Typical setup for calibration

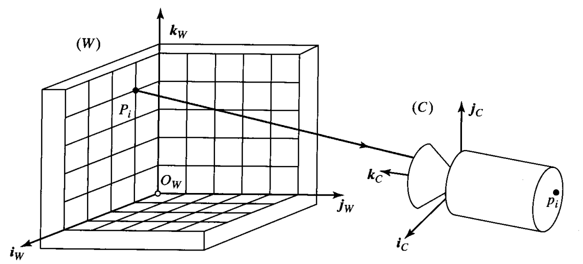

Figure 3.1 Camera calibration setup: In this example, the calibration rig is formed by three grids drawn in orthogonal planes. Other patterns could be used as well, and they may involve lines or other geometric figures.

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{intrinsic parameters} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{extrinsic parameters} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Camera matrix, M

Goal one: find M from image of calibration object

Goal two: given M, find the two matrices

Is goal one feasible?

More specifically, given (X,Y,Z) and corresponding u=(U/W) and v=(V/W), can we compute M?

First observation---if M is a solution then, because of homogeneity, k*M is an equivalent solution.

Thus, it only makes sense to recover M up to a scaling constant, and we can set the scale of M in advance for our convenience.

Is goal one feasible?

Since we are allowed to collect as much data as we need, goal one seems feasible.

(Details to follow soon).

Is goal two feasible?

Reason by counting parameters.

We have 11 numbers, as M is 3 by 4, and we can fix the scale.

The number of parameters (degrees of freedom) are the number of intrinsic parameters *plus* the number of extrinsic parameters.

Extrinsic parameters: ?

Intrinsic parameters: ?

The number of parameters are the number of intrinsic parameters *plus* the number of extrinsic parameters.

Extrinsic parameters:

location	(3)
orientation	(3)

Intrinsic parameters:

focal length	(1)	Or α and β.
pixel aspect ratio	(1)	
principal point	(2)	
skew	(1)	
	11	

Often assume skew is zero

Is goal two feasible?

Yes (provided the points are not "degenerate").

The 11 numbers (knowns) from M match the unknowns (camera parameters).

If some camera parameters are known, then a more robust computation is possible.

Finding M (goal one) (§3)

Find M from an image of calibration object. The equation relating world coordinates to image coordinates is:

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = M \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = MP$$

If we identify enough non-degenerate points whose *world coordinates are known* then we can estimate M from their *location in the image*.

Specifically we have points in space, P, and corresponding observed image coordinates, u=U/W and v=V/W

(§2.2.2, §3.2.1)

We have $\begin{pmatrix} U \\ V \\ W \end{pmatrix} = M\mathbf{P}$

Write $M = \begin{vmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{vmatrix}$ Where $\mathbf{m}_i$ are row vectors

Thus
$$U = \mathbf{m}_1 \cdot \mathbf{P}$$
$$V = \mathbf{m}_2 \cdot \mathbf{P}$$
$$W = \mathbf{m}_3 \cdot \mathbf{P}$$

(§2.2.2, §3.2.1)

From the previous slide

$$U = \mathbf{m}_1 \cdot \mathbf{P}$$
$$V = \mathbf{m}_2 \cdot \mathbf{P}$$
$$W = \mathbf{m}_3 \cdot \mathbf{P}$$

So **each** point, **i**, gives two equations (§2.2.2, §3.2.1)

$$u_i = \frac{\mathbf{m}_1 \cdot \mathbf{P}_i}{\mathbf{m}_3 \cdot \mathbf{P}_i} \qquad v_i = \frac{\mathbf{m}_2 \cdot \mathbf{P}_i}{\mathbf{m}_3 \cdot \mathbf{P}_i}$$

Which become

$$(\mathbf{m}_1 - u_i \mathbf{m}_3) \cdot \mathbf{P}_i = 0$$
$$(\mathbf{m}_2 - v_i \mathbf{m}_3) \cdot \mathbf{P}_i = 0$$

(§2.2.2, §3.2.1)

We have linear equations for the components of M

The components of the matrix M are the *variables* in linear equations

Represent M by a vector $\mathbf{m} = \begin{pmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{pmatrix}$

Note that our camera matrix, M, is the unknown so we want to make it a vector in some matrix equation (where something else is going to be the matrix)---standard thing to do.

(§2.2.2, §3.2.1)

We are representing the matrix M by a vector $\mathbf{m} = \begin{pmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{pmatrix}$

Now rewrite $(\mathbf{m}_1 - u_i \mathbf{m}_3) \cdot \mathbf{P}_i = 0$ as $\begin{pmatrix} \mathbf{P}_i^T & 0 & -u_i \mathbf{P}_i^T \end{pmatrix} \mathbf{m} = 0$

$(\mathbf{m}_2 - v_i \mathbf{m}_3) \cdot \mathbf{P}_i = 0$ as $\begin{pmatrix} 0 & \mathbf{P}_i^T & -v_i \mathbf{P}_i^T \end{pmatrix} \mathbf{m} = 0$

Thus every point leads to two rows of a matrix P.

(§2.2.2, §3.2.1)

From previous slide, each point gives two rows of a matrix P

$$\begin{pmatrix} \mathbf{P}_i^T & 0 & -u_i\mathbf{P}_i^T \end{pmatrix} \mathbf{m} = 0$$
$$\begin{pmatrix} 0 & \mathbf{P}_i^T & -v_i\mathbf{P}_i^T \end{pmatrix} \mathbf{m} = 0$$

So, in general, the 2n by 12 matrix P is:

We call the matrix "P" to be consistent with the book.

$$\begin{pmatrix} \mathbf{P}_1^T & & -u_1\mathbf{P}_1^T \\ & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \\ \cdots & \cdots & \cdots \\ \mathbf{P}_i^T & & -u_i\mathbf{P}_i^T \\ & \mathbf{P}_i^T & -v_i\mathbf{P}_i^T \\ \cdots & \cdots & \cdots \\ \mathbf{P}_n^T & & -u_n\mathbf{P}_n^T \\ & \mathbf{P}_n^T & -v_n\mathbf{P}_n^T \end{pmatrix}$$

(§2.2.2, §3.2.1)

So, we want to solve P**m**=**0** for **m**, where P is 2n by 12

This problem is a bit tricky

Clearly **m**=**0** is a solution (degenerate solution)

There must be another solution (if we believe our imaging model)

If **m** is a solution, then a scalar multiple of **m** is also (homogeneity)

So, we solve P**m**=**0** under the constraint that |**m**|=1

If n>6, then this typically will not have a solution due to error (over-constrained)

To simultaneously deal with this problem, AND to use the information from multiple points, we find a "best" solution, using more than 6 points.

Math aside, #4

Homogenous linear least squares

Thus the problem inspired by our camera calibration problem is

Solve $U\mathbf{x} = \mathbf{0}$ subject to $|\mathbf{x}| = 1$

Again, **there is no exact solution**.

Least squares solution is the value of **x** so that the magnitude of $U\mathbf{x}$ is as close zero as possible.

(still §3.1.1)

Math aside, #4

Homogenous linear least squares

The least squares problem is thus

Minimize $\|U\mathbf{x}\|$ subject to $|\mathbf{x}| = 1$

This is solved by **magic** (see supplementary slides)

Important

Specifically, the minimum is reached when **x** is set to the eigenvector corresponding to the minimum eigenvalue of $U^T U$.

Math aside, #4

Homogenous linear least squares

Pragmatic solution of

$$\text{Solve } U\mathbf{x} = \mathbf{0} \text{ subject to } |\mathbf{x}| = 1$$

In Matlab, form $Y = U^{T} U$

Then use eig() to get the eigenvalues and eigenvectors of Y.

These are provided in order.

Extract the eigenvector that you need.

("proof" --- try 1,000,000 random solutions to see if you can do better)