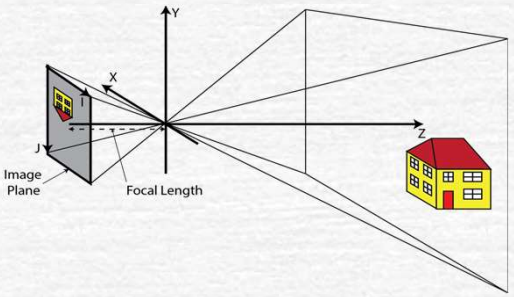
Camera models – Simple Pinhole Model

$$\begin{bmatrix} i.w \\ j.w \\ w \end{bmatrix} = \begin{bmatrix} f_i & 0 & c_i \\ 0 & f_j & c_j \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

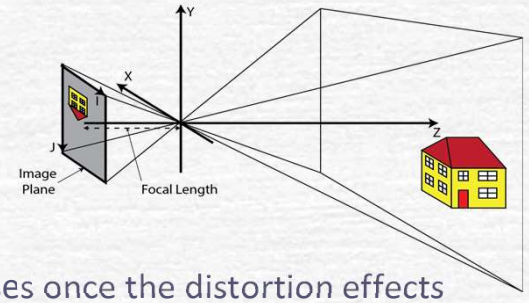


- 3-D point (*x, y, z*)
- 2-D image point *(i, j)*
- Scaling factor *w*
- Combination of focal length and image coordinate system (f_i & f_j)
- Location of the optical centre (c_i & c_j)

3

Camera models – Simple Pinhole Model

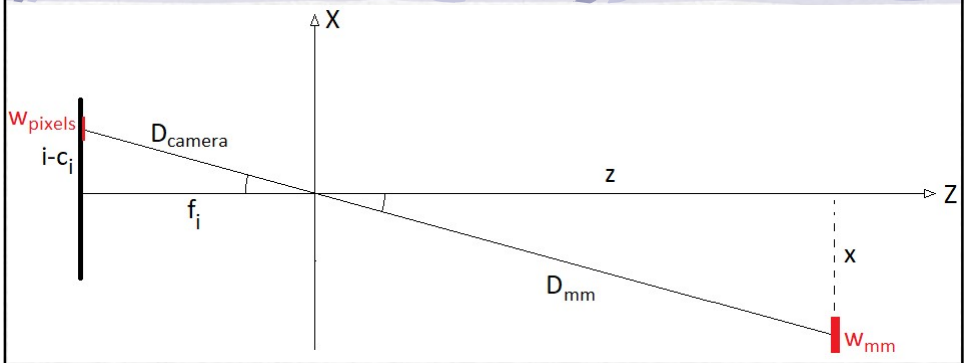
$$\begin{bmatrix} i.w \\ j.w \\ w \end{bmatrix} = \begin{bmatrix} f_i & 0 & c_i \\ 0 & f_j & c_j \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$



- This is what OpenCV uses once the distortion effects are removed.
- $i.w = f_i.x + c_i.z$ $\quad w = z$
- $i = f_i.(x/z) + c_i$
- $j = f_j.(y/z) + c_j$

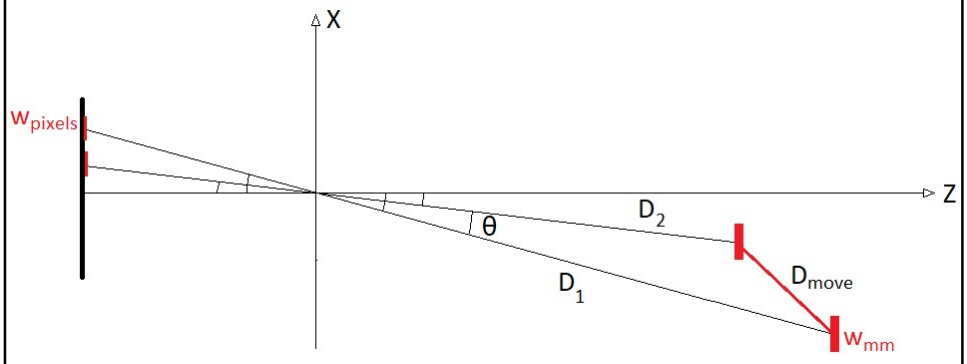
4

Camera models – Distance from the camera



- Just considering the i and X plane, if we have an object of a known width (note it is a little more complex when j and Y are introduced)
- D_{camera} can be computed using Pythagorus for right angled triangles
- $D_{mm} / W_{mm} = D_{camera} / W_{pixels}$
- So $D_{mm} = D_{camera} * W_{mm} / W_{pixels}$

Camera models – Distance travelled



- We can compute the two distances
- We can compute theta (the difference of the two angles)
- We can compute the distance moved using the **law of cosines**