Camera Projection

Background Reading:
Trucco&Verri, Section 2.4
Forsyth&Ponce, Section 2.2

New point of view

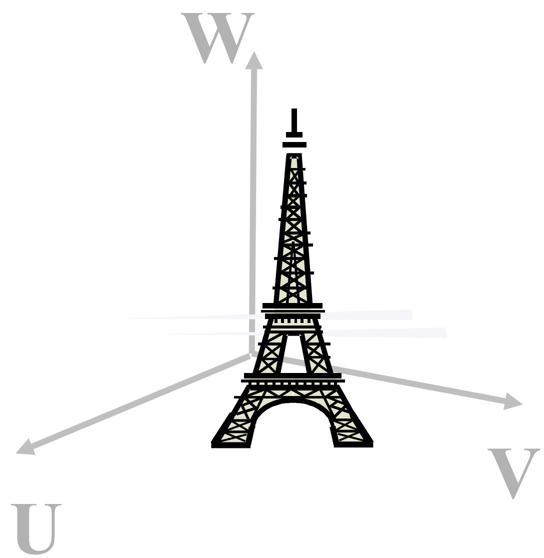
No pun intended!

In addition to being a device for measuring intensity (luminance) values...

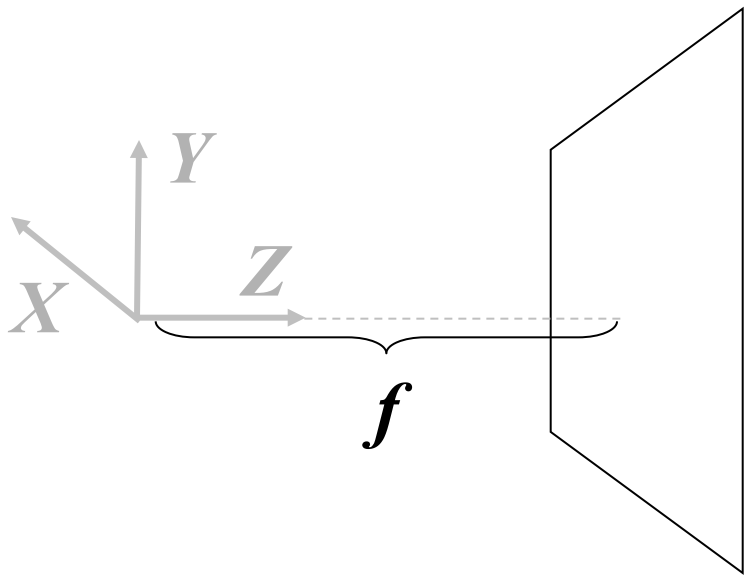
... your camera is also a device for measuring viewing directions / pointing angles.

Imaging Geometry

Object of Interest in World Coordinate System (U,V,W)



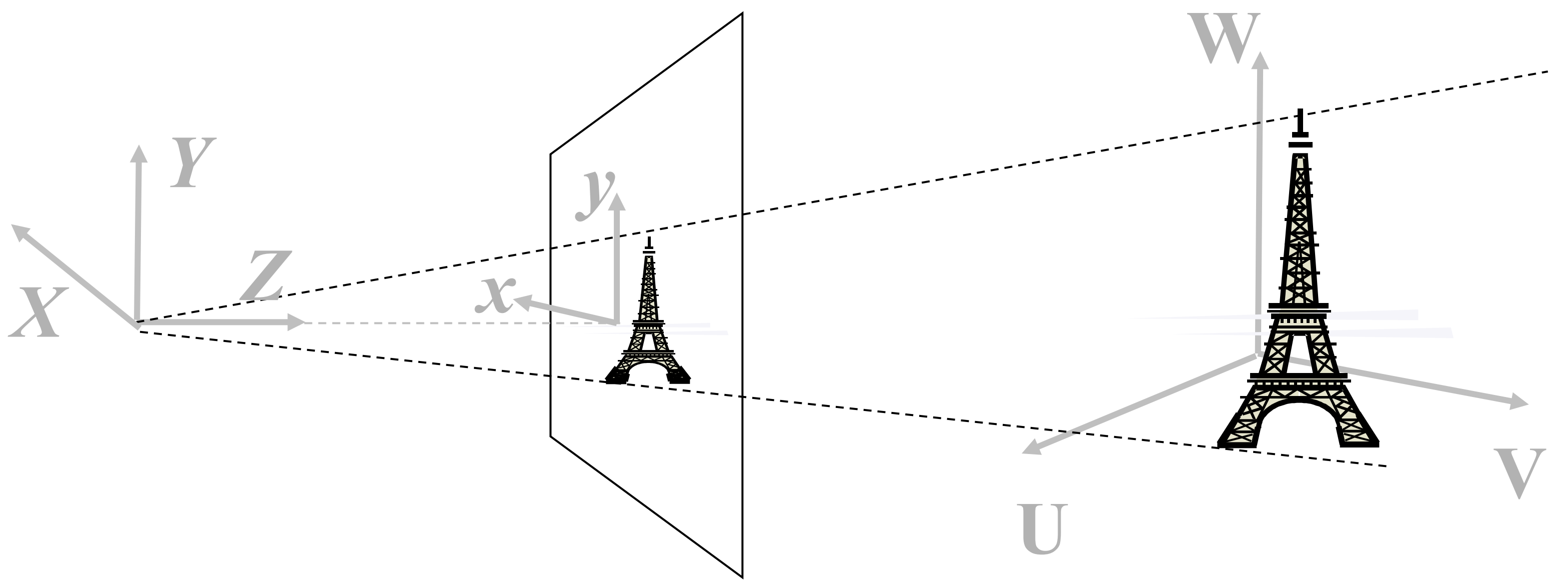
Imaging Geometry



Camera Coordinate System (X,Y,Z).

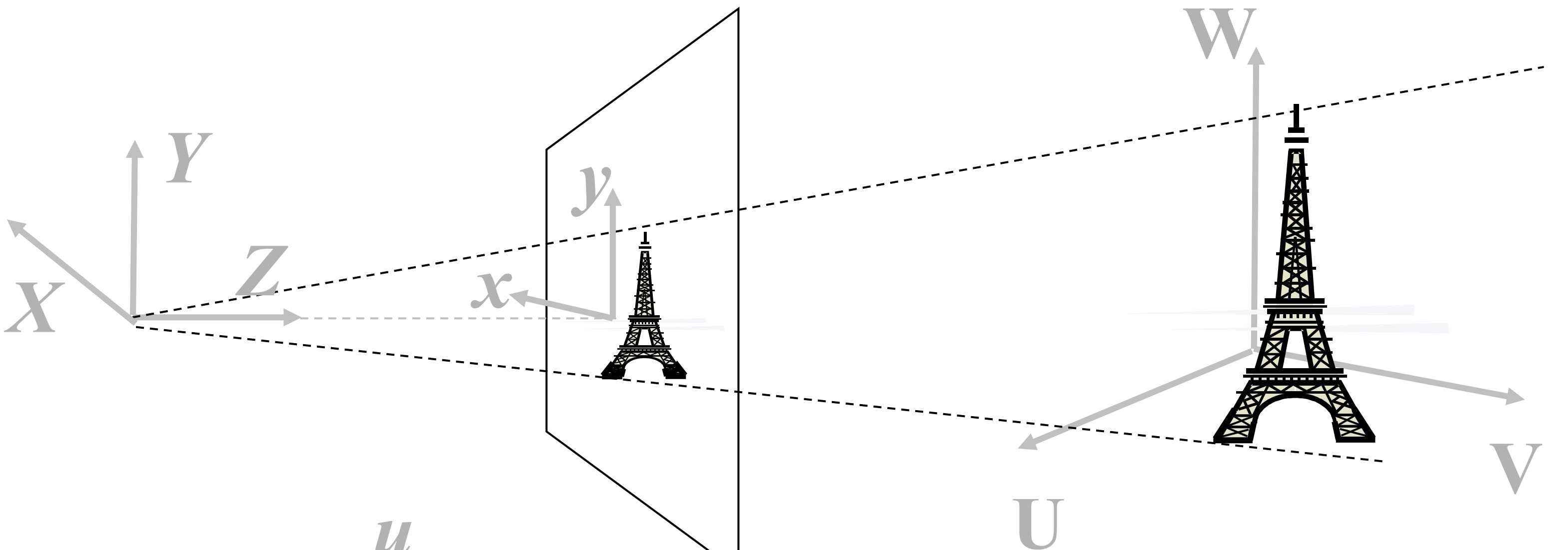
- Z is optic axis
- Image plane located f units out along optic axis
- f is called focal length

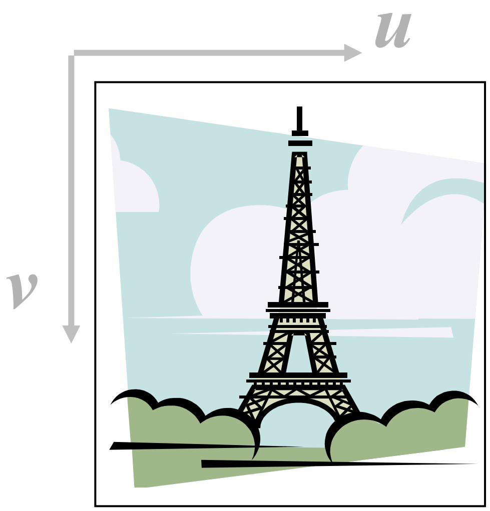
Imaging Geometry



Forward Projection onto image plane.
3D (X,Y,Z) projected to 2D *(x,y)*

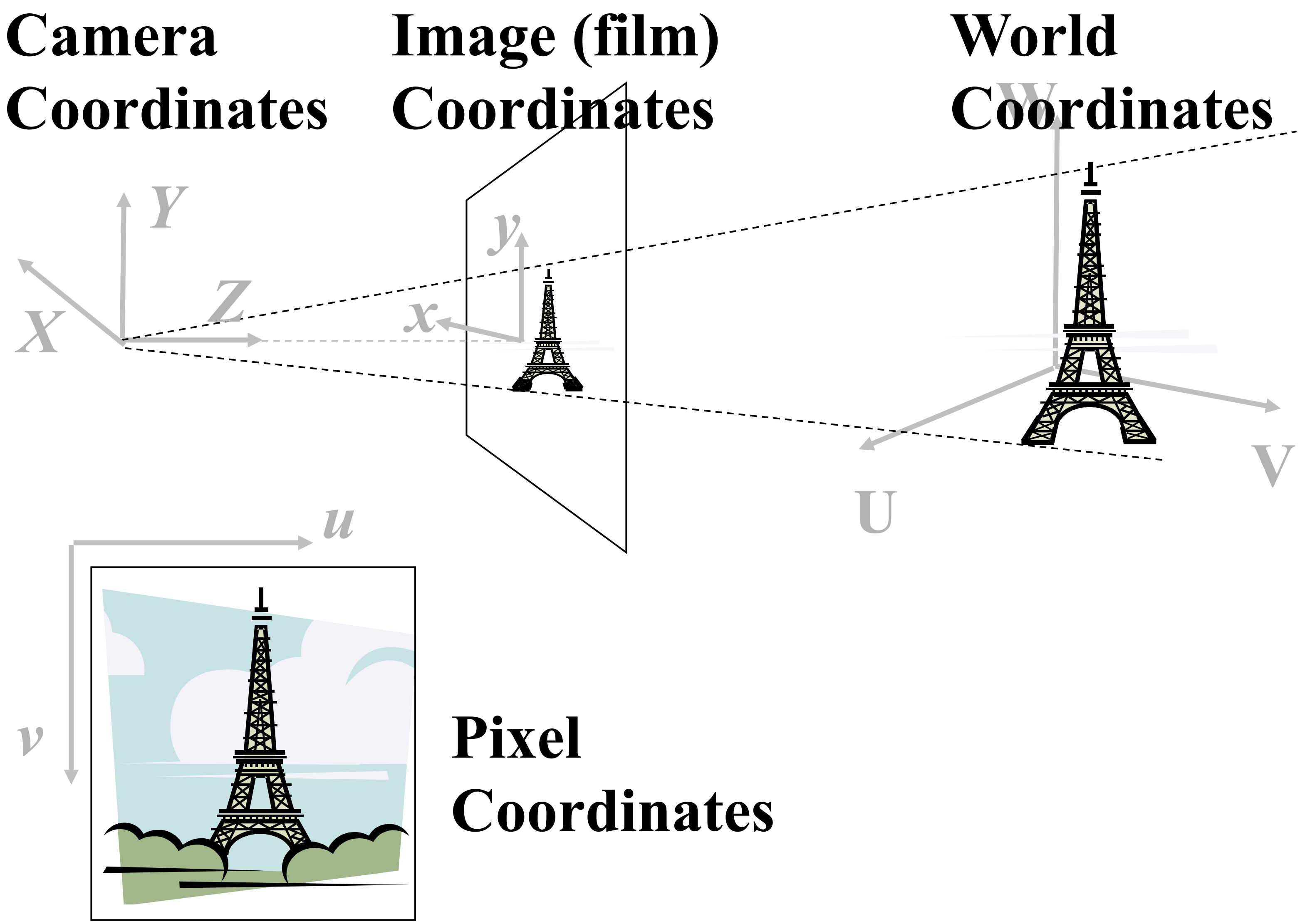
Imaging Geometry





Our image gets digitized into pixel coordinates *(u,v)*

Imaging Geometry



Forward Projection

World Coords		Camera Coords		Film Coords		Pixel Coords
U V W	→	X Y Z	→	x y	→	u v

We want a mathematical model to describe how 3D World points get projected into 2D Pixel coordinates.

Our goal: describe this sequence of transformations by a big matrix equation!

Robert Collins
CMPEN454

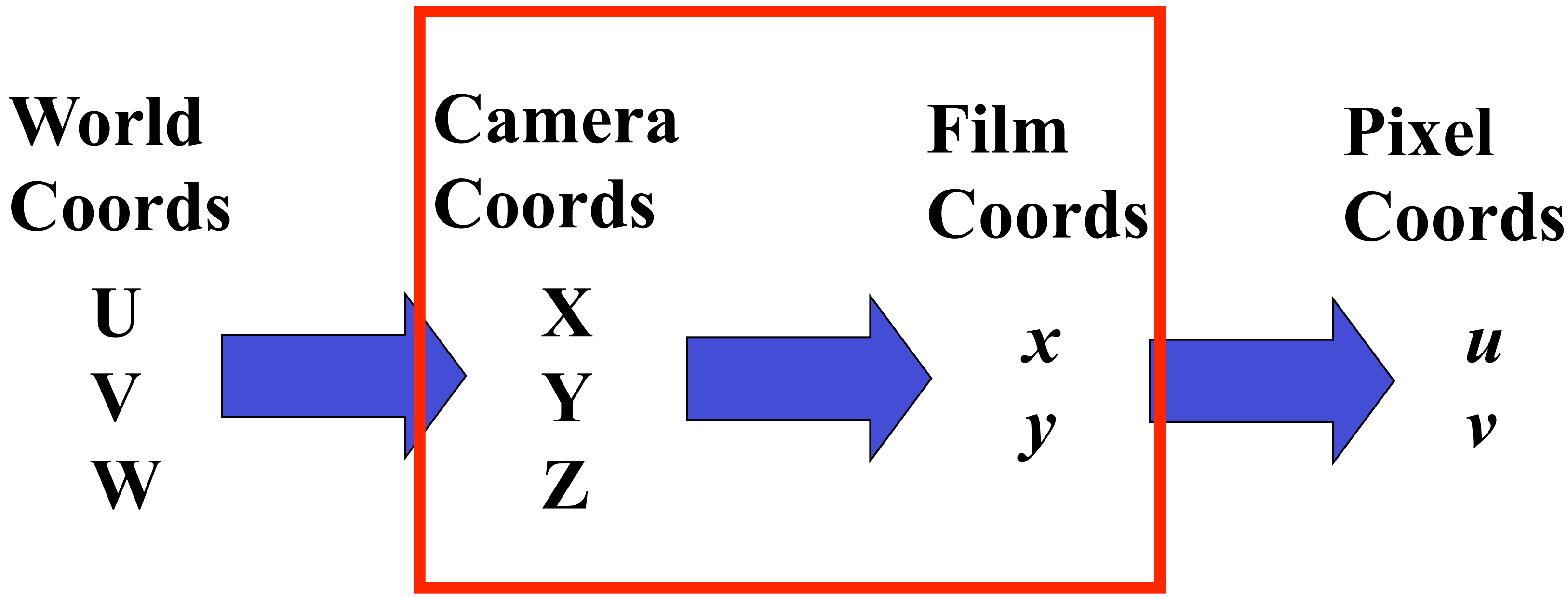
Backward Projection

World Coords		Camera Coords		Film Coords		Pixel Coords
U	←	X	←	x	←	u
V		Y		y		v
W		Z				

Note, much of vision concerns trying to derive backward projection equations to recover 3D scene structure from images (via stereo or motion)

But first, we have to understand forward projection…

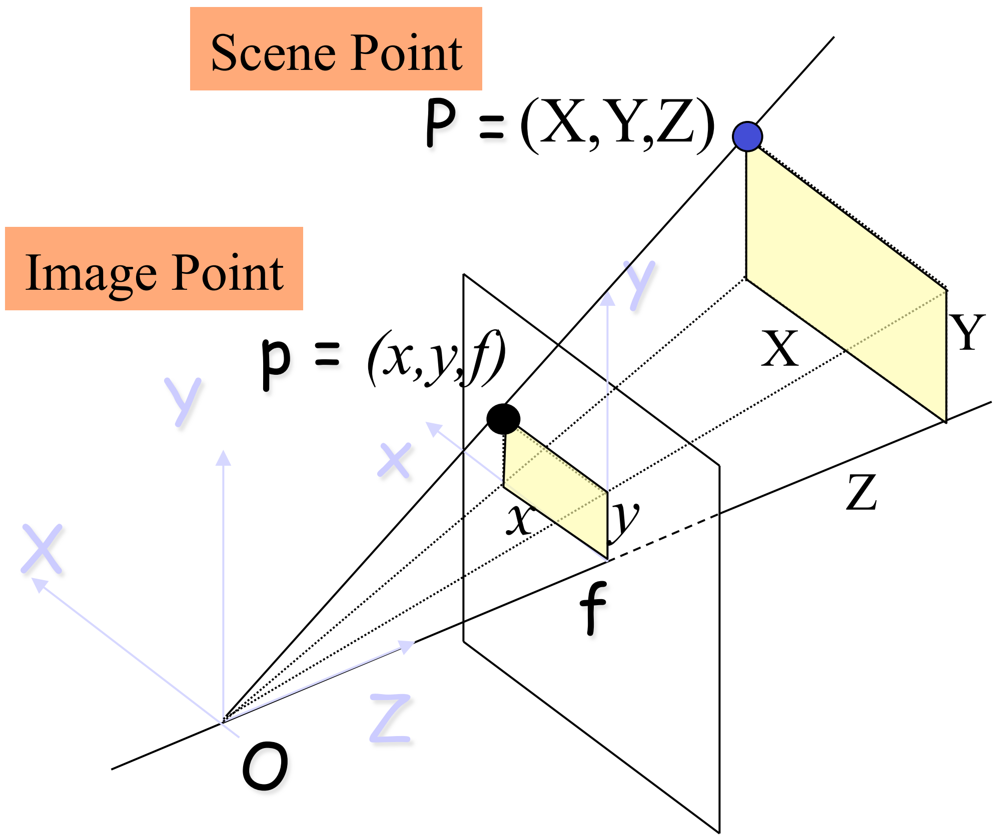
Forward Projection



3D-to-2D Projection
- **perspective projection**

We will start here in the middle, since we have already talked about this when discussing stereo.

Basic Perspective Projection



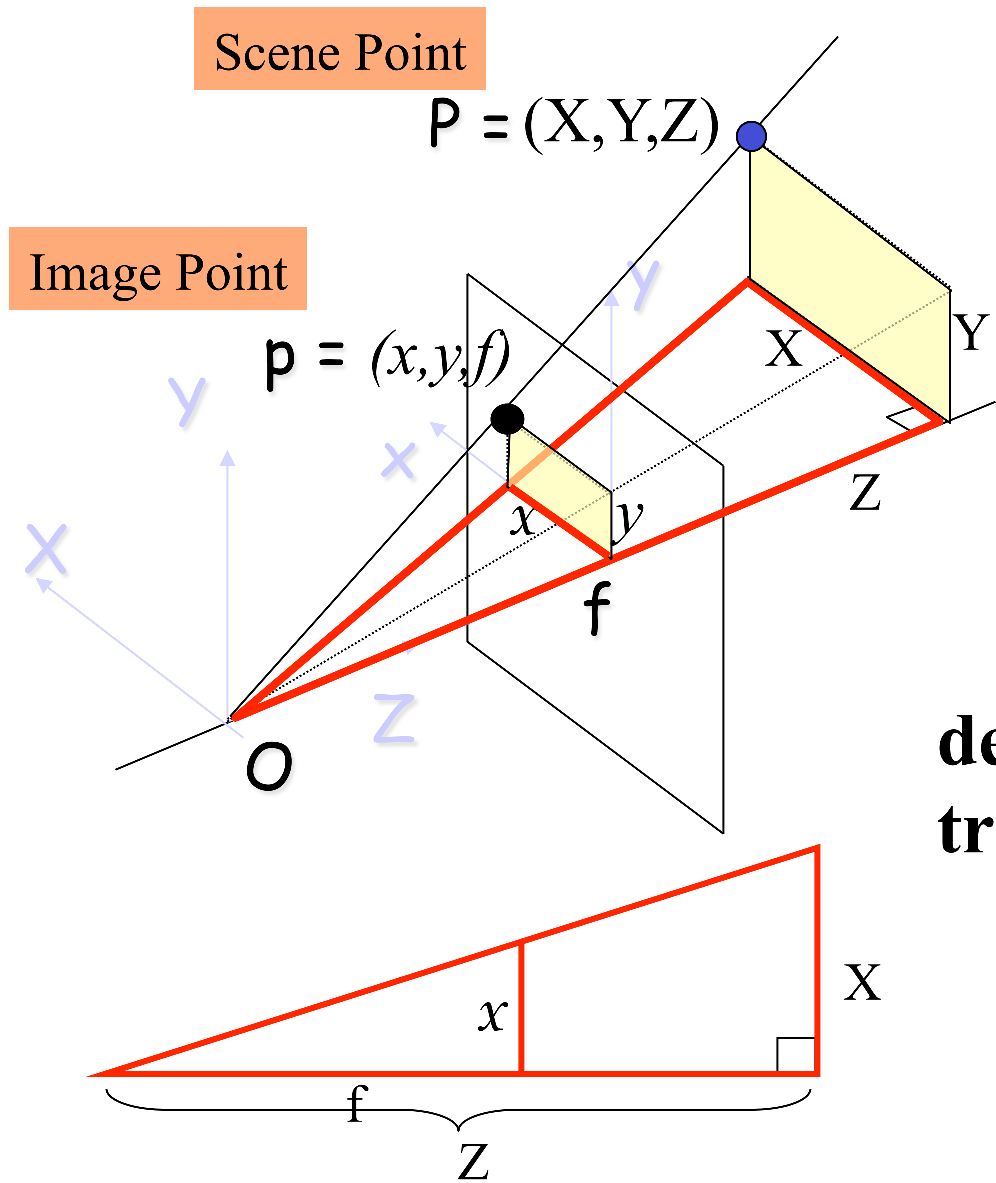
Perspective Projection Eqns

$$x = f\frac{X}{Z}$$

$$y = f\frac{Y}{Z}$$

Robert Collins
CMPEN454

Basic Perspective Projection



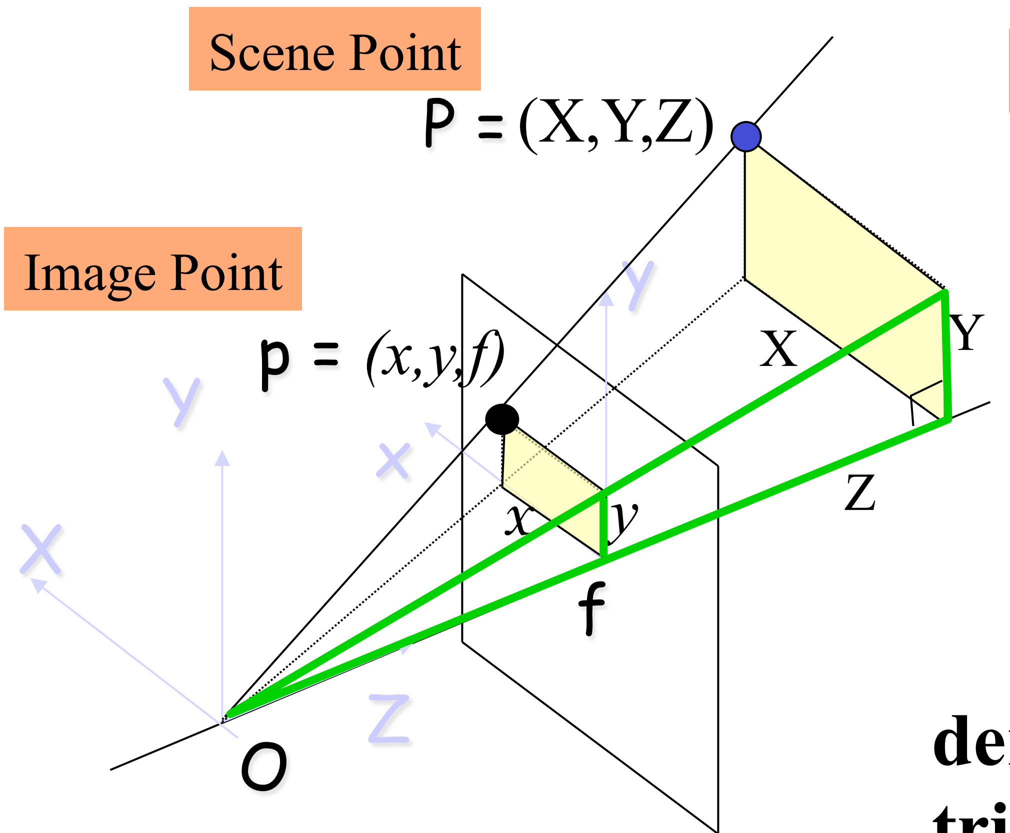
Perspective Projection Eqns

$$x = f\frac{X}{Z}$$

$$y = f\frac{Y}{Z}$$

derived via similar triangles rule

Basic Perspective Projection

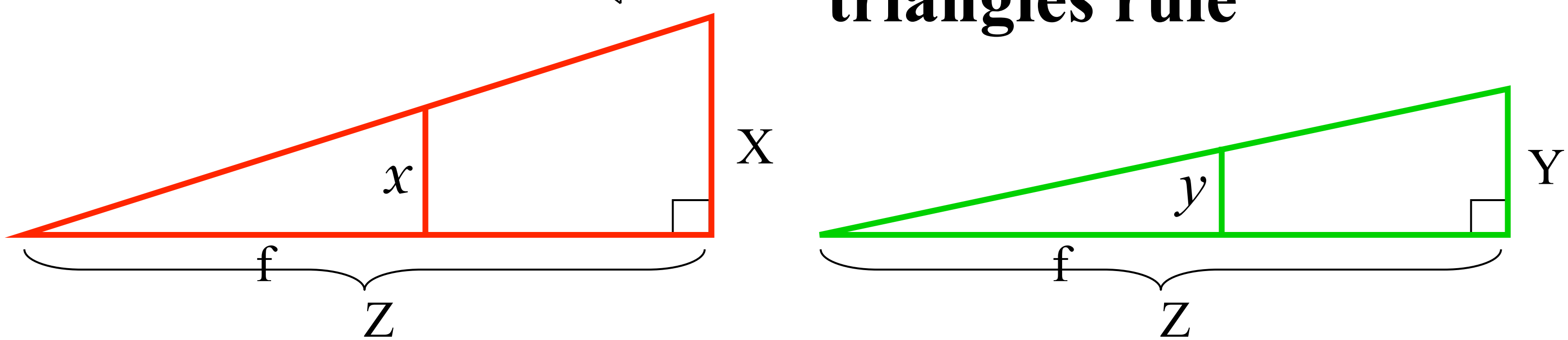


Perspective Projection Eqns

$$x = f\,\frac{X}{Z}$$

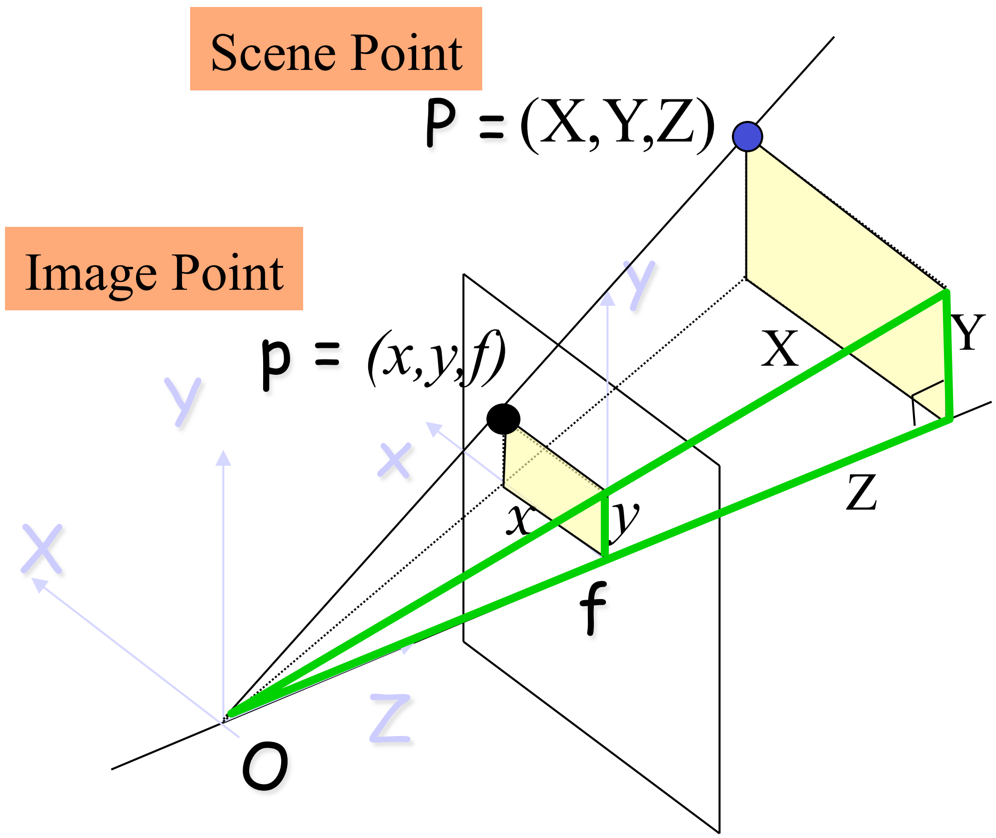
$$y = f\,\frac{Y}{Z}$$

derived via similar triangles rule



Robert Collins
CMPEN454

Basic Perspective Projection



Perspective Projection Eqns

$$x = f\,\frac{X}{Z}$$

$$y = f\,\frac{Y}{Z}$$

So how do we represent this as a matrix equation?
We need to introduce homogeneous coordinates.

Homogeneous Coordinates

Represent a 2D point (x,y) by a 3D point (x',y',z') by adding a "fictitious" third coordinate.

By convention, we specify that given (x',y',z') we can recover the 2D point (x,y) as

$$x = \frac{x'}{z'} \quad y = \frac{y'}{z'}$$

Note: (x,y) = (x,y,1) = (2x, 2y, 2) = (kx, ky, k)
for any nonzero k (can be negative as well as positive)

Perspective Matrix Equation

(in Camera Coordinates)

$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z}$$

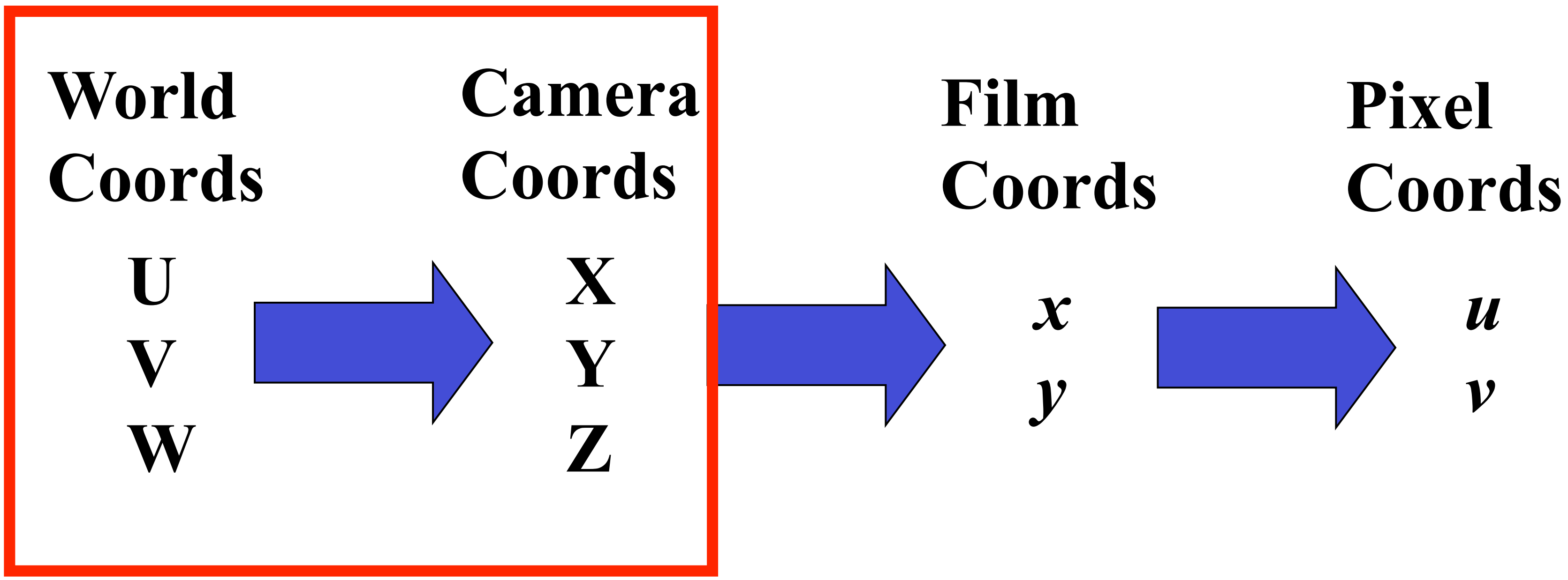
$$\Longleftrightarrow$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Inhomogeneous coordinates

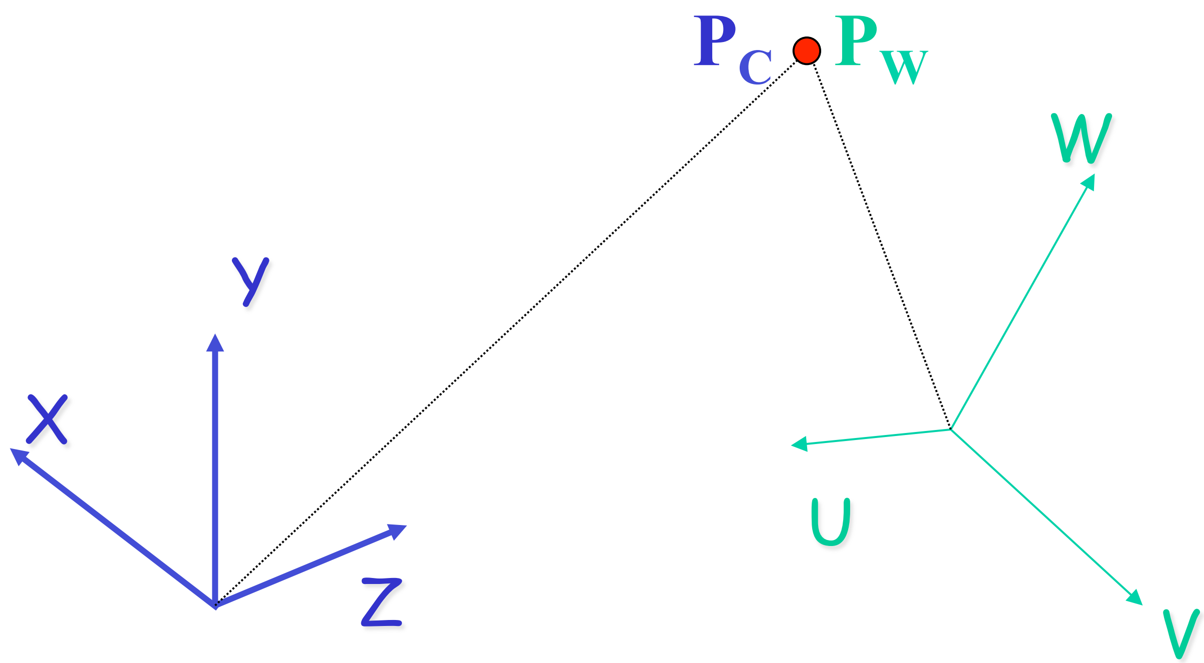
Homogeneous coordinates

Forward Projection



Rigid Transformation (rotation+translation) between world and camera coordinate systems

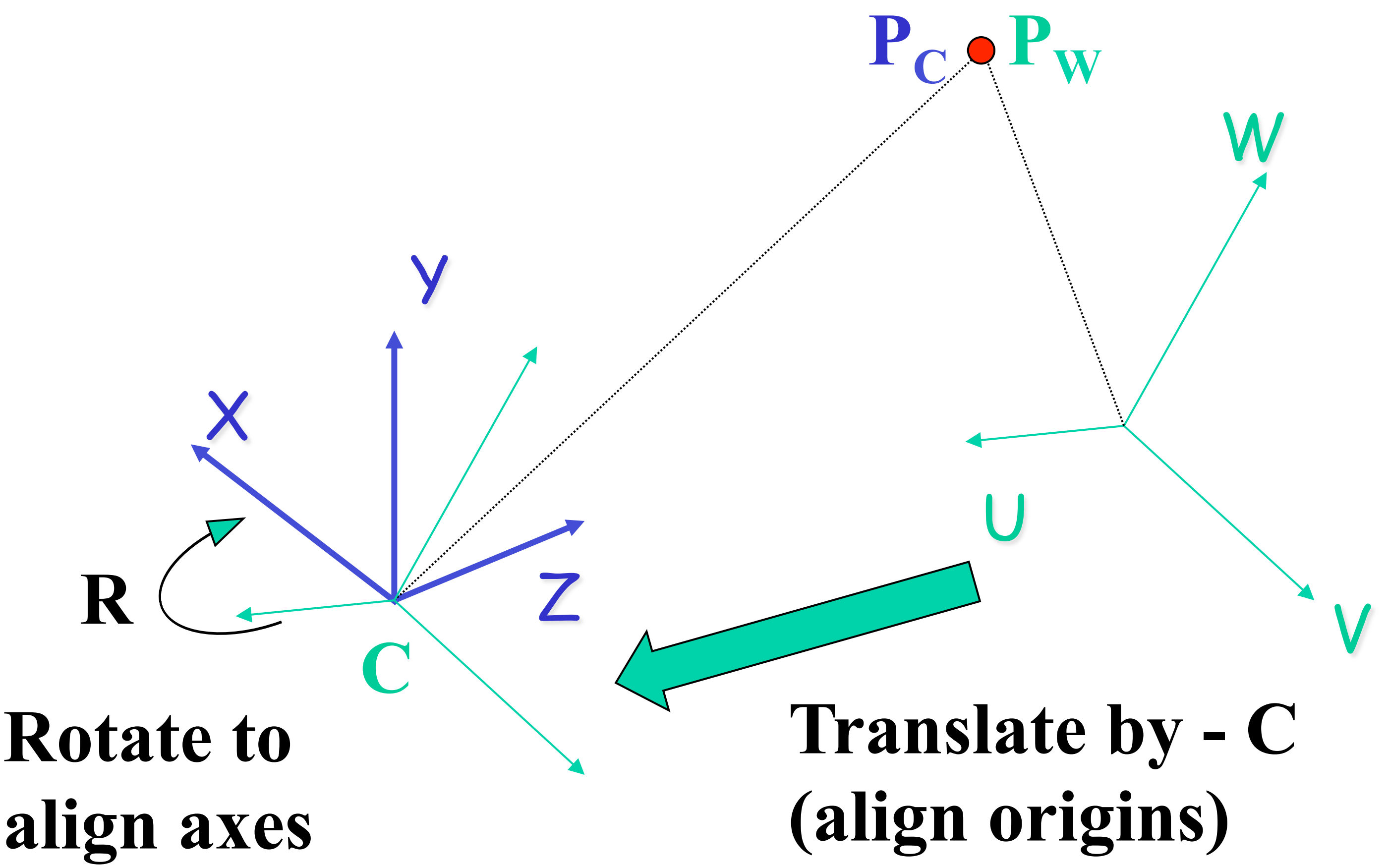
World to Camera Transformation



Avoid confusion: Pw and Pc are not two different points. They are the same physical point, described in two different coordinate systems.

Robert Collins
CMPEN454

World to Camera Transformation



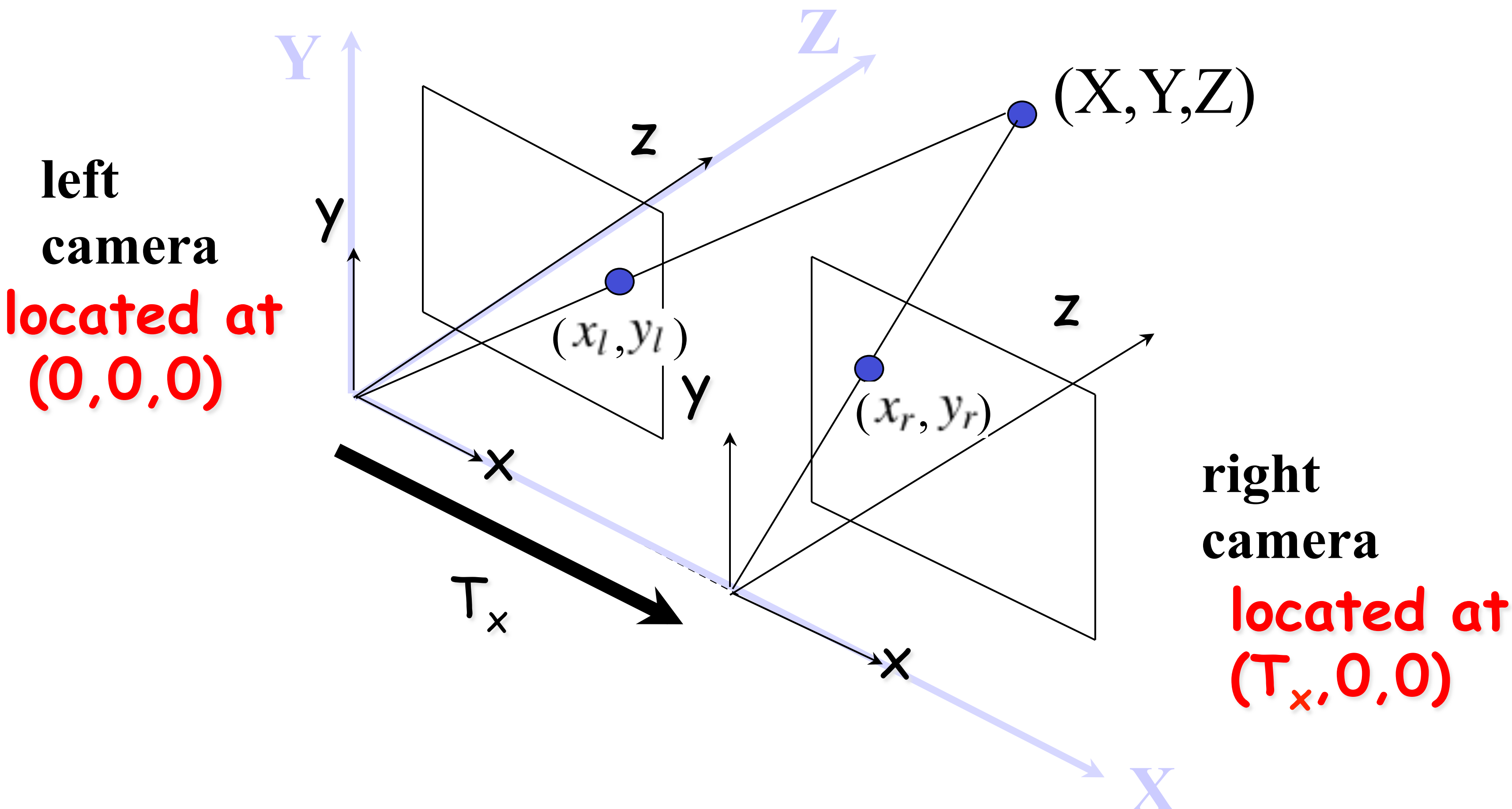
$$P_C = R\,(\,P_W - C\,)$$

Matrix Form, Homogeneous Coords

$$\mathbf{P_C} = \mathbf{R}\,(\,\mathbf{P_W} - \mathbf{C}\,)$$

$$\overset{\mathbf{P_C}}{\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \overset{\mathbf{P_W}}{\begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}}$$

Example: Simple Stereo System



Left camera located at world origin (0,0,0)
and camera axes aligned with world coord axes.

Simple Stereo, Left Camera

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}$$

camera axes aligned with world axes

located at world position (0,0,0)

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

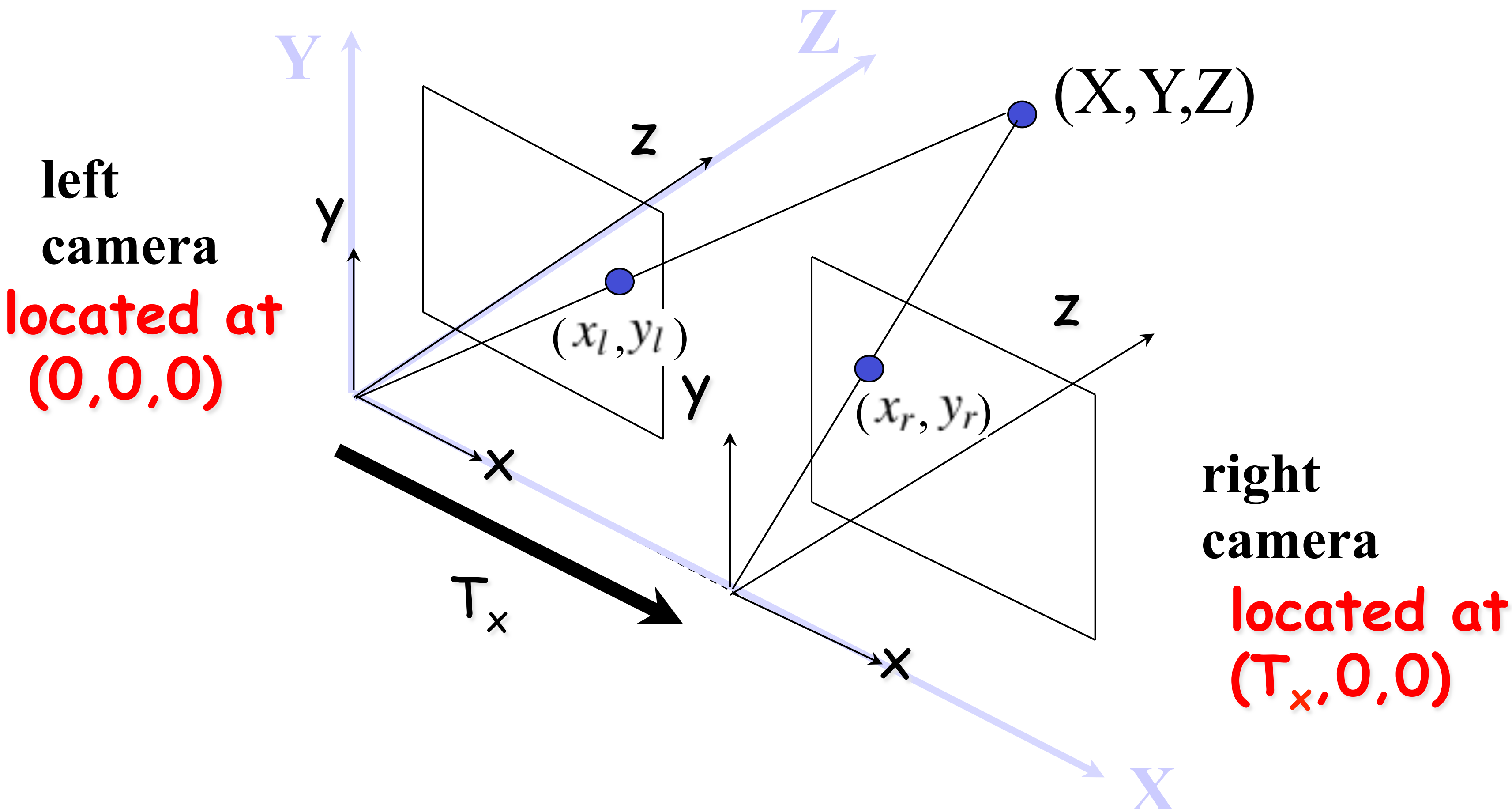
Simple Stereo Projection Equations

Left camera

$$\begin{bmatrix} x_l \\ y_l \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$x_l = f\frac{X}{Z} \qquad y_l = f\frac{Y}{Z}$$

Example: Simple Stereo System



Right camera located at world location (Tx,0,0)
and camera axes aligned with world coord axes.

Simple Stereo, Right Camera

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}$$

camera axes aligned with world axes

located at world position (T_x,0,0)

$$= \begin{bmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Simple Stereo Projection Equations

Left camera

$$\begin{bmatrix} x_l \\ y_l \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$x_l = f\,\frac{X}{Z} \qquad y_l = f\,\frac{Y}{Z}$$

Right camera

$$\begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$x_r = f\,\frac{X - T_x}{Z} \qquad y_r = f\,\frac{Y}{Z}$$

Bob's sure-fire way(s) to figure out the rotation

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cancel{\begin{pmatrix} 1 & 0 & 0 & -c_x \\ & & & \\ & & & \\ 0 & 0 & 0 & 1 \end{pmatrix}} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}$$

forget about this while thinking about rotations

$$\mathbf{P_C = R\ P_W}$$

This equation says how vectors in the world coordinate system (including the coordinate axes) get transformed into the camera coordinate system.

Figuring out Rotations

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}$$

$$\mathbf{P_C = R\ P_W}$$

what if world U axis (1,0,0) corresponds to camera axis (a,b,c)?

$$\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{a} & r_{12} & r_{13} & 0 \\ \mathbf{b} & r_{22} & r_{23} & 0 \\ \mathbf{c} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ 1 \end{pmatrix}$$

we can immediately write down the first column of R!

Figuring out Rotations

and likewise with world V axis and world W axis...

same axis in camera coords

axis in world coords

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}$$

world U axis (1,0,0) in camera coords

world V axis (0,1,0) in camera coords

world W axis (0,0,1) in camera coords

Figuring out Rotations

Alternative approach: sometimes it is easier to specify what camera X,Y,or Z axis is in world coordinates. Then rearrange the equation as follows.

$$\mathbf{P_C = R\,P_W} \Rightarrow \mathbf{R^{-1}P_C = P_W} \Rightarrow \mathbf{R^T P_C = P_W}$$

$$\begin{pmatrix} r_{11} & r_{21} & r_{31} & 0 \\ r_{12} & r_{22} & r_{32} & 0 \\ r_{13} & r_{23} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}$$

Figuring out Rotations

$$\begin{pmatrix} r_{11} & r_{21} & r_{31} & 0 \\ r_{12} & r_{22} & r_{32} & 0 \\ r_{13} & r_{23} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}$$

$$\mathbf{R^T P_C = P_W}$$

what if camera X axis (1,0,0) corresponds to world axis (a,b,c)?

$$\begin{pmatrix} r_{11} & r_{21} & r_{31} & 0 \\ r_{12} & r_{22} & r_{32} & 0 \\ r_{13} & r_{23} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} \mathbf{a} & r_{21} & r_{31} & 0 \\ \mathbf{b} & r_{22} & r_{32} & 0 \\ \mathbf{c} & r_{23} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ 1 \end{pmatrix}$$

we can immediately write down the first column of R^T, *(which is the first row of R).*

Figuring out Rotations

and likewise with camera Y axis and camera Z axis...

same axis in camera coords

axis in world coords

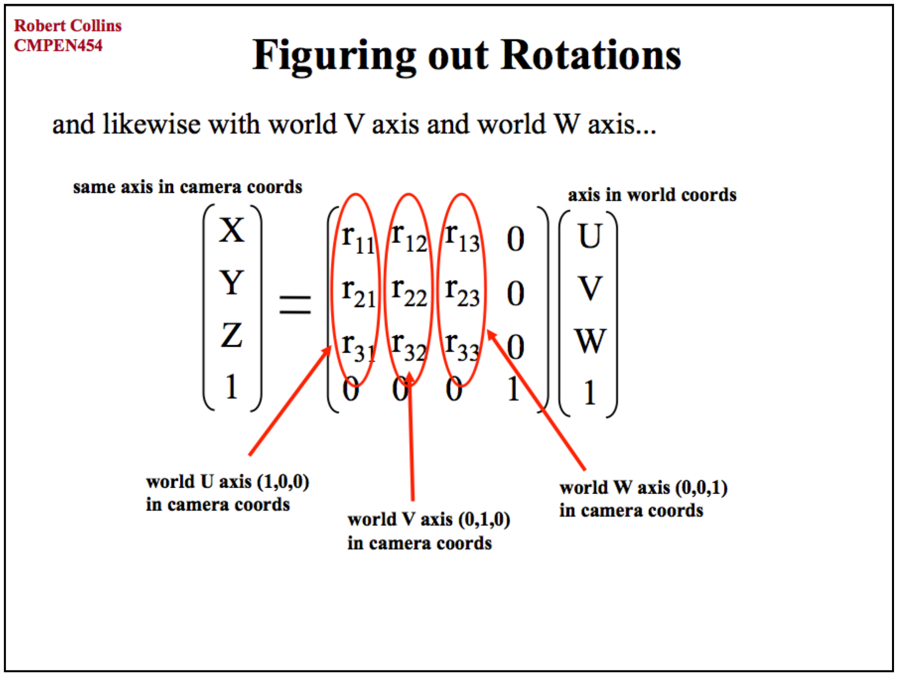
$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}$$

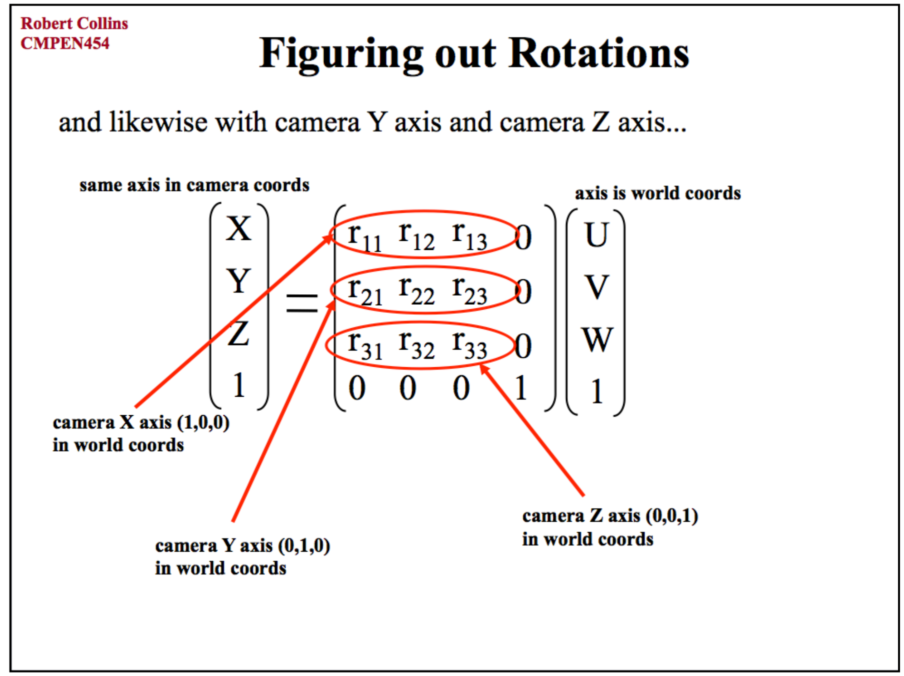
**camera X axis (1,0,0)
in world coords**

**camera Y axis (0,1,0)
in world coords**

**camera Z axis (0,0,1)
in world coords**

Mnemonic

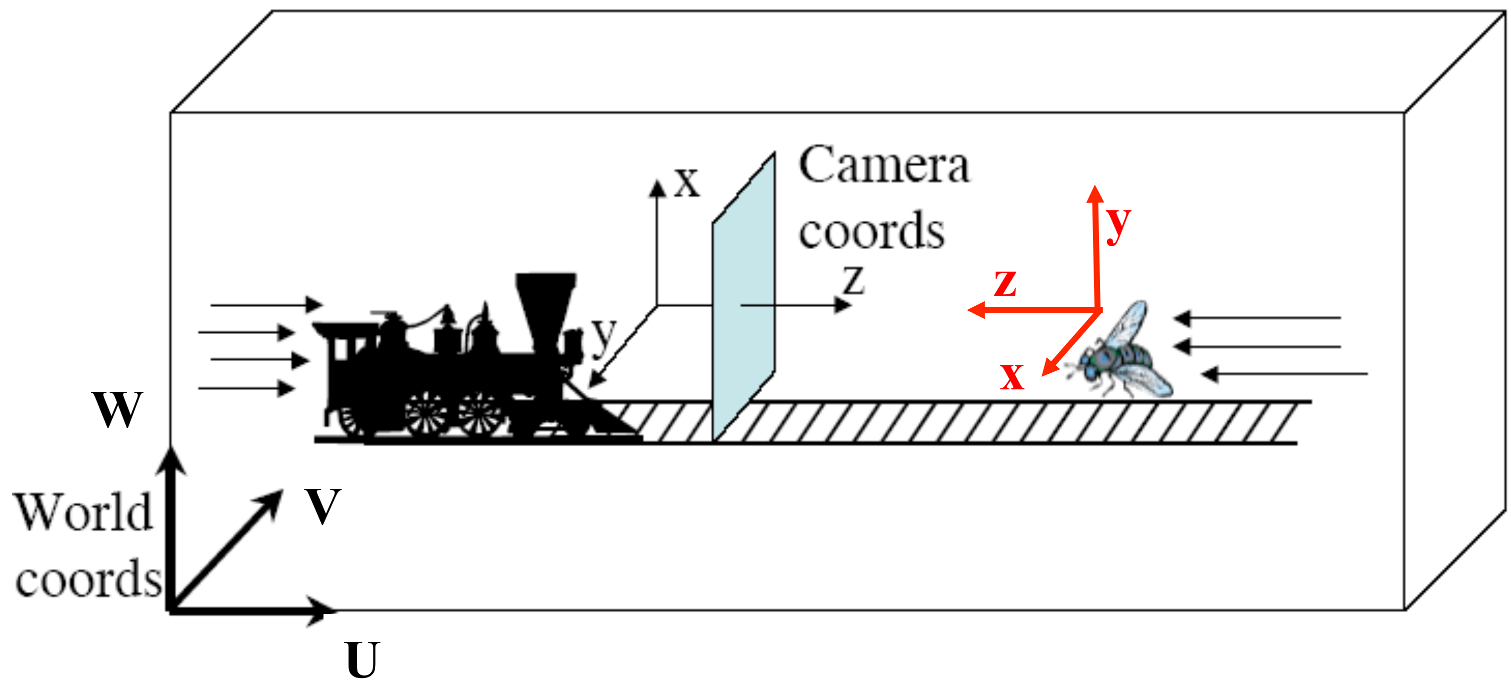




World axes are written down the columns

Camera axes are written across the rows

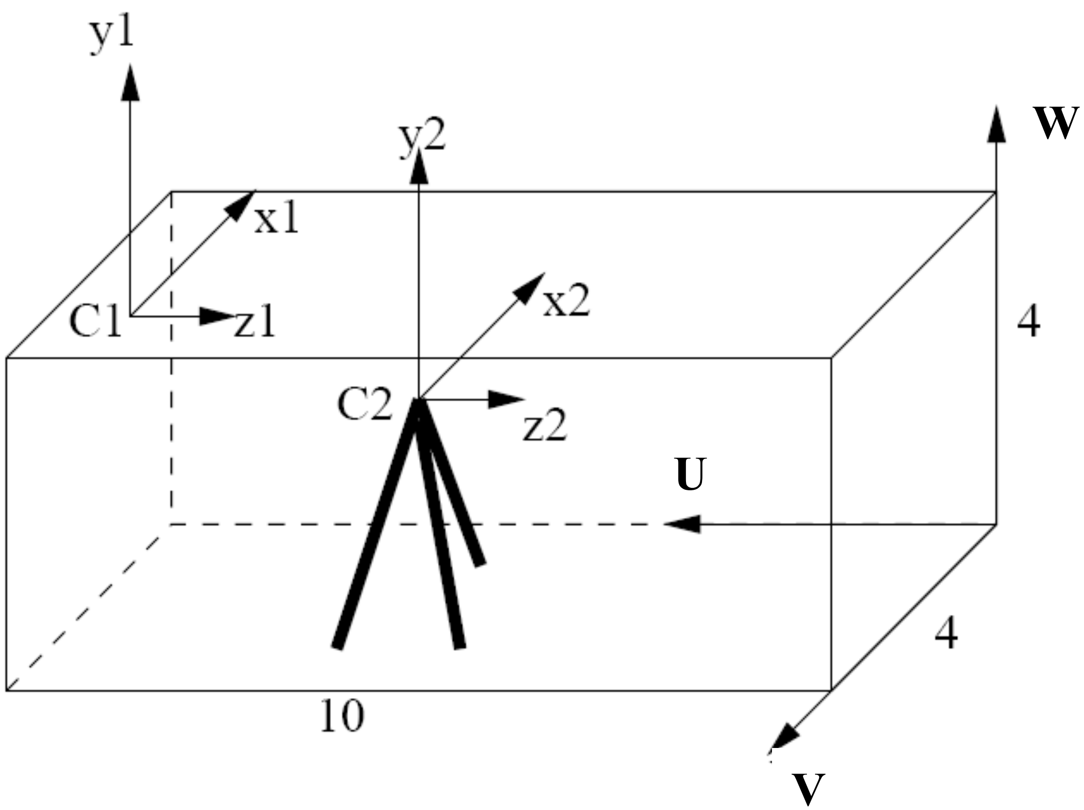
Example



$$R_{train} \begin{matrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{matrix} \qquad R_{fly} \begin{matrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{matrix}$$

Another Example

5. (Old exam problem) Consider a room $10 \times 4 \times 4$ with world coordinate system **(U,V,W)** as shown in Figure 1. The room has a stereo rig with two identical cameras with focal length $f = 1$. Camera 1 is mounted on a wall such that its center of projection is located at the point $C1$ with world coordinates $(10, 1, 3)$. Camera 2 is mounted on a tripod and it has its center of projection located at the point $C2$ with world coordinates $(7, 1, 2)$. The optical axes of both cameras are parallel to the floor of the room, the image axes X_1 and X_2 are parallel to the world axis Y and the image axes Y_1 and Y_2 are parallel to the world axis Z as shown in Figure 1. The image plane of each camera is located at $Z_i = 1$, $i = 1, 2$.



1) what is the transformation from world to camera (rotation and offset matrices), for camera 1?

2) what is the transformation from world to camera (rotation and offset matrices), for camera 2?

Note: External Parameters also often written as R,T

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}$$

$$\mathbf{R}\,(\,\mathbf{P_W} - \mathbf{C}\,)$$

$$= \mathbf{R}\,\mathbf{P_W} - \mathbf{R}\,\mathbf{C}$$

$$= \mathbf{R}\,\mathbf{P_W} + \mathbf{T}$$

Note: External Parameters also often written as R,T

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}$$

$$\mathbf{R}\,(\mathbf{P_W} - \mathbf{C})$$
$$= \mathbf{R}\,\mathbf{P_W} - \mathbf{R}\,\mathbf{C}$$
$$= \mathbf{R}\,\mathbf{P_W} + \mathbf{T}$$

Question: If I give you R and T (e.g. this matrix), how can you compute where the camera is located in the world coord system?

Robert Collins
CMPEN454

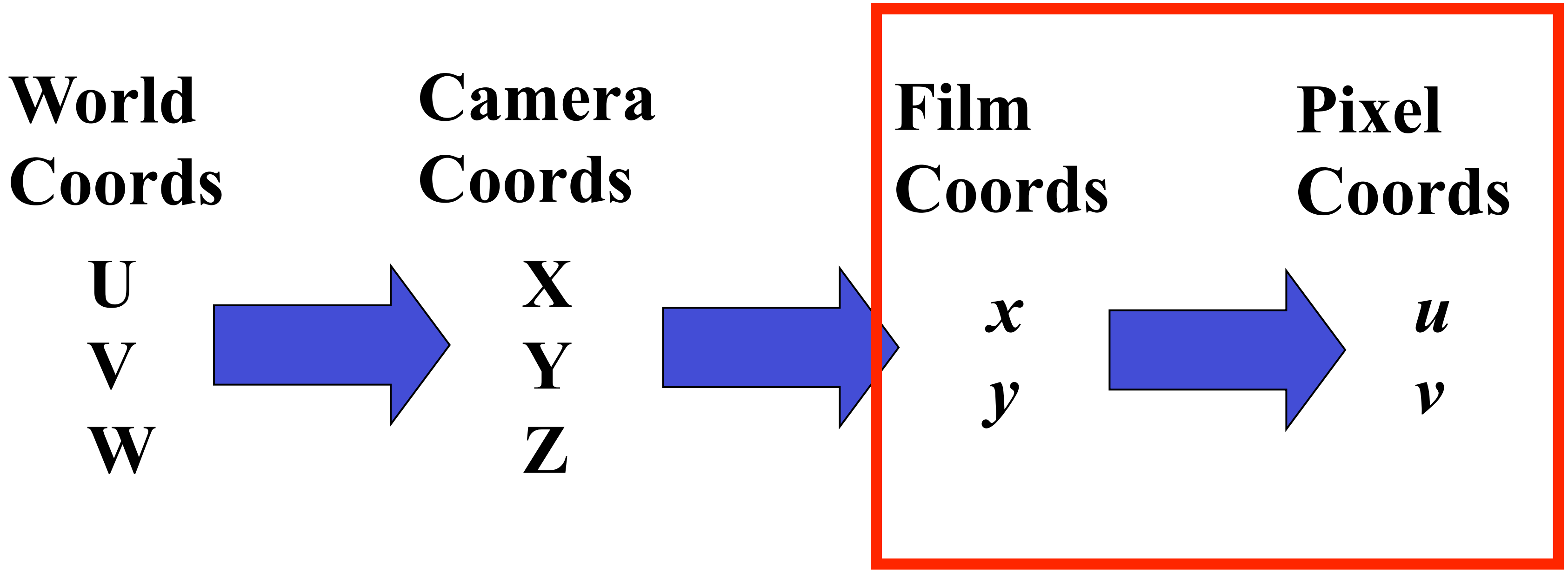
Summary, Extrinsic Parameters

World Coords		Camera Coords		Film Coords		Pixel Coords
U V W	→	X Y Z	→	x y	→	u v

We now know how to transform 3D world coordinate points into camera coords, and then do perspective project to get 2D points in the film plane.

Robert Collins
CMPEN454

Intrinsic Camera Parameters



Now we have to talk about how film coords get transformed/digitized into pixel coordinates (e.g. which column and row in a matlab image).

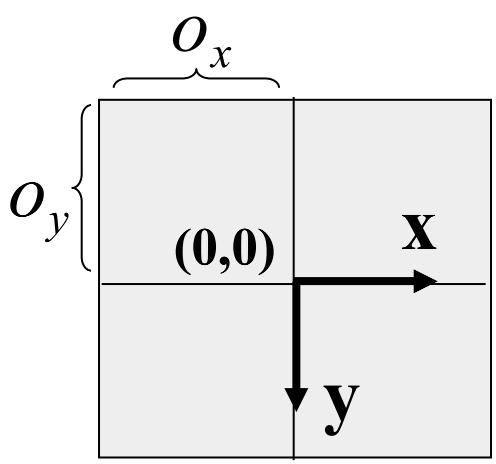
Intrinsic parameters

- Describes coordinate transformation between film coordinates (projected image) and pixel array
- Film cameras: scanning/digitization
- CCD cameras: grid of photosensors

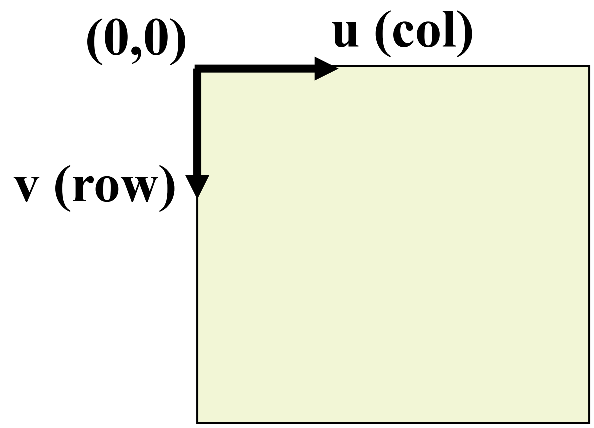
Robert Collins
CMPEN454

Intrinsic parameters (offsets)

film plane
(projected image)



pixel array (matlab)



$$u = x + o_x$$
$$v = y + o_y$$

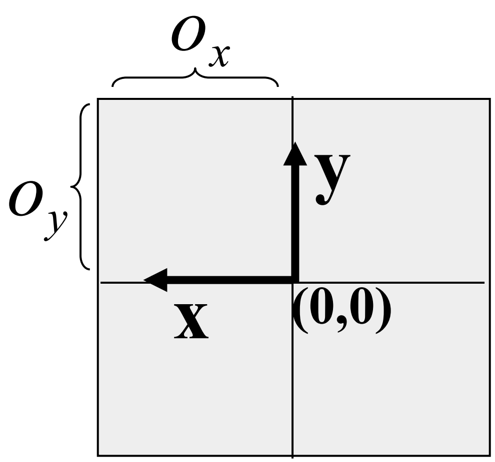
$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & o_x \\ 0 & 1 & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

o_x and o_y called image center or principle point

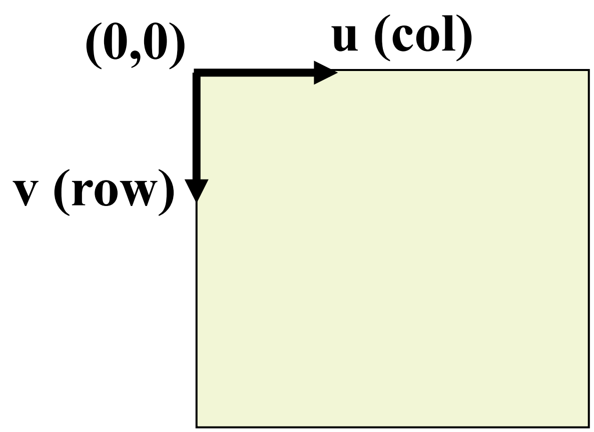
Note:

Sometimes one or more film axes needs to be reversed in direction to form the corresponding camera axis.

film plane



pixel array (matlab)



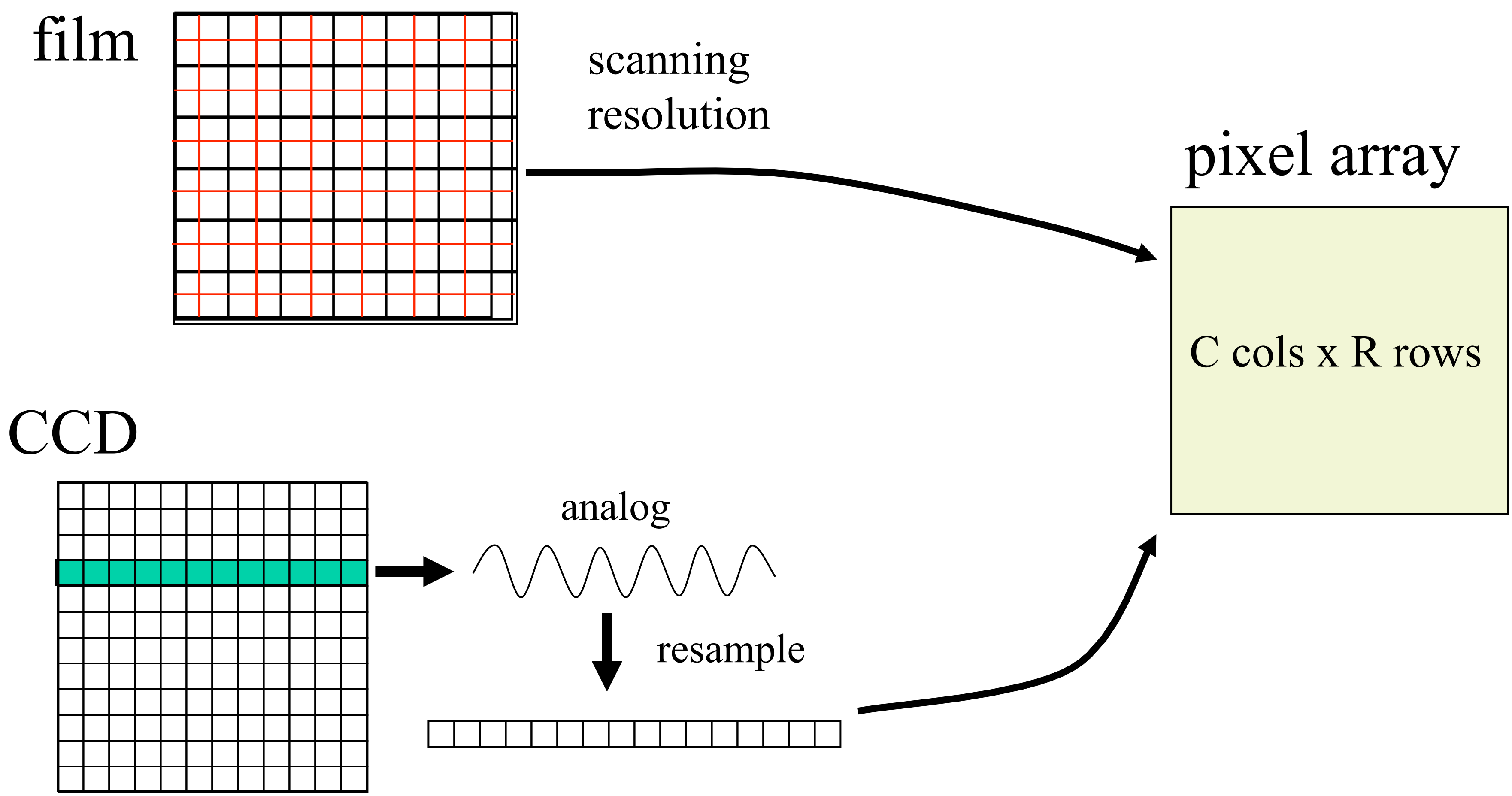
$$u = -x + o_x$$

$$v = -y + o_y$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & o_x \\ 0 & -1 & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Intrinsic parameters (scales)

sampling determines how many rows/cols in the image



Effective Scales: s_x and s_y

$$u = \frac{1}{s_x} x + o_x$$

$$v = \frac{1}{s_y} y + o_y$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/s_x & 0 & o_x \\ 0 & 1/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Note, since we have different scale factors in x and y, we don't necessarily have square pixels!

Aspect ratio is s_y / s_x

Robert Collins
CMPEN454

Perspective projection matrix

Adding the intrinsic parameters into the perspective projection matrix:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \underbrace{\begin{bmatrix} f/s_x & 0 & o_x & 0 \\ 0 & f/s_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\begin{bmatrix} 1/s_x & 0 & o_x \\ 0 & 1/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Perspective projection matrix

Adding the intrinsic parameters into the perspective projection matrix:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} f/s_x & 0 & o_x & 0 \\ 0 & f/s_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

In nonhomogeneous coords:

$$u = \frac{u'}{w'} \qquad v = \frac{u'}{w'}$$

$$\Longrightarrow \quad u = \frac{1}{s_x} f \frac{X}{Z} + o_x \qquad v = \frac{1}{s_y} f \frac{Y}{Z} + o_y$$

Simplify Your Life

Sometimes it's helpful to think of the conversion from film coords (x,y) to pixel coordinates (u,v) as a general 2D affine transformation:

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = \underbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M_{aff}}} \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{M_{proj}}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

This is useful simplification because unless you are trying to do camera calibration:
1) you typically know what the values in M_{aff} are already (practical applications), or
2) just knowing it is some affine transformation is good enough (theoretical derivations)

Side Note

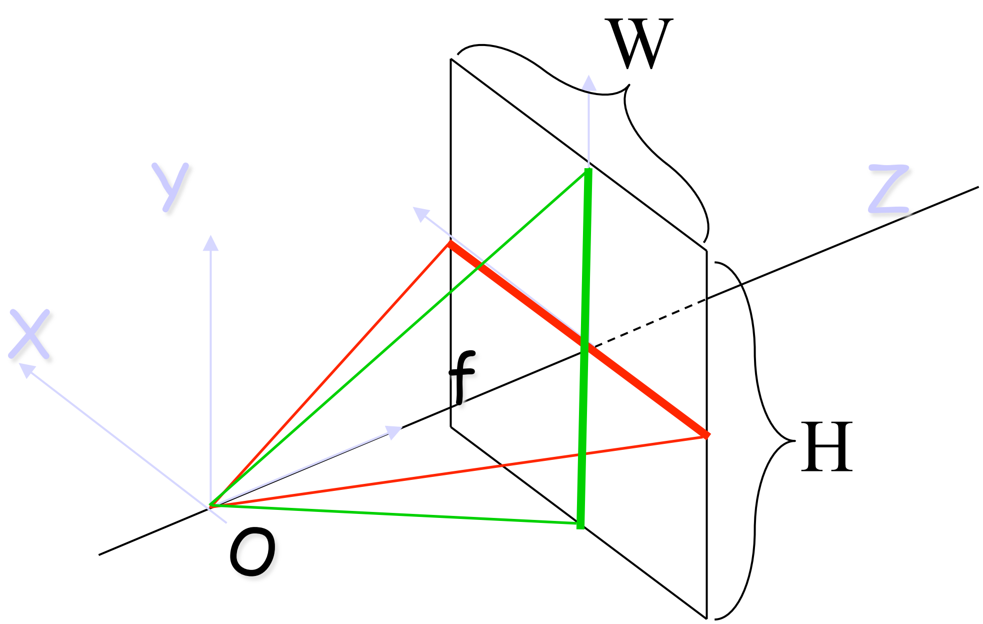
Sometimes you'll run into the idea that there are two different focal lengths, one horizontal and one vertical. Just keep in mind that it makes no sense physically, and it is just an equivalent way mathematically to combine focal length with non-square aspect ratio.

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \cancel{f/s_x}\,\mathbf{f_x} & 0 & o_x & 0 \\ 0 & \cancel{f/s_y}\,\mathbf{f_y} & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Robert Collins
CMPEN454

Side Note 2

It is common to talk about field of view (FOV) of the camera. This is the angle between opposite sides of the image. It **does** make sense to talk about two different fields of view, horizontal and vertical (even if pixels are square, the image is typically rectangular).

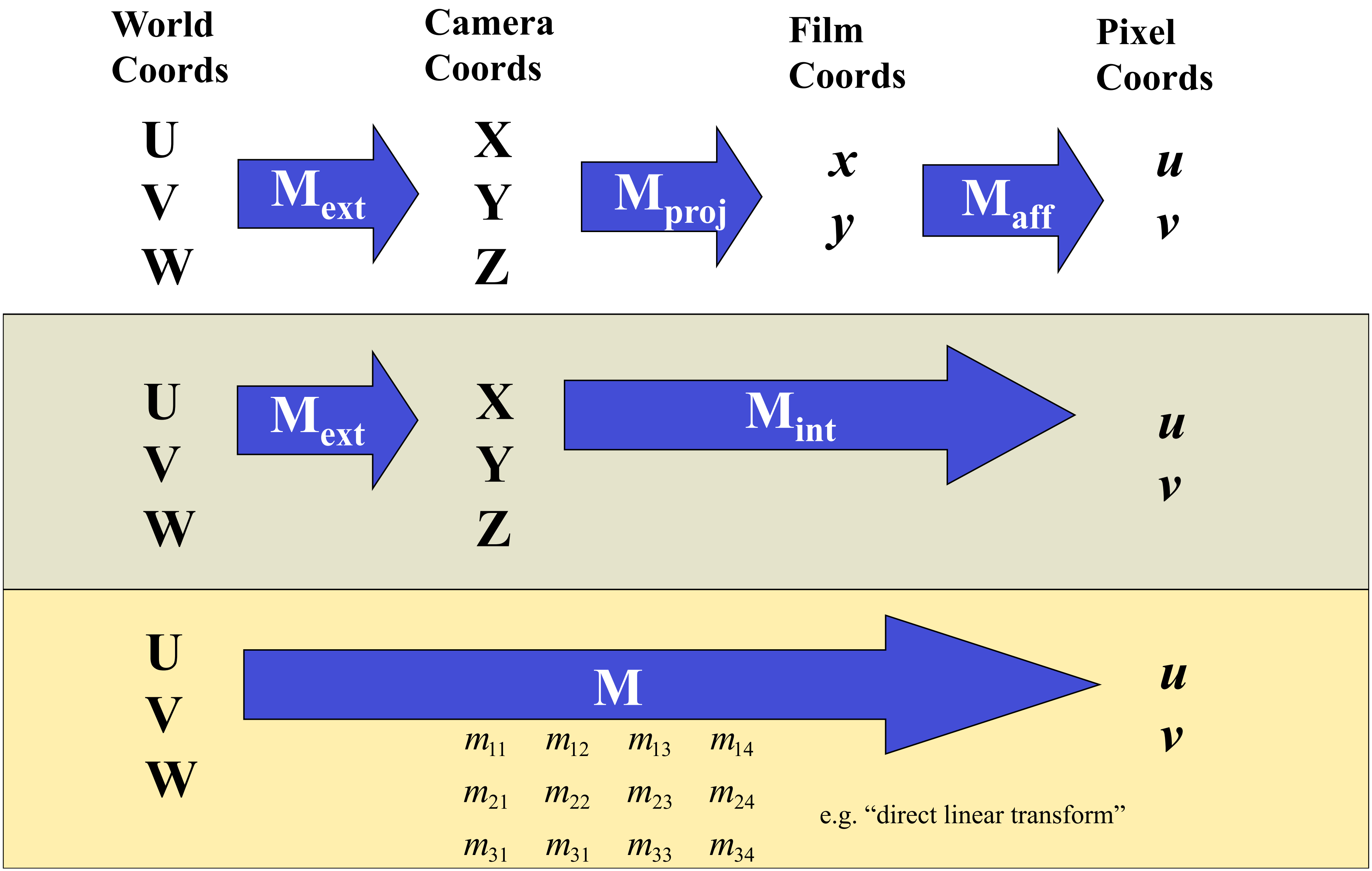


$$Fov_x = 2\ \mathrm{atan}(W/(2f))$$

$$Fov_y = 2\ \mathrm{atan}(H/(2f))$$

Robert Collins
CMPEN454

Summary : Forward Projection



Summary: Projection Equation

Pixel location | Film plane to pixels | Perspective projection | World to camera | World point

$$\begin{bmatrix}u\\v\\1\end{bmatrix} \sim \underbrace{\underbrace{\underbrace{\begin{bmatrix}\pm1/s_x & 0 & o_x\\0&\pm1/s_y&o_y\\0&0&1\end{bmatrix}}_{\mathbf{M}_{\mathbf{aff}}}\underbrace{\begin{bmatrix}f&0&0&0\\0&f&0&0\\0&0&1&0\end{bmatrix}}_{\mathbf{M}_{\mathbf{proj}}}}_{\mathbf{M}_{\mathbf{int}}}\underbrace{\begin{pmatrix}r_{11}&r_{12}&r_{13}&0\\r_{21}&r_{22}&r_{23}&0\\r_{31}&r_{32}&r_{33}&0\\0&0&0&1\end{pmatrix}\begin{pmatrix}1&0&0&-c_x\\0&1&0&-c_y\\0&0&1&-c_z\\0&0&0&1\end{pmatrix}}_{\mathbf{M}_{\mathbf{ext}}}}_{\mathbf{M}}\begin{bmatrix}U\\V\\W\\1\end{bmatrix}$$

Cavaet

We have totally ignored nonlinear lens distortion. Why? Because you can't represent that as matrix multiplication in the 3D to 2D projection chain, even using homogeneous coords. If you have nonlinear distortion you can/should calibrate and correct for it prior to applying the pinhole camera model.