Camera Models and Parameters

We will discuss camera geometry in more detail. Particularly, we will outline what parameters are important within the model. These parameters are important to several key computer vision tasks and must be computed (*calibrated*) using approaches we will discuss in later lectures.

Important Definitions

- *Frame* of reference: a measurements are made with respect to a particular coordinate system called the frame of reference.
- *World Frame*: a fixed coordinate system for representing objects (points, lines, surfaces, etc.) in the world.
- *Camera Frame*: coordinate system that uses the camera center as its origin (and the optic axis as the Z-axis)
- *Image or retinal plane*: plane on which the image is formed, note that the image plane is measured in camera frame coordinates (mm)
- *Image Frame:* coordinate system that measures *pixel* locations in the image plane.
- *Intrinsic Parameters*: Camera parameters that are internal and fixed to a particular camera/digitization setup
- *Extrinsic Parameters*: Camera parameters that are external to the camera and may change with respect to the world frame.

Camera Models Overview

- *Extrinsic Parameters*: define the location and orientation of the camera with respect to the world frame.
- *Intrinsic Parameters*: allow a mapping between camera coordinates and pixel coordinates in the image frame.
- Camera model in general is a mapping from world to image coordinates.
- This is a 3D to 2D transform and is dependent upon a number of independent parameters.

Pinhole Model Revisited

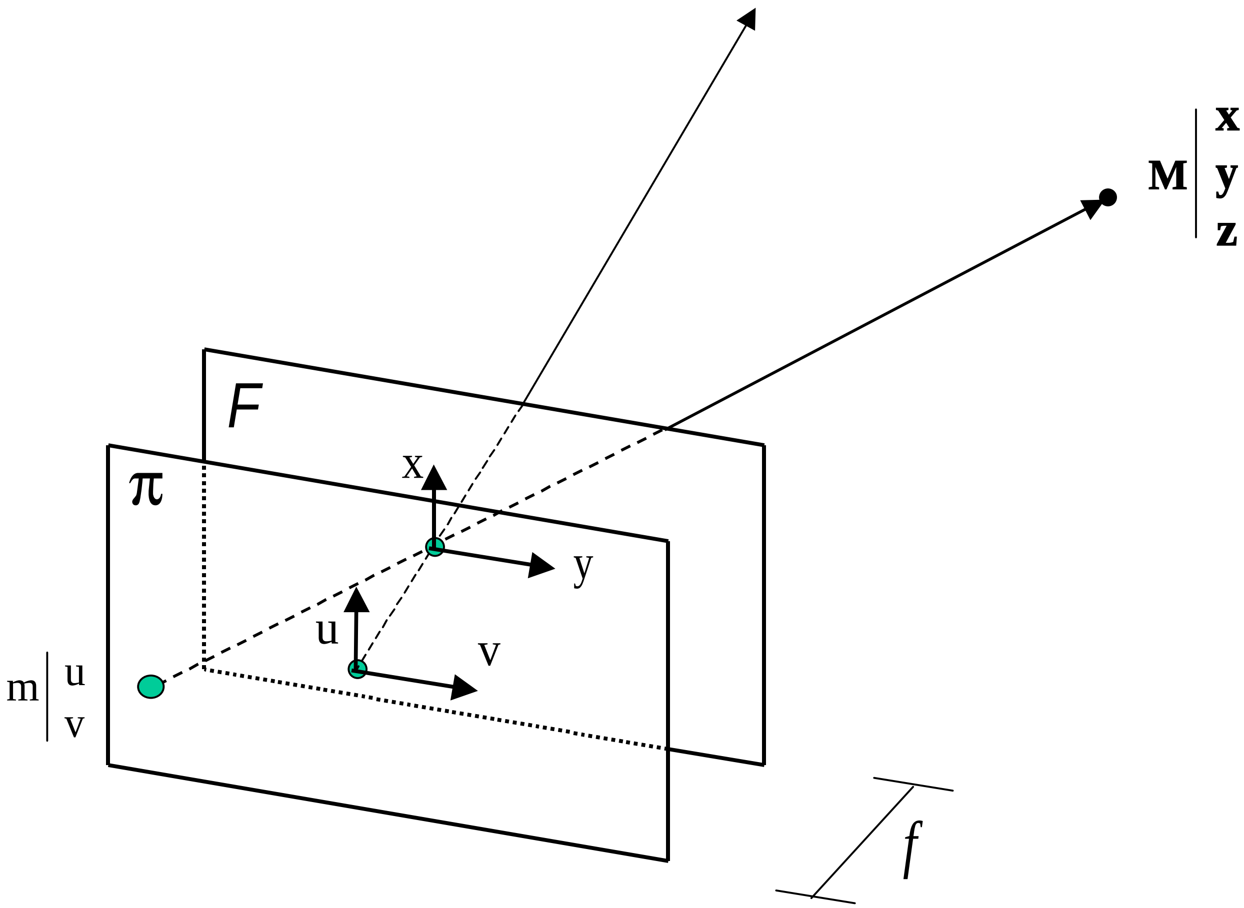
- Select a coordinate system *(,x,y,z)* for the three-dimensional space to be imaged
- Let *(u,v)* be the retinal plane π
- Then, the two are related by:

$$-\frac{f}{z} = \frac{u}{x} = \frac{v}{y}$$

- Which is written linearly in homogeneous coordinates as:

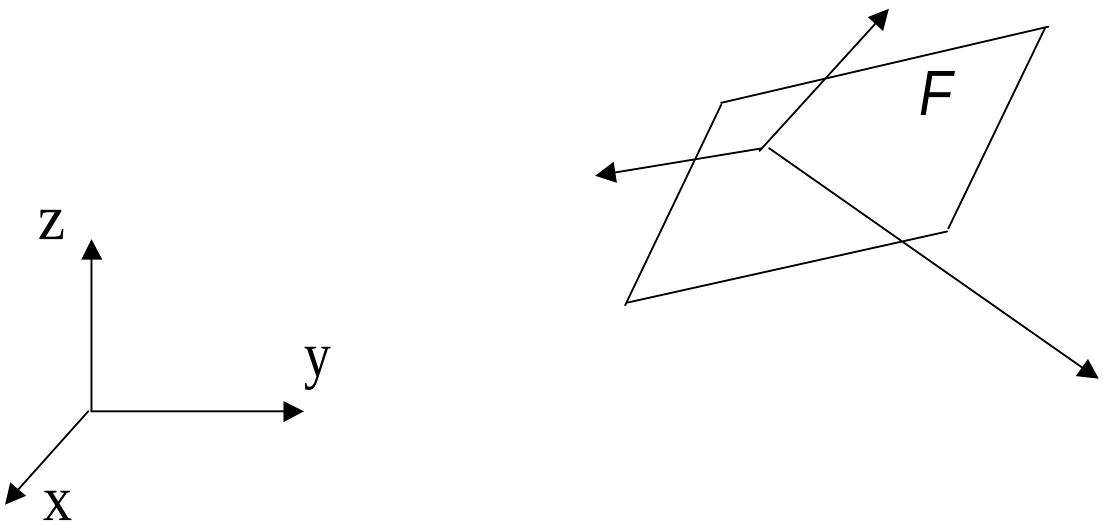
$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

The Retinal Plane



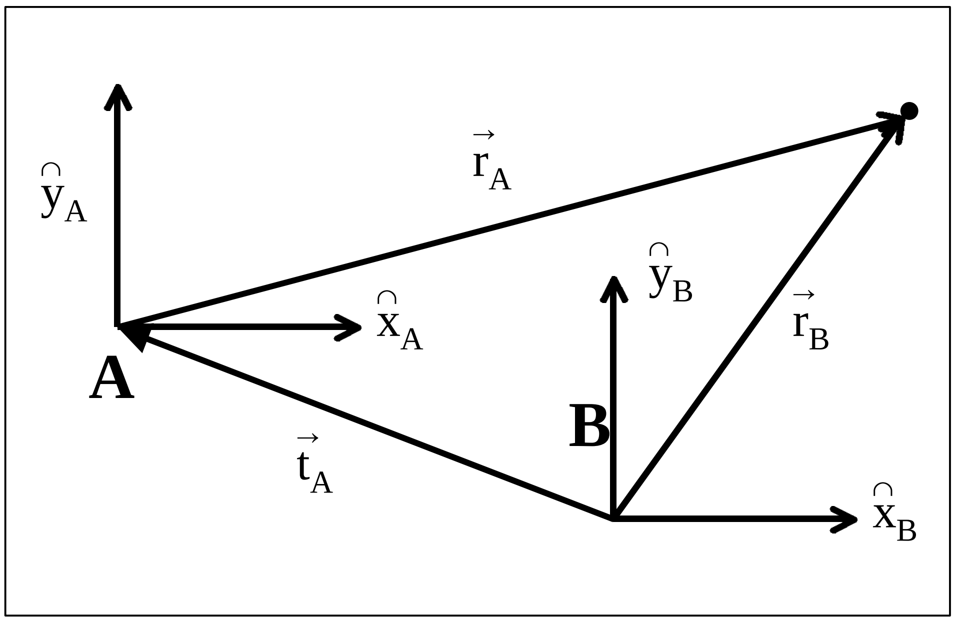
Camera orientation in the world

- The position of the camera in the world must be recovered
  - rotational component
  - translation component
- Describes absolute position of the focal plane in the world coordinate system
- This is a Euclidean transform from one coordinate system to another
- -translation, rotation



Translation

Frames A and B are related through a pure translation

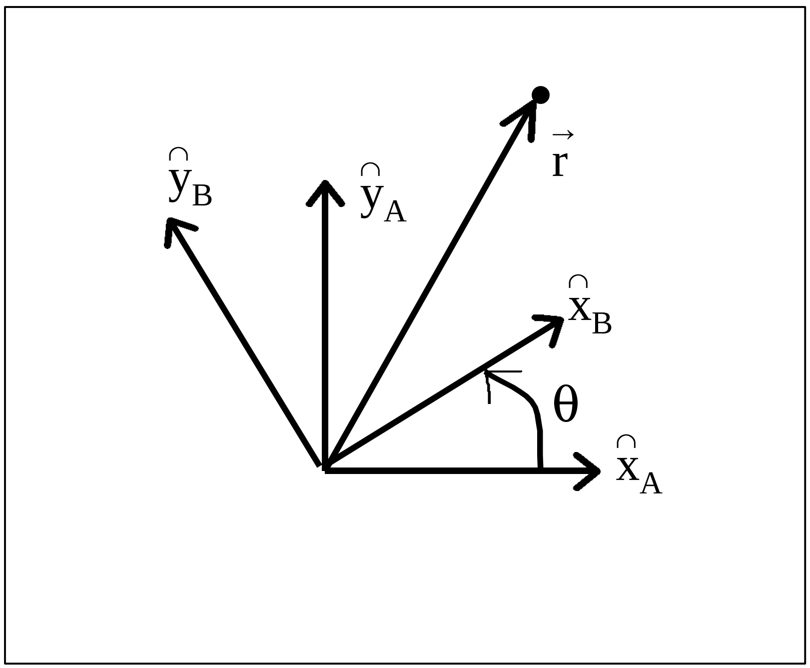


$$\vec{\mathbf{r}}_{\mathbf{A}} = \vec{\mathbf{r}}_{\mathbf{B}} + \vec{\mathbf{t}}_{\mathbf{A}}$$

where $\vec{\mathbf{t}}_{\mathbf{a}}$ represents the pure translation from frame A to frame B written in frame A coordinates

Rotations

Frames A and B are related through a pure rotation



A position vector *r*, in frame B, can be expressed in the A coordinate frame by employing the 3 X 3 transformation matrix ${}_A\mathbf{R}_B$ . . .

Rotation Matrix

$$\vec{r}_A = {}_A R_B \vec{r}_B$$

$$\begin{bmatrix} rx_A \\ ry_A \\ rz_A \end{bmatrix} = \begin{bmatrix} \hat{i}_A \cdot \hat{i}_B & \hat{i}_A \cdot \hat{j}_B & \hat{i}_A \cdot \hat{k}_B \\ \hat{j}_A \cdot \hat{i}_B & \hat{j}_A \cdot \hat{j}_B & \hat{j}_A \cdot \hat{k}_B \\ \hat{k}_A \cdot \hat{i}_B & \hat{k}_A \cdot \hat{j}_B & \hat{k}_A \cdot \hat{k}_B \end{bmatrix} \begin{bmatrix} rx_B \\ ry_B \\ rz_B \end{bmatrix}$$

This projection of frame B onto frame A clearly converts a position vector, $\vec{r}_B$, written in frame B, into the corresponding coordinates in frame A, $\vec{r}_A$.

Interpreting the Rotation Matrix

To interpret the rotation matrix for this transformation:

- the rows of ${}_A\mathbf{R}_B$ represent the projection of the basis vectors for frame A onto the basis vectors of frame B
- the columns of ${}_A\mathbf{R}_B$ represent the basis vectors of frame B projected onto the basis vectors of frame A

Rotations

One way to specify the rotation matrix ${}_A\mathbf{R}_B$ is to write the base vectors

$$\left(\hat{i}\,,\,\hat{j}\,,\hat{k}\,\right)_B$$

in frame A coordinates and to enter the result into the columns of ${}_A\mathbf{R}_B$

Rotations

If $\widehat{x}_B^A$ is a column vector representing the $\overline{x}$ axis of frame B written in frame A coordinates, then

$$ {}_A\mathrm{R}_B = \begin{bmatrix} \widehat{x}_B^A & \widehat{y}_B^A & \widehat{z}_B^A \end{bmatrix} = \begin{bmatrix} \cos(q) & -\sin(q) & 0 \\ \sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

Rotations: x

For completeness, we will look at the rotation matrix for rotations about all three axes:

$$rot(\widehat{x},q) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(q) & -\sin(q) \\ 0 & \sin(q) & \cos(q) \end{bmatrix}$$

Rotations: y

For completeness, we will look at the rotation matrix for rotations about all three axes:

$$rot(\hat{y}, q) = \begin{bmatrix} \cos(q) & 0 & \sin(q) \\ 0 & 1 & 0 \\ -\sin(q) & 0 & \cos(q) \end{bmatrix}$$

Rotations: z

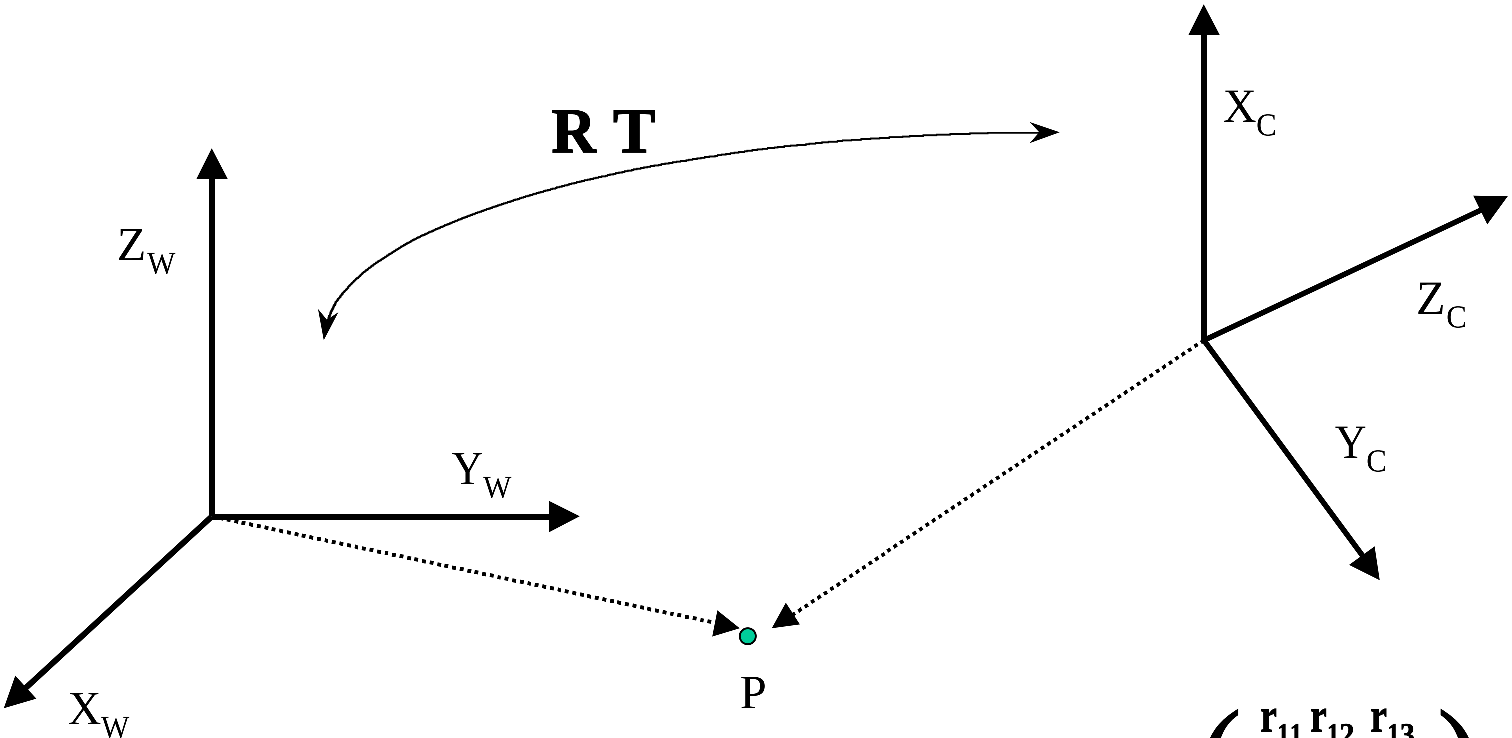
For completeness, we will look at the rotation matrix for rotations about all three axes:

$$rot(\hat{z},q) = \begin{bmatrix} \cos(q) & -\sin(q) & 0 \\ \sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Extrinsic Parameters

- Recall the *fundamental equations of perspective projection*
  - assumed the orientation of the camera and world frame known
  - this is actually a difficult problem known as *extrinsic pose* problem
  - using only image information recover the relative position and orientation of the camera and world frames
- This transformation is typically defined by:
  - 3-D translation vector $\mathbf{T=[x,y,z]}^T$
    - defines relative positions of each frame
  - 3x3 rotation matrix, **R**
    - rotates corresponding axes of each frame into each other
    - **R** is *orthogonal:* ($\mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}$)

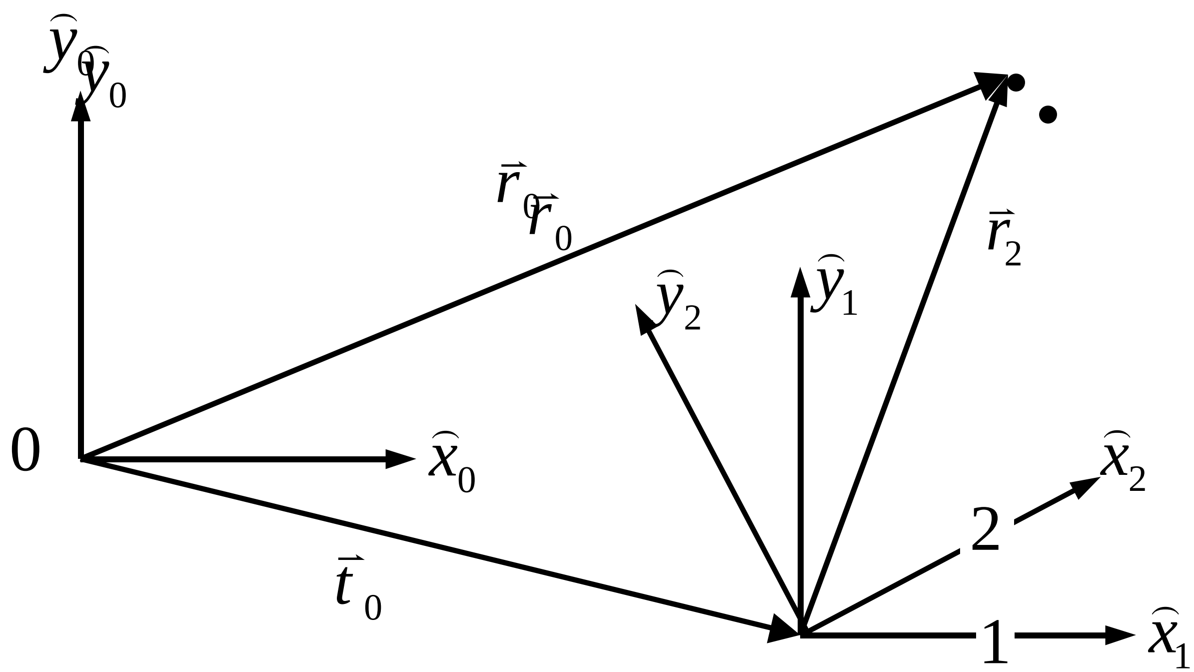
Extrinsic Parameters



Note: we write $\mathbf{R} = \begin{pmatrix} \mathbf{r}_{11} & \mathbf{r}_{12} & \mathbf{r}_{13} \\ \mathbf{r}_{21} & \mathbf{r}_{22} & \mathbf{r}_{23} \\ \mathbf{r}_{31} & \mathbf{r}_{32} & \mathbf{r}_{33} \end{pmatrix}$

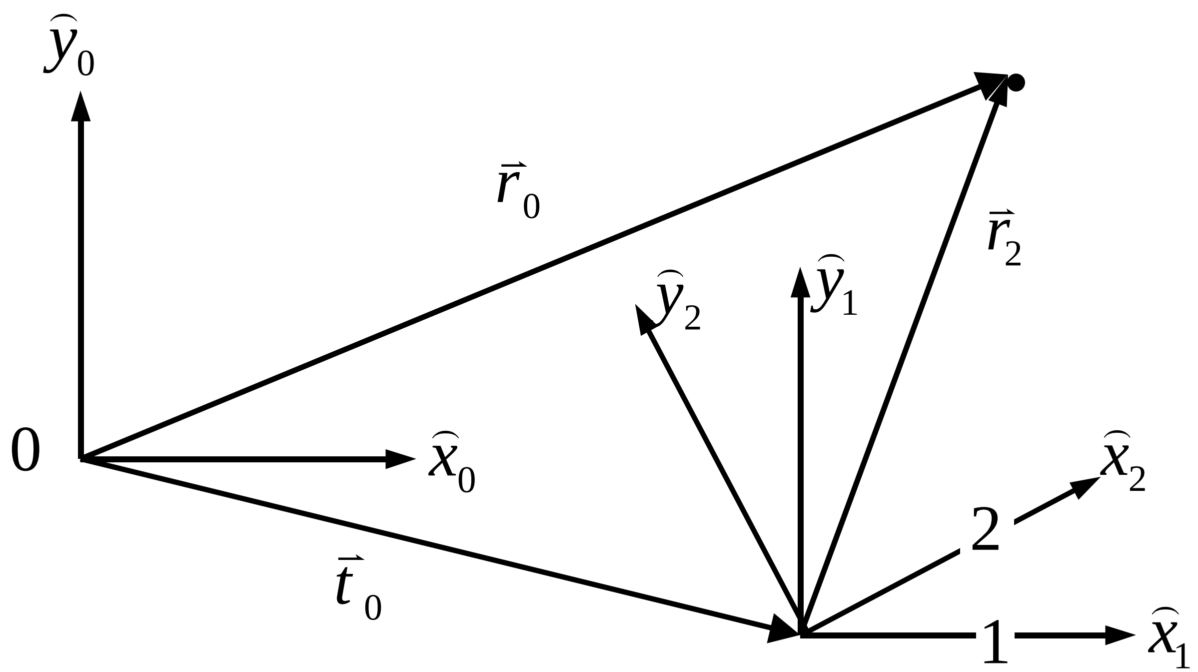
How to write *both* rotation and translation as a single, composed transform?

Homogeneous Transformations



We consider the general case where frame 0 and frame 2 are related to one another through both rotation and translation

Homogeneous Transformations



The homogeneous transform is a mechanism for expressing this form of compound transformation

Homogeneous Transforms

- Expand the dimensionality of the domain space
- Same transformation now can be expressed in a linear fashion
- Linear transforms can be easily composed and written as a single matrix multiply
- Vectors, in homoeneous space take on a new parameter r. This is the scale of the vector along the new axis and is arbitrary: $[x\ y\ z\ r]$
- Normalization, after the transform has been applied is accomplished simply by dividing each vector component by r $[x\ y\ z\ 1] = [x'/r\ \ y'/r\ \ z'/r\ \ r/r]$

Homogeneous Transformations

- Let ${}_0\mathbf{T}_2$ be the compound transformation consisting of a translation from 0 to 1, followed by a rotation from 1 to 2

Homogeneous Transformations

- In vector notation, this homogeneous transformation and corresponding homogeneous position vectors are written:

$$
{}_0\mathrm{T}_2 = \left[\begin{array}{ccc|c} & {}_1\mathrm{R}_2 & & \vec{t}_0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]
\qquad
\vec{r}_2 = \begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix}_2
$$

$$
\text{Then, } \vec{r}_0 = {}_0\mathrm{T}_2 r_2 = {}_1\mathrm{R}_2 \vec{r}_2 + \vec{t}_0
$$

Composing Transformations

The homogeneous transform provides a convenient means of *constructing* compound transformations

Composing Transformations

Example: Suppose

$$ {}_0\mathrm{T}_4 = {}_0\mathrm{T}_1 \; {}_1\mathrm{T}_2 \; {}_2\mathrm{T}_3 \; {}_3\mathrm{T}_4 $$

where:

$$ {}_0\mathrm{T}_1 = translation(\hat{x}_0,\ 1.0) $$

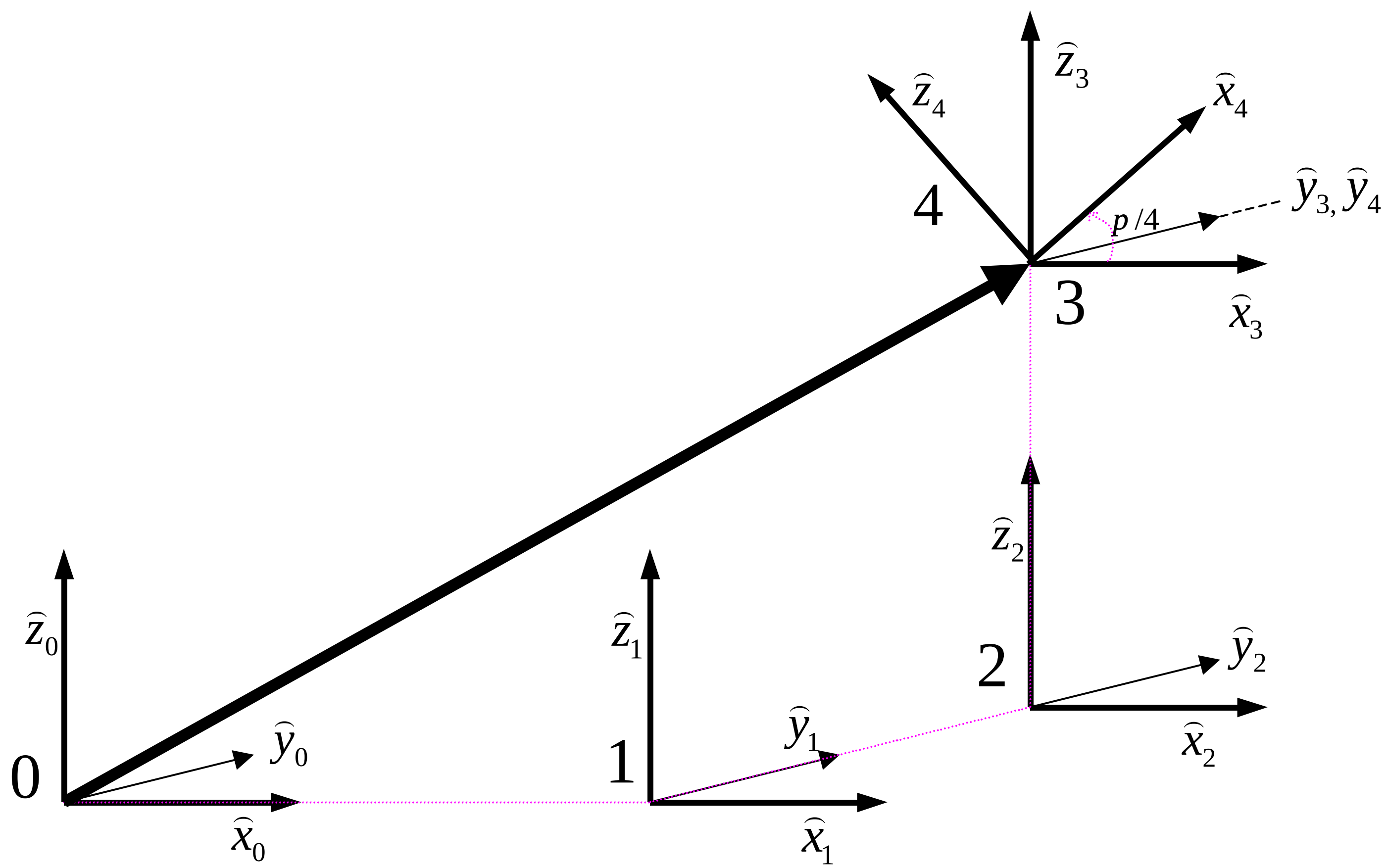
$$ {}_1\mathrm{T}_2 = translation(\hat{y}_1,\ 1.0) $$

$$ {}_2\mathrm{T}_3 = translation(\hat{z}_2,\ 1.0) $$

$$ {}_3\mathrm{T}_4 = rotation(\hat{y}_3,\ -p\ /\ 4) $$

Composing Transformations

Example: ${}_0T_4 = {}_0T_1 \; {}_1T_2 \; {}_2T_3 \; {}_3T_4$

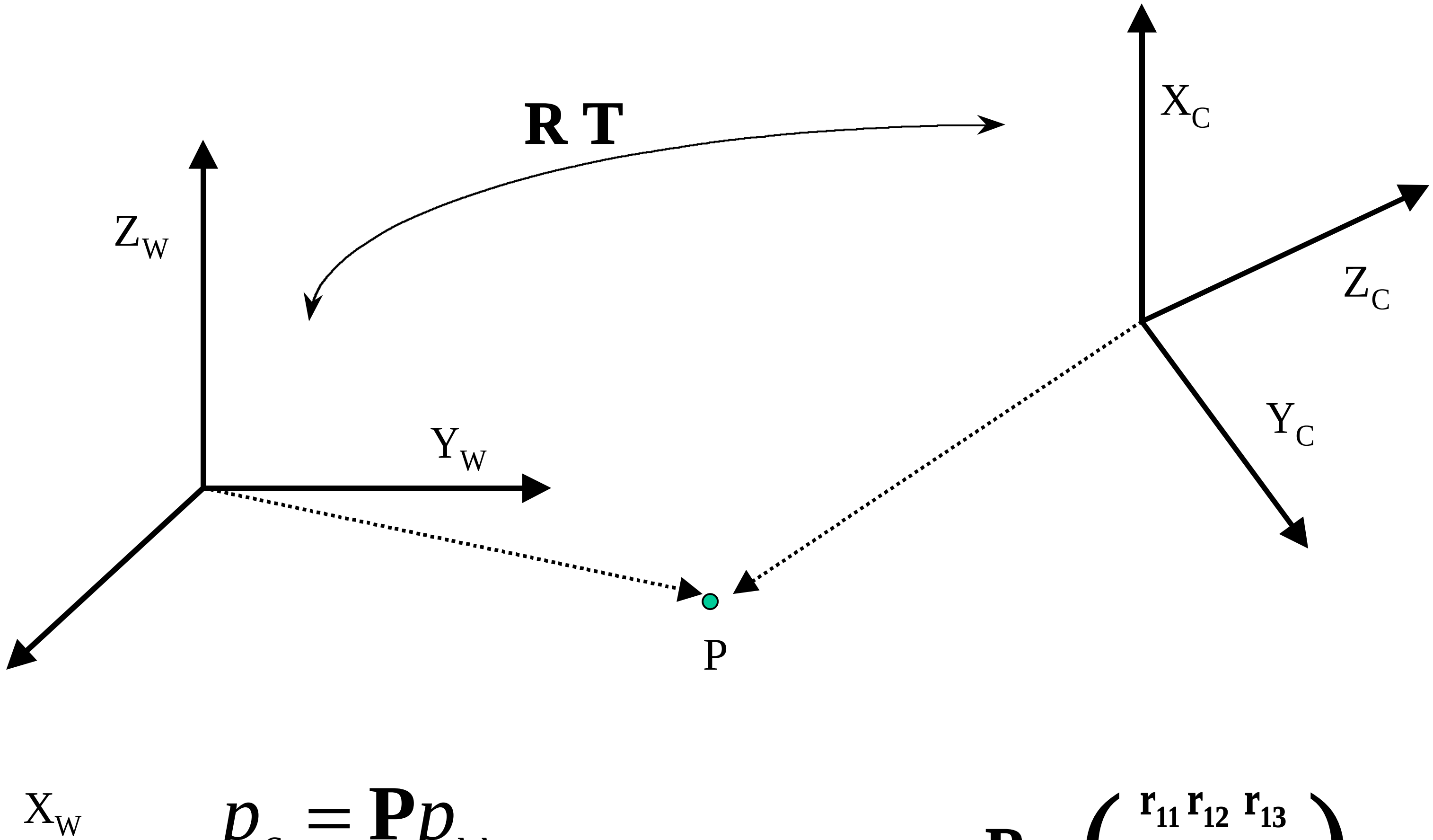


Composing Transformations

The resulting compound transformation is

$$
{}_{0}T_{4} = \begin{bmatrix} 0.707 & 0 & -0.707 & 1 \\ 0 & 1 & 0 & 1 \\ 0.707 & 0 & 0.707 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Extrinsic Parameters



$$p_c = \mathbf{P} p_w$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0\,0\,0 & 1 \end{bmatrix}$$

$$\mathbf{R} = \begin{pmatrix} \mathbf{r}_{11} & \mathbf{r}_{12} & \mathbf{r}_{13} \\ \mathbf{r}_{21} & \mathbf{r}_{22} & \mathbf{r}_{23} \\ \mathbf{r}_{31} & \mathbf{r}_{32} & \mathbf{r}_{33} \end{pmatrix}$$

$$\mathbf{T} = \begin{bmatrix} T_x & T_y & T_z \end{bmatrix}^T$$

Intrinsic Parameters

- Characterize the optical, geometric, and digital characteristics of the camera
- Defined by:
  - perspective projection: focal length ***f***
  - transformation between camera frame and pixel coordinates
  - geometric distortion introduced by the lens
- Transform between camera frame and pixels:

$$x = -(x_{im} - o_x)s_x$$

$$y = -(y_{im} - o_y)s_y$$

- (o_x,o_y) image center (principle point)
- (s_x,s_y) effective size of pixels in mm in horizontal and vertical directions

Camera Lens Distortion

- Optical system itself a source of distortions
  - evident at the image periphery
  - worsened by large field of view
- Modeled accurately as *radial distortion*

$$x = x_d\,(1 + k_1 r^2 + k_2 r^4)$$

$$y = y_d\,(1 + k_1 r^2 + k_2 r^4)$$

- (x_d, y_d) distorted points, and $r^2 = x^2_d + y^2_d$
- note: this is a radial displacement of the image points
- because $k_2 << k_1, \quad k_2$ is often ignored.

Camera models (again)

Using the equations from previous slides, we are able to transform pixel coordinates to world points this is our camera model.

- Recall $x = f\,\dfrac{X}{Z}$ $y = f\,\dfrac{Y}{Z}$
- Linear Perspective Projection Equations:

$$-(x_{im} - o_x)s_x = f\,\frac{\mathbf{R}^T{}_1(\mathbf{P}_w - \mathbf{T})}{\mathbf{R}^T{}_3(\mathbf{P}_w - \mathbf{T})}$$

$$-(x_{im} - o_x)s_x = f\,\frac{\mathbf{R}^T{}_2(\mathbf{P}_w - \mathbf{T})}{\mathbf{R}^T{}_3(\mathbf{P}_w - \mathbf{T})}$$

$\mathbf{R}_i$, i=1,2,3 is a 3D vector formed by the I-th row of **R**

Linear Matrix Representation

- If we neglect radial distortion, we can rewrite the linear perspective transform as a matrix product.
- A matrix is used for intrinsic and extrinsic parameters.

$$M_{int} = \begin{pmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$M_{ext} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & -\mathbf{R}^{T}{}_{1}\mathbf{T} \\ r_{21} & r_{22} & r_{23} & -\mathbf{R}^{T}{}_{1}\mathbf{T} \\ r_{31} & r_{32} & r_{33} & -\mathbf{R}^{T}{}_{1}\mathbf{T} \end{pmatrix}$$

Using the Perspective Equations

- 3x3 M_{int} only depends on the intrinsic parameters
- 3x4 M_{ext} depends on extrinsic parameters
- We make use of these by introducing *homogeneous coordinates* to point vectors in the world.
  - $\mathbf{P}_w$ must be expressed in homogenous coordinates to allow direct multiplication to M_{int} and M_{ext}

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = M_{int} M_{ext} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

- $[x_1,x_2,x_3]^T$ is the projected point, using the vector we compute image coordinates:

$$x_1/x_3 = x_{im}$$
$$x_2/x_3 = y_{im}$$

The Perspective Transform

- M_{ext} : from world to camera frame
- M_{int} : from camera to image
- Can be viewed, formally, as a relation between a 3D point and its perspective projection on the image plane.
  - Maps points in *projective space*, space of vectors $[x_w,y_w,z_w]^T$ to the *projective plane*, space of vectors $[x_1,y_1,z_1]^T$.
  - defined up to a
  - 11 independent parameters

Camera models from the Projective Equations

- Various models can be derived by setting appropriate constraints on the projection equations
- **The Perspective Model**
  - $o_x = o_y = 0$
  - $s_x = s_y = 1.0$
- **The Weak-Perspective Model**
  - note that image point **p** of world point **P** is given by:

$$\mathbf{p} = M \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = \begin{pmatrix} f\,\mathbf{R}^T{}_1(\mathbf{T} - \mathbf{P}) \\ f\,\mathbf{R}^T{}_2(\mathbf{T} - \mathbf{P}) \\ f\,\mathbf{R}^T{}_3(\mathbf{T} - \mathbf{P}) \end{pmatrix}$$

Weak-Perspective Continued

$\mathbf{R}^T_3(\mathbf{P} - \mathbf{T})$ is the distance of **P** from the perspective center along the optical axis.

Therefore:

$$\left| \frac{\mathbf{R}^T_3(\mathbf{P_i} - \mathbf{P})}{\mathbf{R}^T_3(\mathbf{P} - \mathbf{T})} \right| << 1$$

Is the weak-perspective approximation. Using this approximation the perspective matrix can be written to eliminate negligible terms.