Lecture 3

Camera Models 2 & Camera Calibration

Professor Silvio Savarese

Computational Vision and Geometry Lab

Lecture 3

Camera Models 2 & Camera Calibration

- Recap of camera models
- Camera calibration problem
- Camera calibration with radial distortion
- Example

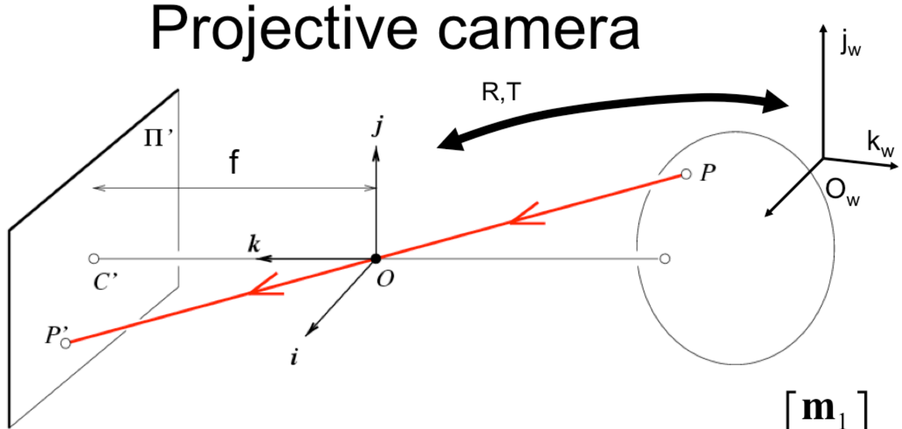
Reading: [FP] Chapter 1 “Geometric Camera Calibration”
[HZ] Chapter 7 “Computation of Camera Matrix P”

Some slides in this lecture are courtesy to Profs. J. Ponce, F-F Li

Silvio Savarese Lecture 3 - 15-Apr-16

In this lecture, we will discuss the topic of camera calibration. We will start with a recap of camera models. Next, we will formulate the camera calibration problem and investigate how to estimate the unknown parameters. We will see how radial distortion affects our calibration process, and how we can model it. Finally, we will end with an example calibration session in MATLAB.

Projective camera

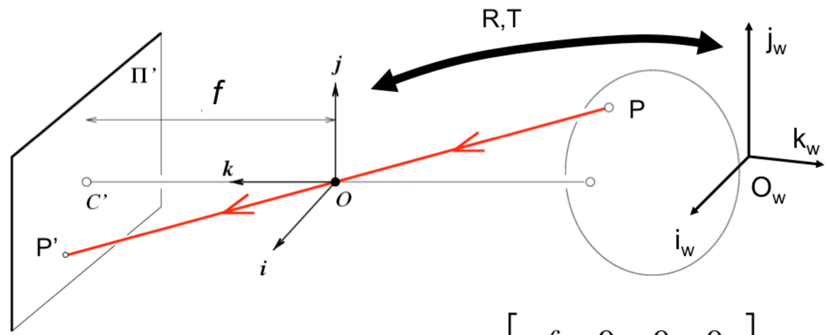


$$P'_{3\times1} = M\,P_w \quad = K_{3\times3}\begin{bmatrix} R & T \end{bmatrix}_{3\times4} P_{w\,4\times1} \qquad M = \begin{bmatrix}\mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3\end{bmatrix}$$

$$= \begin{bmatrix}\mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3\end{bmatrix} P_{\mathrm{W}} = \begin{bmatrix}\mathbf{m}_1 P_{\mathrm{W}} \\ \mathbf{m}_2 P_{\mathrm{W}} \\ \mathbf{m}_3 P_{\mathrm{W}}\end{bmatrix} \xrightarrow{\mathbf{E}} P'_E = \left(\frac{\mathbf{m}_1 P_w}{\mathbf{m}_3 P_w}, \frac{\mathbf{m}_2 P_w}{\mathbf{m}_3 P_w}\right)$$

This slide summarizes the main equation that describes the projective transformation from point P_w in the world coordinate system (in homogenous coordinates) into a point P_E in the image pixels (in Euclidean coordinate systems). This equation was also presented in lecture 2.

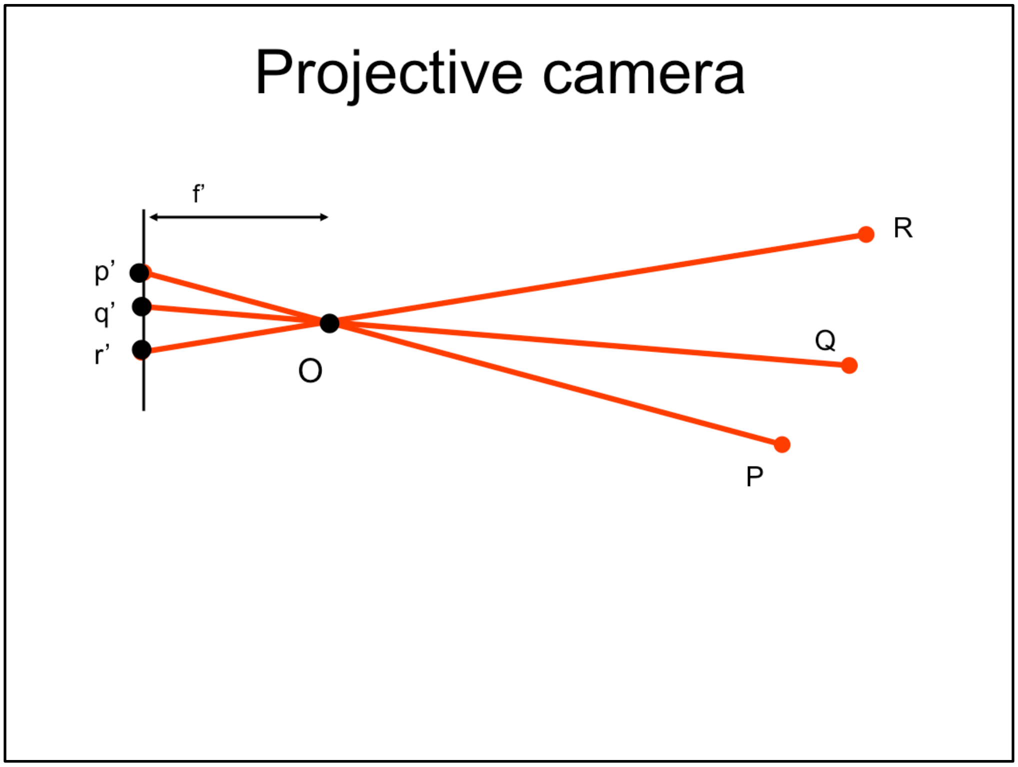
Exercise!



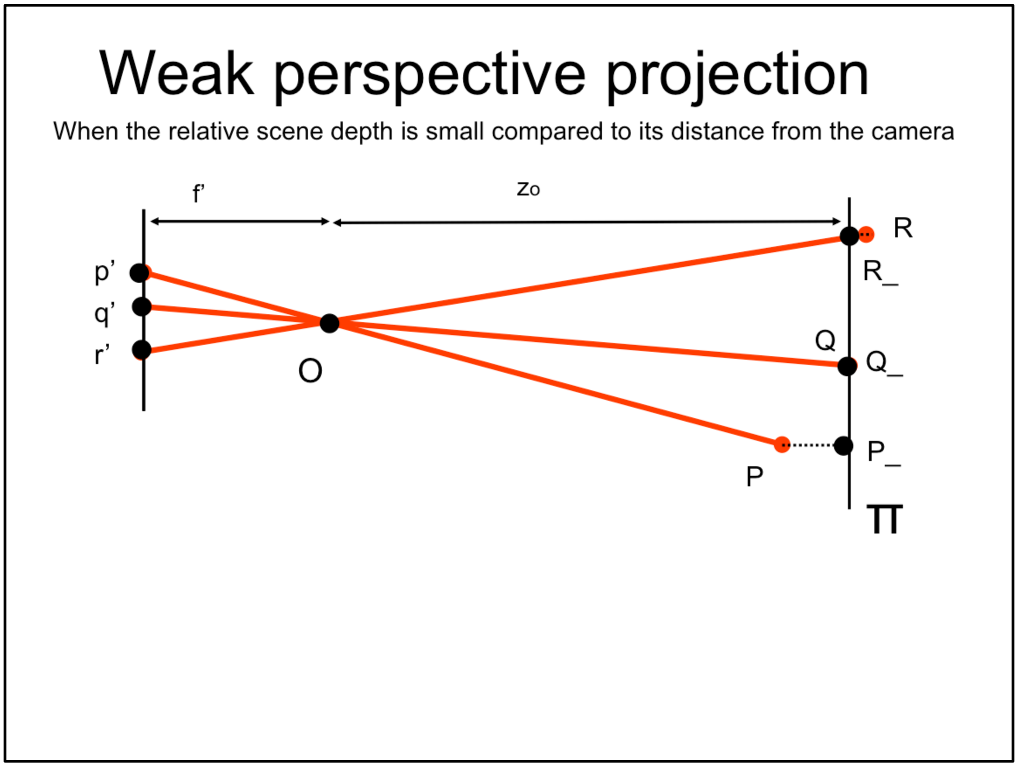
$$M = K\begin{bmatrix} R & T \end{bmatrix} = K\begin{bmatrix} I & 0 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\rightarrow P'_E = \left(\frac{\mathbf{m}_1 P_w}{\mathbf{m}_3 P_w}, \frac{\mathbf{m}_2 P_w}{\mathbf{m}_3 P_w}\right) = \left(f\,\frac{x_w}{z_w}, f\,\frac{y_w}{z_w}\right) \qquad P_w = \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

Let's consider now the following exercise: suppose that the two reference systems coincide ($R=\mathbf{I}_{3x3}$ and $T=\mathbf{0}_{3x1}$) and the camera model has focal length *f*, zero-skew, no offset and square pixels. Can we write a simplified expression for P'_E?
Notice that the result is exactly what we derived for the pinhole camera in lecture 2.



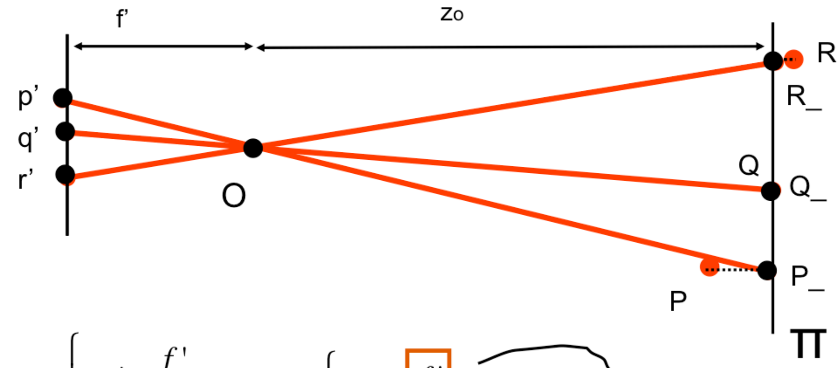
So far, we have discussed a projective camera model. Given a point in the world, its image coordinates depends on its depth.



We will now describe a simpler model known as the *Weak Perspective Projection* model.
In the weak perspective points are: 1) first projected to a reference plane (using orthogonal projection) and 2) then projected to the image plane using a projective transformation.

1) As the slide shows, given a reference plane π at a distance z_0 from the center of the camera, the 3 points P, Q and R are first projected to the plane π using an orthogonal projection; this generates the points R_, Q_ and P_.
(this is equivalent to assigning the z-coordinate of the points P, Q and R to z_o)
This is a reasonable approximation when deviations in depth from the plane are small compared to the distance from the camera.

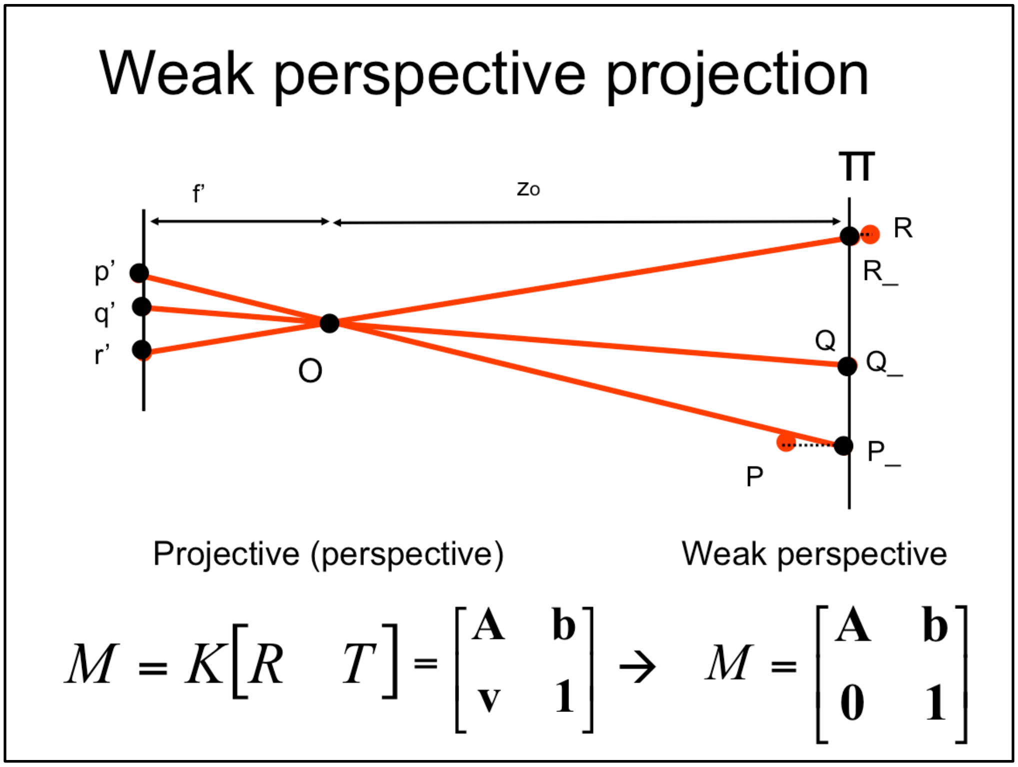
Weak perspective projection



$$\begin{cases} x' = \dfrac{f'}{z}x \\ y' = \dfrac{f'}{z}y \end{cases} \rightarrow \begin{cases} x' = \boxed{\dfrac{f'}{z_0}}x \\ y' = \dfrac{f'}{z_0}y \end{cases}$$

Magnification m

2) Then, the points R_, Q_ and P_ are projected to the image plane using a regular projective transformation producing the points p', q', r'. Notice, however, that because we have approximated the depth of each point to $z_{0,}$ the projection has been reduced to a simple (constant) magnification. The magnification is equal to the focal length f' (which is a constant) is divided by z_0 (which is also a constant).



This also simplifies the projection matrix **M**. In general, **M** is expressed as a block matrix [**A** **b**; **v** 1], whereby **A** is 2x3, **b** is 2x1 and **v** is 1x3.

In the weak perspective case, the last row of M is [0 0 0 1] (written as [**0** 1] using the block matrix notation above). We do not prove this result and leave it to students as an exercise. We can observe this simplification algebraically in the next slide.

$$P' = M\,P_w = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} P_W = \begin{bmatrix} \mathbf{m}_1 P_W \\ \mathbf{m}_2 P_W \\ \mathbf{m}_3 P_W \end{bmatrix} \qquad M = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{v} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

$$\xrightarrow{\mathbf{E}} \left(\frac{\mathbf{m}_1 P_w}{\mathbf{m}_3 P_w}, \frac{\mathbf{m}_2 P_w}{\mathbf{m}_3 P_w}\right)$$

Perspective

$$P' = M\,P_w = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} P_W = \begin{bmatrix} \mathbf{m}_1 P_W \\ \mathbf{m}_2 P_W \\ 1 \end{bmatrix} \qquad M = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} = \begin{bmatrix} & \mathbf{m}_1 & & \\ & \mathbf{m}_2 & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\xrightarrow{\mathbf{E}} (\mathbf{m}_1 P_w, \mathbf{m}_2 P_w)$$

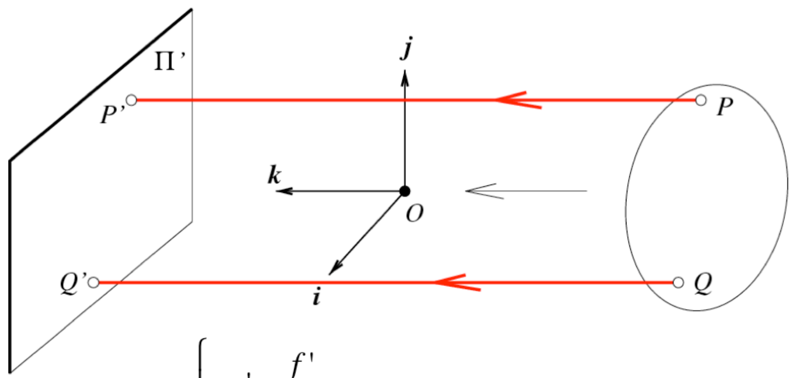
magnification

Weak perspective

In the equations above, m_1, m_2, and m_3 denote the rows of the projection matrix. In our full perspective model, m_3 is equal to [v 1], where v is some non-zero 1x3 vector. On the other hand, m_3 = [0 0 0 1] for the weak perspective model. This results in the denominator term $m_3 P_w$ to be 1. As result, the non linearity of the projective transformation disappears and the weak perspective transformation acts as a mere magnifier.

Orthographic (affine) projection

Distance from center of projection to image plane is infinite



$$\begin{cases} x' = \dfrac{f'}{z} x \\ y' = \dfrac{f'}{z} y \end{cases} \rightarrow \begin{cases} x' = x \\ y' = y \end{cases}$$

Further simplification leads to the orthographic (or affine) projection model. In this case, the optical center is located at infinity. The projection rays are now perpendicular to the retinal plane (parallel to the optical axis). As a result, this model ignores depth altogether. It's often used for architecture and industrial design.

Pros and Cons of These Models

- Weak perspective results in much simpler math.
  - Accurate when object is small and distant.
  - Most useful for recognition.
- Pinhole perspective is much more accurate for modeling the 3D-to-2D mapping.
  - Used in structure from motion or SLAM.

Weak perspective projection

The Kangxi Emperor's Southern Inspection Tour (1691-1698) By Wang Hui

Painted by Wang Hui (1632-1717) and assistants, it was executed before Western perspective was introduced into Chinese art. In the video D. Hockney comments the properties of the prospective geometry depicted in the painting.

Weak perspective projection

The Kangxi Emperor's Southern Inspection Tour (1691-1698) By Wang Hui

Lecture 3

Camera Calibration

- Recap of camera models
- Camera calibration problem
- Camera calibration with radial distortion
- Example

Reading: [FP] Chapter 1 “Geometric Camera Calibration”
[HZ] Chapter 7 “Computation of Camera Matrix P”

Some slides in this lecture are courtesy to Profs. J. Ponce, F-F Li

Silvio Savarese   Lecture 3 -   15-Apr-16

In this section, we will introduce the problem of camera calibration and discuss its importance. Next, we will look at some common methods for solving the calibration problem.

Projective camera

$$P' = M\,P_w = K\begin{bmatrix} R & T \end{bmatrix} P_w$$

Internal parameters

External parameters

$$\mathcal{M} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta}\boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{pmatrix}_{3\times 4}$$

$$K = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_o \\ 0 & \dfrac{\beta}{\sin\theta} & v_o \\ 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \qquad T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Recall that our camera is modeled using internal, or intrinsic, parameters (*K)* and external, or extrinsic, parameters ([*R T*]). Their product gives us the full 3x4 projection matrix M, shown above in the expanded form. It has 11 degrees of freedom (we discussed that in lecture 2). R is expressed as function of r_1^T, r_2^T and r_3^T which are row vectors. T has coordinates t_x, t_y, t_z

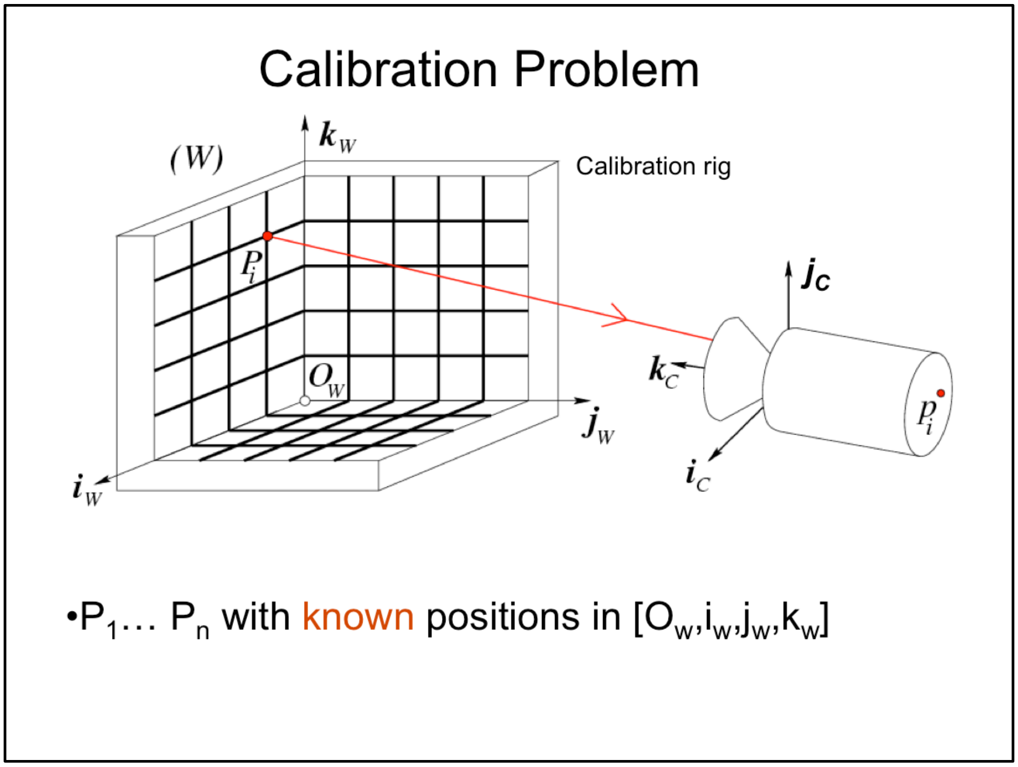
Goal of calibration

$$P' = M\,P_w = \underbrace{K}_{\text{Internal parameters}}\underbrace{\begin{bmatrix} R & T \end{bmatrix}}_{\text{External parameters}} P_w$$

Estimate intrinsic and extrinsic parameters from 1 or multiple images

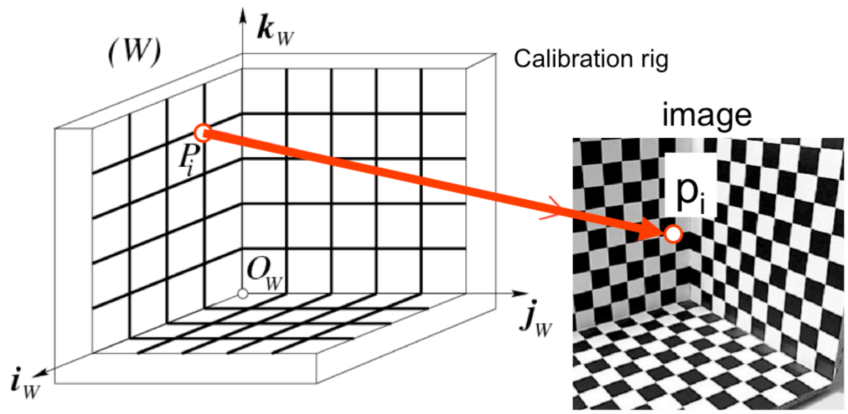
Change notation:
$P = P_w$
$p = P'$

We now pose the following problem: given one or more images taken by a camera, estimate its intrinsic and extrinsic parameters. Notice that from this point on we change the notation for P_w which now is denoted as P, and P' which is now denoted as p.



We can describe this problem more precisely using a calibration rig, such as the one shown above. The rig usually consists of a pattern (usually a checkerboard) with known dimensions. The rig also defines our world reference coordinate frame ($[O_w,i_w,j_w,k_w]$ as shown in the figure.

Calibration Problem

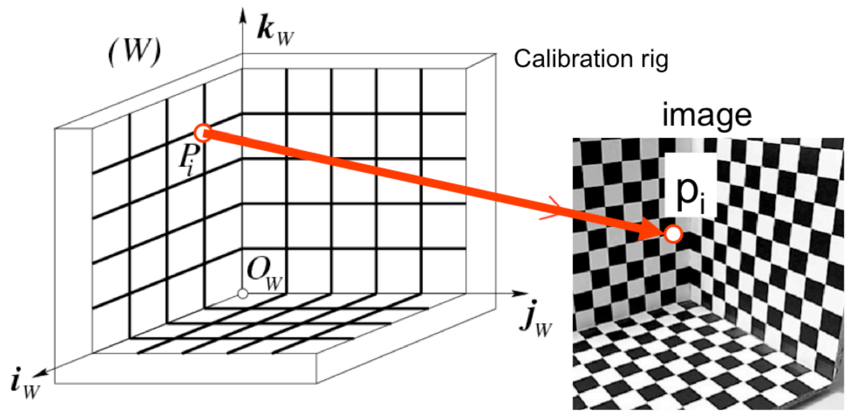


- $P_1 \dots P_n$ with known positions in $[O_w, i_w, j_w, k_w]$
- $p_1, \dots p_n$ known positions in the image

Goal: compute intrinsic and extrinsic parameters

Now, suppose we are given *n* points on the calibration rig, $P_1 \dots P_n$, along with their corresponding coordinates, $p_1, \dots p_n$ in the image. Using these correspondences, our goal is to estimate both intrinsic and extrinsic parameters. Notice that $P_1 \dots P_n$ as well as $p_1, \dots p_n$ are known!

Calibration Problem

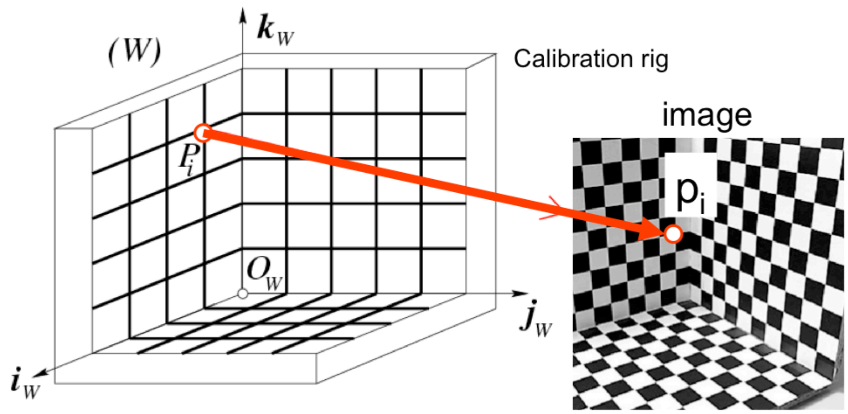


How many correspondences do we need?

• M has 11 unknowns • We need 11 equations • 6 correspondences would do it

How many of such correspondences would we need to compute both intrinsics and extrinsics? Our projection matrix has 11 degrees of freedom (5 for the intrinsics and 6 for the extrinsics), so we need at least 11 equations. Each correspondence gives us two equations (one for x and y each). Therefore, we would need at least 6 correspondences.

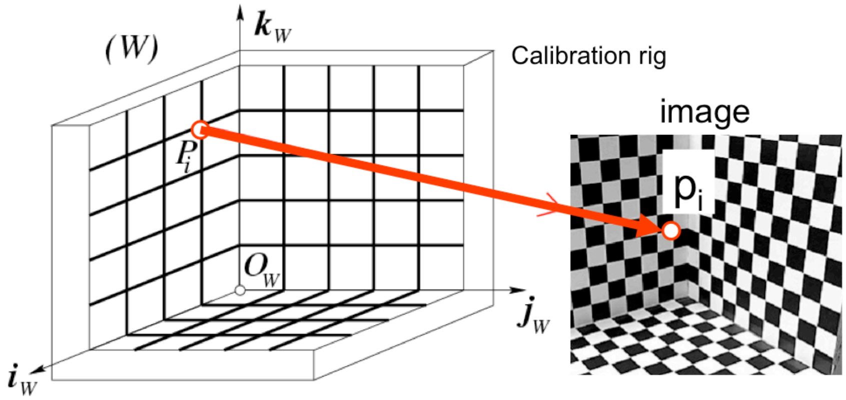
Calibration Problem



In practice, using more than 6 correspondences enables more robust results

As always, our correspondences are not perfect and susceptible to noise. Having more than the minimal number of correspondences allows us to be more robust to these imprecisions.

Calibration Problem



$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{m}_1 \, P_i}{\mathbf{m}_3 \, P_i} \\ \dfrac{\mathbf{m}_2 \, P_i}{\mathbf{m}_3 \, P_i} \end{bmatrix} = M \, P_i \qquad M = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

in pixels

[Eq. 1]

In the next slides, we will set up a linear system of equations from n of such correspondences and propose a procedure for solving it against the unknowns (intrinsic and extrinsic parameters). Let's consider the i^{th} correspondence pair defined by a point P_i in the calibration rig (in the world reference system) and its observation p_i in the image. u_i and v_i are the coordinates of p_i (measured in pixels and in Euclidean coordinates); the relationship between u_i and v_i and P_i is expressed by (Eq. 1) as also illustrated in the following slide.

Calibration Problem

[Eq. 1]

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \dfrac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix}$$

$$u_i = \frac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \rightarrow u_i(\mathbf{m}_3 P_i) = \mathbf{m}_1 P_i \rightarrow u_i(\mathbf{m}_3 P_i) - \mathbf{m}_1 P_i = 0$$

$$v_i = \frac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \rightarrow v_i(\mathbf{m}_3 P_i) = \mathbf{m}_2 P_i \rightarrow v_i(\mathbf{m}_3 P_i) - \mathbf{m}_2 P_i = 0$$

[Eqs. 2]

From Eq.1, we can derive a pair of equations that relate u_i with P_i and v_i with P_i (the system of Eqs. 2)

Calibration Problem

$$\begin{cases} u_1(\mathbf{m}_3\, P_1) - \mathbf{m}_1\, P_1 = 0 \\ v_1(\mathbf{m}_3\, P_1) - \mathbf{m}_2\, P_1 = 0 \\ \vdots \\ u_i(\mathbf{m}_3\, P_i) - \mathbf{m}_1\, P_i = 0 \\ v_i(\mathbf{m}_3\, P_i) - \mathbf{m}_2\, P_i = 0 \\ \vdots \\ u_n(\mathbf{m}_3\, P_n) - \mathbf{m}_1\, P_n = 0 \\ v_n(\mathbf{m}_3\, P_n) - \mathbf{m}_2\, P_n = 0 \end{cases}$$

[Eqs. 3]

Assuming that we have *n* of such correspondences, we can set up a system of *2n* equations as illustrated above [Eqs. 3].

Block Matrix Multiplication

$$\mathrm{A} = \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix} \qquad \mathrm{B} = \begin{bmatrix} \mathrm{B}_{11} & \mathrm{B}_{12} \\ \mathrm{B}_{21} & \mathrm{B}_{22} \end{bmatrix}$$

What is *AB* ?

$$AB = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

Before we move forward, let's briefly discuss the concept of block matrix multiplication (from matrix theory 101). In order to make computation more concise and efficient, large matrices such as A or B can be partitioned into smaller blocks (or *submatrices*) – for instance the block A_{ij}. Given two such block-structured matrices (such as A and B), we can define their product (e.g. A B) in terms of their submatrices (assuming that the partitioning is such that the product is feasible) as shown n the example. This way of handling matrices is very compact and helps us make the ensuing derivation easier to describe.

Calibration Problem

$$\begin{cases} -u_1(\mathbf{m}_3\ P_1)+\mathbf{m}_1\ P_1 = 0 \\ -v_1(\mathbf{m}_3\ P_1)+\mathbf{m}_2\ P_1 = 0 \\ \vdots \\ -u_n(\mathbf{m}_3\ P_n)+\mathbf{m}_1\ P_n = 0 \\ -v_n(\mathbf{m}_3\ P_n)+\mathbf{m}_2\ P_n = 0 \end{cases} \longrightarrow \boxed{\overset{\text{known}}{\mathbf{P}}\ \overset{\text{unknown}}{\mathbf{m}} = 0} \quad \text{[Eq. 4]}$$

Homogenous linear system

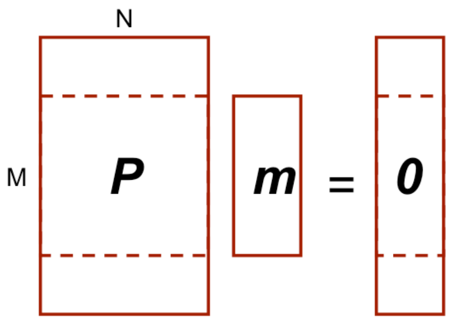
$$\mathbf{P} \overset{\text{def}}{=} \begin{pmatrix} \mathrm{P}_1^T & \mathbf{0}^T & -u_1\overset{1\text{x}4}{\boxed{\mathrm{P}_1^T}} \\ \mathbf{0}^T & \mathrm{P}_1^T & -v_1\mathrm{P}_1^T \\ & \vdots & \\ \mathrm{P}_\mathrm{n}^T & \mathbf{0}^T & -u_n\mathrm{P}_\mathrm{n}^T \\ \mathbf{0}^T & \mathrm{P}_\mathrm{n}^T & -u_n\mathrm{P}_\mathrm{n}^T \end{pmatrix}_{2\text{n x }12} \qquad \boldsymbol{m} \overset{\text{def}}{=} \begin{pmatrix} \overset{4\text{x}1}{\boxed{\mathbf{m}_1^\mathrm{T}}} \\ \mathbf{m}_2^\mathrm{T} \\ \mathbf{m}_3^\mathrm{T} \end{pmatrix}_{12\text{x}1}$$

Returning to the camera calibration problem, the 2n equations obtained from the correspondences constitute a homogeneous linear system [Eqs. 3], which we can express it in terms of a matrix equation of the form **Pm = 0** [Eq. 4]**.** **P** comprises the known coefficients (from our correspondences) while **m** comprises the unknown parameters we wish to estimate.

Using block matrix notation, we can concisely write down **P** and **m** as shown above. Note that **m** is a vectorized version of the 3x4 projection matrix, where each 4x1 block corresponds to a row in the original matrix.

Homogeneous M x N Linear Systems

M=number of equations = 2n
N=number of unknown = 11



Rectangular system (M>N)

- 0 is always a solution
- To find non-zero solution

Minimize $|\mathbf{P}\ \mathbf{m}|^2$

under the constraint $|\mathbf{m}|^2 = 1$

When *2n>11*, our homogeneous linear system is overdetermined. For such a system, zero is always a (trivial) solution. Furthermore, if **m** is a solution, then so is ***k*** **m**, where ***k*** is an arbitrary scaling factor. Therefore, to constrain our optimization, we minimize $|\mathbf{P}\ \mathbf{m}|^2$ subject to the constraint that $|\mathbf{m}|^2 = 1$ (see linear algebra review session for details). The boxes around **P, m** and **0** give a qualitative illustration of the dimensionality of the matrix associated to each of the variables. The dashed sides illustrate a situation where M=N.

Calibration Problem

$$\mathbf{P}\ \mathbf{m} = 0$$

- How do we solve this homogenous linear system?
- Via SVD decomposition!

Recall that for homogeneous linear system such as **Pm = 0**, the SVD gives us the least-squares solution, subject to |**m| = 1.**

Calibration Problem

$$\boxed{\mathbf{P}}\,\mathbf{m} = 0$$

SVD decomposition of P

$$\boxed{U_{2n\times 12}\ D_{12\times 12} V^{T}_{12\times 12}}$$

Last column of V gives $\boldsymbol{m}$

Why? See pag 592 of HZ

$$\boldsymbol{m} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{m}_1^{\mathrm{T}} \\ \mathbf{m}_2^{\mathrm{T}} \\ \mathbf{m}_3^{\mathrm{T}} \end{pmatrix}$$

$$\boldsymbol{m} \longrightarrow M$$

We can obtain the least-squares solution for **m** by factorizing **P** to $\mathbf{UDV}^{\mathbf{T}}$ using SVD and then taking the last column of **V**. The derivation as for why this is true goes beyond the scope of this lecture. Please refer to Sec. A5.3 HZ (Pag. 592, 593) for details.

Once **m** is computed, we can repackage it into M and compute all the camera parameters as we'll see next;

Extracting camera parameters

$$M = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \frac{\beta}{\sin\theta}\boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \frac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{pmatrix} \rho$$

Once we have estimated the combined parameters in **m**, the next step is to extract the actual intrinsics and extrinsics from it.

Note that M has 11 degrees of freedom and can be computed up to a scale parameter. We explicitly identify the scale parameter as ρ. Notice that ρ it's not a real unknown. We can estimate ρ by using the fact that |**m**| must be =1 (or, equivalently, the Frobenius norm of M = 1).

Extracting camera parameters

See [FP], Sec. 1.3.1

$$\frac{M}{\rho} = \left(\begin{array}{c|c} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta} \boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{array} \right) = \frac{K}{\rho}\left[\, R \quad T \,\right]$$

$$\underbrace{}_{A} \qquad \underbrace{}_{\mathbf{b}}$$

$$K = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_o \\ 0 & \dfrac{\beta}{\sin\theta} & v_o \\ 0 & 0 & 1 \end{bmatrix}$$

Box 1

$$\mathrm{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \mathbf{a}_3^T \end{bmatrix} \qquad \mathrm{b} = \begin{bmatrix} \mathrm{b}_1 \\ \mathrm{b}_2 \\ \mathrm{b}_3 \end{bmatrix}$$

Estimated values

Intrinsic

$$\rho = \frac{\pm 1}{|\mathbf{a}_3|} \qquad u_o = \rho^2 (\mathbf{a}_1 \cdot \mathbf{a}_3) \qquad \mathrm{v}_o = \rho^2 (\mathbf{a}_2 \cdot \mathbf{a}_3)$$

$$\cos\theta = \frac{(\mathbf{a}_1 \times \mathbf{a}_3) \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}{|\mathbf{a}_1 \times \mathbf{a}_3| \cdot |\mathbf{a}_2 \times \mathbf{a}_3|}$$

It is relatively straightforward to derive all the intrinsics and extrinsics from the 3x4 matrix M (whose entries we estimated in the previous slides). We refer to [FP], Sec. 1.3.1, for the details or leave it as an exercise. To avoid overcomplicating the notation, we rename M/ρ as [**A** **b**], and provide a solution for the intrinsics and extrinsics as function of the elements of **A** and **b** (as defined in the box 1 in the slide).
A solution for the scale ρ, offset and skew are reported in the slide.

Theorem (Faugeras, 1993)

Let $\mathcal{M} = (\mathcal{A} \quad \boldsymbol{b})$ be a 3×4 matrix and let $\boldsymbol{a}_i^T$ $(i = 1, 2, 3)$ denote the rows of the matrix $\mathcal{A}$ formed by the three leftmost columns of $\mathcal{M}$.

- A necessary and sufficient condition for $\mathcal{M}$ to be a perspective projection matrix is that $\mathrm{Det}(\mathcal{A}) \neq 0$.
- A necessary and sufficient condition for $\mathcal{M}$ to be a zero-skew perspective projection matrix is that $\mathrm{Det}(\mathcal{A}) \neq 0$ and
$$(\boldsymbol{a}_1 \times \boldsymbol{a}_3) \cdot (\boldsymbol{a}_2 \times \boldsymbol{a}_3) = 0.$$
- A necessary and sufficient condition for $\mathcal{M}$ to be a perspective projection matrix with zero skew and unit aspect-ratio is that $\mathrm{Det}(\mathcal{A}) \neq 0$ and
$$\begin{cases} (\boldsymbol{a}_1 \times \boldsymbol{a}_3) \cdot (\boldsymbol{a}_2 \times \boldsymbol{a}_3) = 0, \\ (\boldsymbol{a}_1 \times \boldsymbol{a}_3) \cdot (\boldsymbol{a}_1 \times \boldsymbol{a}_3) = (\boldsymbol{a}_2 \times \boldsymbol{a}_3) \cdot (\boldsymbol{a}_2 \times \boldsymbol{a}_3). \end{cases}$$

Interestingly, the equation that provides a solution for the skew angle theta, supplies an easy way for verifying the second claim of the Faugeras theorem that we introduced earlier.

Extracting camera parameters

$$\frac{\mathcal{M}}{\rho} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \frac{\beta}{\sin\theta}\boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \frac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{pmatrix} = \mathrm{K}\begin{bmatrix} \mathrm{R} & \mathrm{T} \end{bmatrix}$$

A   **b**

$$\mathrm{A} = \begin{bmatrix} \mathbf{a}_1^{\mathrm{T}} \\ \mathbf{a}_2^{\mathrm{T}} \\ \mathbf{a}_3^{\mathrm{T}} \end{bmatrix} \qquad \mathrm{b} = \begin{bmatrix} \mathrm{b}_1 \\ \mathrm{b}_2 \\ \mathrm{b}_3 \end{bmatrix}$$

Estimated values

Intrinsic

$$\alpha = \rho^2 |\mathbf{a}_1 \times \mathbf{a}_3| \sin\theta$$

$$\beta = \rho^2 |\mathbf{a}_2 \times \mathbf{a}_3| \sin\theta$$

Here we have solution for alpha and beta and, thus, for the focal length.

Extracting camera parameters

$$\frac{\mathcal{M}}{\rho} = \begin{pmatrix} \underbrace{\begin{matrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T \\ \frac{\beta}{\sin\theta} \boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T \\ \boldsymbol{r}_3^T \end{matrix}}_{\mathrm{A}} & \underbrace{\begin{matrix} \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \frac{\beta}{\sin\theta} t_y + v_0 t_z \\ t_z \end{matrix}}_{\mathbf{b}} \end{pmatrix} = \mathrm{K}\begin{bmatrix} \mathrm{R} & \mathrm{T} \end{bmatrix}$$

$$\mathrm{A} = \begin{bmatrix} \mathbf{a}_1^{\mathrm{T}} \\ \mathbf{a}_2^{\mathrm{T}} \\ \mathbf{a}_3^{\mathrm{T}} \end{bmatrix} \qquad \mathrm{b} = \begin{bmatrix} \mathrm{b}_1 \\ \mathrm{b}_2 \\ \mathrm{b}_3 \end{bmatrix}$$

Estimated values

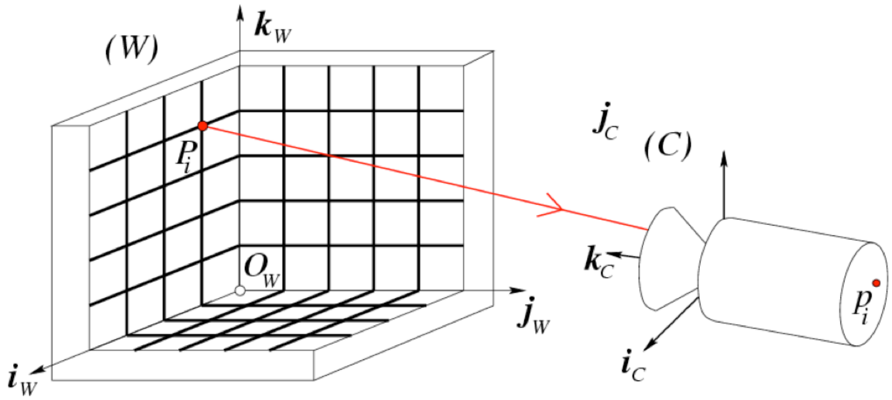
Extrinsic

$$\mathbf{r}_1 = \frac{(\mathbf{a}_2 \times \mathbf{a}_3)}{|\mathbf{a}_2 \times \mathbf{a}_3|} \qquad \mathbf{r}_3 = \frac{\pm \mathbf{a}_3}{|\mathbf{a}_3|}$$

$$\mathbf{r}_2 = \mathbf{r}_3 \times \mathbf{r}_1 \qquad \mathrm{T} = \rho\, \mathrm{K}^{-1} \mathbf{b}$$

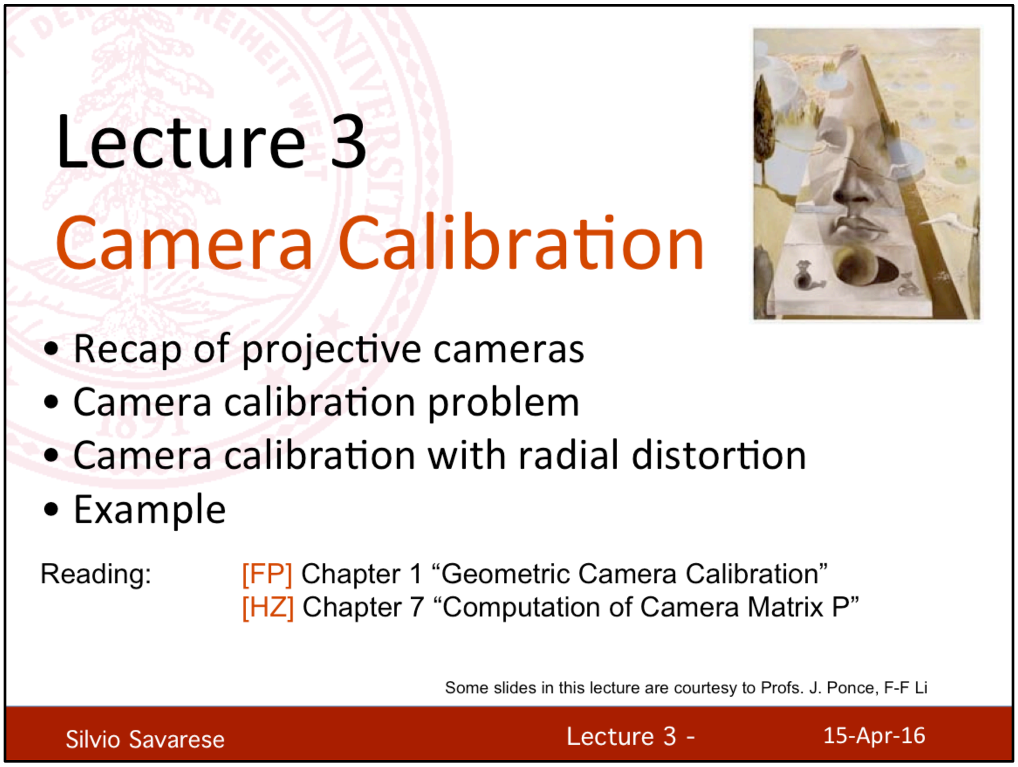
Here we have solutions for the extrinsics R and T.

Degenerate cases



- P_i's cannot lie on the same plane!
- Points cannot lie on the intersection curve of two quadric surfaces

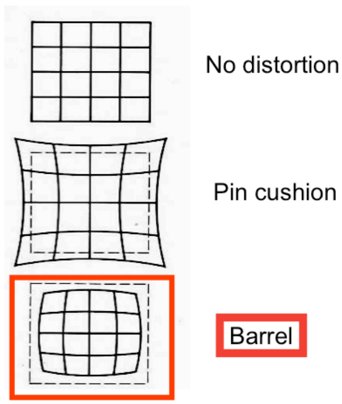
Is the homogenous system introduced in Eq 4 solvable regardless of how points P_i are distributed in 3D? No…. There are certain configurations that won't allow the system to be solved – these are called **degenerate configurations**. For instance, P_i's cannot lie on the same plane. Or points cannot lie on the intersection curve of two quadric surfaces (see [FP] section 1.3, page 25)



We will now discuss the more complex scenario where real world lenses introduce radial distortion. We will analyze a few common types of radial distortions and incorporate them in our model.

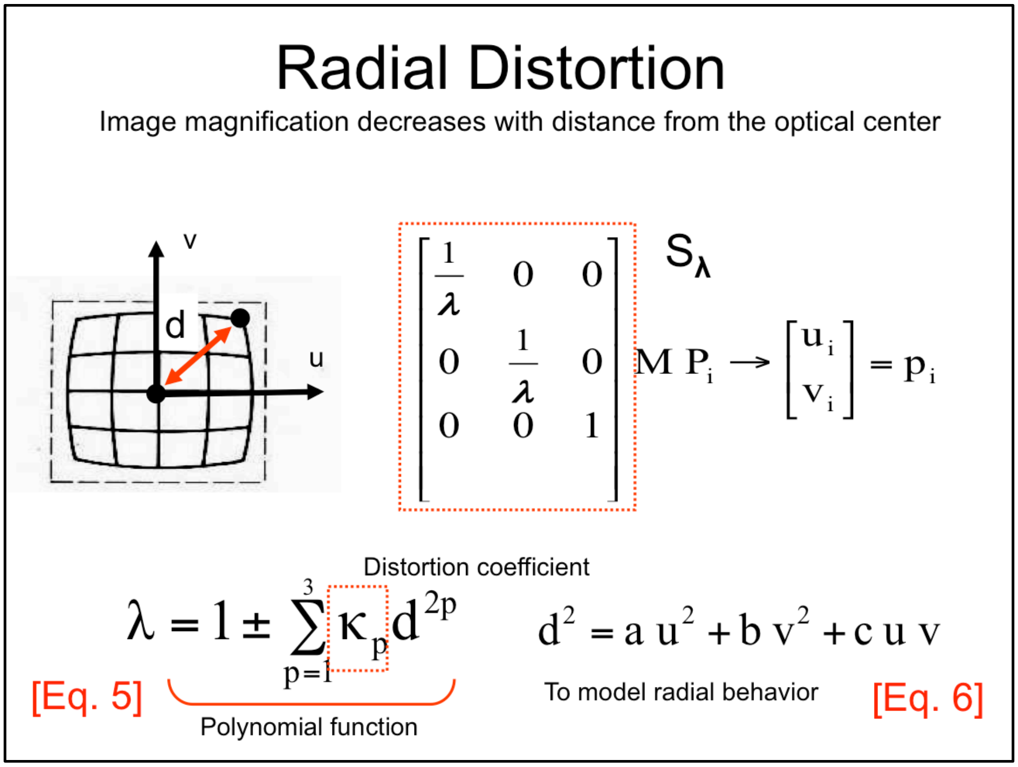
Radial Distortion

- Image magnification (in)decreases with distance from the optical axis
- Caused by imperfect lenses
- Deviations are most noticeable for rays that pass through the edge of the lens



So far, we have been working with ideal lenses which are free from any distortion. However, real lenses can deviate from rectilinear projection. The resulting distortions are often radially symmetric, which can be attributed to the physical symmetry of the lens.

The image above exhibits barrel distortion. This occurs when the image magnification decreases with distance from the optical axis. Fisheye lenses usually produce this type of distortion.



The radial distortion can be modeled using an isotropic transformation S_λ as shown in the slide. This transformation is regulated by the distortion factor $\boldsymbol{\lambda}$. Notice that unlike a traditional scale transformation (of the type [s 0 0;0 s 0; 0 0 1]), where s is constant, $\boldsymbol{\lambda}$ is a function of the distance ***d*** from the center (since the distortion is radially symmetric about the center) and thus function of the coordinates u_i, v_i . This causes the distortion transformation to introduce a non-linearity to the mapping from P_i to p_i

We approximate $\boldsymbol{\lambda}$ using a polynomial expansion (Eq. 5). The resulting coefficients, κ_p, are known as the **distortion coefficients.** In order to model the radial behavior, the distance can be expressed as quadratic function of the coordinates u,v in the image plane (Eq. 6).

Radial Distortion

$$\underbrace{\begin{bmatrix} \frac{1}{\lambda} & 0 & 0 \\ 0 & \frac{1}{\lambda} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{Q\text{ (with } M)} M\, P_i \rightarrow \begin{bmatrix} u_i \\ v_i \end{bmatrix} = p_i \qquad Q = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix}$$

Is this a linear system of equations?

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 P_i}{\mathbf{q}_3 P_i} \\ \dfrac{\mathbf{q}_2 P_i}{\mathbf{q}_3 P_i} \end{bmatrix} \rightarrow \begin{cases} u_i \mathbf{q}_3 P_i = \mathbf{q}_1 P_i \\ v_i \mathbf{q}_3 P_i = \mathbf{q}_2 P_i \end{cases}$$

No! why? [Eqs.7]

We can rewrite our projection equations by using the block matrix notation and defining Q as S_λ M. We follow the same procedure that led to Eq. 1 and Eqs.2 (see previous slides). Unlike for the system defined by Eqs.2, however, the system of equations Eqs. 7 is no longer a linear one.

General Calibration Problem

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 \, \mathrm{P_i}}{\mathbf{q}_3 \, \mathrm{P_i}} \\ \dfrac{\mathbf{q}_2 \, \mathrm{P_i}}{\mathbf{q}_3 \, \mathrm{P_i}} \end{bmatrix} \longrightarrow X = f(Q) \quad \text{[Eq .8]}$$

i=1…n

X: measurements; Q: parameters

f( ) is the nonlinear mapping

-Newton Method

-Levenberg-Marquardt Algorithm

- Iterative, starts from initial solution
- May be slow if initial solution far from real solution
- Estimated solution may be function of the initial solution (because of local minima)
- Newton requires the computation of J, H
- Levenberg-Marquardt doesn't require the computation of H

If n correspondences are available, all these constraints can be packaged into a (non-linear) multi-variable function f which relates all (the unknown) parameters Q with the observations/measurements [Eq .8]. A common way to solve this is to resort to non-linear optimization techniques. Two common ones include the Newton's method and the Levenberg-Marquardt algorithm.

The slide lists some of the advantages and trade-offs of Levenberg-Marquardt. J and H refer to the Jacobian and Hessian, respectively.

For more details about optimization methods, please refer to [FP] Sec. 22.2 (page 669-672) or any reference text books.

General Calibration Problem

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 \mathrm{P_i}}{\mathbf{q}_3 \mathrm{P_i}} \\ \dfrac{\mathbf{q}_2 \mathrm{P_i}}{\mathbf{q}_3 \mathrm{P_i}} \end{bmatrix} \longrightarrow X = f(Q) \quad \text{[Eq .8]}$$

i=1…n

X: measurements; Q: parameters

f( ) is the nonlinear mapping

A possible algorithm

1. Solve linear part of the system to find approximated solution
2. Use this solution as initial condition for the full system
3. Solve full system using Newton or L.M.

A possible simple algorithm that people use in practice is:

- solve the system using n correspondences (as we did for Eq. 4) by assuming that there is no distortion.
- Use this solution as initial condition for solving the problem in Eq. 8.
- Solve the full problem in Eq. 8 using Newton or L.M.

General Calibration Problem

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 \, \mathrm{P_i}}{\mathbf{q}_3 \, \mathrm{P_i}} \\ \dfrac{\mathbf{q}_2 \, \mathrm{P_i}}{\mathbf{q}_3 \, \mathrm{P_i}} \end{bmatrix} \longrightarrow X = f(\mathrm{Q}) \quad \text{[Eq .8]}$$

measurements

parameters

f( ) is the nonlinear mapping

Typical assumptions:

- zero-skew, square pixel
- u_o, v_o = known center of the image

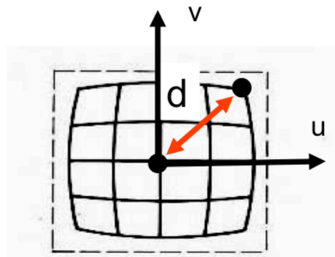
We can simplify the calibration problem if we make certain assumptions. Some typical assumptions include zero skew and square pixels (which is reasonable for many modern cameras), a known image center, and negligible distortion. Under these assumptions, the dimensionality of the problem is reduced and the optimization problem is simplified.

Radial Distortion

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{q}_1 P_i}{\mathbf{q}_3 P_i} \\ \dfrac{\mathbf{q}_2 P_i}{\mathbf{q}_3 P_i} \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \dfrac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \dfrac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix}$$

Can we estimate m_1 and m_2 and ignore the radial distortion?

Hint:



$$\frac{u_i}{v_i} = \text{slope}$$

An alternative and very elegant approach, that doesn't require the use of non-linear optimization, follows next.
We note that the ratio between two coordinates u_i and v_i of the point p_i in the image is not affected by the distortion (remember the barrel distortion acts radially).

Radial Distortion

Tsai [87]

Estimating m_1 and m_2…

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{1}{\lambda}\begin{bmatrix} \dfrac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \dfrac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix} \longrightarrow \frac{u_i}{v_i} = \frac{\dfrac{(\mathbf{m}_1 P_i)}{(\mathbf{m}_3 P_i)}}{\dfrac{(\mathbf{m}_2 P_i)}{(\mathbf{m}_3 P_i)}} = \frac{\mathbf{m}_1 P_i}{\mathbf{m}_2 P_i} \quad \text{[Eq .9]}$$

[Eq .10]

$$\begin{cases} v_1(\mathbf{m}_1 P_1) - u_1(\mathbf{m}_2 P_1) = 0 \\ v_i(\mathbf{m}_1 P_i) - u_i(\mathbf{m}_2 P_i) = 0 \\ \vdots \\ v_n(\mathbf{m}_1 P_n) - u_n(\mathbf{m}_2 P_n) = 0 \end{cases}$$

[Eq .11]

$$L\,\mathbf{n} = 0$$

$\downarrow$

Get $\mathbf{m}_1$ and $\mathbf{m}_2$ by SVD

$$\mathbf{L} \stackrel{\text{def}}{=} \begin{pmatrix} v_1 \mathrm{P}_1^T & -u_1 \mathrm{P}_1^T \\ v_2 \mathrm{P}_2^T & -u_2 \mathrm{P}_2^T \\ \vdots & \vdots \\ v_n \mathrm{P}_n^T & -u_n \mathrm{P}_n^T \end{pmatrix} \qquad \mathbf{n} = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \end{bmatrix}$$

Thus, let's compute this ratio of the each p_i. This leads us to Eq.9.
By assuming that n correspondences are available, we can set up the system in Eq. 10.
Similarly to the derivation we used to compute Eq. 4, we can obtain the homogenous system in Eq. 11. This can be solved using SVD and allows a solution for m_1 and m_2.

Radial Distortion

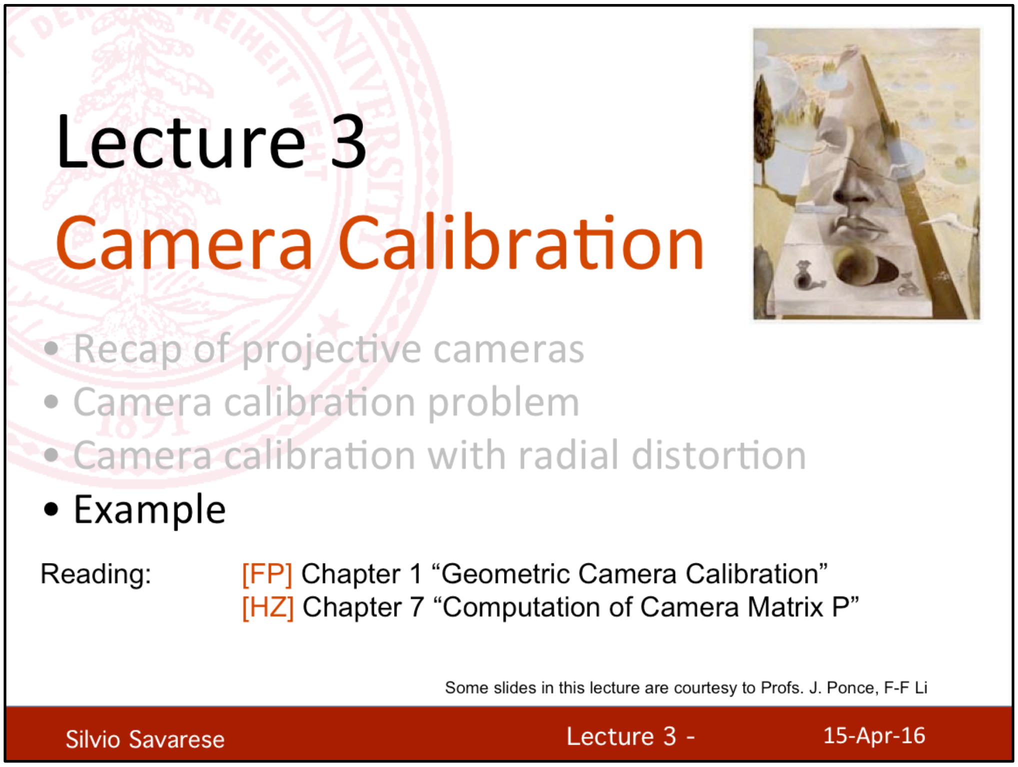
Once that $\mathbf{m}_1$ and $\mathbf{m}_2$ are estimated…

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \dfrac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \dfrac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix}$$

$\mathbf{m}_3$ is non linear function of $\mathbf{m}_1$, $\mathbf{m}_2$, λ

There are some degenerate configurations for which m_1 and m_2 cannot be computed

Once that $\mathbf{m}_1$ and $\mathbf{m}_2$ are estimated $\mathbf{m}_3$ can be expressed a non linear function of $\mathbf{m}_1$ and $\mathbf{m}_2$ and lambda. This still requires to solve an non-optimization problem whose complexity, however, is very much simplified.

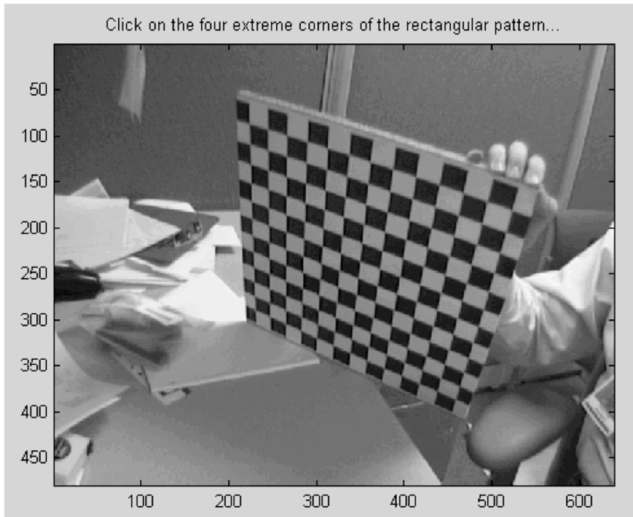


We will now go over an example of camera calibration using MATLAB.

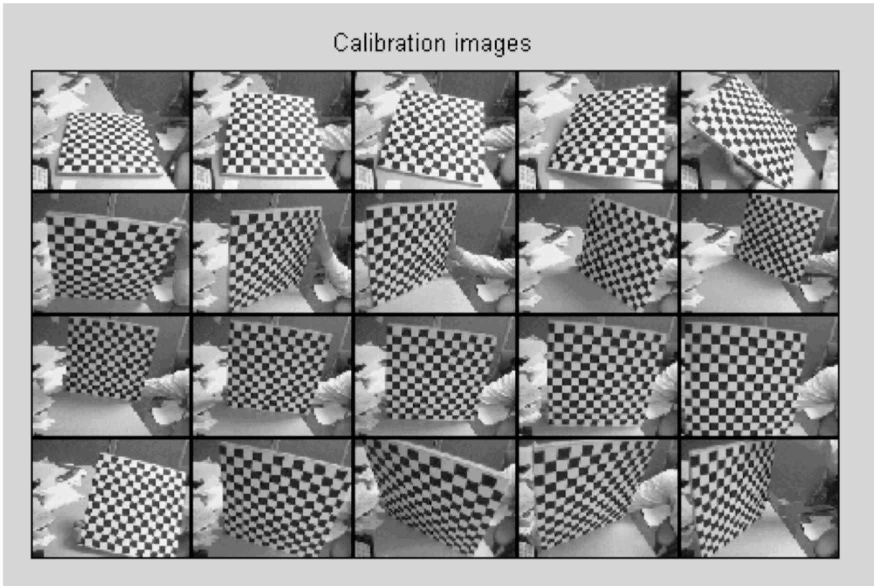
Calibration Procedure

Camera Calibration Toolbox for Matlab
J. Bouguet – [1998-2000]

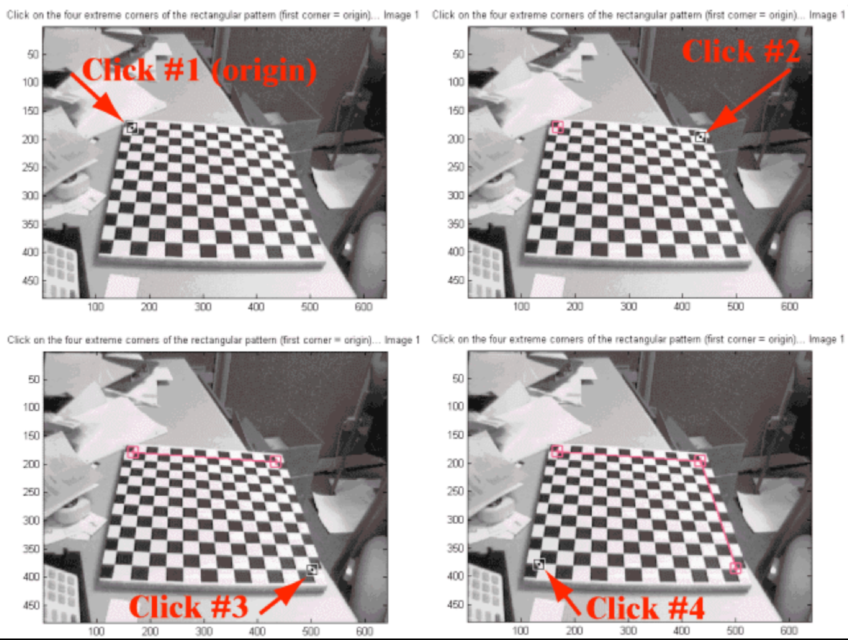
http://www.vision.caltech.edu/bouguetj/calib_doc/index.html#examples



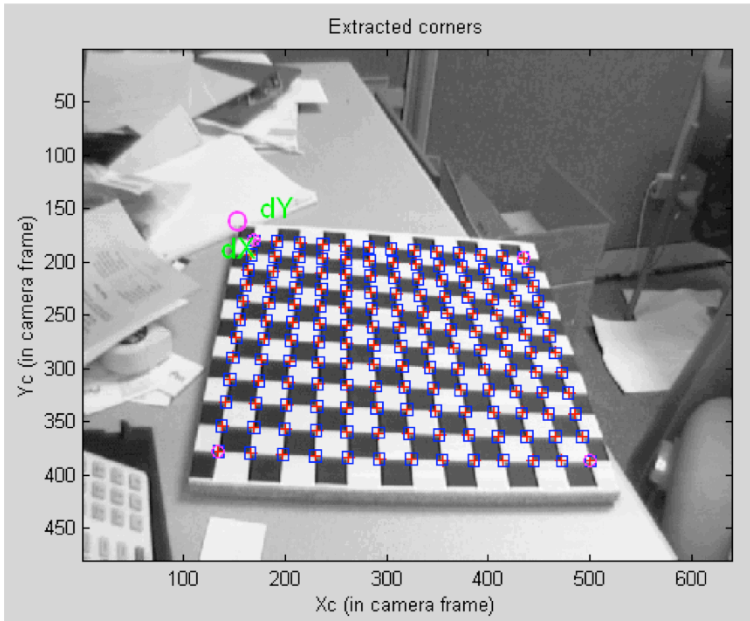
Calibration Procedure



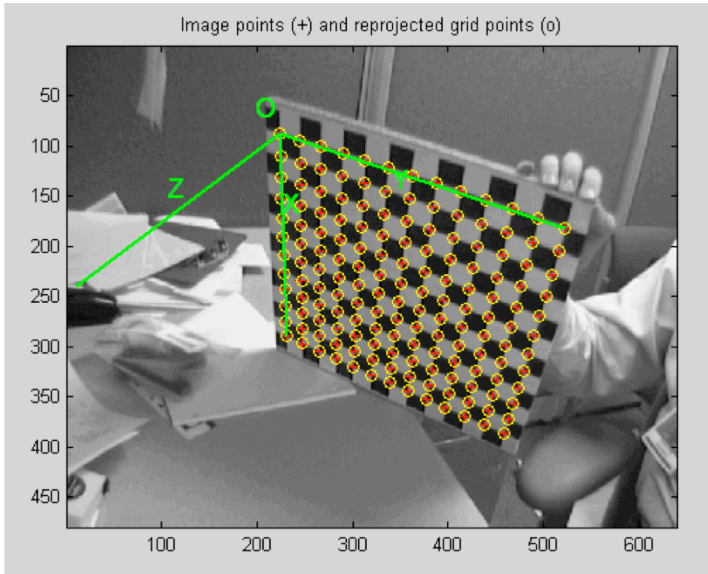
Calibration Procedure



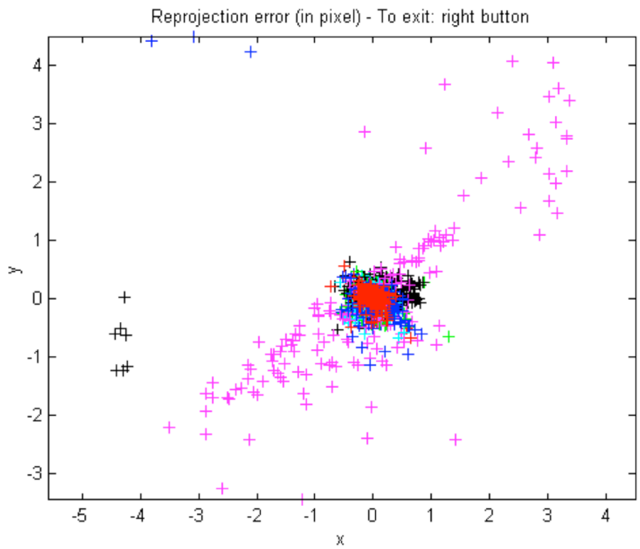
Calibration Procedure



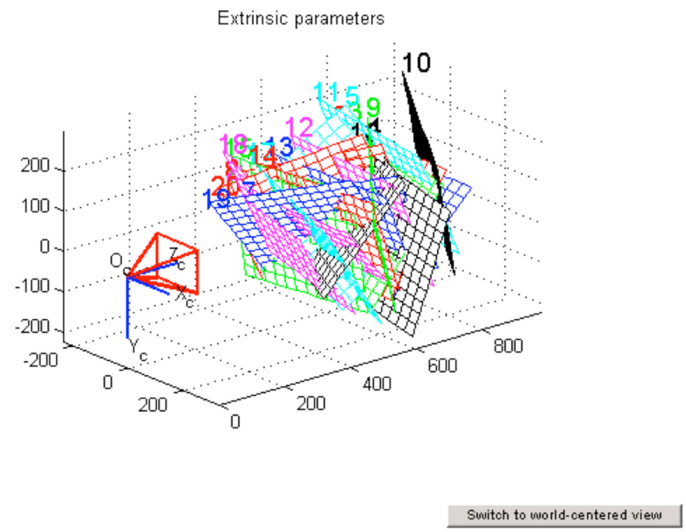
Calibration Procedure



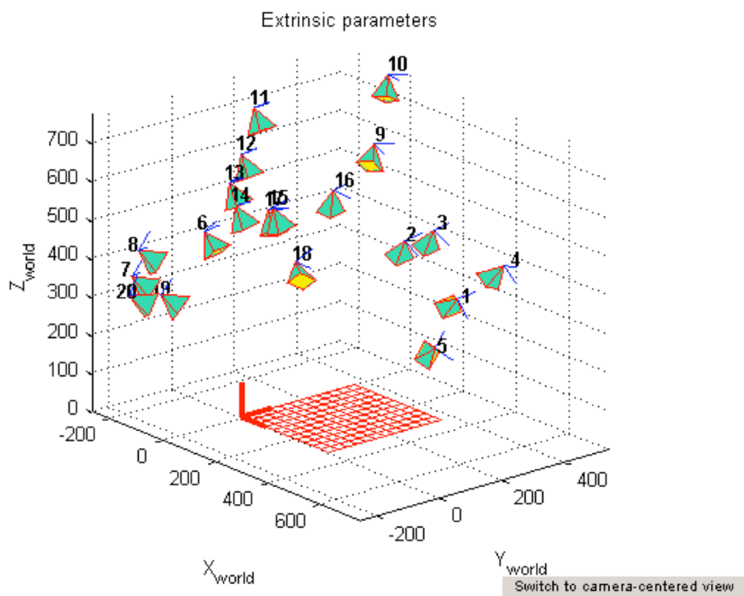
Calibration Procedure



Calibration Procedure



Calibration Procedure



Next lecture

- Single view reconstruction

Eigenvalues and Eigenvectors

Eigendecomposition

$$A = S\Lambda\, S^{-1} = S \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \cdot & \\ & & & \lambda_N \end{bmatrix} S^{-1}$$

Eigenvectors of A are columns of S

$$S = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_N \end{bmatrix}$$

Singular Value decomposition

$$A = U\,\Sigma\,V^{-1} \qquad \Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \cdot & \\ & & & \sigma_N \end{bmatrix}$$

U, V = orthogonal matrix

$$\sigma_i = \sqrt{\lambda_i}$$

σ = singular value
λ = eigenvalue of $A^t A$