CSCI 4830 / 5722
Computer Vision

University of Colorado **Boulder**

CSCI 4830 / 5722
Computer Vision

Dr. Ioana Fleming
Spring 2019
Lecture 6

University of Colorado **Boulder**

Reminders

Submissions:

- Homework 2: later this week
- Homework 3: later this week

Readings:

- Szeliski:
  - chapter 3 (filters, changing resolution, Laplacian pyramids, warping)
  - chapter 4.1 (points) and 4.2 (edge detection)
- P&F Ch. 4,5
- Camera Calibration paper

Today

- Camera Calibration paper – posted on Moodle
- Estimating homographies

Camera Parameters

- Summary:
  - points expressed in external frame
  - points are converted to canonical camera coordinates
  - points are projected
  - points are converted to pixel units

$$\begin{pmatrix}U\\V\\W\end{pmatrix} = \begin{pmatrix}\text{Transformation}\\\text{representing}\\\text{intrinsic parameters}\end{pmatrix}\begin{pmatrix}\text{Transformation}\\\text{representing}\\\text{projection model}\end{pmatrix}\begin{pmatrix}\text{Transformation}\\\text{representing}\\\text{extrinsic parameters}\end{pmatrix}\begin{pmatrix}X\\Y\\Z\\T\end{pmatrix}$$

point in pixel coords.

point in metric image coords.

point in cam. coords.

point in world coords.

Degrees of freedom

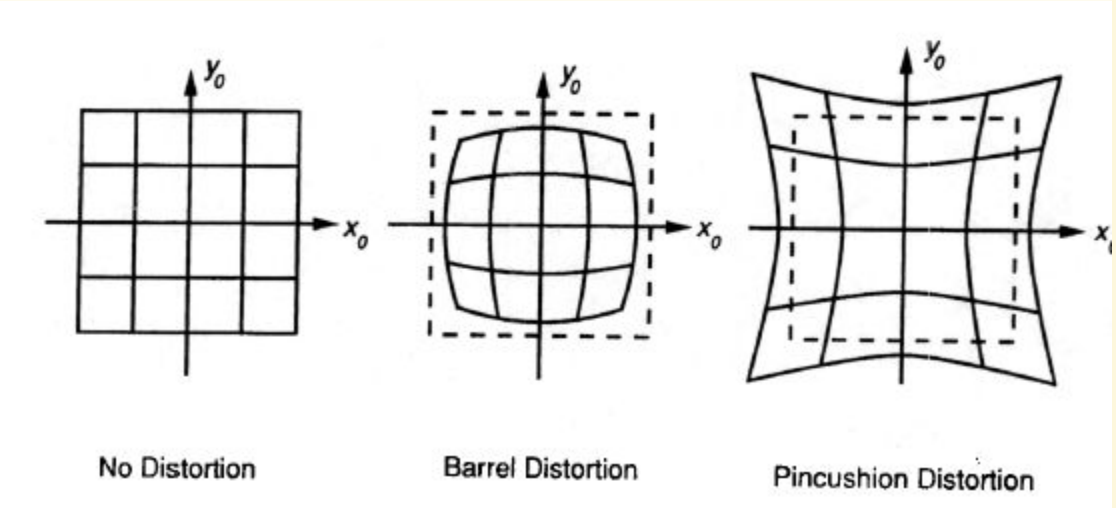
Q: How many known points are needed to estimate this?

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$w\begin{bmatrix}u\\v\\1\end{bmatrix} = \overset{5}{\begin{bmatrix}\alpha & s & u_0\\0 & \beta & v_0\\0 & 0 & 1\end{bmatrix}}\overset{6}{\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\r_{21} & r_{22} & r_{23} & t_y\\r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

Beyond Pinholes: Radial Distortion

The radial distortion model says that coordinates in the observed images are displaced away(barrel distortion) or towards (pincushion distortion) the image center by an amount proportional to their radial distance

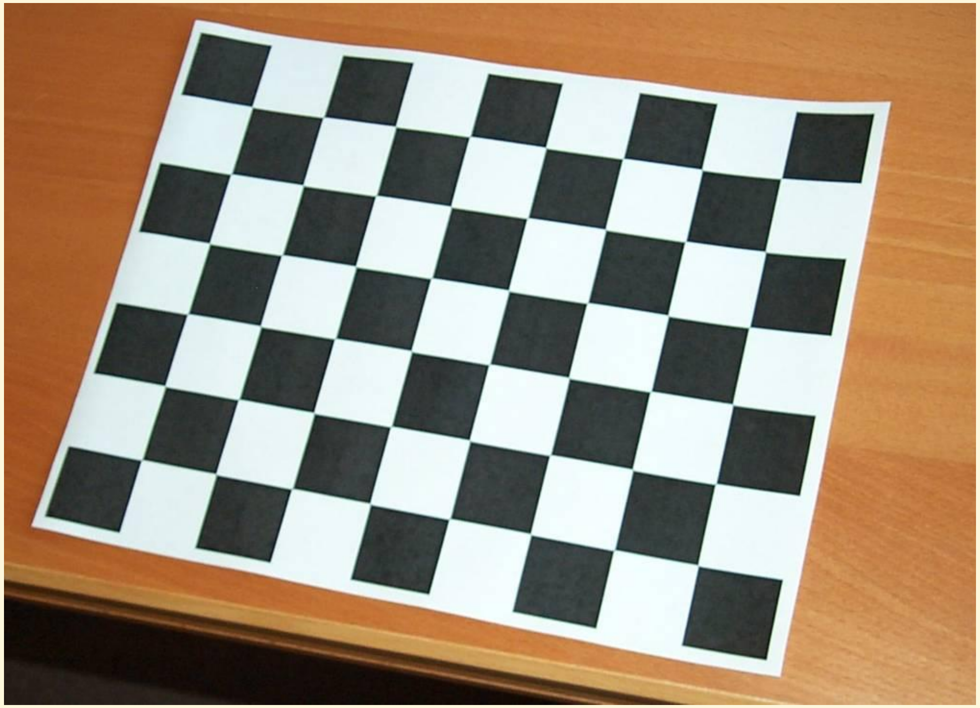


Corrected Barrel Distortion

Image from Martin Habbecke

Camera Calibration: Problem Statement

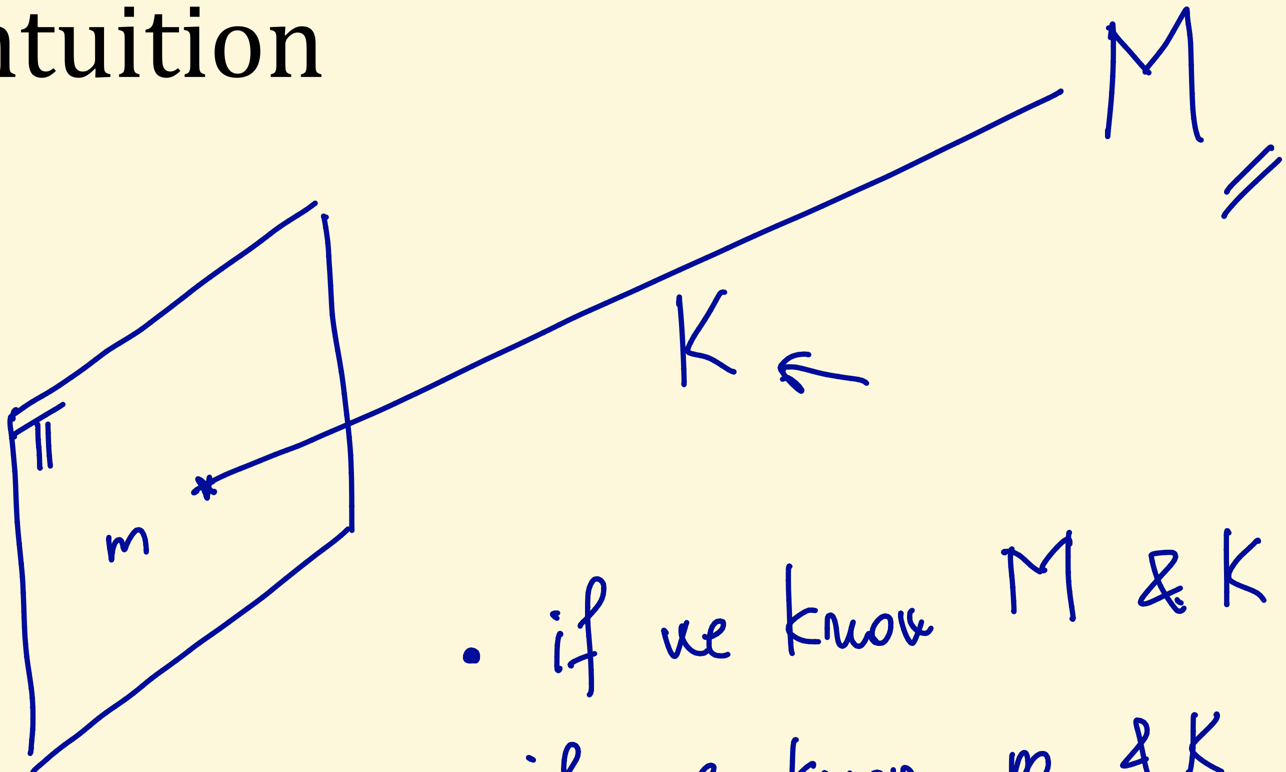
Given one or more images of a calibration pattern, estimate the camera intrinsic (4 or more) and extrinsic parameters (6) using only observed camera data.



Motivation

- Good calibration is important when we need to:
  - Reconstruct a world model: virtual navigation project
  - Interact with the world
    - Robotics
    - Hand-eye coordination
    - Autonomous driving

Intuition



- if we know M & K $\longrightarrow$ m
- if we know m & K

Camera calibration - I

- Intrinsic parameters of the camera:
  - Position of image center in the image
    - It is typically not at (width/2, height/2) of image
  - Focal length
  - Scaling factors for row and column pixels
  - Skew factor
  - Lens distortion (pin-cushion/barrel effect)

Camera calibration - II

- General strategy:
  - view calibration object; identify image points in image
    - positions are known in some fixed world coordinate system
  - optimization process: *discrepancy between observed image features and their theoretical position is minimized with respect to the camera's intrinsic and extrinsic parameters*
  - obtain camera matrix by minimizing error
  - obtain intrinsic parameters from camera matrix

A Quick Aside: Least Squares

Algebraically, find x s.t. $Ax = b$.

1) A is square
   is invertable] $\Rightarrow$ $x = A^{-1} b$

$$\underset{m \times n}{\underset{A}{\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}}} \underset{n \times 1}{\underset{x}{\begin{bmatrix} \\ \\ \\ \end{bmatrix}}} = \underset{b}{\begin{bmatrix} \\ \\ \\ \end{bmatrix}}$$

2) A is not square
   ↳ $m \times n$

a) $m < n$
   underdetermined
   $\Rightarrow$ solution is not unique

$n \rightarrow$ # parameters
$m \rightarrow$ # data points

b) $m > n$ ⟶ A → "tall" matrix

minimize $\|Ax - b\|$

A Quick Aside: Least Squares

Algebraically, find x s.t. $Ax = b$.

$f(x) = \| Ax - b \|$

error

$= (Ax - b)^T (Ax - b)$

$= x^T A^T A x - x^T A^T b - b^T A x + b^T b$

All 4 terms are scalars

$(x^T A^T b)^T \longrightarrow b^T A x \implies \underline{x^T A^T b} = b^T A x$

$f(x) = x^T A^T A x - 2\, b^T A x + b^T b$

Take derivative

$\dfrac{\partial f(x)}{\partial x} = \; . \; . \; . \; . \; .$

A Quick Aside: Least Squares

Algebraically, find x s.t. $Ax = b$.

$f(x)$ is a function of $x_1, x_2, \ldots, x_n$

$$\frac{\partial}{\partial x} f(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{bmatrix}$$

$$\frac{\partial}{\partial x} x^T b = b$$

$$\Rightarrow \frac{\partial}{\partial x} b^T x = b$$

$$C = \begin{bmatrix} 3 & 2 \\ 2 & 5 \end{bmatrix}$$

$$x^T C x = [x_1 \ x_2] \begin{bmatrix} 3 & 2 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} =$$

$$= 3x_1^2 + 4x_1 x_2 + 5x_2^2$$

$$\frac{\partial}{\partial x_1} = 6x_1 + 4x_2$$

$$\frac{\partial}{\partial x_2} = 4x_1 + 10x_2$$

$$\frac{\partial x^T C x}{\partial x} = 2Cx$$

$$f(x) = x^T A^T A x$$

Least Squares – courtesy of Rene Vidal

A Quick Aside: Least Squares

Trick:

$$Ax \approx b$$
$$A^\top Ax \approx A^\top b$$
$$x \approx (A^\top A)^{-1} A^\top b$$

$$f(x) = x^T A^T A x - 2 x^T A^T b$$

$$\frac{\partial f(x)}{\partial x} = 2 A^T A x - 2 A^T b =$$
$$= 2 A^T (Ax - b) \longrightarrow 0$$

$$A^T A x = A^T b$$

$$x = (A^T A)^{-1} A^T b \quad , \quad (A^T A)^{-1} = \text{pseudoinverse}$$

Matlab $\rightarrow$ pinv

Least Squares – courtesy of Rene Vidal

Camera Calibration

- General strategy:
  - view calibration object
  - identify image points
  - obtain camera matrix by minimizing error
  - obtain intrinsic parameters from camera matrix
- Most modern systems employ the multi-plane method
  - avoids knowing absolute coordinates of calibration points

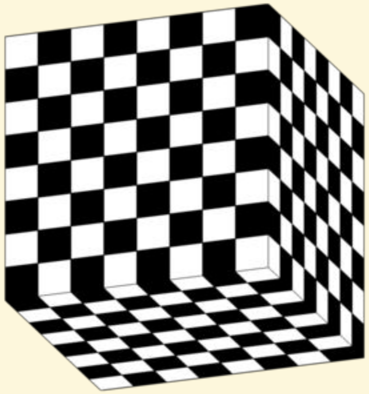
Methods:

- Photogrammetric Calibration
- Self Calibration
- **Multi-Plane Calibration**

Think of it like of a least squares problem (error minimization):

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix}_{pix} = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} & R & & T \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}_{world}$$

Camera Calibration



Methods:

- Photogrammetric Calibration
  - needs rig/cube with pattern
  - very good 3D precision
  - expensive, hard to set up
- Self Calibration
  - move camera in static scene
  - rigidity of the scene helps estimate camera parameters
  - too many to estimate, can give inaccurate results
- **Multi-Plane Calibration**

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix}_{pix} = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} & R & & T \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}_{world}$$

Multi-Plane Calibration

- Hybrid method: Photogrammetric & Self-Calibration.
- Planar pattern imaged multiple times (inexpensive).
- Used widely in practice; many implementations.
- Based on a group of projective transformations called *homographies.*

Notation

$$\widetilde{\mathbf{m}} = [u, v, 1]^T \text{ and } \widetilde{\mathtt{M}} = [X, Y, Z, 1]^T.$$

$$s\tilde{m} = A[R \quad T]\tilde{M}$$

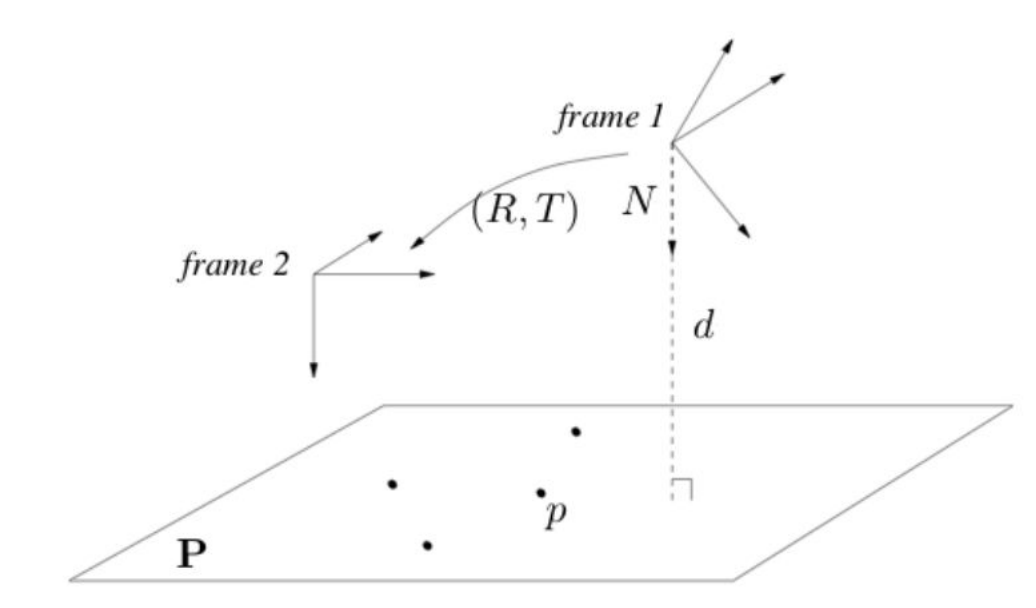
$$\mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

- s = arbitrary scale factor
- (u0, v0) = coordinates of the principal point
- α & β = scale factors in the u and v axes
- γ = skew factor between the two axes

Planar Homographies

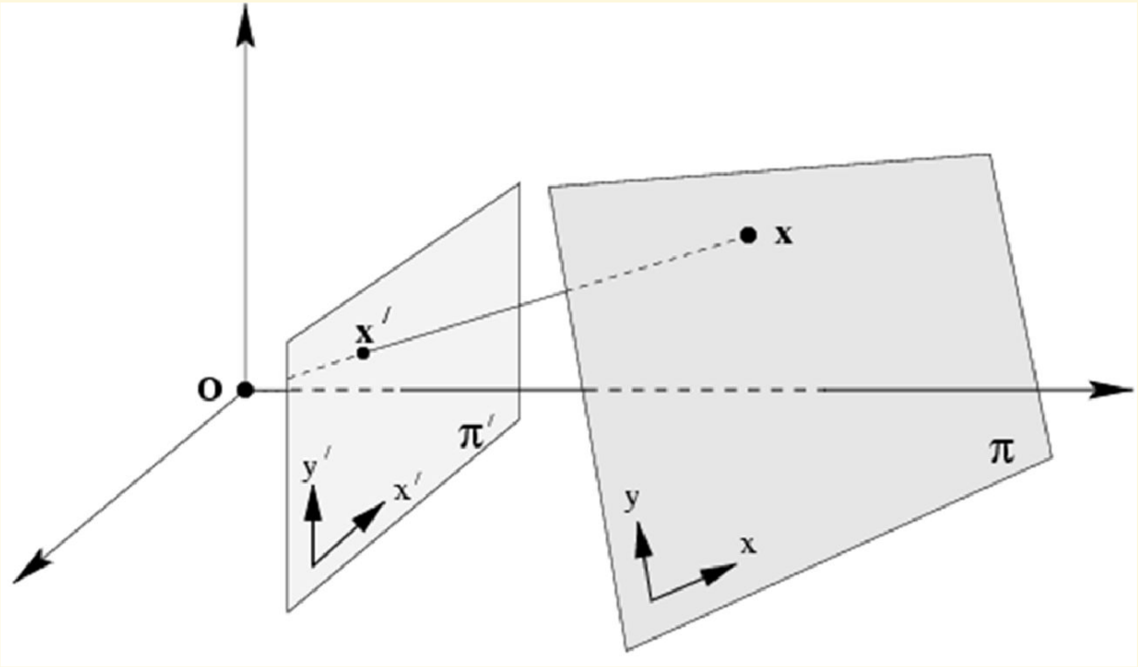
First Fundamental Theorem of Projective Geometry: There exists a unique homography H that performs a change of basis between two projective spaces of the same dimension. $s\tilde{m} = H\tilde{M}$

If two cameras image planar points from the same scene, then those cameras are related through a *homography*.

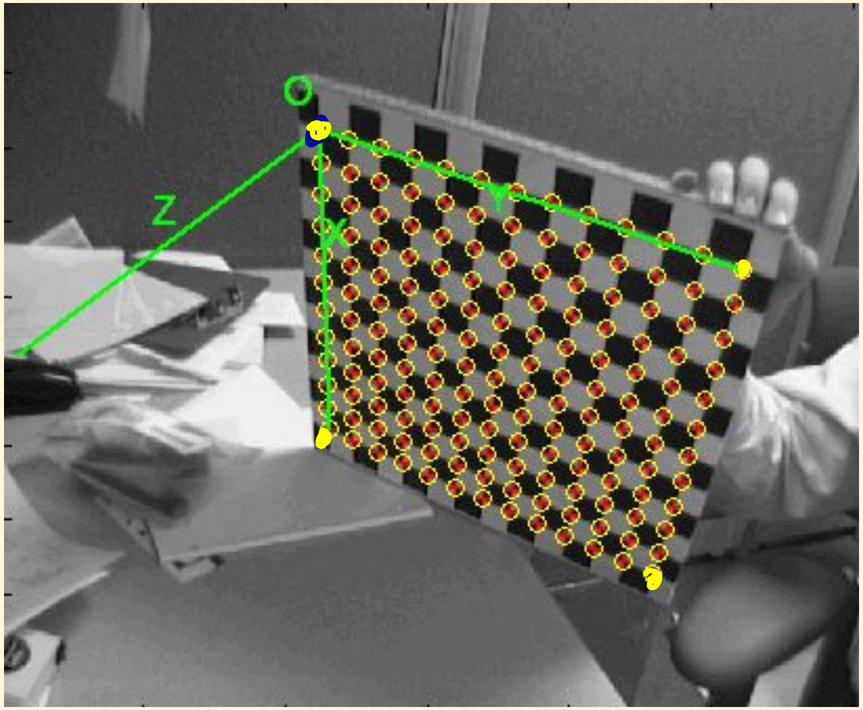


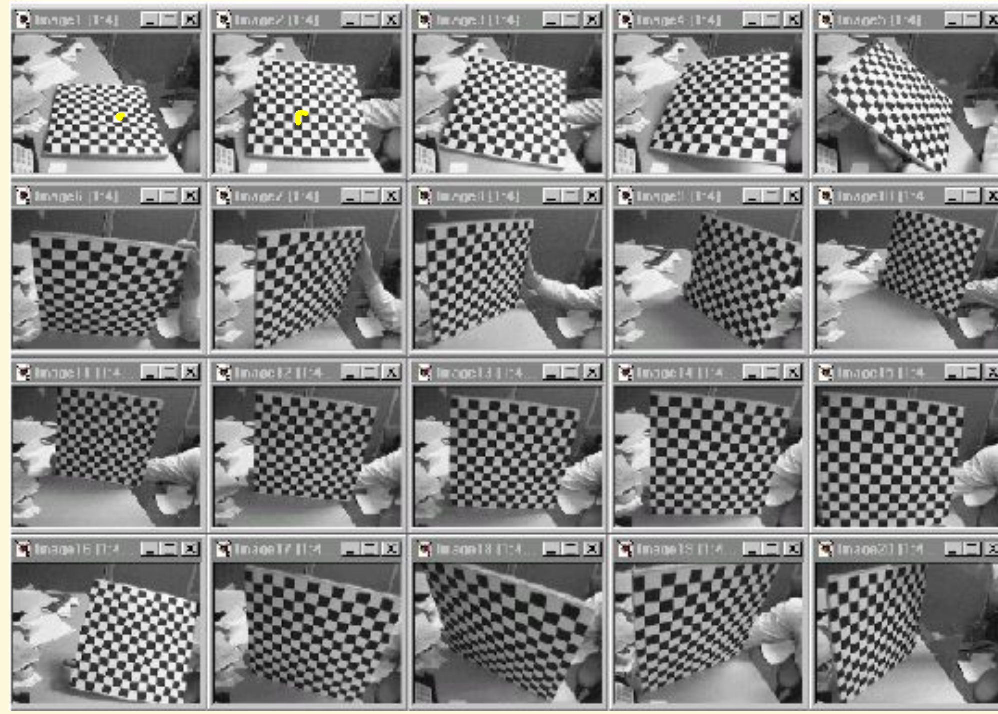
Planar Homographies

- Homography = transformation between 2 spaces of the same dimension
- In this case, 2 planes



Multi-Plane Calibration





Trick: set the world coordinate system to the corner of the checkerboard

Planar Homographies

$$s[u \quad v \quad 1]^T = A[r_1 \quad r_2 \quad r_3 \quad t][X \quad Y \quad Z \quad 1]^T$$

$$s[u \quad v \quad 1]^T = A[r_1 \quad r_2 \quad r_3 \quad t][X \quad Y \quad 0 \quad 1]^T$$

$$s[u \quad v \quad 1]^T = A[r_1 \quad r_2 \quad t][X \quad Y \quad 1]^T$$

$$s[u \quad v \quad 1]^T = H[X \quad Y \quad 1]^T$$

$$s\tilde{m} = H\tilde{M}$$

Multi-Plane Approach In Action

- …if we can get matlab to work…