## Camera matrix, $M$

Actual pixel coords are
(u,v) = (U/W, V/W)

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

**Projection**. We use **f=1** and let the intrinsic parameters absorb the focal length.

First part makes it so that we are in standard camera coords where we know how to project.

## Camera parameters (§2.2)

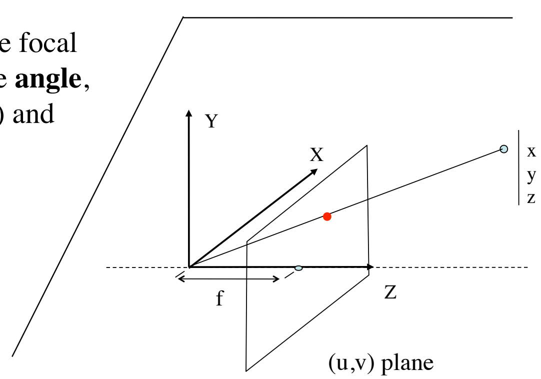
- Extrinsic parameters
  - position of the camera (3)
  - orientation of the camera (3)
- Intrinsic parameters (more natural)
  - focal length (1)
  - aspect ratio (ratio of pixel horizontal size to vertical size) (1)
  - principal point (intersection of viewing direction with camera plane) (2)
  - angle between axes of image plane---usually very close to 90 degrees (1)
- Alternative intrinsic parameters (more convenient)
  - focal length in units of horizontal pixels (1)
  - focal length in units of vertical pixels (1)
  - principal point (intersection of viewing direction with camera plane) (2)

## Intrinsic parameters (focal length)

Recall that u = f * (x/z) and v = f * (y/z)

The natural, easy to measure, units for (u,v) are pixels.

This means that the focal length transfers the **angle**, as encoded in (x/z) and (y/z) into **pixels**.




## Intrinsic parameters (focal length)

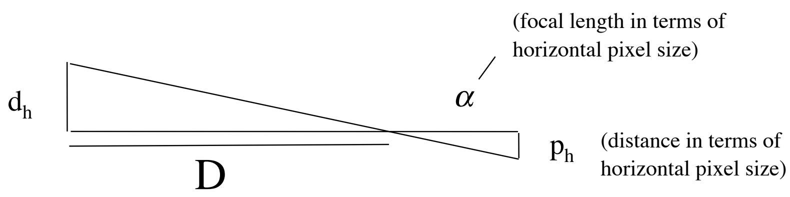
To transform a focal length in meters to pixels you would need to know the size of a pixel in meters. But you can easily measure the focal length in pixels (which is usually what you want)*.

However pixels are not always square. The ratio of width to height is called the aspect ratio.

Hence it is common to instead use two other parameters, $\alpha$ and $\beta$, which are the focal length in term of horizontal and vertical pixel units.

* If you have the focal length both in pixels and in meters, then you can compute the size of a pixel (if you wanted it for some reason)

# Measuring focal lengths




$$\alpha = D\frac{p_h}{d_h}$$

# Intrinsic parameter matrix

Conversion of projected coords into pixel units is achieved by simple **scalings** based on α and β.

**Translate** coords to so that line through pinhole (or center of lens) perpendicular to projection plane is at origin.

Compensate for non-perpendicular pixel axis (if needed, usually this is OK) using a **shear** transformation.

Since these operations are all achievable with matrices, we see that a camera can be modeled, as claimed, with M composed of the three parts (intrinsic, projection, extrinsic).

# Camera calibration (§3)

- Want to find out:
  - what is the camera matrix? (intrinsic+extrinsic)
  - what are intrinsic parameters of the camera?
- General strategy:
  - view calibration object
  - identify image points
  - obtain camera matrix by minimizing error
  - obtain intrinsic parameters from camera matrix

Typical setup for calibration

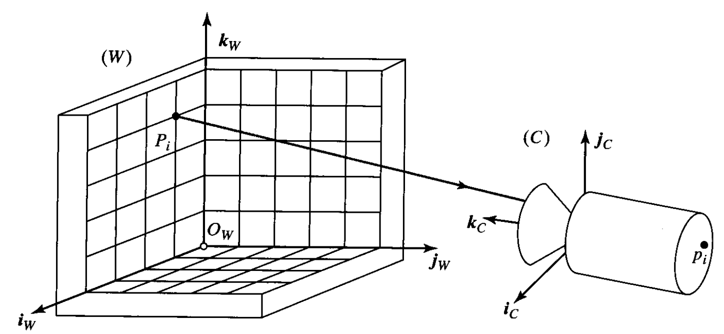



**Figure 3.1** Camera calibration setup: In this example, the calibration rig is formed by three grids drawn in orthogonal planes. Other patterns could be used as well, and they may involve lines or other geometric figures.

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \underbrace{\begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{intrinsic parameters} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{extrinsic parameters} \end{pmatrix}}_{\text{Camera matrix, } M} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

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 $\alpha$ and $\beta$. |
| 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

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

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

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



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

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.



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

Now rewrite

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

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

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