is true that $h(\mathbf{x}) = \mathtt{H}\mathbf{x}$.*

Hartley & Zisserman 2011

Homography == Projective Transformation

Definition 2.11. Projective transformation. A planar projective transformation is a linear transformation on homogeneous 3-vectors represented by a non-singular 3×3 matrix:

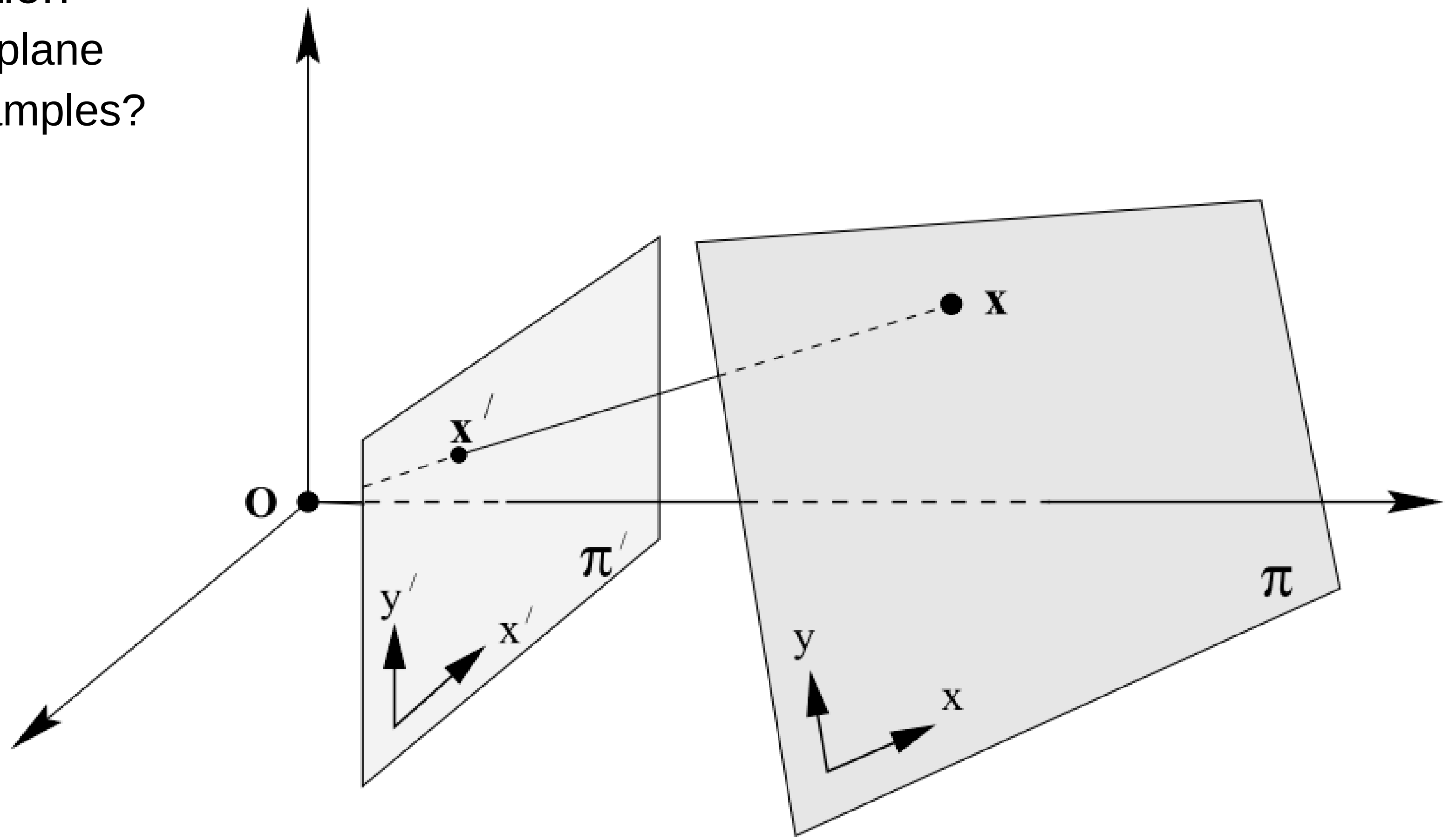
$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \tag{2.5}$$

or more briefly, $\mathbf{x}' = \mathtt{H}\mathbf{x}$.

Hartley & Zisserman 2011

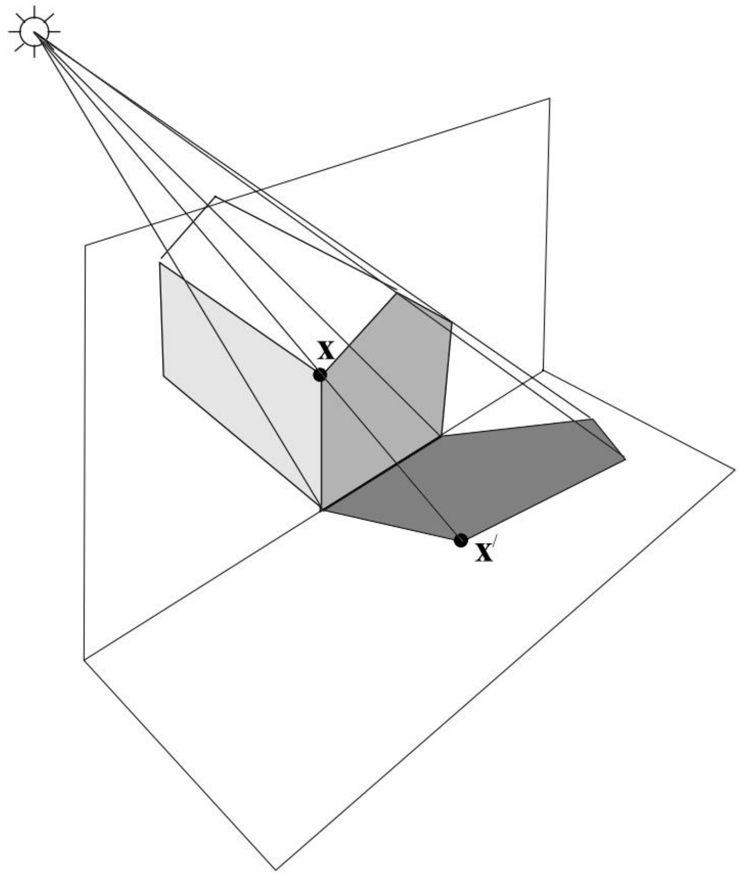
Perspective Transformation ⊂ Homography

- One way to geometrically construct a homography between two planes:
  - Perspective transformation
    - i.e., taking photo of a plane
    - Any other real-life examples?



Hartley & Zisserman 2011

Real-life Example of Perspectivity

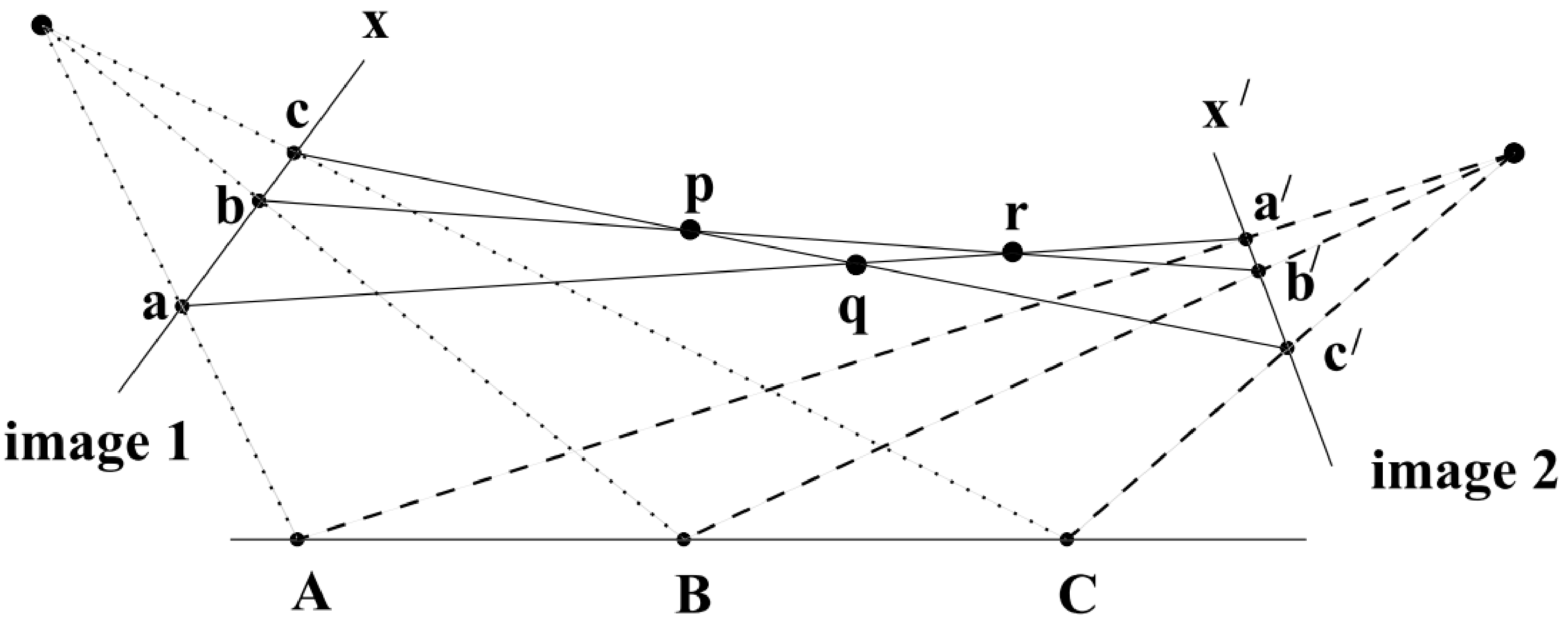


Hartley & Zisserman 2011

Homography vs. Perspectivity

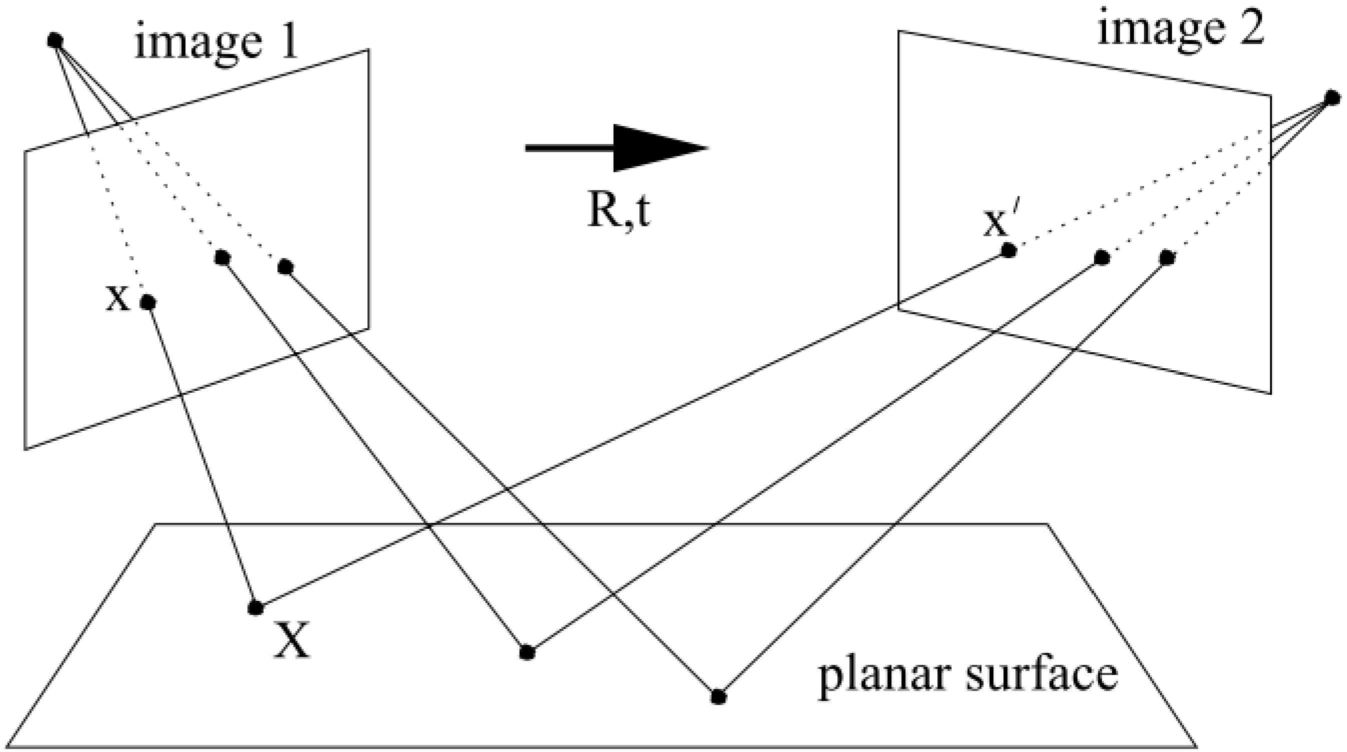
homography

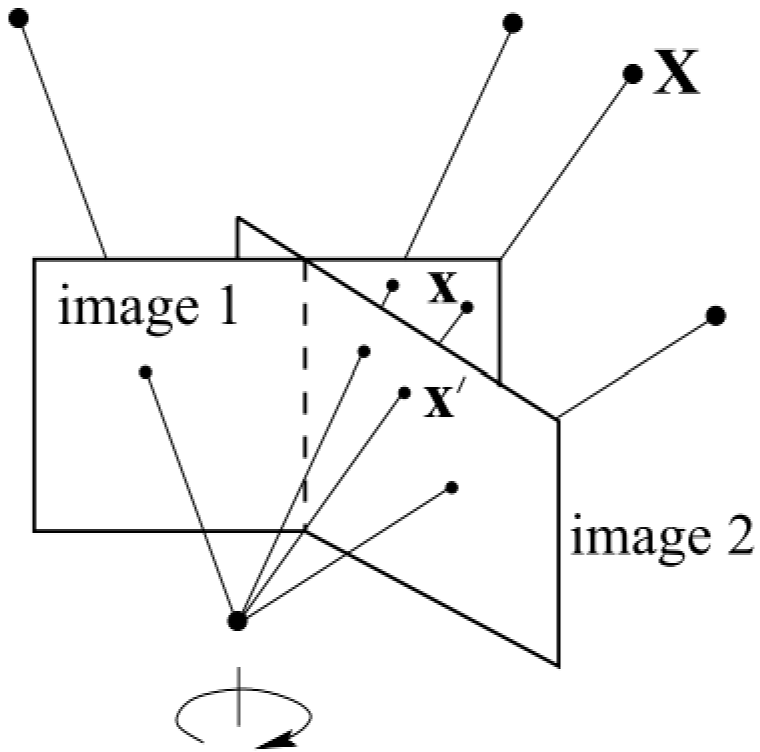
The composition of two (or more) perspectivities is a projectivity, but not, in general, a perspectivity.



Hartley & Zisserman 2011

Which Is a Non-Perspective Homography?



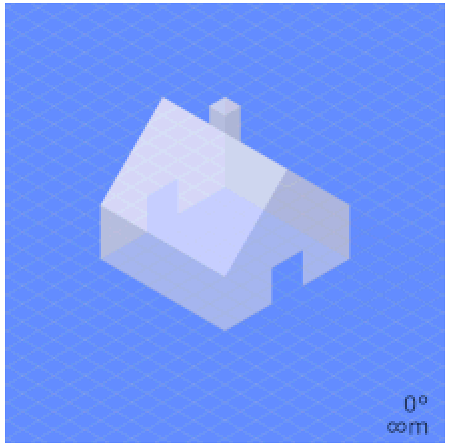


Hartley & Zisserman 2011

Application of Perspective Homography

Hartley & Zisserman 2011

Another Application: Removing Perspective Distortion

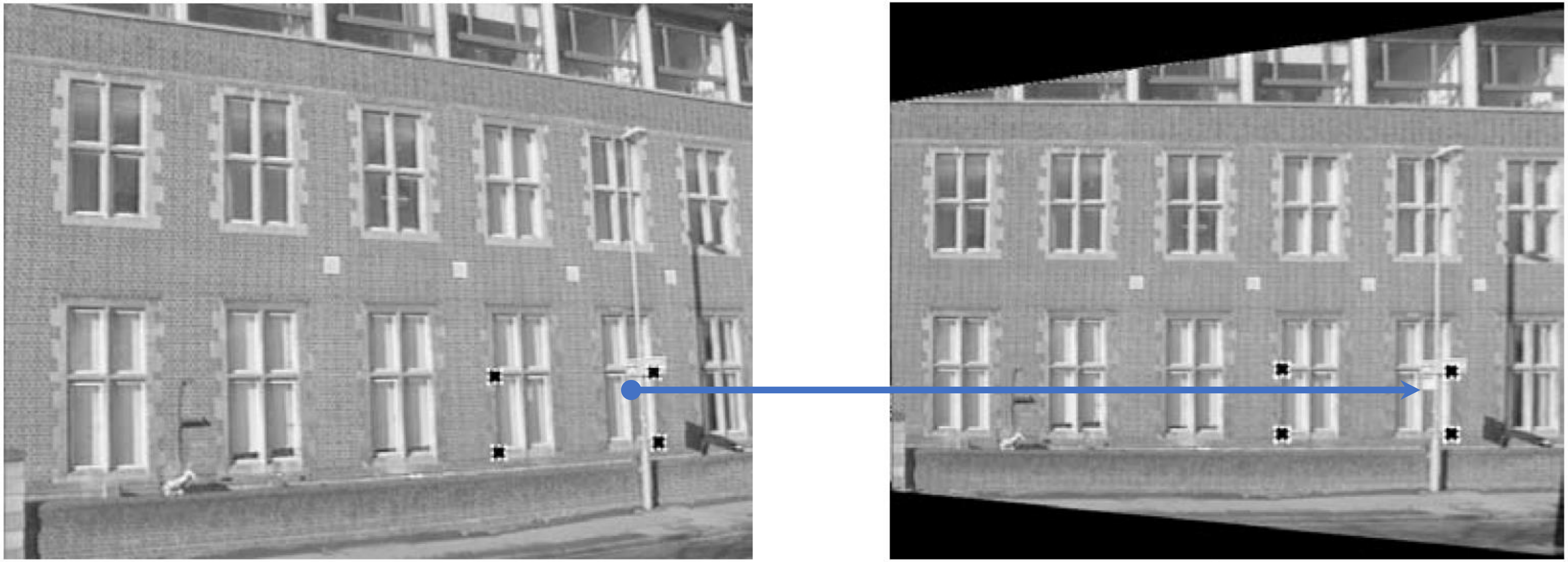


Perspective Distortion

Hartley & Zisserman 2011

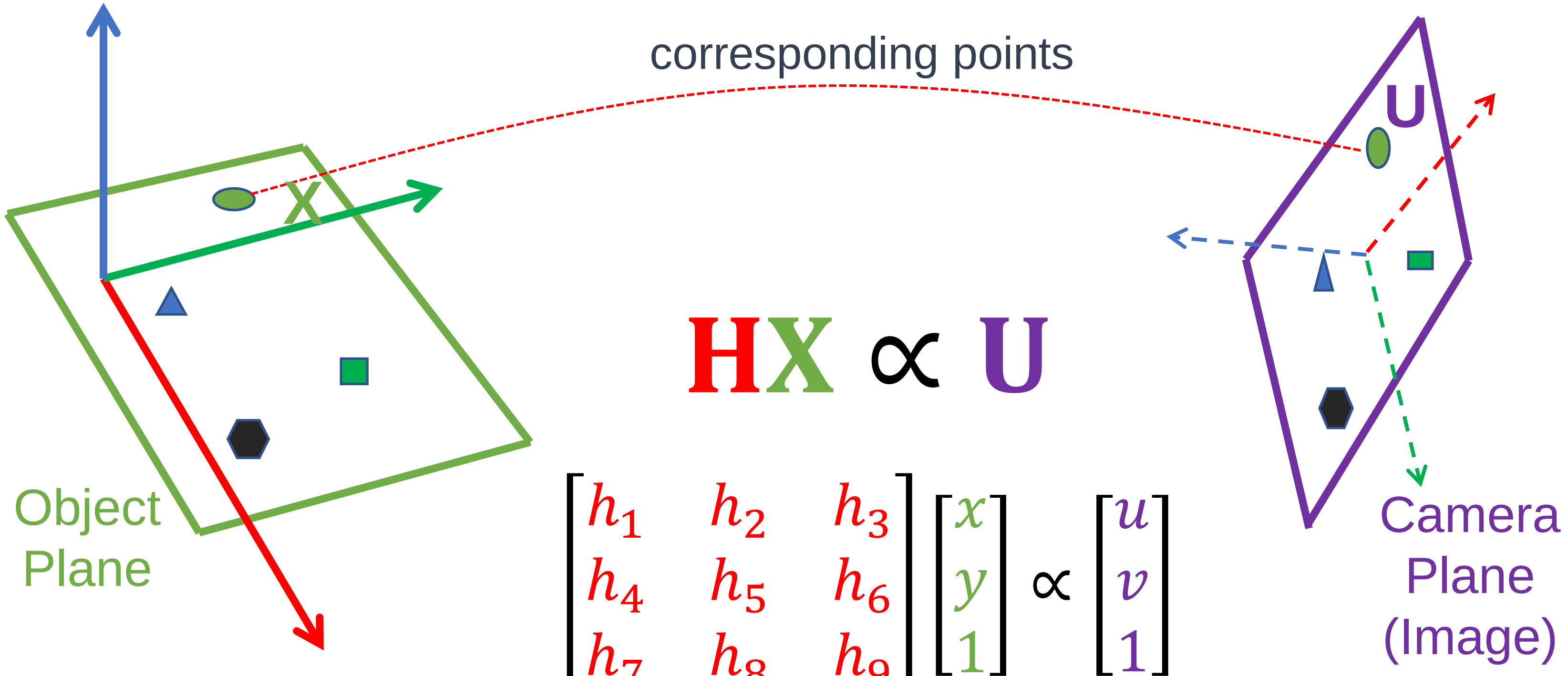
Removing Perspective Distortion via Estimating the Homography

- This could be used to create “Live Spot Map” from surveillance videos
  - Useful for indoor robotics applications



Hartley & Zisserman 2011

How to Estimate a Homography?



$$\mathbf{HX} \propto \mathbf{U}$$

$$\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \propto \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Estimating Homography: DLT

- 2D direct linear transformation (DLT) algorithm
- Find multiple $\mathbf{X} \leftrightarrow \mathbf{U}$ correspondences (≥ 4) between a planar object and an image
- Each correspondence leads to 2 independent linear equations with homography as unknown parameters:

$$u(h_7x + h_8y + h_9) = h_1x + h_2y + h_3$$
$$v(h_7x + h_8y + h_9) = h_4x + h_5y + h_6$$

- This leads to a homogeneous system of linear equations

$$\begin{aligned} xh_1 + yh_2 + h_3 \qquad\qquad\qquad\qquad - uxh_7 - uyh_8 - uh_9 &= 0 \\ xh_4 + yh_5 + h_6 - vxh_7 - vyh_8 - vh_9 &= 0 \end{aligned}$$

$$\begin{bmatrix} x & y & 1 & & & & -ux & -uy & -u \\ & & & x & y & 1 & -vx & -vy & -v \end{bmatrix} [h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7\ h_8\ h_9]^T = 0$$

$$\mathbf{Ah} = \mathbf{0}$$

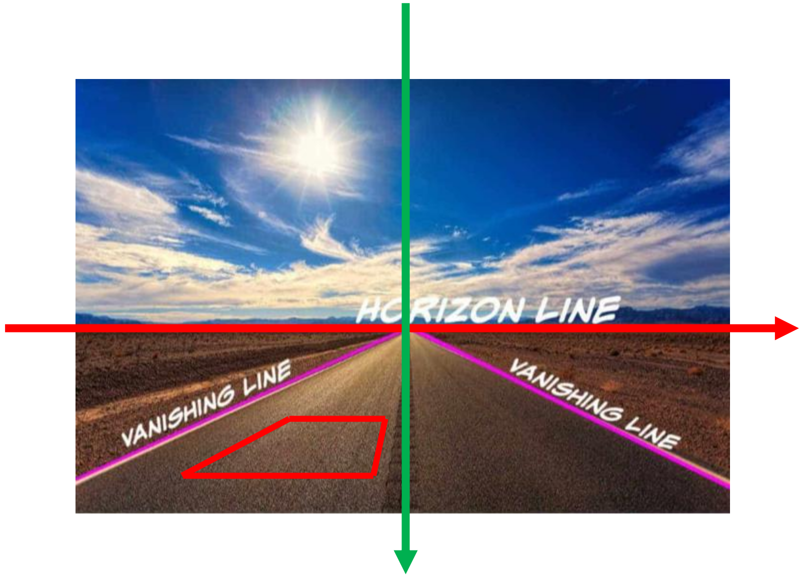
2D DLT Using Inhomogeneous Homography

- Set $h_9 = 1$, $\quad \tilde{\boldsymbol{h}} = [h_1, h_2, \cdots, h_8]$

$$\begin{bmatrix} x & y & 1 & & & & -ux & -uy \\ & & & x & y & 1 & -vx & -vy \end{bmatrix} [h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7\ h_8]^T = \begin{bmatrix} u \\ v \end{bmatrix}$$

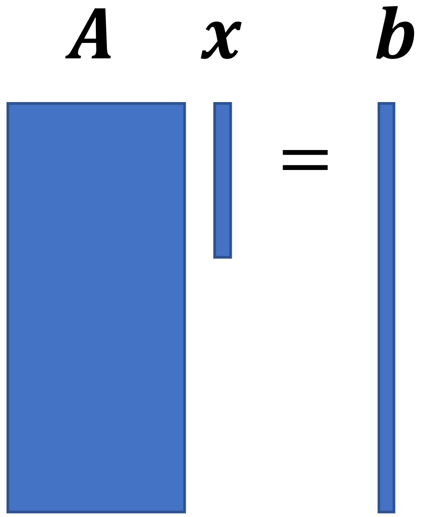
- Solve by least square: $\left(A^T A\right)^{-1} A^T b$

- Potential issues
  - What if $h_9 = 0$?
  - Does this happen often?

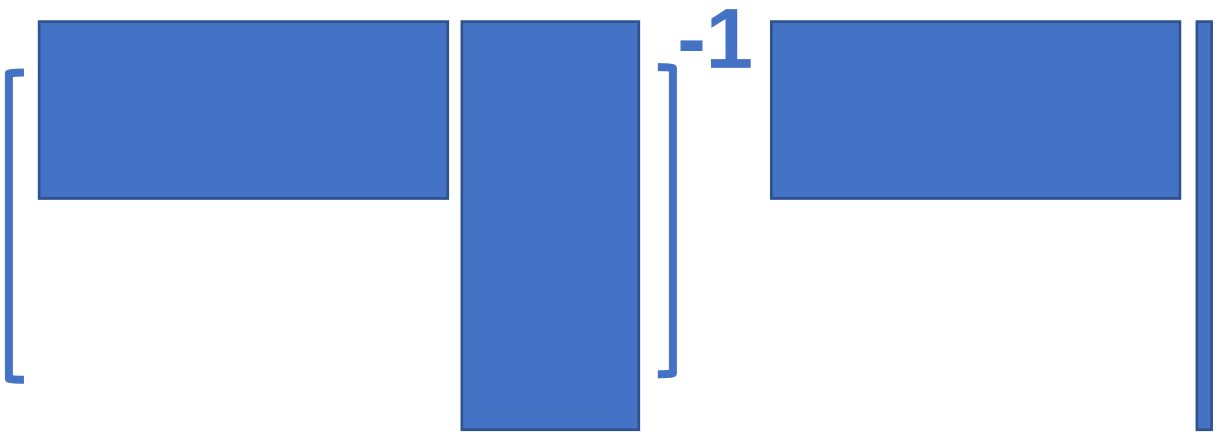


Solving Ax=b

- A: design matrix
  - shape: m x n
  - m>>n
  - Typically, full column-rank
- x: unknowns
  - shape: n x 1
- b: observed data
  - shape: m x 1
- Solve by least squares: x* = inv(A'A)A'b
  - Solving normal equation: A'Ax=A'b
  - Least squares residual:
    - Observed - Predicted
    - b - Ax*
  - Residual is usually not zero in real world problems!

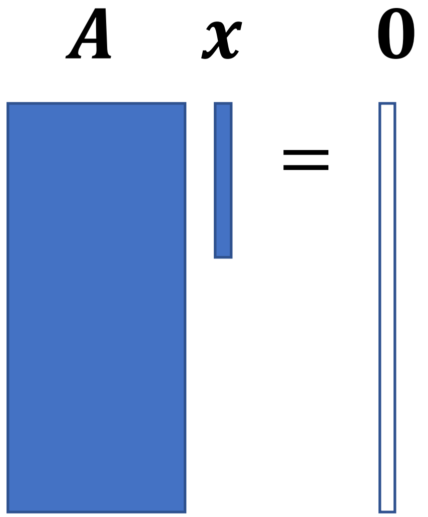


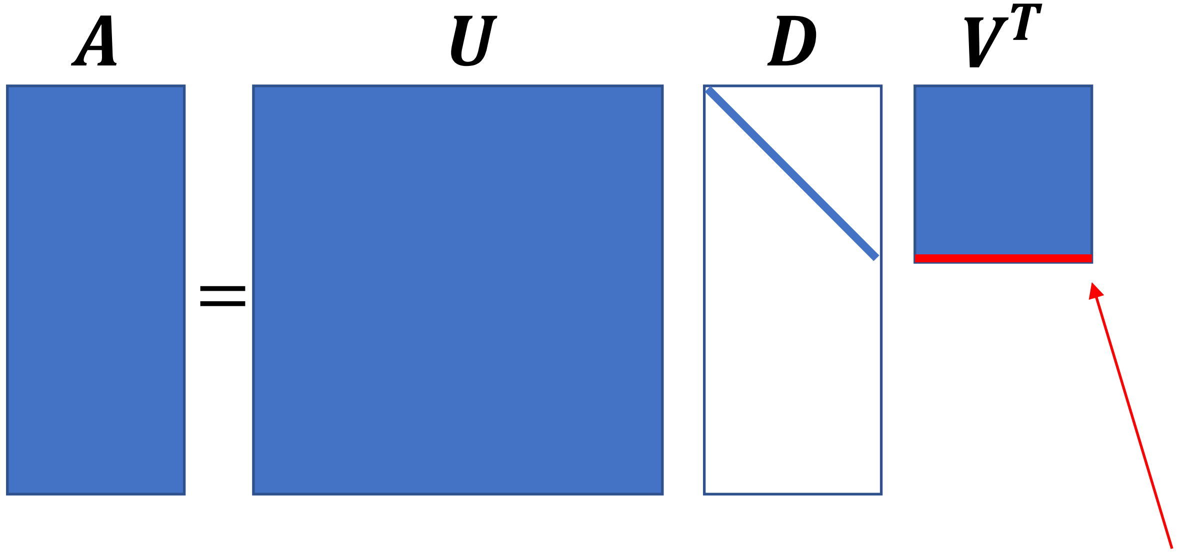
Least squares solution $\boldsymbol{x}^*$



Solving Ax=0

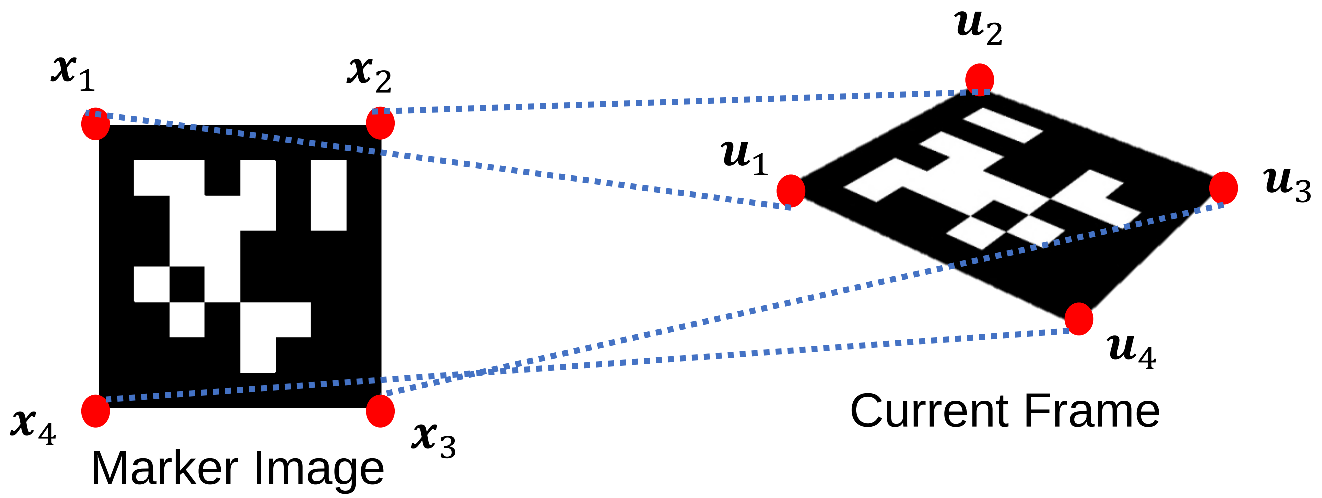
- A: data matrix
  - shape: m x n
  - m >> n
  - rank(A) = n when data contains noise: full column-rank
- x: unknowns
  - shape: n x 1
- Seek for an approximation instead of exact non-trivial solutions
- Add a constraint: ||x||=1
- Solve by SVD: A=UDV’
  - x*=last column of V, if diag(D) is descending order
  - diag(D): non-negative
    - called singular values





Least squares solution x^*

AprilTags Provide Point Correspondences

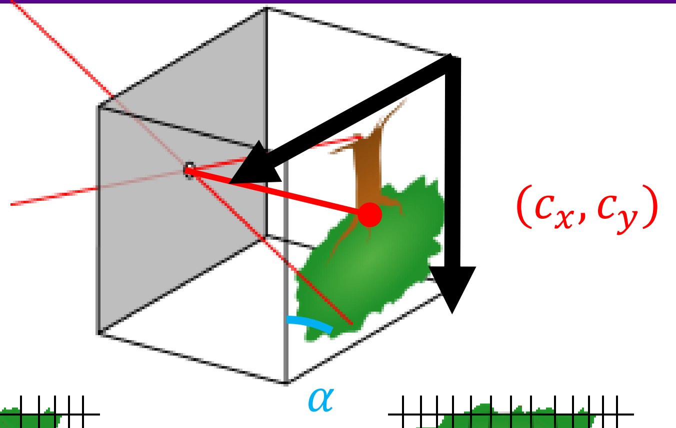


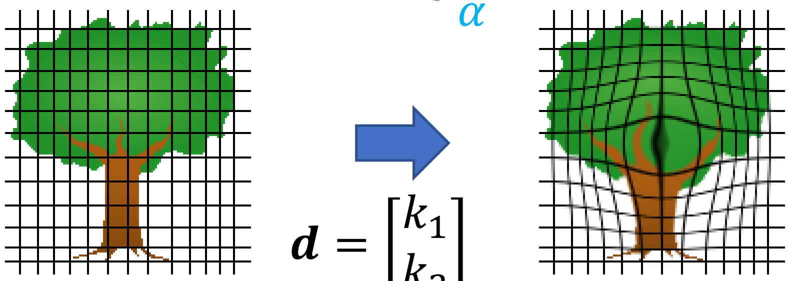
- Tags can help us calibrate our camera

Camera Calibration

- To find out intrinsic parameters of a camera
  - **Linear:** $\mathbf{K} = ?$
  - **Non-linear:** $\mathbf{d} = ?$
- Intrinsic parameter values are generally static
  - Only need to be calibrated once
  - Unless the camera has been shipped for long distances
- Why?
  - Reduce uncertainties/unknowns in the projection system
  - Improve accuracy
- How?
  - $\mathbf{K}$ and $\mathbf{d}$ can not be easily measured directly
  - Has to be solved using perspective projection equation indirectly

$$\mathbf{K} = \begin{bmatrix} f_x & \alpha & c_x \\ & f_y & c_y \\ & & 1 \end{bmatrix}$$



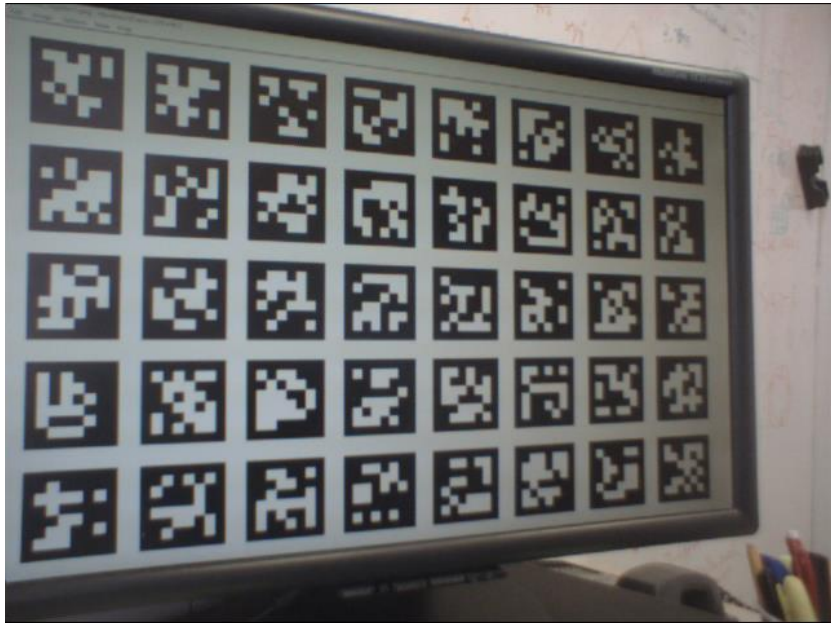


$$\boldsymbol{d} = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}$$

Calibration with a 2D Rig

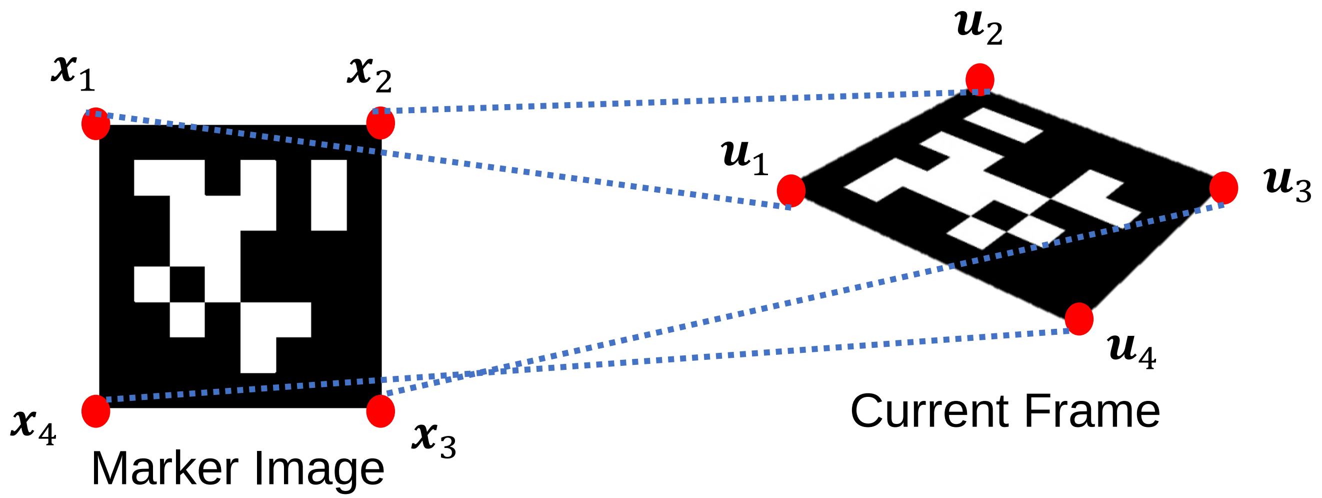
- Using 2D Calibration Rig
  1. All markers on a same plane
  2. Measure each marker’s 2D position
  3. Take multiple images
  4. Solve by Zhang’s method
  5. Refine by bundle adjustment

- Advantages
  - Measuring 2D position is easy
  - Easy to setup planar rig
    - Print it out
    - Use a flat screen



Z. Zhang, "A flexible new technique for camera calibration'", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.22, No.11, pages 1330–1334, 2000

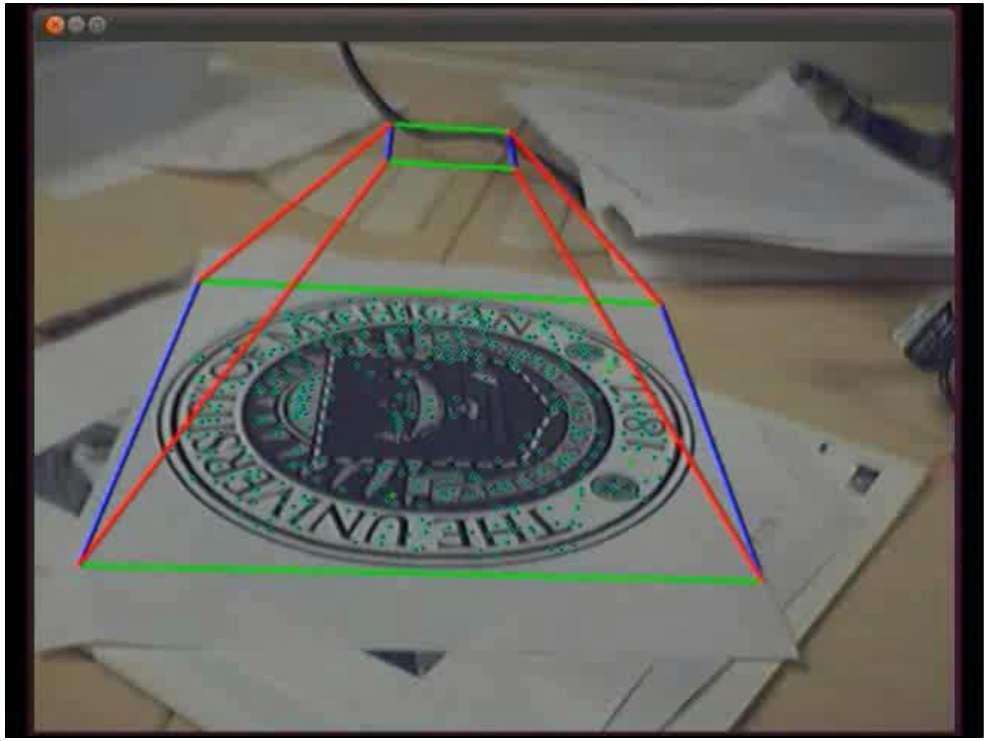
AprilTags Provide Point Correspondences



- Tags can also help us perform augmented reality

Augmented Reality (AR)

- AR = rendering virtual component on real images/videos
- AR can help robot and human to understand each other
- A key robot vision problem to realize AR:
  - Camera Pose Estimation
  - [R, t]?
- Then we can calculate where to draw on the image
  - p ~ K[R,t]X
  - p is the homogeneous coordinate of where you should draw



https://youtu.be/8Y8MIh7jhsY

Homography Decomposition

1) Compute the $\hat{R}_1$, $\hat{R}_2$ and $\hat{t}$ in the following equation:

$$[\hat{R}_1\ \hat{R}_2\ \hat{t}] = K_c^{-1} H_w^c \tag{3}$$

Note that $\hat{R}_1$, $\hat{R}_2$ and $\hat{t}$ represent the first, second and third column of the computed matrix $K_c^{-1} H_w^c$.

2) Obtain the U, V matrix in SVD:

$$USV^T = [\hat{R}_1\ \hat{R}_2\ \hat{R}_1 \times \hat{R}_2] \tag{4}$$

3) Calculate the refined R_w^c, t_w^c which satisfy the constraint of $R_1^T R_2 = 0$ and $||R_1|| = ||R_2|| = 1$ (R_1, R_2 represent the first and second column of R_w^c respectively):

$$R_w^c = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det(UV^T) \end{bmatrix} V^T,\ t_w^c = \hat{t}/||\hat{R}_1|| \tag{5}$$

Perspective-n-point (PnP) Problem

- Solving PnP problem = Estimating a camera pose from 2D-to-3D correspondences
- **A calibrated camera is an angular sensor**
  - $\hat{\boldsymbol{x}}_i = \boldsymbol{K}^{-1}\boldsymbol{x}_i / \left\| \boldsymbol{K}^{-1}\boldsymbol{x}_i \right\|$
- OpenCV provides the API

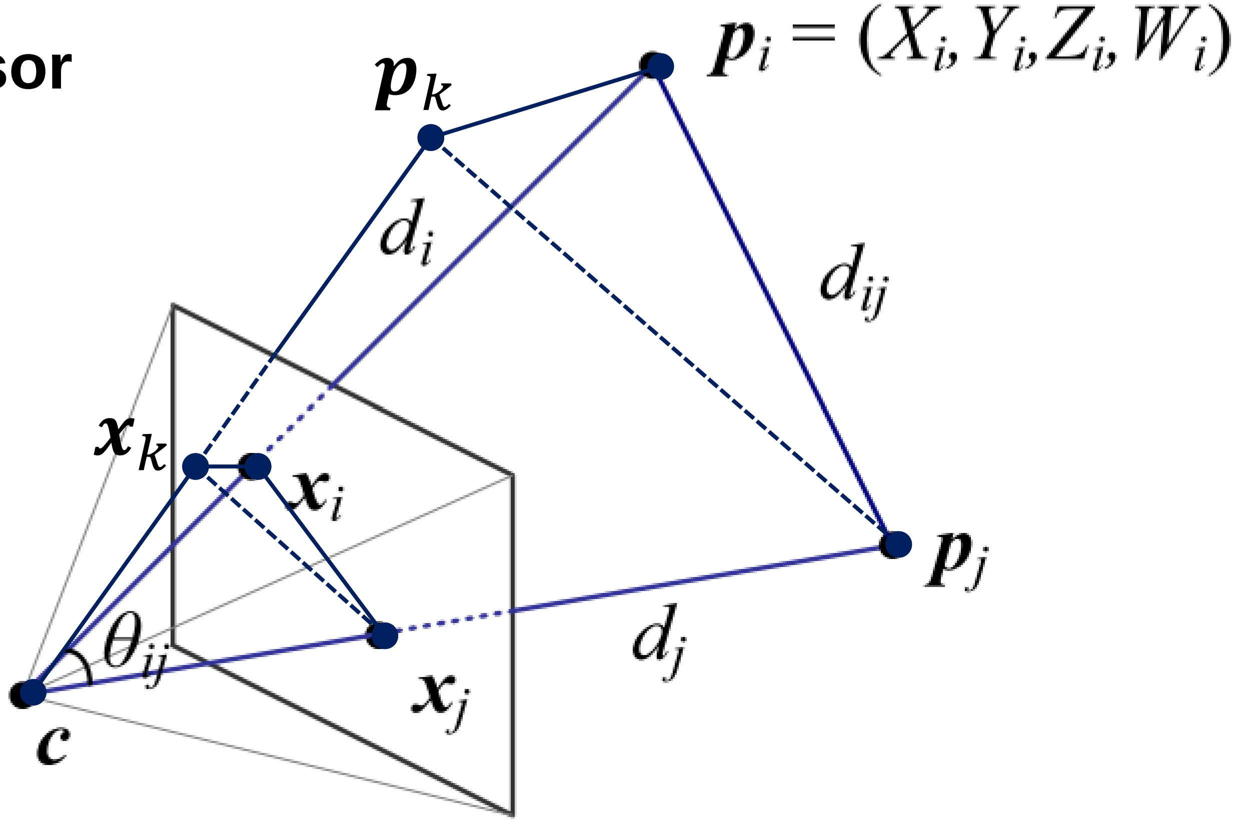


Image modified from Szeliski 2011

References for Next Week

- Hartley & Zisserman 2003:
  - Section 9.1, 9.2, 9.5, 9.6
- Corke 2011:
  - Section 14.2, 14.3
- Forsyth & Ponce 2011
  - Chapter 7
- Szeliski 2011:
  - Section 11.1, 12.2, 12.3