EECE 5639 Computer Vision I

Lecture 3

Image Formation & Camera Model

Next Class

Color, Filtering

Projective Plane

The projective plane P2 is the set of equivalence classes of triplets of numbers (not all zero) where two triplets (x,y,z) and (x',y',z') are equivalent if and only if there is a real number k such that

$$(x,y,z) = k(x',y',z')$$

Projective 3D Space

The projective space P3 is the set of equivalence classes of quadruplets of numbers (not all zero) where two quadruplets (x,y,z,w) and (x',y',z',w') are equivalent if and only if there is a real number k such that

$$(x,y,z,w) = k(x',y',z',w')$$

What happens if k=0?

k=0 can be used to represent points “at infinity”. All points at infinity in the 2D projective space lie on the line “at infinity”. Points at infinity are also called **IMPROPER POINTS or IDEAL POINTS**.

In projective space **ALL lines intersect at a point**. Some lines intersect at a point on the infinity line. We call these lines PARALLEL.

Duality Principle

To **any theorem** of 2-dimensional projective geometry there corresponds a **dual theorem**, which may be derived by **interchanging the role of points and lines in the original theorem**

$$\mathbf{x} \longleftrightarrow \mathbf{l}$$

$$\mathbf{x}^{\mathsf{T}}\mathbf{l} = 0 \longleftrightarrow \mathbf{l}^{\mathsf{T}}\mathbf{x} = 0$$

$$\mathbf{x} = \mathbf{l} \times \mathbf{l}' \longleftrightarrow \mathbf{l} = \mathbf{x} \times \mathbf{x}'$$

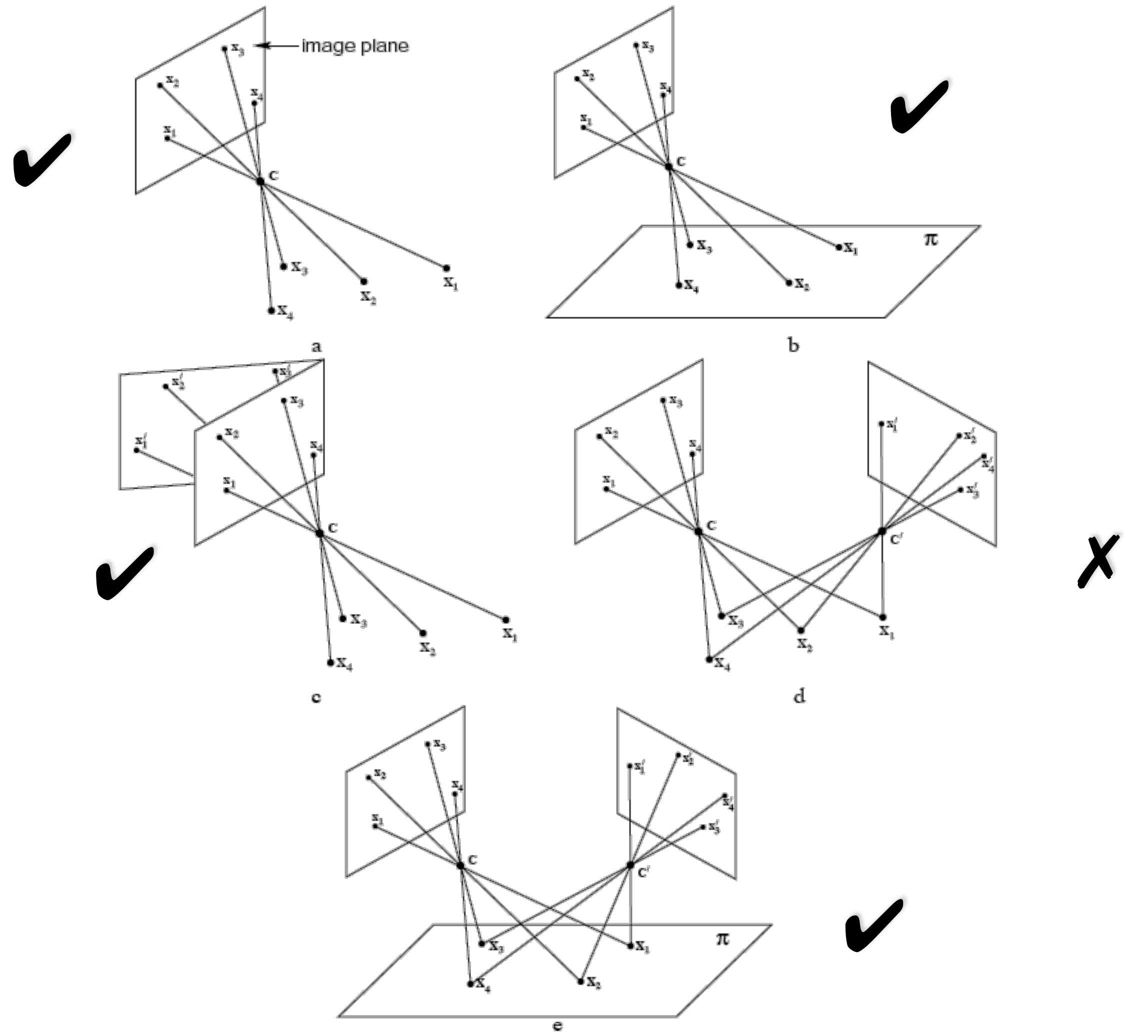
Projective Transformations

A projective transformation is a LINEAR TRANSFORMATION between projective spaces.

Two important classes:

- Linear Invertible of Pn into themselves (n=1,2,3) (i.e. P2 to P2)
- Transformations between P3 and P2 (that model image formation)

Projective Transformations



Theorem:

A projective transformation of Pn onto itself is completely determined by its action on n+2 points.

Proof

Consider two corresponding points p and p' in a P2 projectivity T:

$$p = Tp'$$

We want to show that T (which has 9 entries) can be determined by 4 correspondences.

Proof

Consider the following 4 points and their images:

These points are the “standard basis” of P2

$$
\begin{array}{ll}
p_1 = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T & p'_1 = \lambda\begin{pmatrix} x'_1 & y'_1 & z'_1 \end{pmatrix}^T \\
p_2 = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^T & p'_2 = \mu\begin{pmatrix} x'_2 & y'_2 & z'_2 \end{pmatrix}^T \\
p_3 = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T & p'_3 = \nu\begin{pmatrix} x'_3 & y'_3 & z'_3 \end{pmatrix}^T \\
p_4 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T & p'_4 = \rho\begin{pmatrix} x'_4 & y'_4 & z'_4 \end{pmatrix}^T
\end{array}
$$

An use the fact that T is invertible $p' = T^{-1}p$

Proof

$$\begin{pmatrix} \lambda x'_1 \\ \lambda y'_1 \\ \lambda z'_1 \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} t_{11} \\ t_{21} \\ t_{31} \end{pmatrix}$$

Etc ... Doing the same for p2 and p3:

$$\begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} = \begin{pmatrix} \lambda x'_1 & \mu x'_2 & \nu x'_3 \\ \lambda y'_1 & \mu y'_2 & \nu y'_3 \\ \lambda z'_1 & \mu z'_2 & \nu z'_3 \end{pmatrix}$$

Proof

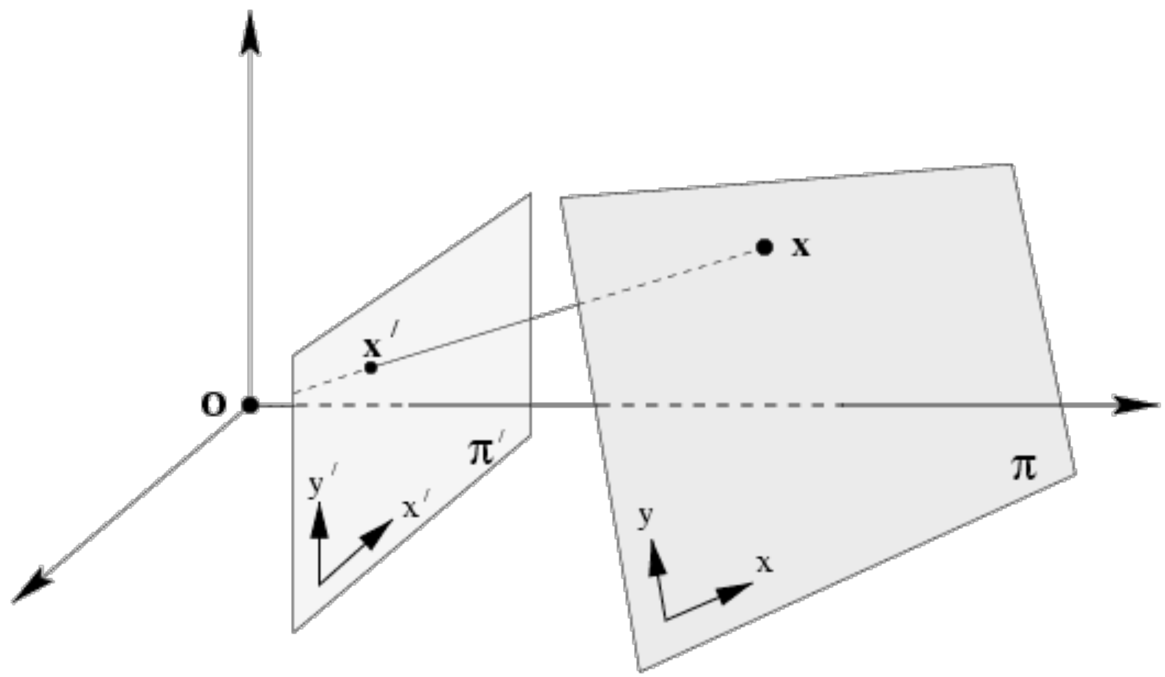
Now using p4:

$$\begin{pmatrix} \rho x'_4 \\ \rho y'_4 \\ \rho z'_4 \end{pmatrix} = \begin{pmatrix} \lambda x'_1 & \mu x'_2 & \nu x'_3 \\ \lambda y'_1 & \mu y'_2 & \nu y'_3 \\ \lambda z'_1 & \mu z'_2 & \nu z'_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

We can find the entries of the inverse of T up to a constant!

$$\boxed{\begin{aligned} \lambda x'_1 + \mu x'_2 + \nu x'_3 &= \rho x'_4 \\ \lambda y'_1 + \mu y'_2 + \nu y'_3 &= \rho y'_4 \\ \lambda z'_1 + \mu z'_2 + \nu z'_3 &= \rho z'_4 \end{aligned}}$$

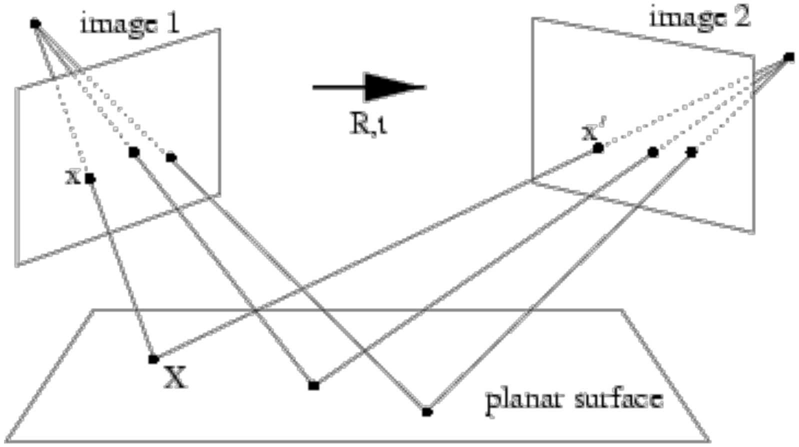
Mapping between planes

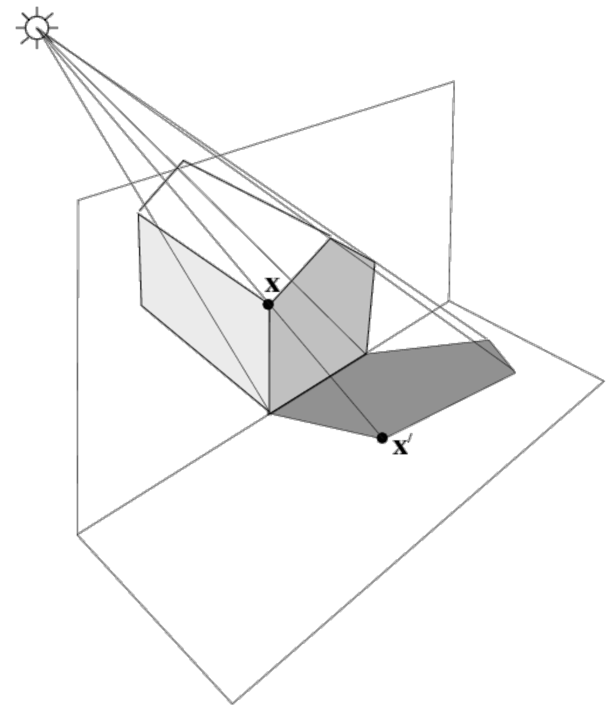


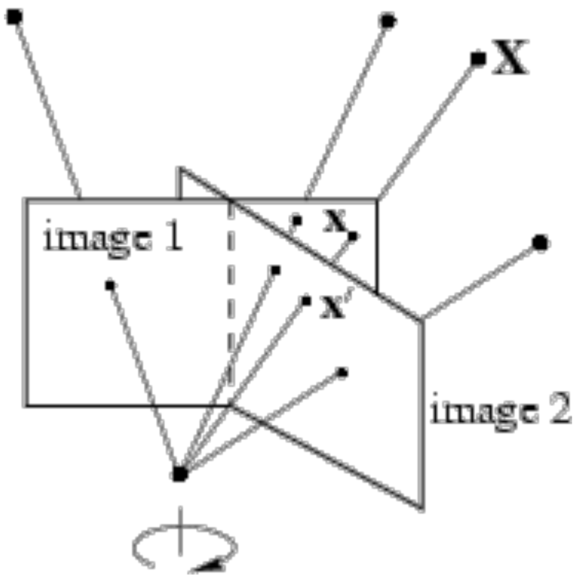
central projection may be expressed by x'=Hx

(application of theorem)

More examples







A Hierarchy of 2D Transformations

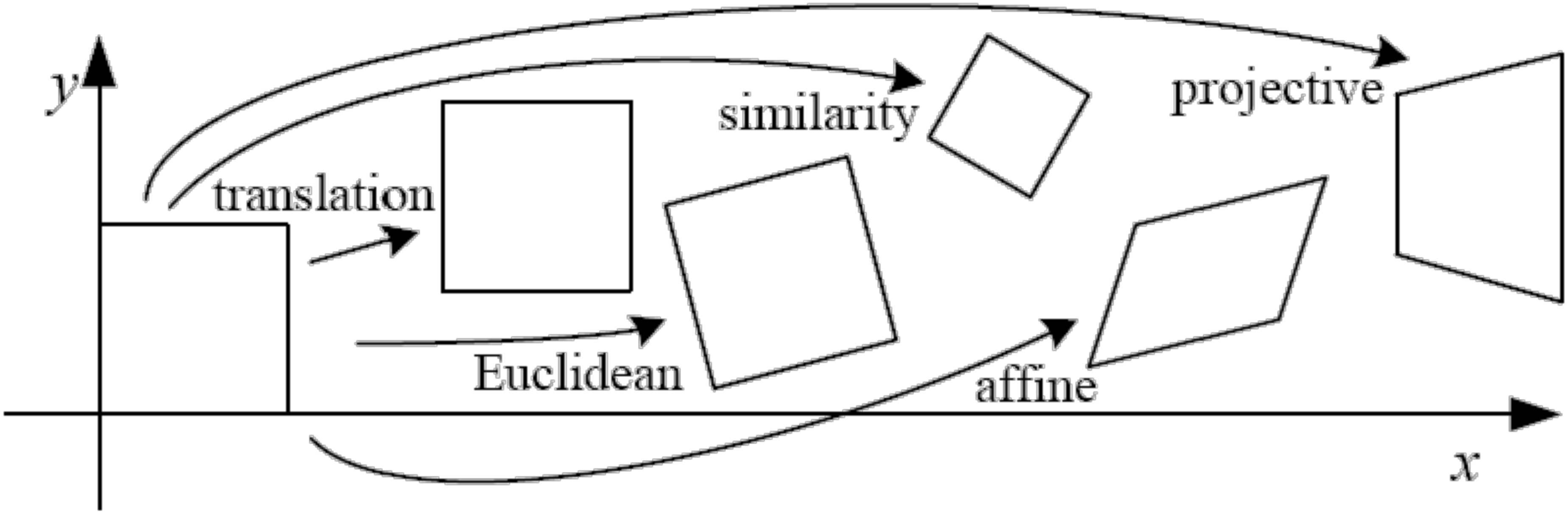
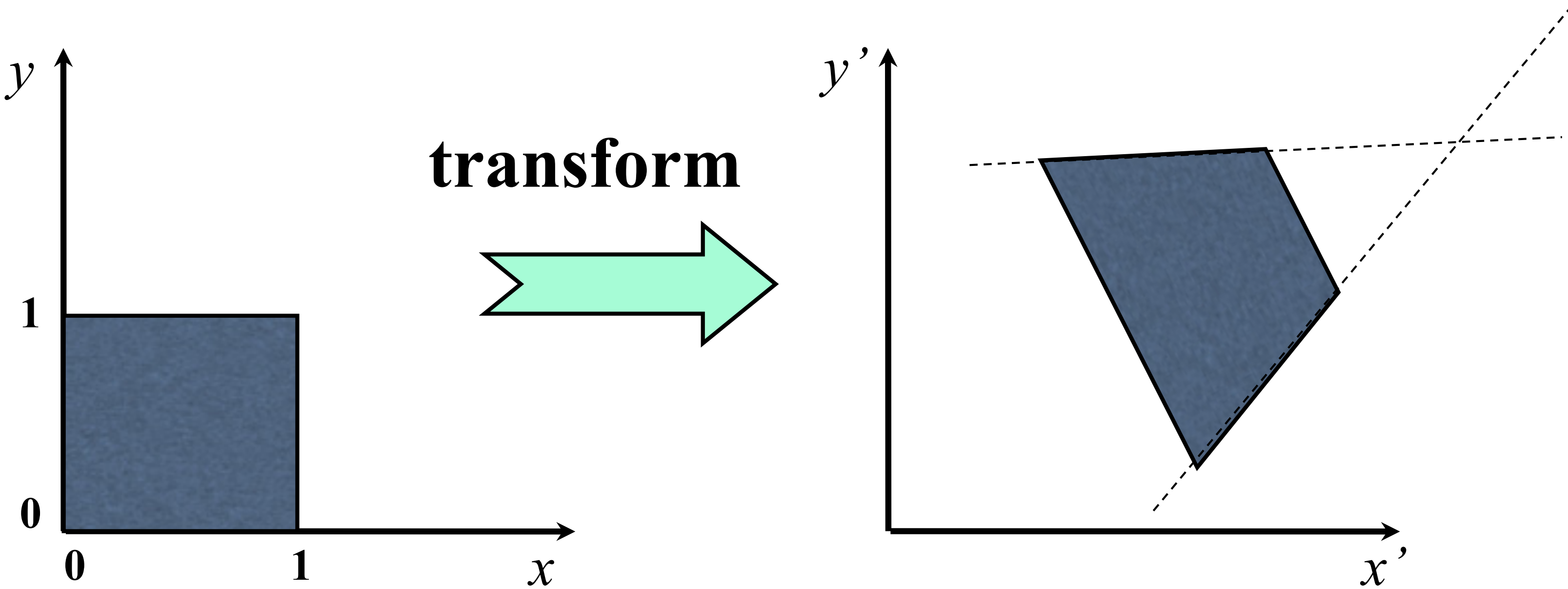


FIGURE 1. Basic set of 2D planar transformations

from R.Szeliski

Projective



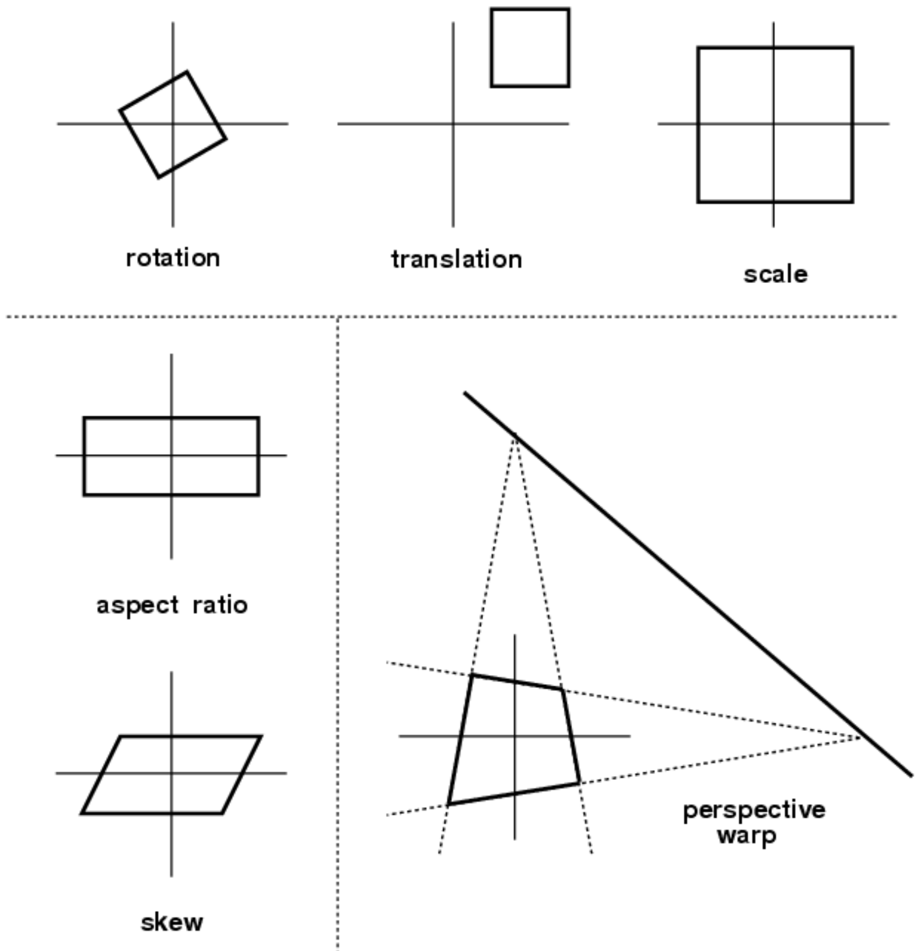
Note!

$$p' = \frac{Ap+b}{c^T p+1}$$

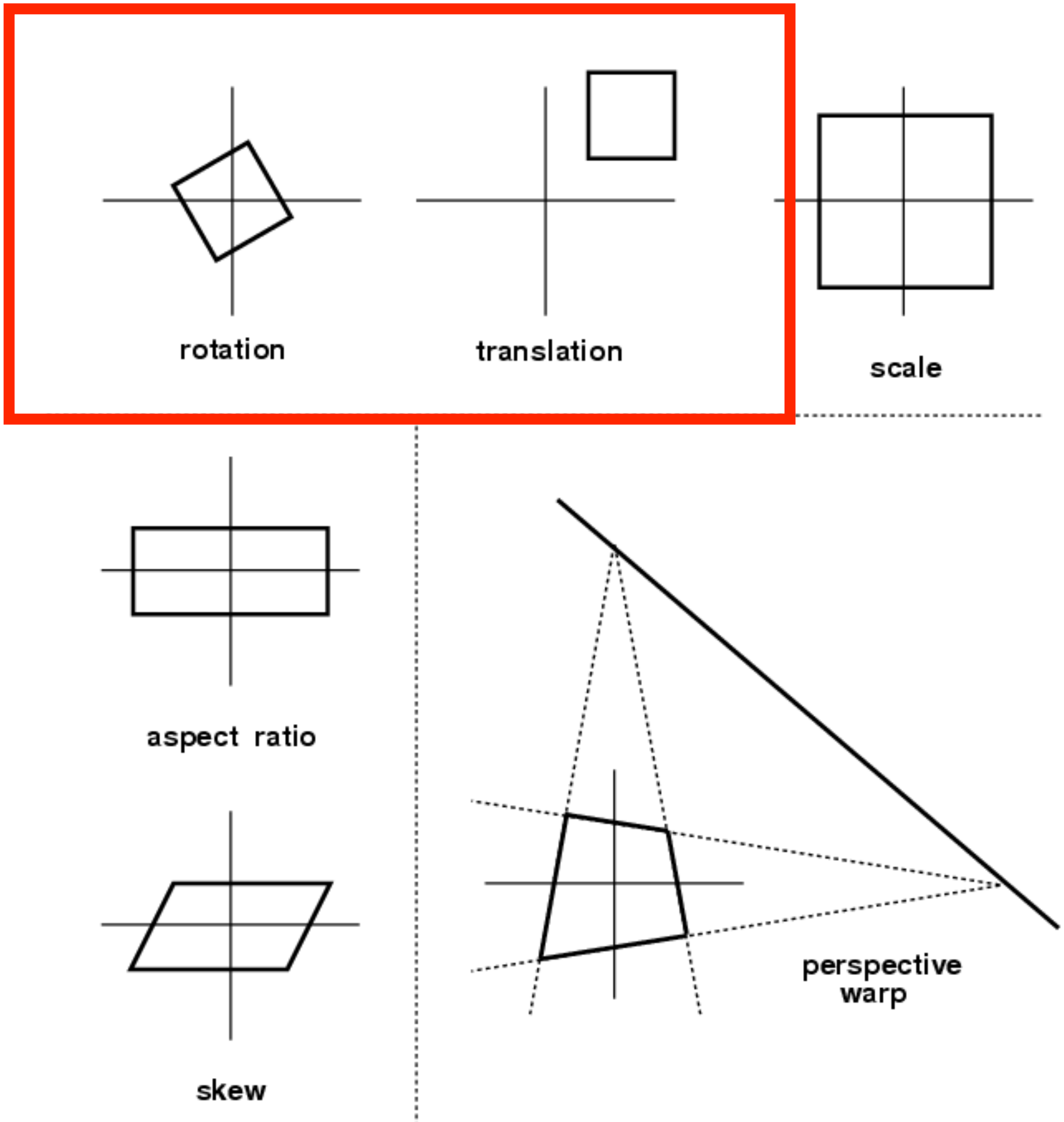
$$\begin{bmatrix} p' \\ 1 \end{bmatrix} \sim \begin{bmatrix} A & b \\ c^T & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix}$$

equations **matrix form**

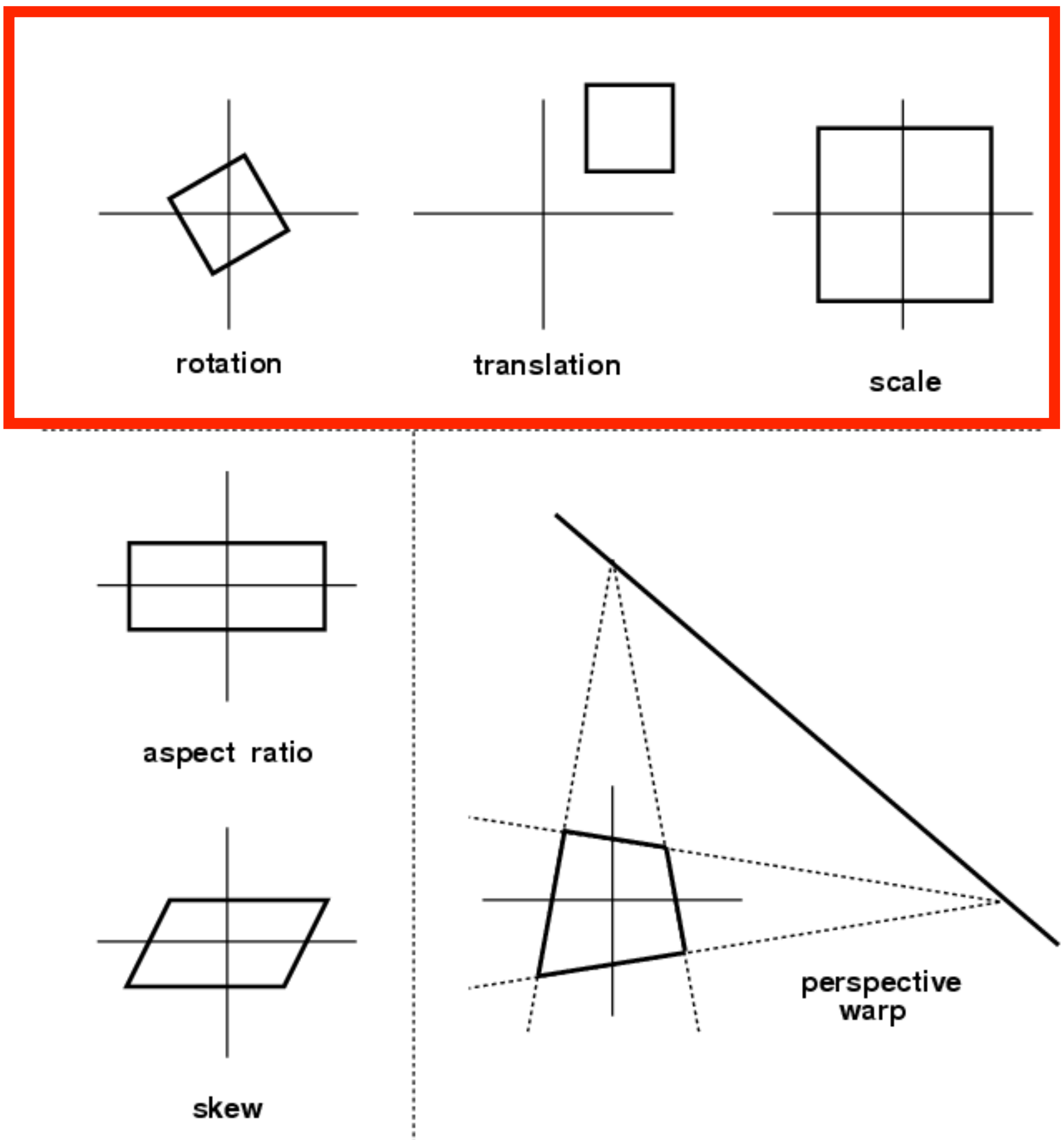
Summary of 2D Transformations



Euclidean

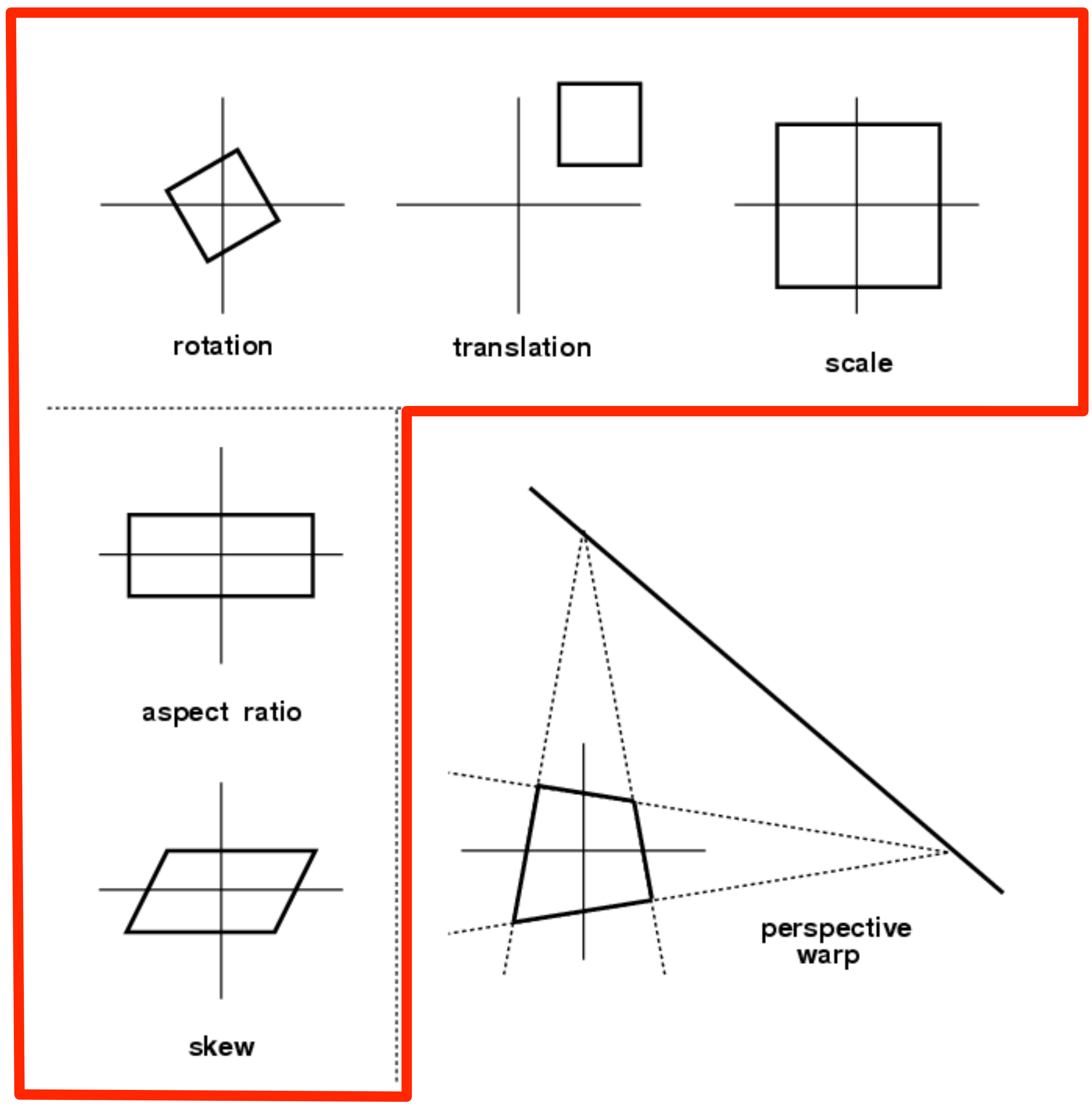

rotation
translation
scale
aspect ratio
skew
perspective
warp

Similarity

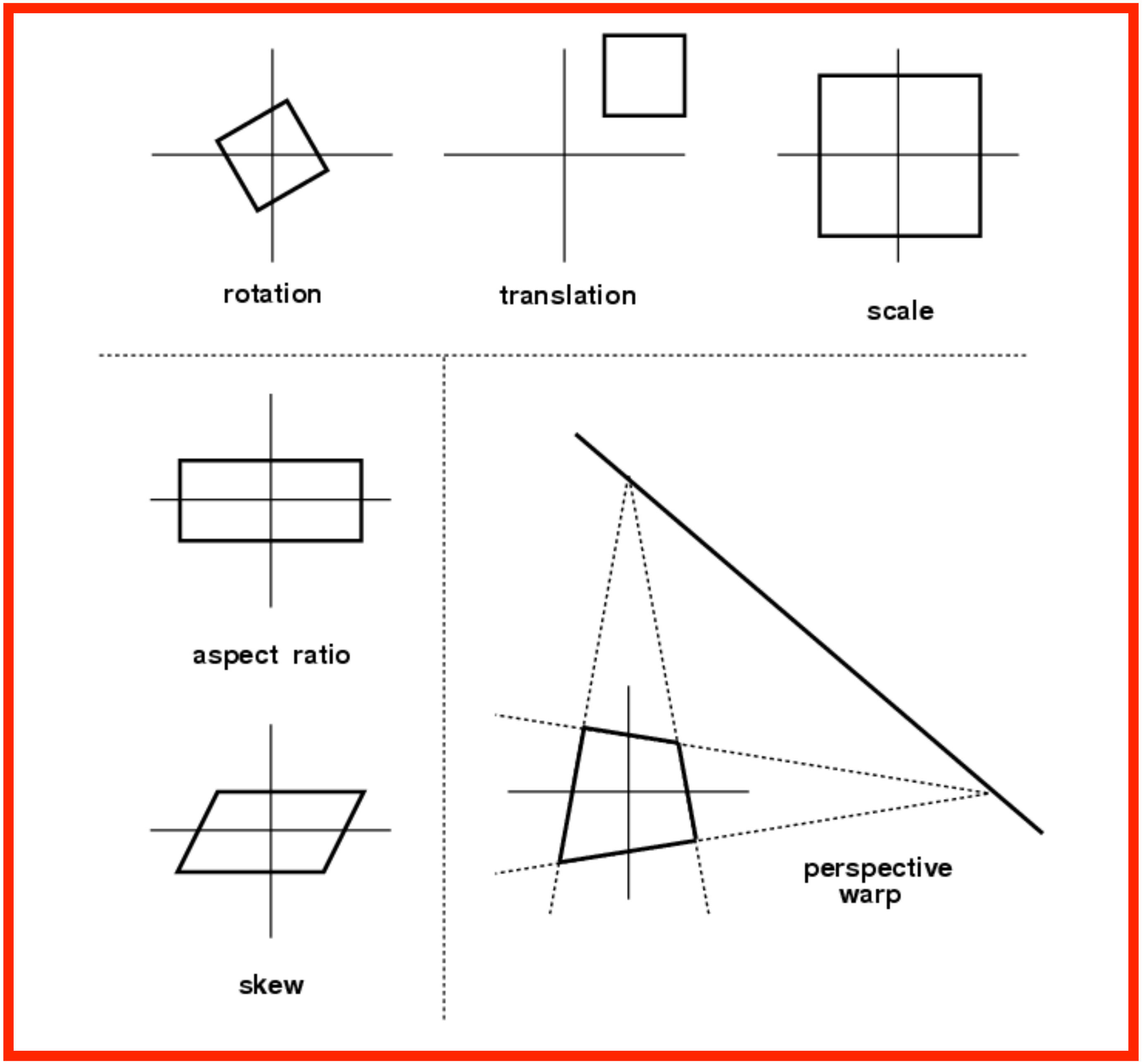


rotation
translation
scale
aspect ratio
skew
perspective
warp

Affine

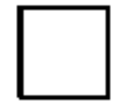
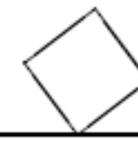

rotation
translation
scale
aspect ratio
skew
perspective
warp

Projective



rotation
translation
scale
aspect ratio
skew
perspective
warp

Summary of 2D Transformations

Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation $+\cdots$	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths $+\cdots$	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles $+\cdots$	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism $+\cdots$	
projective	$\begin{bmatrix} \boldsymbol{H} \end{bmatrix}_{3\times 3}$	8	straight lines	

from R.Szeliski

Transformation Groups

A mathematical **group** G is composed of a set of elements and an associative operator * such that:

1) The set is closed under operator *

$$A \in G \text{ and } B \in G \rightarrow A * B \in G$$

2) There exists an identity element I such that

$$A * I = I * A = A$$

3) Each element A has an inverse A^{-1} such that

$$A^{-1} * A = A * A^{-1} = I$$

Example of a Group

Claim: translations matrices form a group under composition (matrix multiplication operator)

Group element: matrices of form:

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Operator: matrix multiplication *

Note: matrix multiplication is indeed associative
A * (B * C) = (A * B) * C

Translation Group (cont)

Verify: closed under composition

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & s_x \\ 0 & 1 & s_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & s_x + t_x \\ 0 & 1 & s_y + t_y \\ 0 & 0 & 1 \end{bmatrix} \checkmark$$

Translation Group (cont)

Verify: existence of identity element

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} * \; ? \; = \; ? \; * \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \checkmark$$

Translation Group (cont)

Verify: existence of inverse for every element

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} * \, ? = \, ? * \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \checkmark$$

Example2: Euclidean Group

Closed under composition

$$\begin{bmatrix} R_1 & t_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_2 & t_2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_1 R_2 & R_1 t_2 + t_1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

Identity exists

$$\begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

Inverse Exists

$$\begin{bmatrix} R^T & -R^T t \\ 0 & 1 \end{bmatrix}$$

Check it !

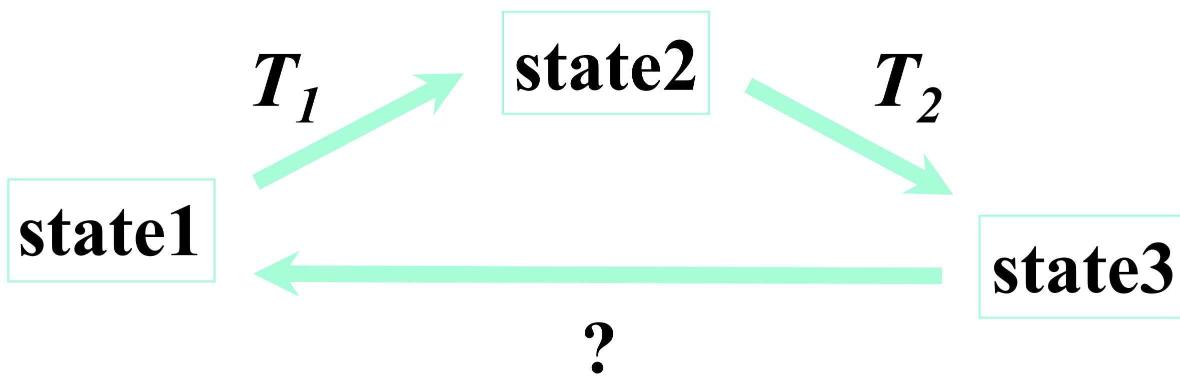
Transformation Groups

We have verified that translations form a group.

Why does it matter?

Groups are very well-behaved. When doing computations with groups we can freely compose transformations and assume inverses exist.

Example

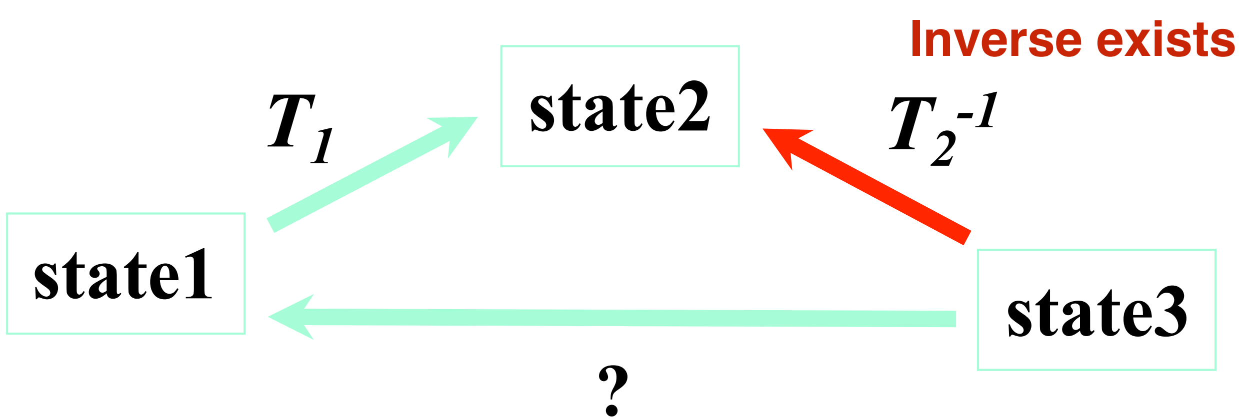


Transformation Groups

We have verified that translations form a group.

Why does it matter?

Groups are very well-behaved. When doing computations with groups we can freely compose transformations and assume inverses exist.

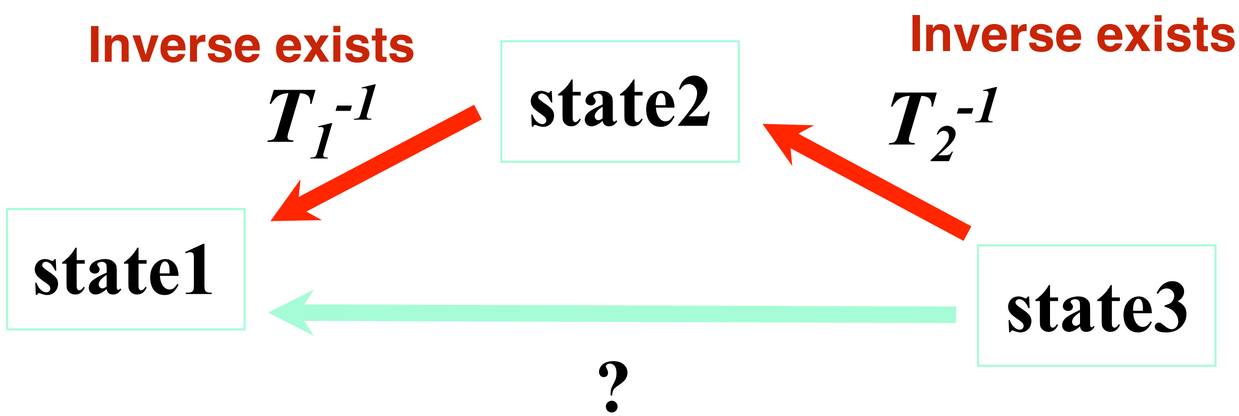


Transformation Groups

We have verified that translations form a group.

Why does it matter?

Groups are very well-behaved. When doing computations with groups we can freely compose transformations and assume inverses exist.

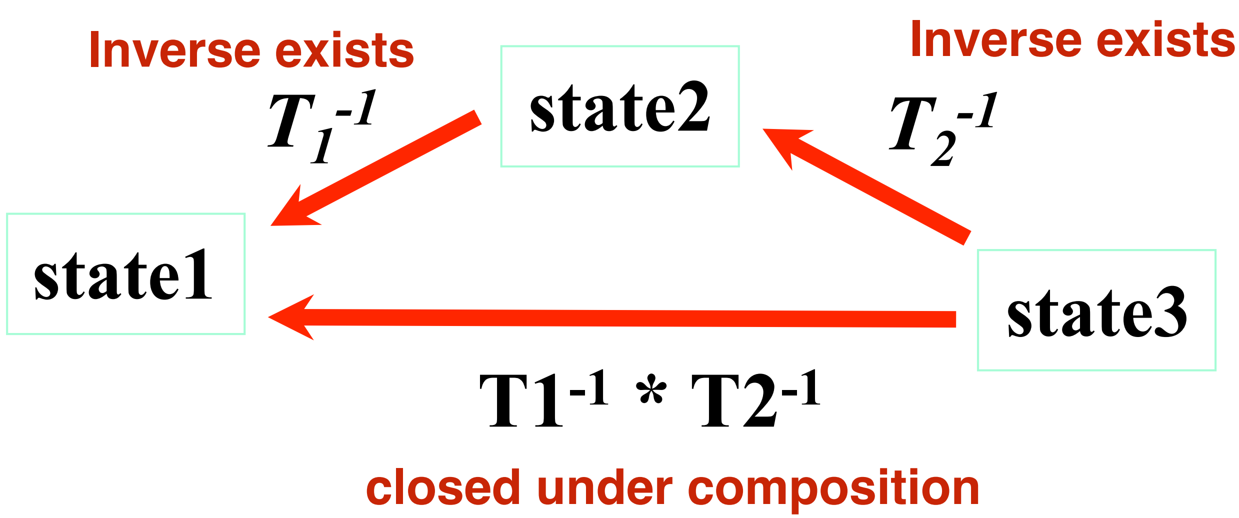


Transformation Groups

We have verified that translations form a group.

Why does it matter?

Groups are very well-behaved. When doing computations with groups we can freely compose transformations and assume inverses exist.



Hierarchy of Transformations

$\begin{bmatrix} A & b \\ c^T & d \end{bmatrix}$	**projective**
	⇩
$\begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix}$	**affine**
	⇩
$\begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix}$	**similarity**
	⇩
$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$	**Euclidean**

subgroup hierarchy

Composition in a Hierarchy

$\begin{bmatrix} A & b \\ c^T & d \end{bmatrix}$ **projective**

$\Downarrow$

$\begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix}$ **affine**

$\Downarrow$

$\begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix}$ **similarity**

$\Downarrow$

$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$ **Euclidean**

similarity * similarity = similarity

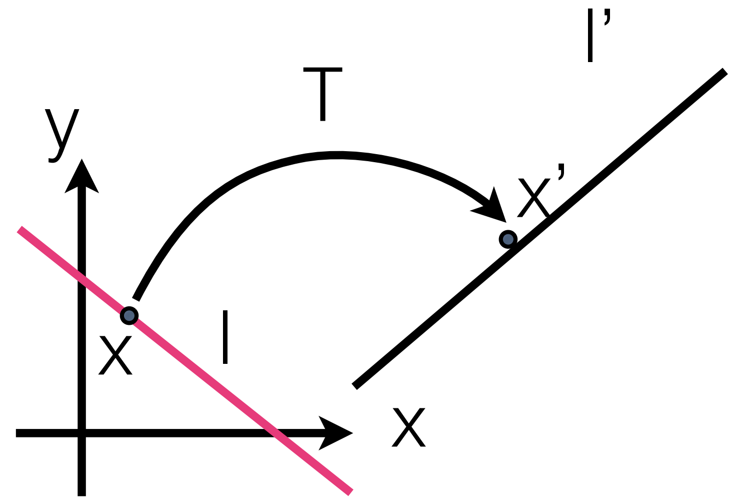
similarity * affine = affine

Euclidean * affine = affine

any * projective = projective

Co-Vectors

$$\mathbf{l} \cdot \mathbf{x} = 0 \qquad \mathbf{l}^T \mathbf{x} = 0$$



Transforming line equations:

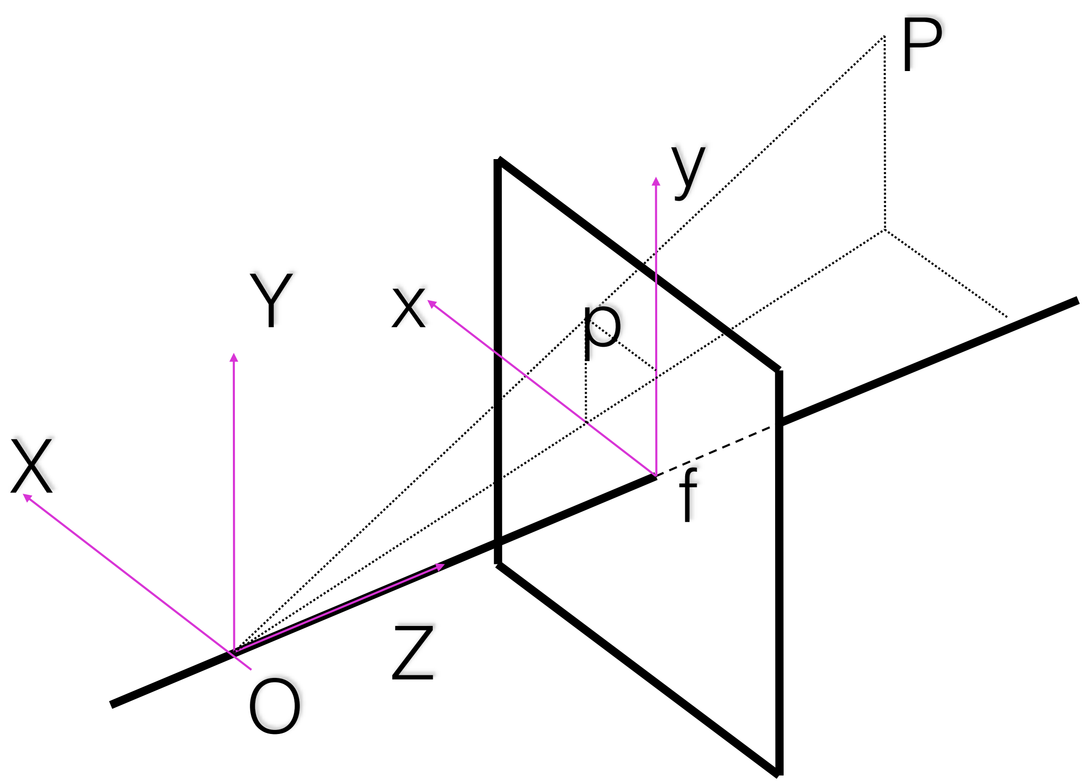
$$\mathbf{l}^T \mathbf{x} = 0 \qquad \boxed{\mathbf{l}^T \mathbf{T}^{-1} \mathbf{T} \mathbf{x}} = 0 \qquad \mathbf{l}'^T \mathbf{x}' = 0$$

$$\mathbf{x}' = \mathbf{T}\mathbf{x}$$

$$\mathbf{l}'^T = \mathbf{l}^T \mathbf{T}^{-1}$$

$$\mathbf{l}' = (\mathbf{l}^T \mathbf{T}^{-1})^T = \mathbf{T}^{-T} \mathbf{l}$$

Pinhole Camera Model
(Camera Coordinates)



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

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

- Non-linear equations
- Any point on the ray OP has image p !!

3D to 2D Perspective Matrix Equation

(Camera Coordinates)

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

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

Using homogeneous coordinates:

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

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

Perspective Matrix Equation

(Camera Coordinates)

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

$$p = M_{\text{int}} \times P$$

Pinhole Camera Properties

Non-linear equations

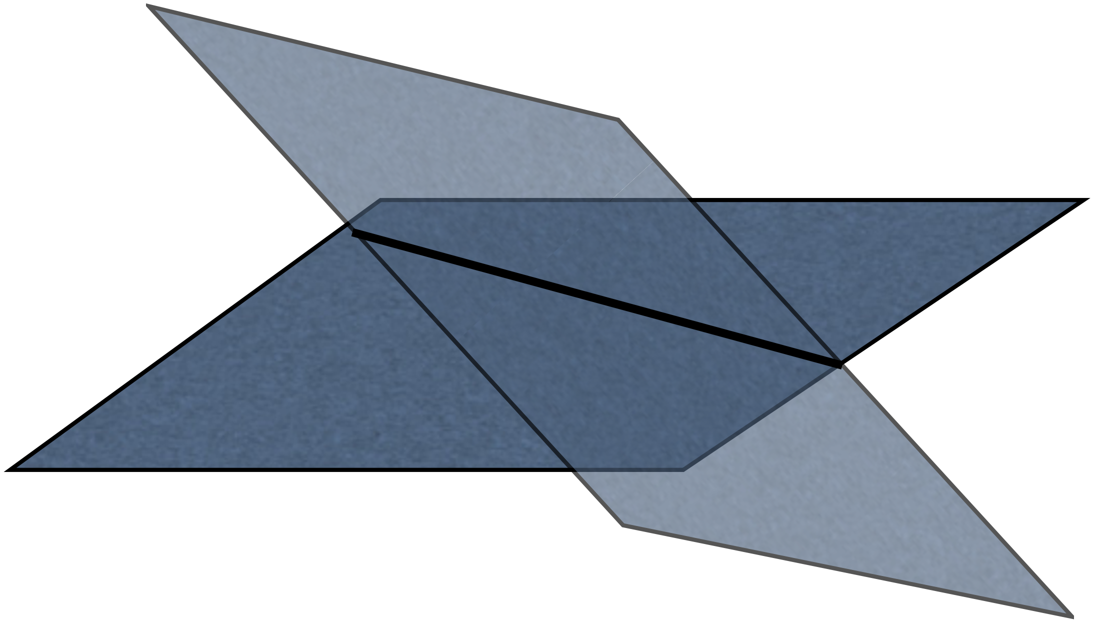
Lines project into lines

Does not preserve angles

Circles project into ellipses

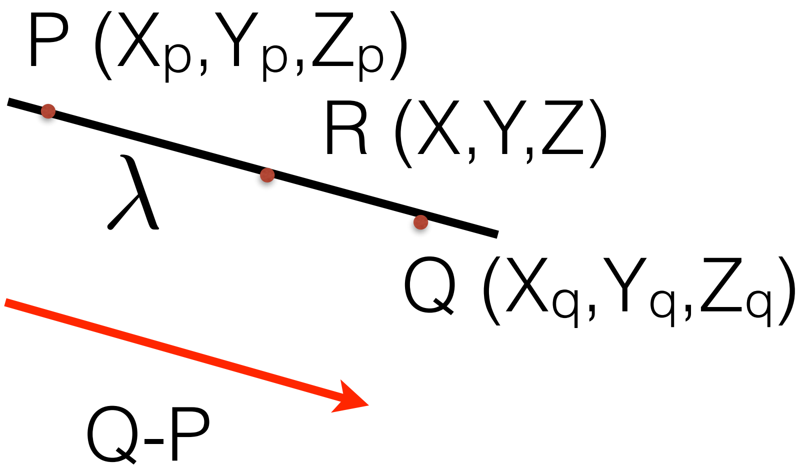
Farther objects appear smaller

3D Line



$$\begin{cases} aX + bY + cZ + d = 0 \\ a'X + b'Y + c'Z + d' = 0 \end{cases}$$

3D Line



$$R = P + \lambda(Q - P)$$

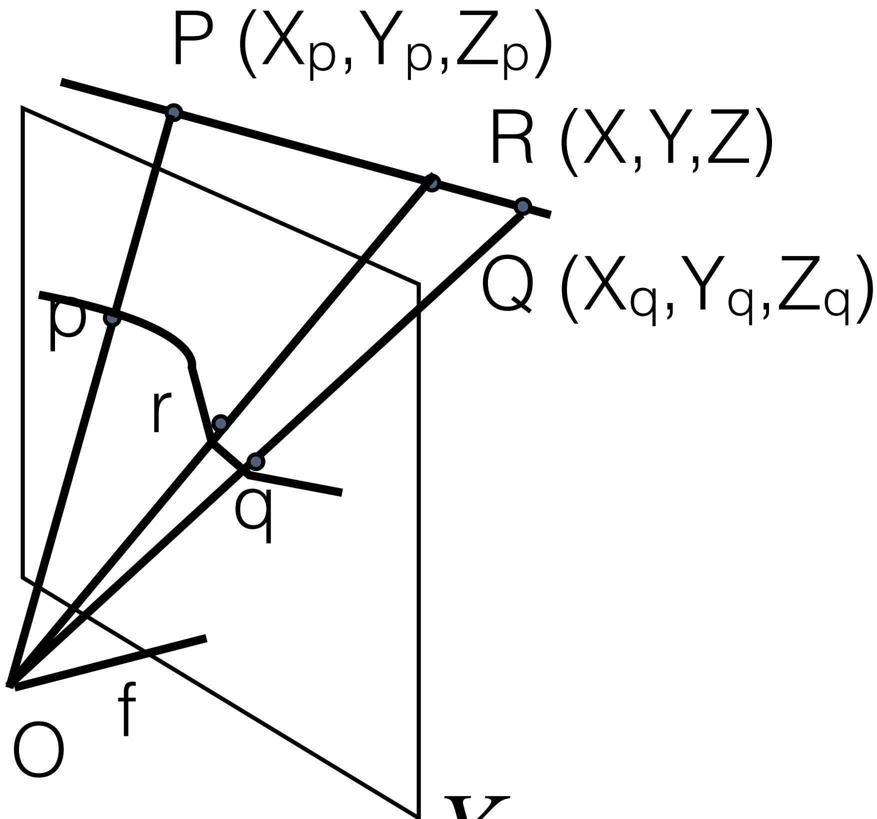
$$\begin{aligned} X &= X_p + \lambda(X_q - X_p) \\ Y &= Y_p + \lambda(Y_q - Y_p) \\ Z &= Z_p + \lambda(Z_q - Z_p) \end{aligned}$$

3D Line

The 2D perspective image of a 3D line is a line.

The 2D perspective images of 3D parallel lines that are not parallel to the image plane are not parallel and intersect at the image of the ideal point of the lines. This point is called the “vanishing point”.

Image of a 3D Line



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

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

$$R = P + \lambda(Q - P)$$

$$X = X_p + \lambda(X_q - X_p)$$

$$Y = Y_p + \lambda(Y_q - Y_p)$$

$$Z = Z_p + \lambda(Z_q - Z_p)$$

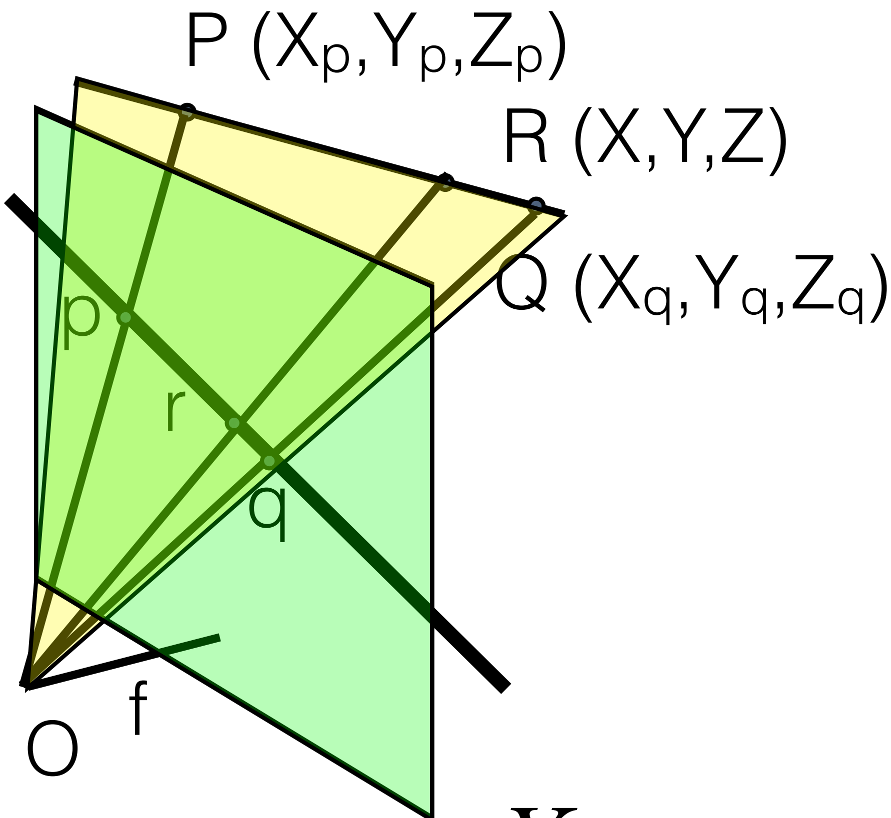
$$x = f\frac{X_p + \lambda(X_q - X_p)}{Z_p + \lambda(Z_q - Z_p)}$$

$$y = f\frac{Y_p + \lambda(Y_q - Y_p)}{Z_p + \lambda(Z_q - Z_p)}$$

$$z = f$$

Are these the equations of a 2D line?

Image of a 3D Line



$$R = P + \lambda(Q - P)$$

$$\begin{aligned} X &= X_p + \lambda(X_q - X_p) \\ Y &= Y_p + \lambda(Y_q - Y_p) \\ Z &= Z_p + \lambda(Z_q - Z_p) \end{aligned}$$

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

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

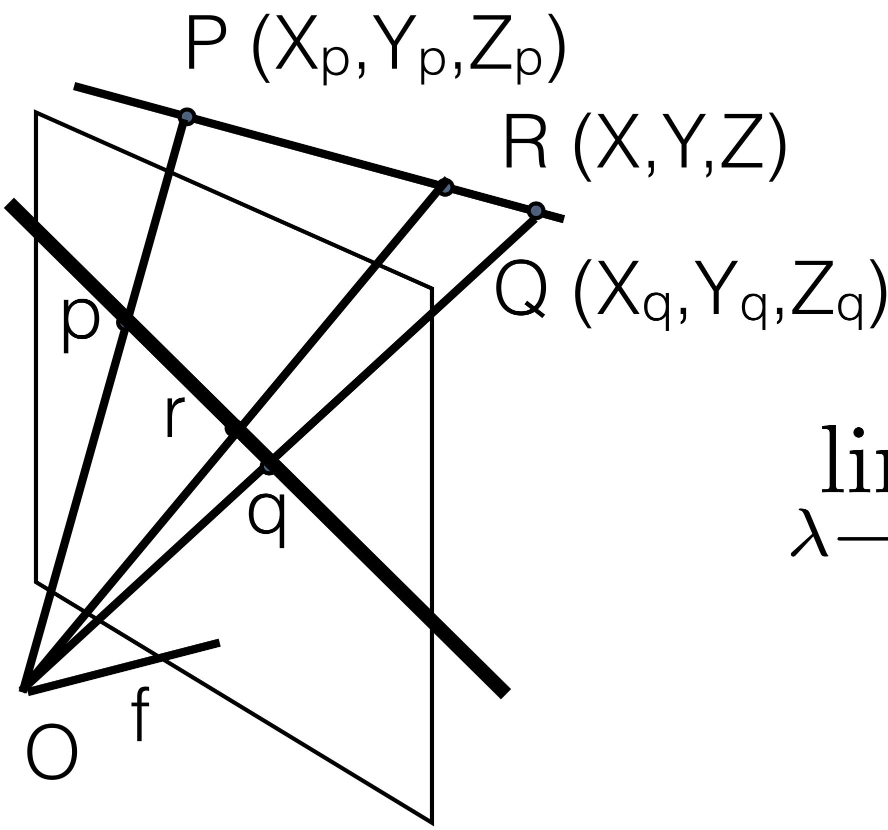
$$x = f\frac{X_p + \lambda(X_q - X_p)}{Z_p + \lambda(Z_q - Z_p)}$$

$$y = f\frac{Y_p + \lambda(Y_q - Y_p)}{Z_p + \lambda(Z_q - Z_p)}$$

$$z = f$$

The image lies in the intersection of two planes, hence it is a line.

Image of an IDEAL point



$$R = P + \lambda(Q - P)$$

$$\lim_{\lambda \to \infty} \boxed{\begin{aligned} x &= f\frac{X_p + \lambda(X_q - X_p)}{Z_p + \lambda(Z_q - Z_p)} \\ y &= f\frac{Y_p + \lambda(Y_q - Y_p)}{Z_p + \lambda(Z_q - Z_p)} \\ z &= f \end{aligned}}$$

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

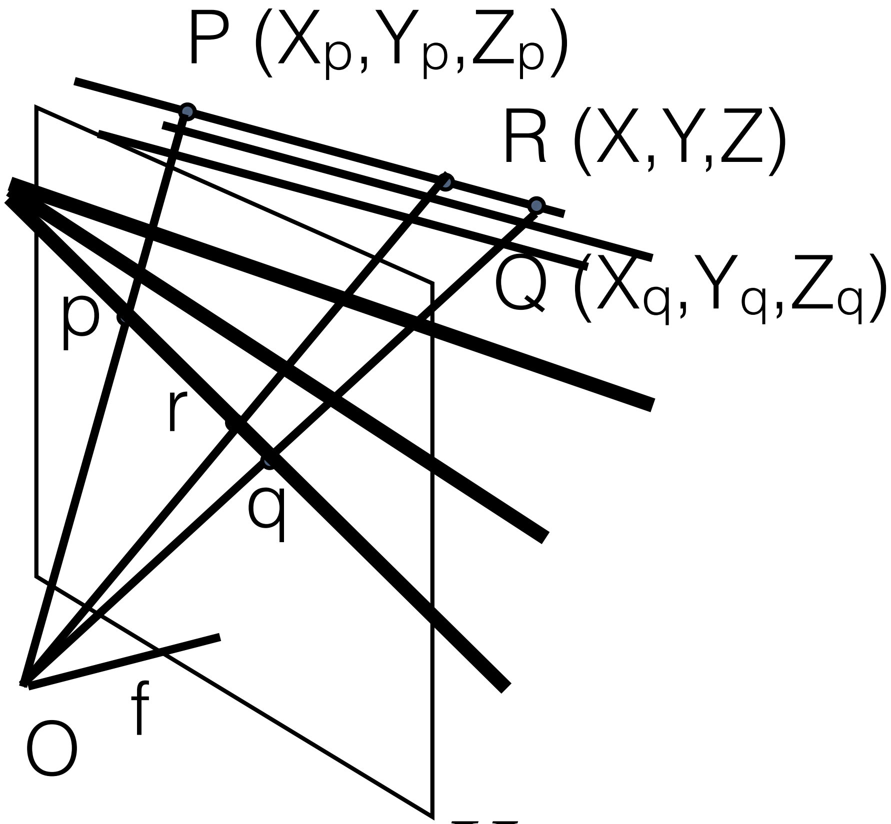
$$y = f\frac{Y}{Z}$$

$$\lim_{\lambda \to \infty} x \to f\frac{X_q - X_p}{Z_q - Z_p}$$

$$\lim_{\lambda \to \infty} y \to f\frac{Y_q - Y_p}{Z_q - Z_p}$$

$$\lim_{\lambda \to \infty} z \to f \text{ assuming that } Z_q \neq Z_p$$

Image of an IDEAL point



$$R = P + \lambda(Q - P)$$

$$\lim_{\lambda\to\infty} x \to f\frac{X_q - X_p}{Z_q - Z_p}$$

$$\lim_{\lambda\to\infty} y \to f\frac{Y_q - Y_p}{Z_q - Z_p}$$

$$\lim_{\lambda\to\infty} z \to f$$

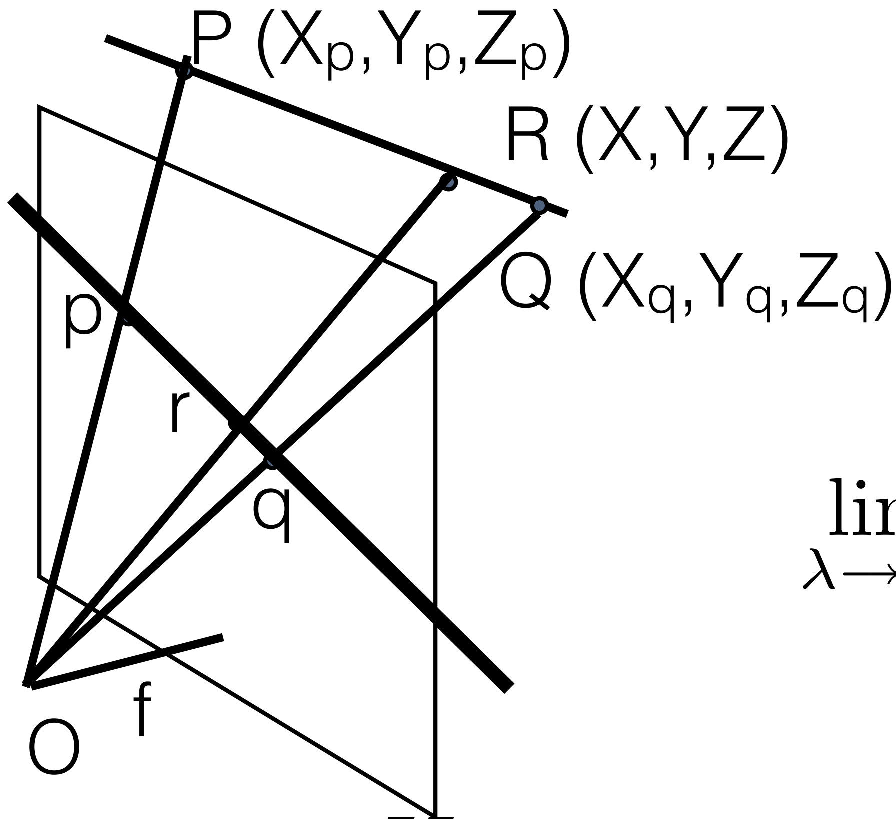
assuming that $Z_q \neq Z_p$

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

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

The image of the ideal point is not ideal and depends only on the direction of the line (Q-P)

Image of an IDEAL point



$$R = P + \lambda(Q - P)$$

$$\lim_{\lambda \to \infty} \boxed{\begin{aligned} x &= f\frac{X_p + \lambda(X_q - X_p)}{Z_p + \lambda(Z_q - Z_p)} \\ y &= f\frac{Y_p + \lambda(Y_q - Y_p)}{Z_p + \lambda(Z_q - Z_p)} \\ z &= f \end{aligned}}$$

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

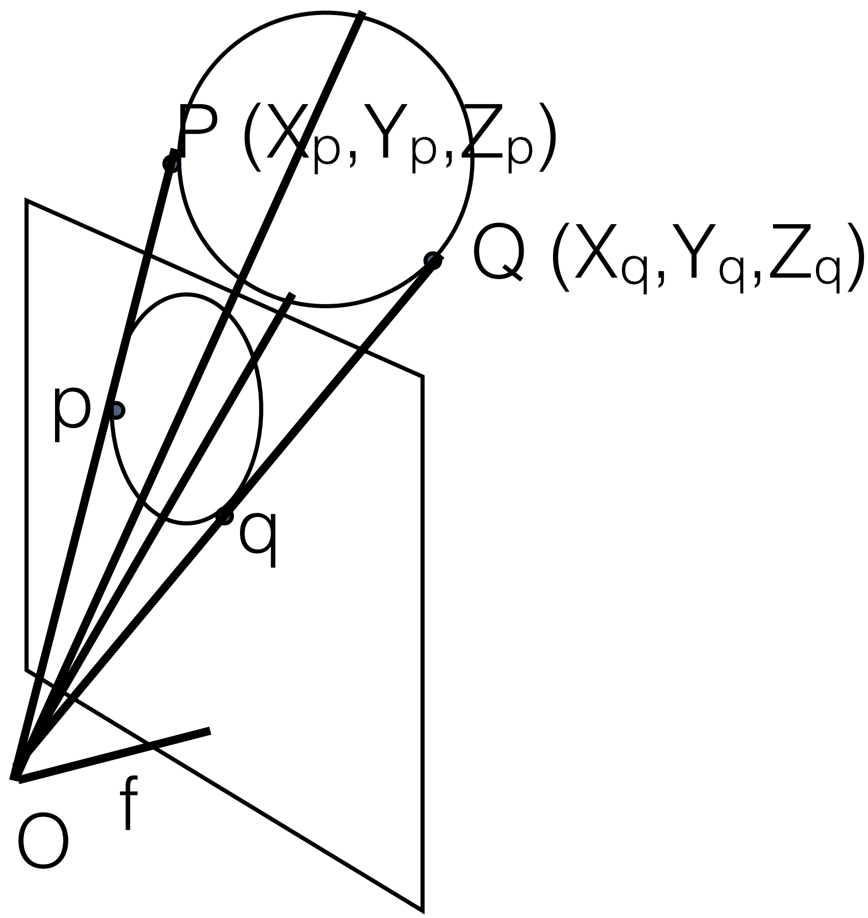
$$y = f\frac{Y}{Z}$$

What happens when $Z_q = Z_p$

$$\lim_{\lambda \to \infty} x \to \infty \quad \lim_{\lambda \to \infty} y \to \infty \quad \lim_{\lambda \to \infty} z \to f$$

The image of the IDEAL point of a line parallel to the image plane is an IDEAL point.

Image of a Circle

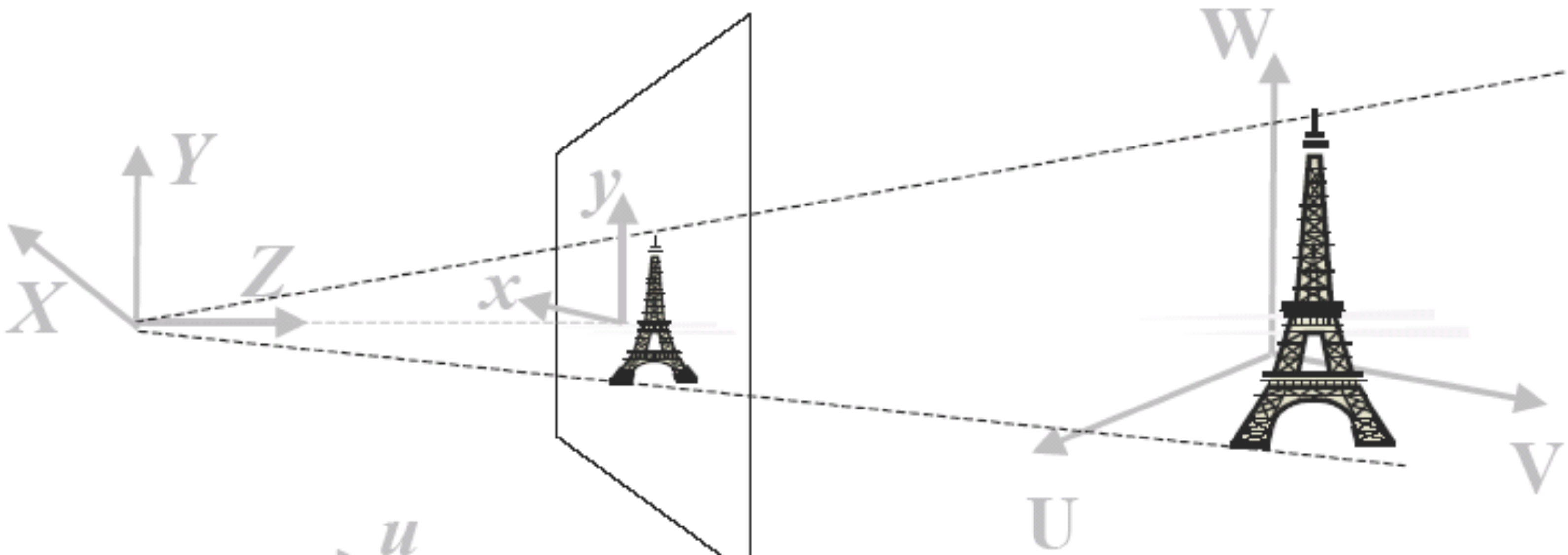


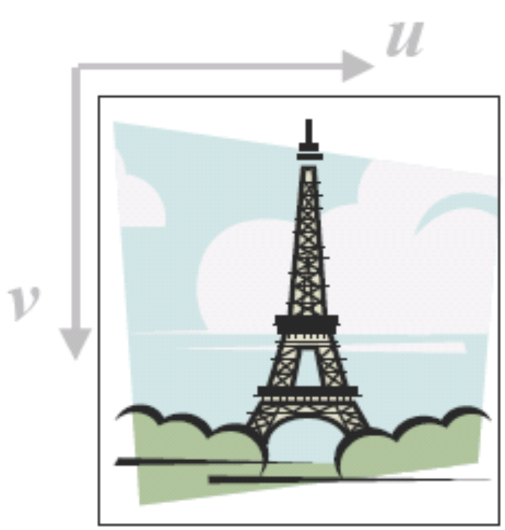
$$x = f\frac{X}{Z}$$

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

The image of a circle is the intersection of a cone and the image plane and it is in general an ellipse.

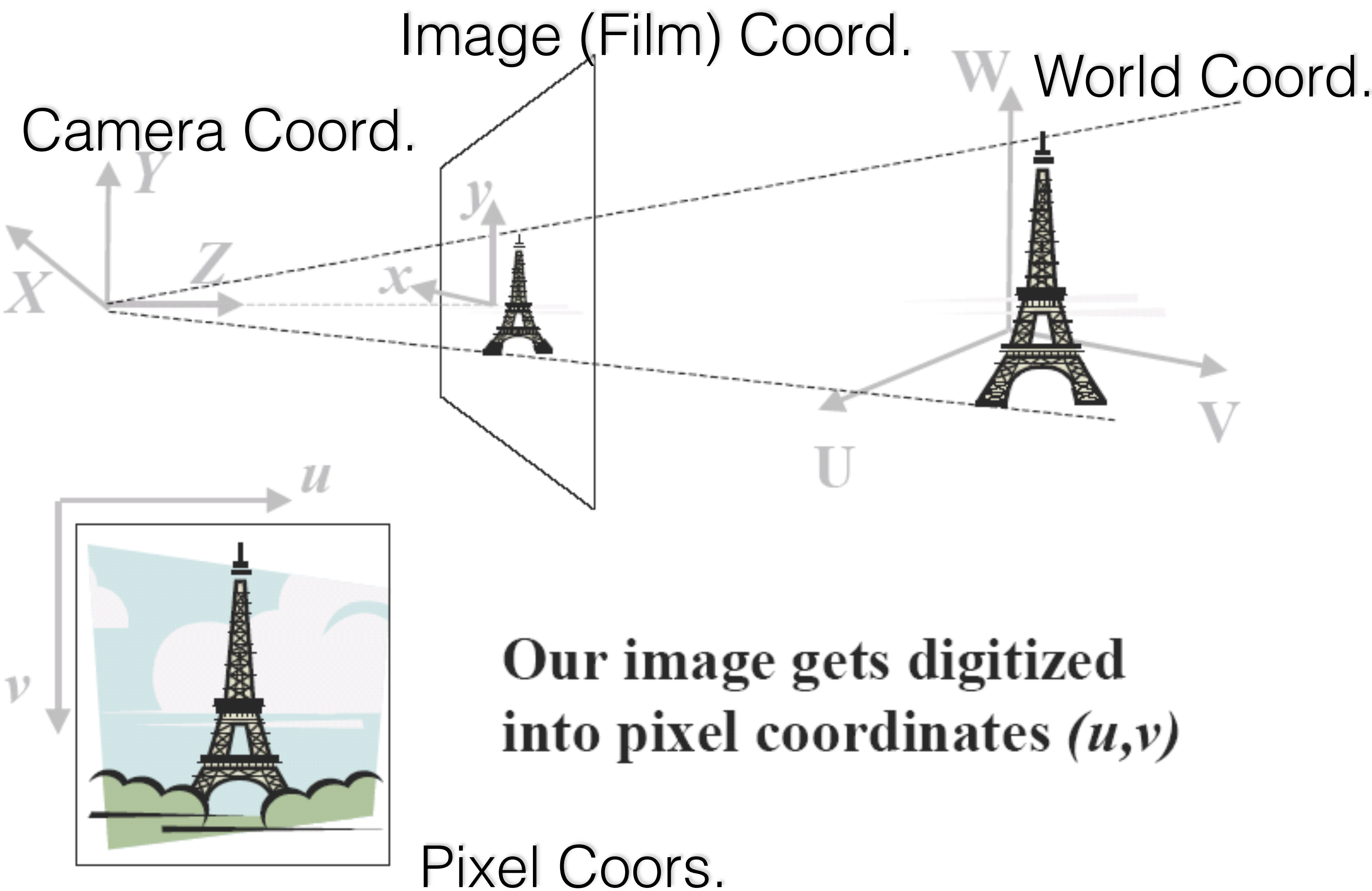
Imaging Geometry





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

Imaging Geometry



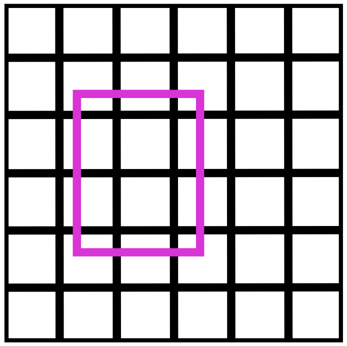
Our image gets digitized into pixel coordinates *(u,v)*

More intrinsic parameters:

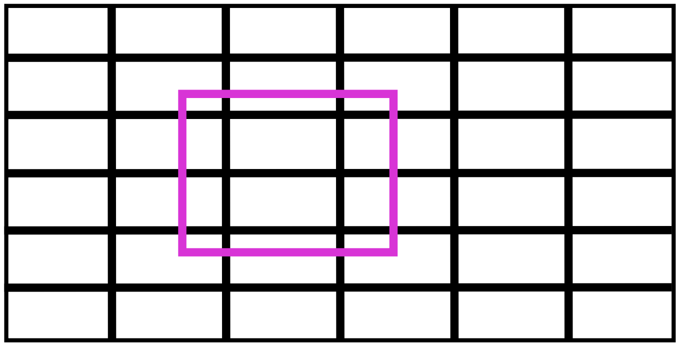
The CCD sensor is made of a rectangular grid nxm of photosensors.

Each photosensor generates an analog signal that is digitized by a frame grabber into an array of NxM pixels.

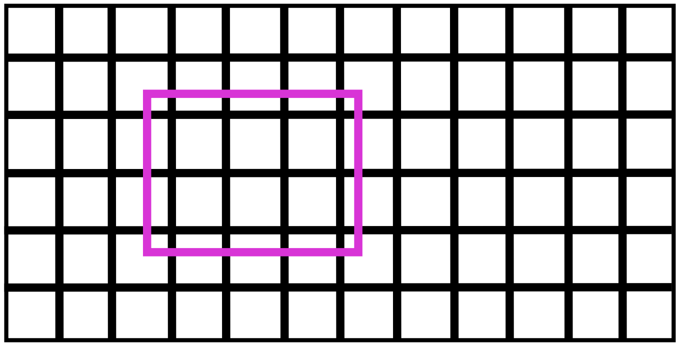
Intrinsic Parameters



NxN pixels Imaged Grid



nxn CCD elements
n:m aspect ratio



mxn CCD elements
n:n aspect ratio

Effective Sizes: s_x and s_y

In practice, we will assume that there is a
1-1 correspondence between CCD elements and pixels.

$$x = \quad f\frac{X}{Z} = (x_{im} - c_x)s_x$$

$$y = \quad f\frac{Y}{Z} = (y_{im} - c_y)s_y$$

Where c_x and c_y are the coordinates of the image center

A more complete Mint

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f/s_x & 0 & c_x & 0 \\ 0 & f/s_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$p = M_{\text{int}} \times P$$

PROBLEM: In general, the camera coordinate system is not aligned with the world coordinate system!

SOLUTION: Find a transformation between coordinate systems.

Coordinate Systems

World Coords.		Camera Coords.		Film Coords.		Image Coords.
X_w		x		x		u
Y_w	$\Rightarrow$	y	$\Rightarrow$	y	$\Rightarrow$	v
Z_w		z				

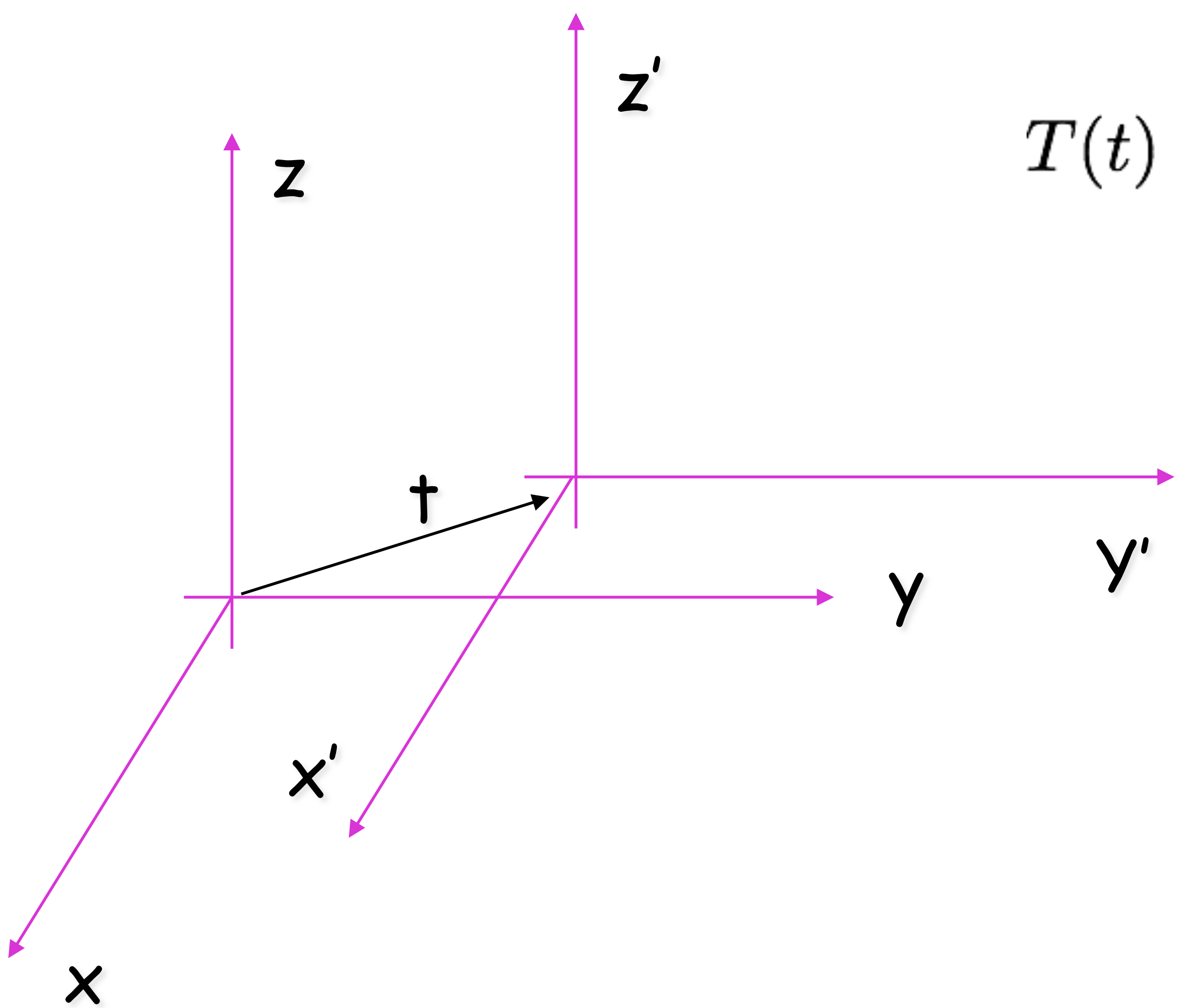
Coordinate Systems

World Coords.		Camera Coords.		Film Coords.		Image Coords.
X_w, Y_w, Z_w	→	x, y, z	→	x, y	→	u, v

Rigid transformation: rotation & translation

3D Translation of Coordinate Systems

Translate by a vector $t=(t_x,t_y,t_z)'$

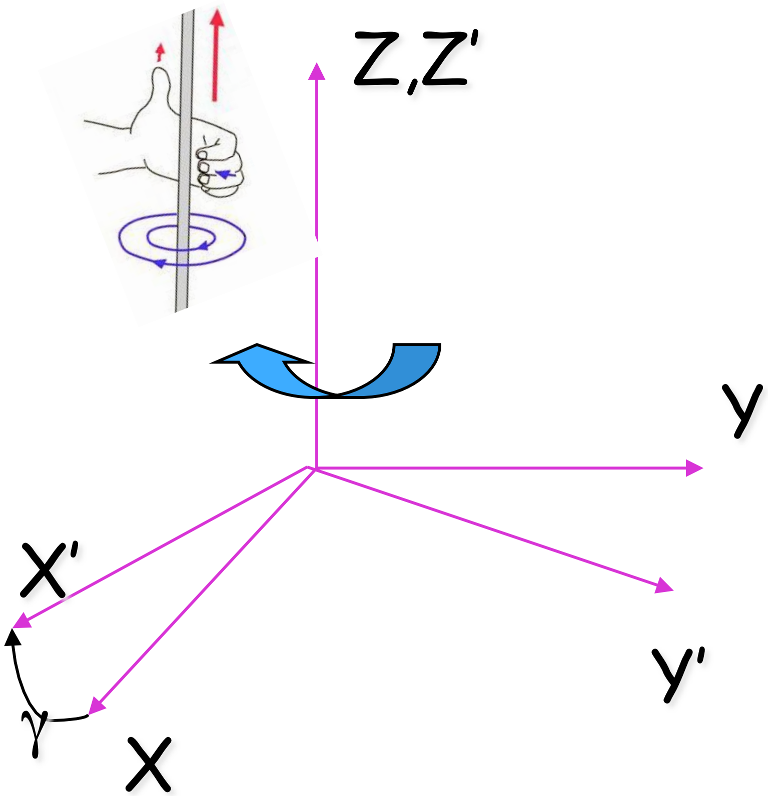


$$T(t) = \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = T * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

3D Rotation of Coordinate Systems

CLOCKWISE Rotation around the coordinate axes (left hand):



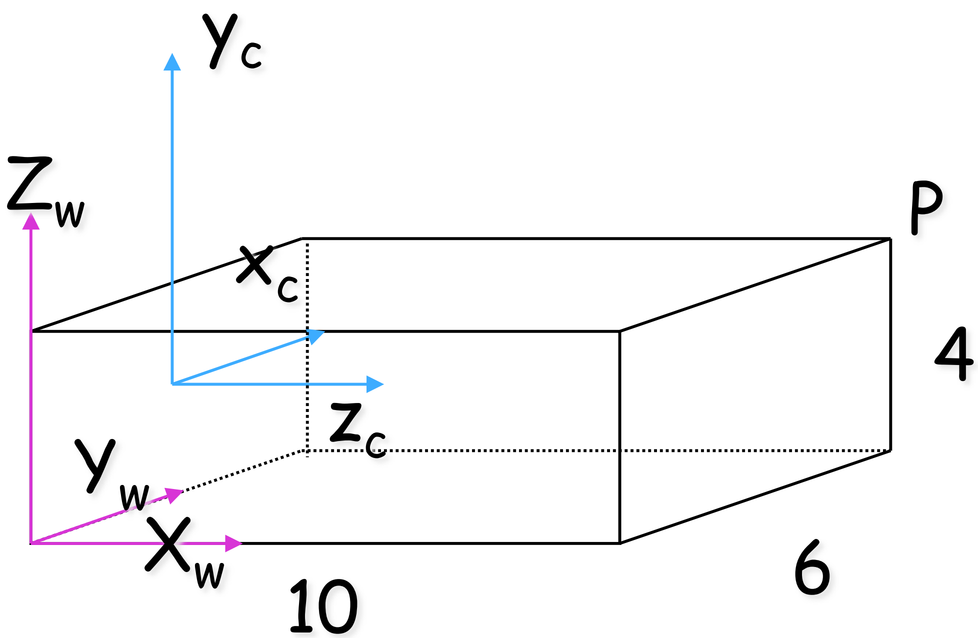
$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

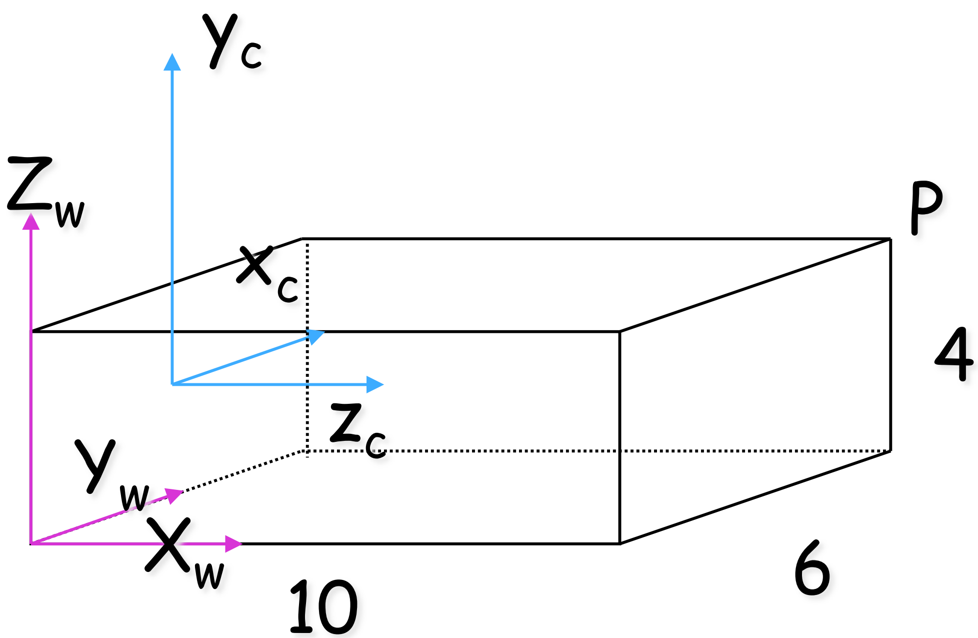
Example



$$P_w = \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix} \quad P_c = \begin{bmatrix} 3 \\ 2 \\ 10 \\ 1 \end{bmatrix}$$

$$P_c = M_{\text{ext}} . P_w$$

Example

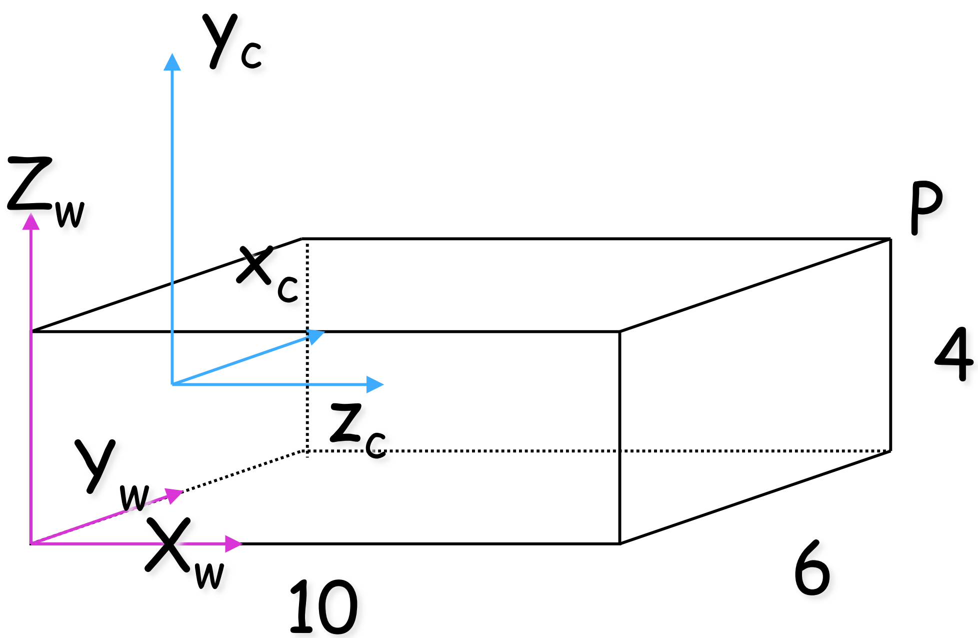


$$P_w = \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix} \quad P_c = \begin{bmatrix} 3 \\ 2 \\ 10 \\ 1 \end{bmatrix}$$

$$P_c = M_{\text{ext}} \cdot P_w$$

$$\begin{aligned} x_c &= y_w - 3 \\ y_c &= z_w - 2 \\ z_c &= x_w \end{aligned}$$

Example



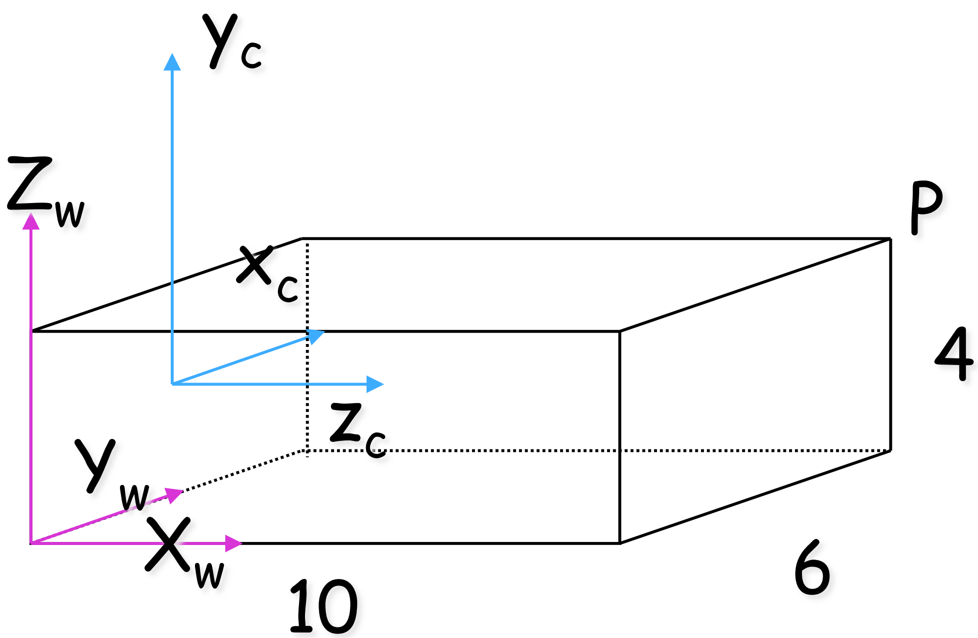
$$P_w = \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix} \qquad P_c = \begin{bmatrix} 3 \\ 2 \\ 10 \\ 1 \end{bmatrix}$$

$$P_c = M_{\text{ext}} \cdot P_w$$

$$\begin{aligned} x_c &= y_w - 3 \\ y_c &= z_w - 2 \\ z_c &= x_w \end{aligned}$$

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

Example



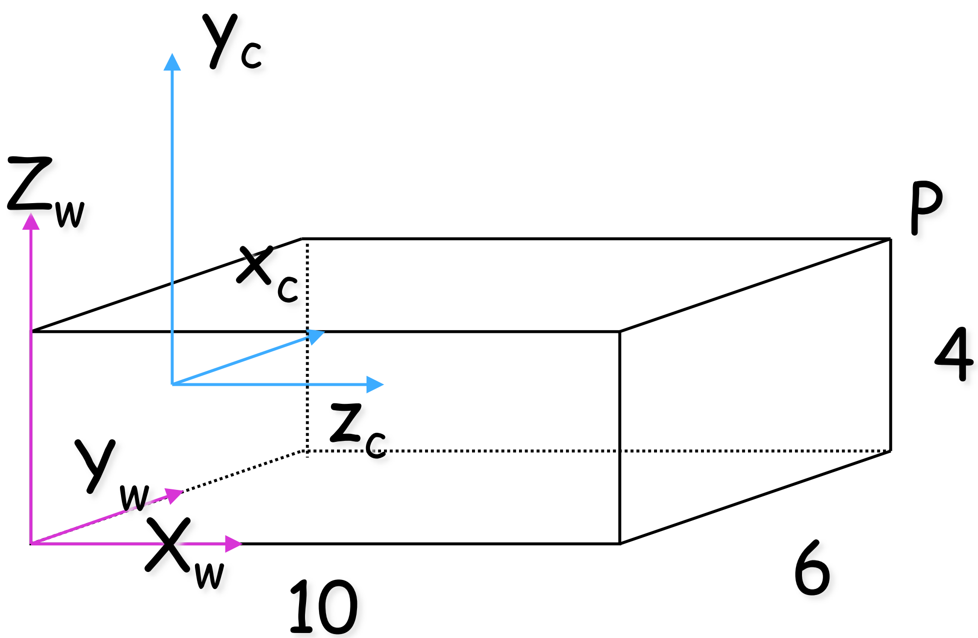
$$P_w = \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix} \qquad P_c = \begin{bmatrix} 3 \\ 2 \\ 10 \\ 1 \end{bmatrix}$$

$$P_c = M_{\text{ext}} \cdot P_w$$

$$\begin{aligned} x_c &= y_w - 3 \\ y_c &= z_w - 2 \\ z_c &= x_w \end{aligned}$$

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{array}{c} \begin{array}{cc} \mathbf{R} & \qquad \mathbf{T} \end{array} \\ \begin{bmatrix} \boxed{\begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{matrix}} & \boxed{\begin{matrix} -3 \\ -2 \\ 0 \end{matrix}} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{bmatrix} \end{array} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

Example



$$P_w = \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix} \qquad P_c = \begin{bmatrix} 3 \\ 2 \\ 10 \\ 1 \end{bmatrix}$$

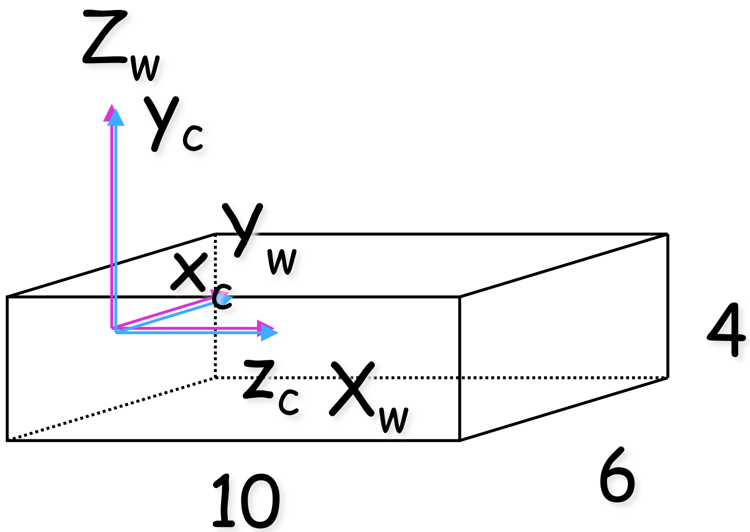
$$P_c = M_{\text{ext}} \cdot P_w$$

First, translate W to C

$t = (0, 3, 2)'$

Expressed in the ***current*** *coordinate system!*

Example



$$P_w = \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix} \qquad P_c = \begin{bmatrix} 3 \\ 2 \\ 10 \\ 1 \end{bmatrix}$$

$$P_c = M_{\text{ext}} \cdot P_w$$

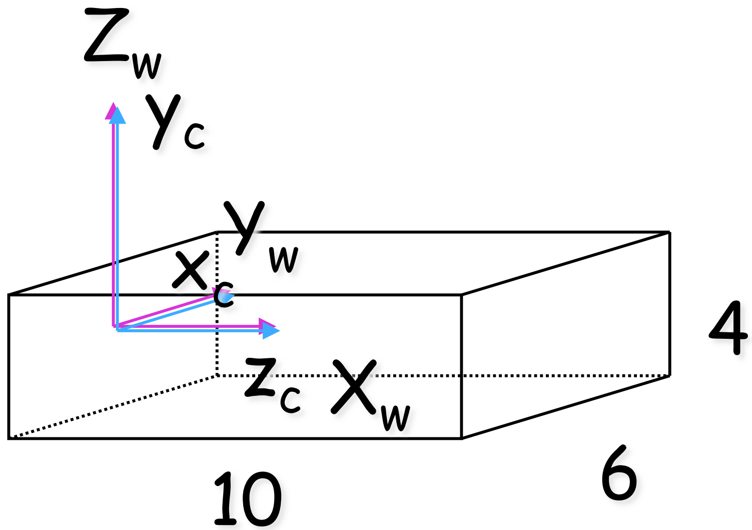
First, translate W to C

$$t = (0, 3, 2)'$$

Expressed in the ***current*** *coordinate system!*

$$\begin{array}{c} \mathbf{T} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

Example

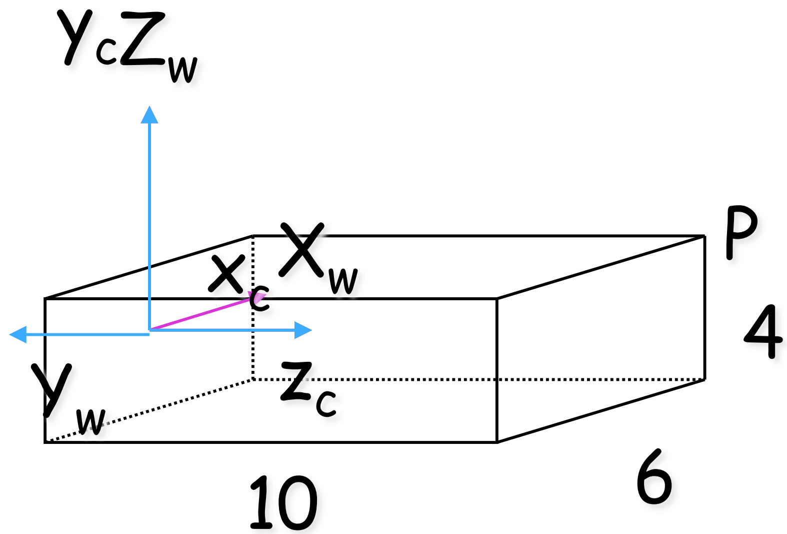


$$P_w = \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix} \quad P_c = \begin{bmatrix} 3 \\ 2 \\ 10 \\ 1 \end{bmatrix}$$

$$P_c = M_{\text{ext}} \cdot P_w$$

Next, rotate W' around Z_w, 90^o CCW (-90^o, CW)

Example



$$P_w = \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix} \quad P_c = \begin{bmatrix} 3 \\ 2 \\ 10 \\ 1 \end{bmatrix}$$

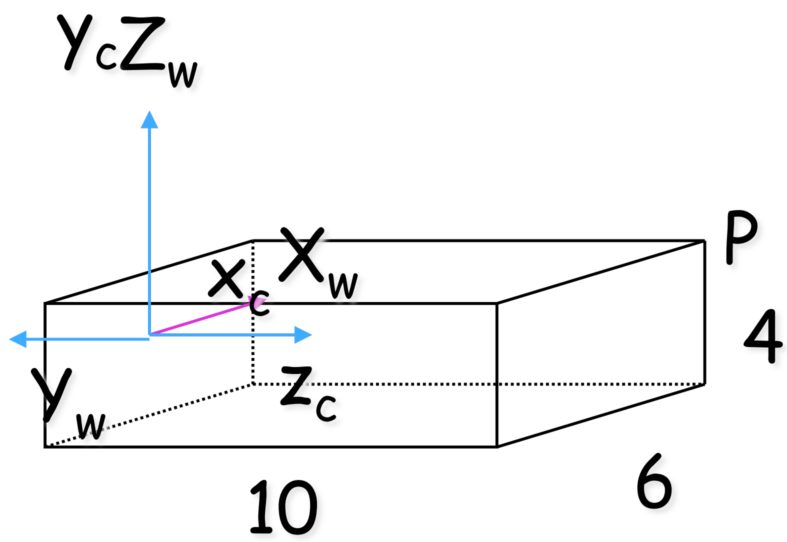
$$P_c = M_{\text{ext}} \cdot P_w$$

Next, rotate W' around Z_w, 90° CCW (-90°, CW)

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{R_z}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

Example

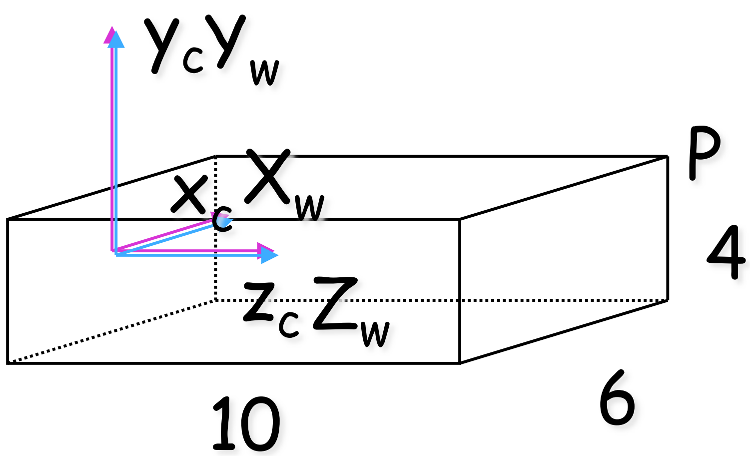


$$P_w = \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix} \quad P_c = \begin{bmatrix} 3 \\ 2 \\ 10 \\ 1 \end{bmatrix}$$

$$P_c = M_{\text{ext}} \cdot P_w$$

Next, rotate W'' around X_w, 90^o CCW (-90^o, CW)

Example



$$P_w = \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix} \quad P_c = \begin{bmatrix} 3 \\ 2 \\ 10 \\ 1 \end{bmatrix}$$

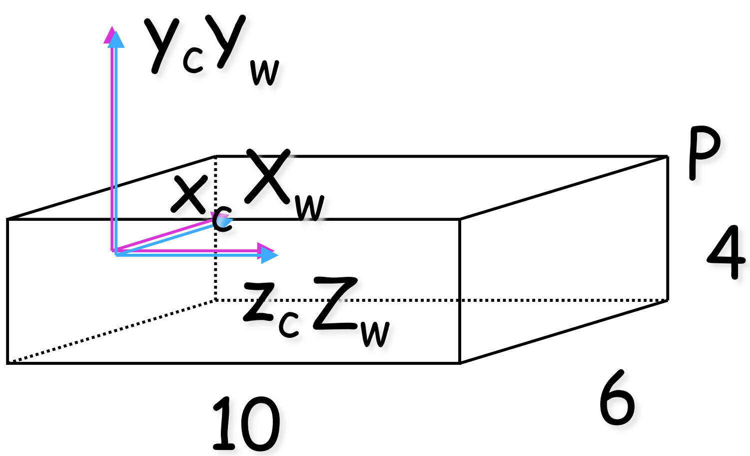
$$P_c = M_{\text{ext}} \cdot P_w$$

Next, rotate W'' around X_w, 90° CCW (-90°, CW)

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$\begin{array}{cccc} \mathbf{R_x} & \mathbf{R_z} & \mathbf{T} & \\ \left[\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\right. & \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \left.\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}\right] & \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \end{array}$$

Example

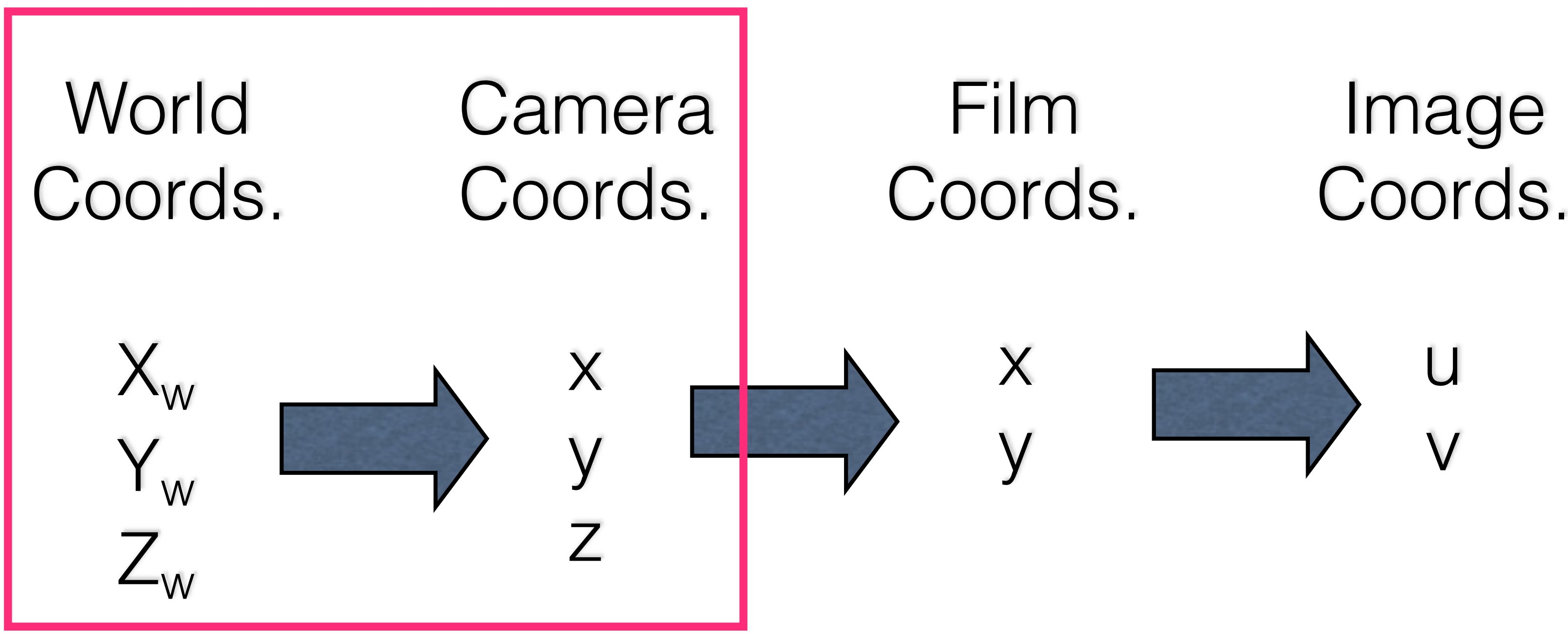


$$P_w = \begin{bmatrix} 10 \\ 6 \\ 4 \\ 1 \end{bmatrix} \quad P_c = \begin{bmatrix} 3 \\ 2 \\ 10 \\ 1 \end{bmatrix}$$

$$P_c = M_{\text{ext}} \cdot P_w$$

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{array}{c} \begin{array}{cc} \mathbf{R} & \mathbf{T} \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

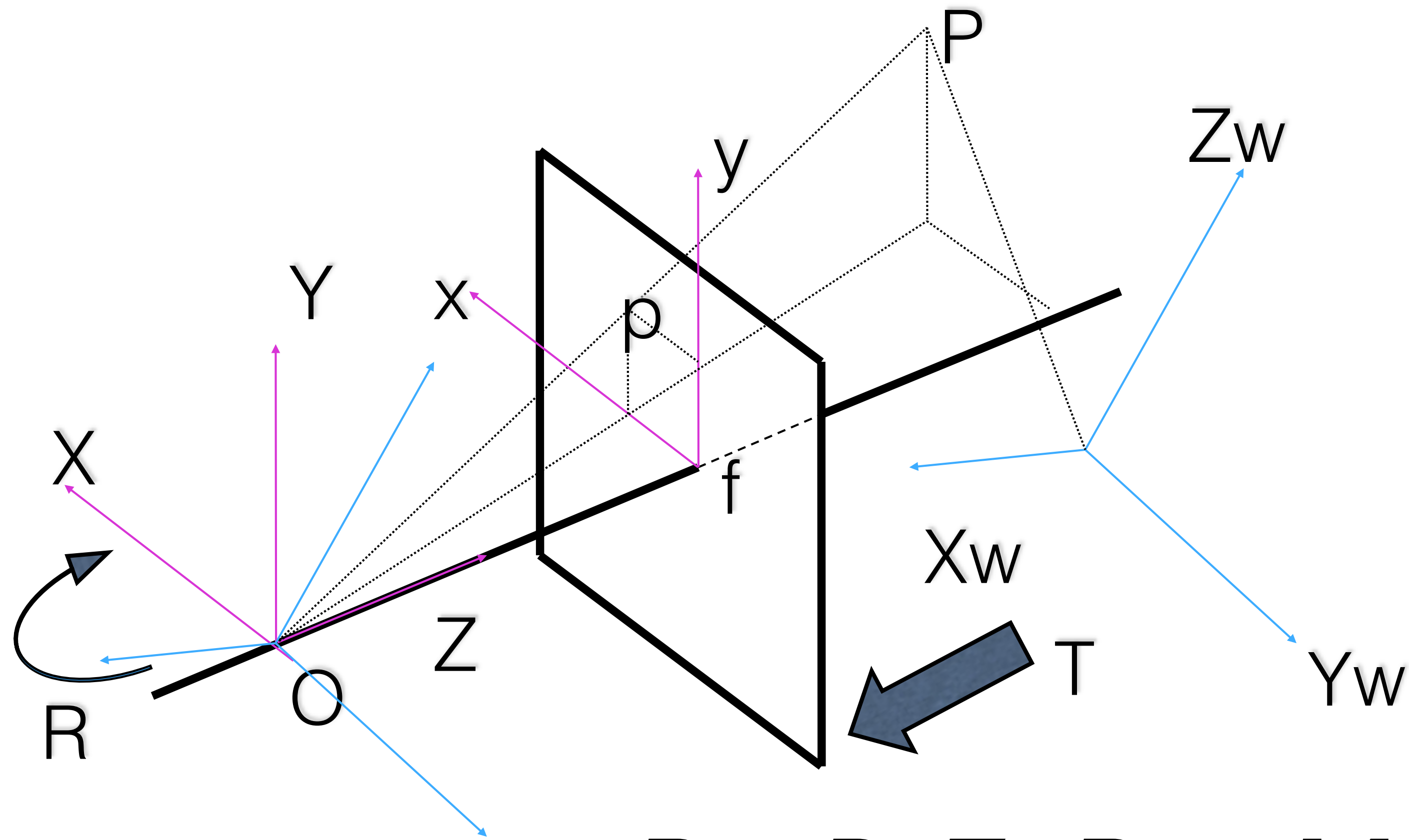
Coordinate Systems



Rigid transformation: rotation & translation

Pinhole Camera Model

(World Coordinates)



$$P = R \times T \times P_w = M_{ext} \times P_w$$

$$p = M_{int} P = M_{int} M_{ext} \times P_w$$

Putting it all together:

- Extrinsic parameters (R, T):

$$P = R \times T \times P_w = M_{\text{ext}} \times P_w$$

- Intrinsic parameter (f):

$$p = M_{\text{int}} P = M_{\text{int}} M_{\text{ext}} \times P_w$$

$$p = M \times P_w$$

M is 3x4
M has 6 dof
(assuming f is known)

How do we find M?

Each image point (x, y) must satisfy:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{31} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$x = x'/z' \qquad y = y'/z'$$

$$xz' = m_{11}X + m_{12}Y + m_{13}Z + m_{14}$$

$$yz' = m_{21}X + m_{22}Y + m_{23}Z + m_{24}$$

$$z' = m_{31}X + m_{32}Y + m_{33}Z + m_{34}$$

$$0 = m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}Xx - m_{32}Yx - m_{33}Zx - m_{34}x$$

$$0 = m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}Xy - m_{32}Yy - m_{33}Zy - m_{34}y$$

Finding M:

M has 12 entries, but only 6 dof.

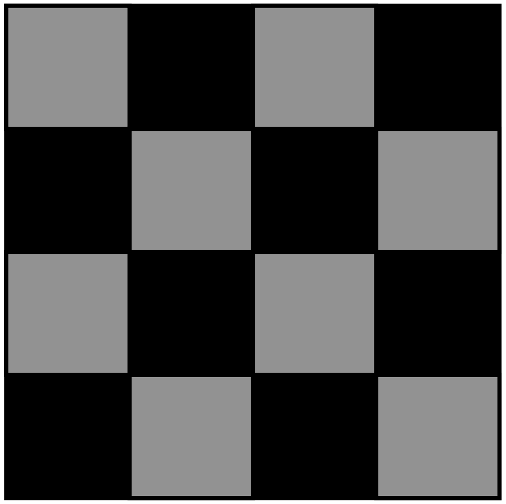
Each image point provides 2 equations.

Solve a system of linear equations.

$$0 = m_{11}X + m_{12}Y + m_{13}Z + m_{14} - m_{31}Xx - m_{32}Yx - m_{33}Zx - m_{34}x$$
$$0 = m_{21}X + m_{22}Y + m_{23}Z + m_{24} - m_{31}Xy - m_{32}Yy - m_{33}Zy - m_{34}y$$

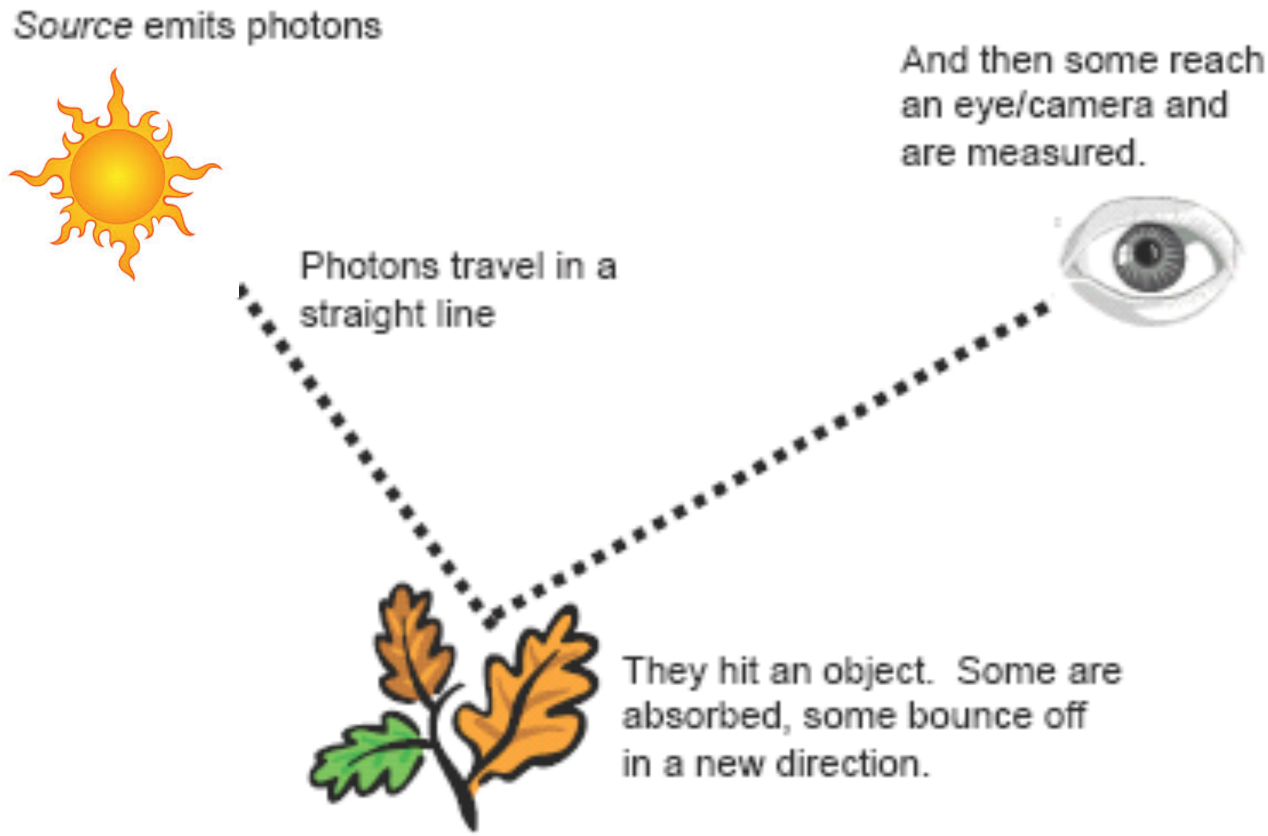
How do we find point correspondences?

Use special calibrating pattern:

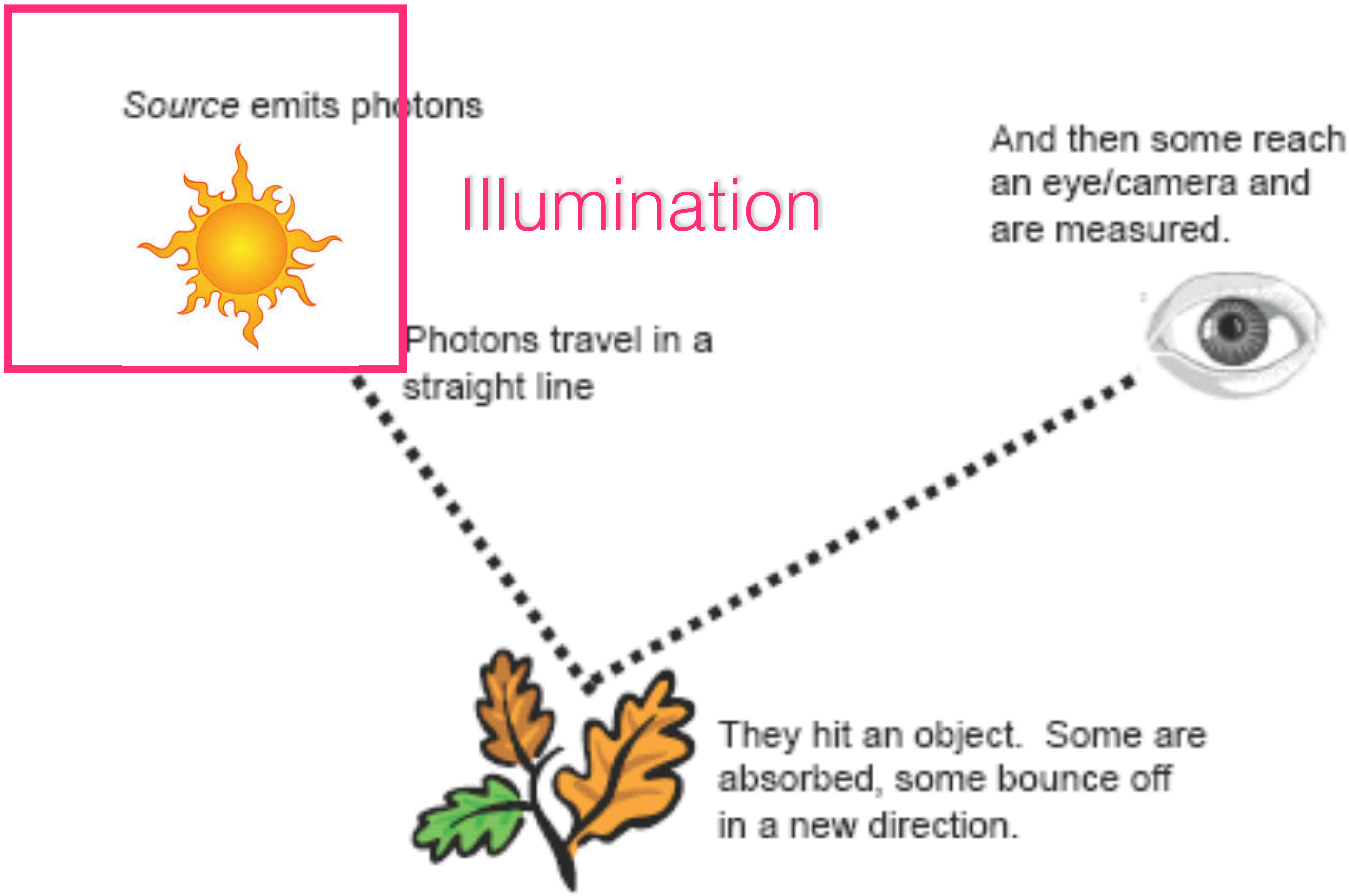


Corners are “easy” to detect and “identify”.

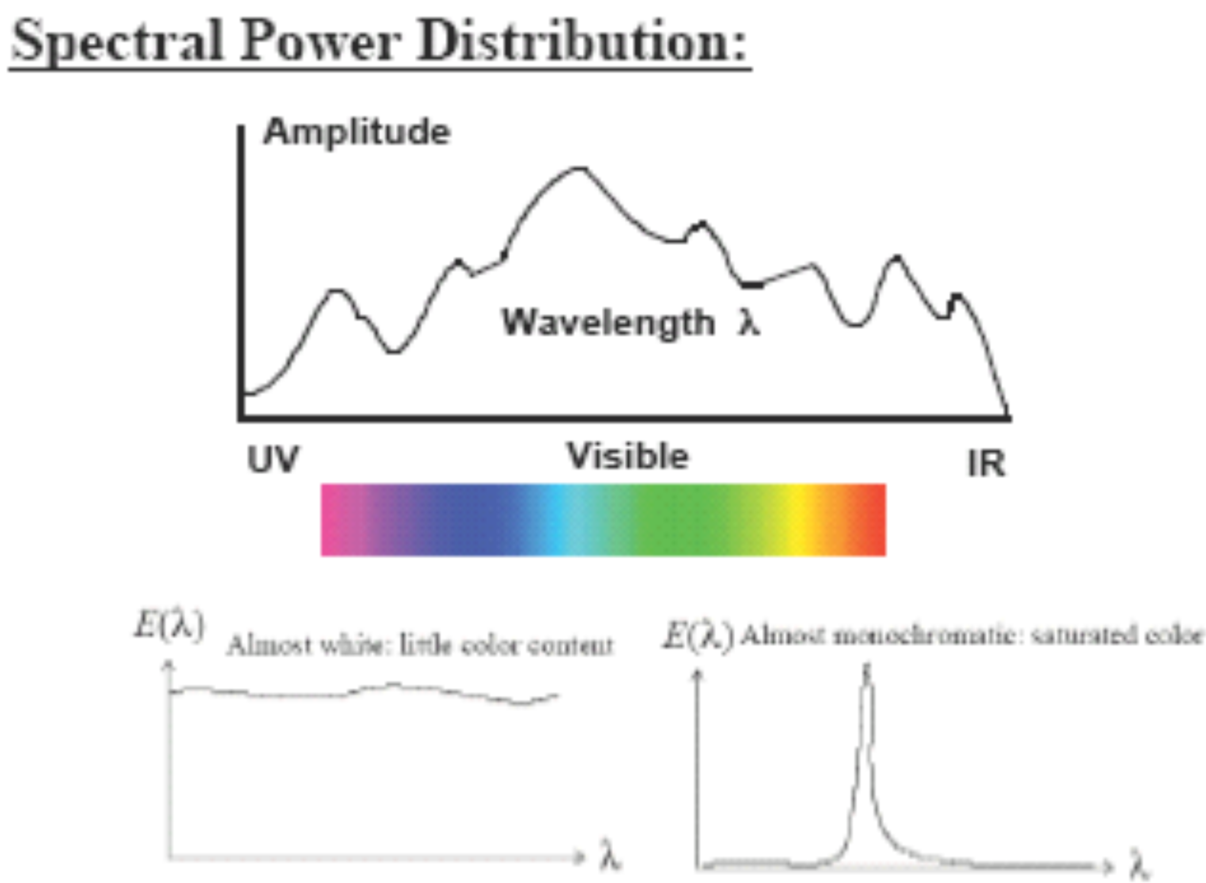
Photometry Overview



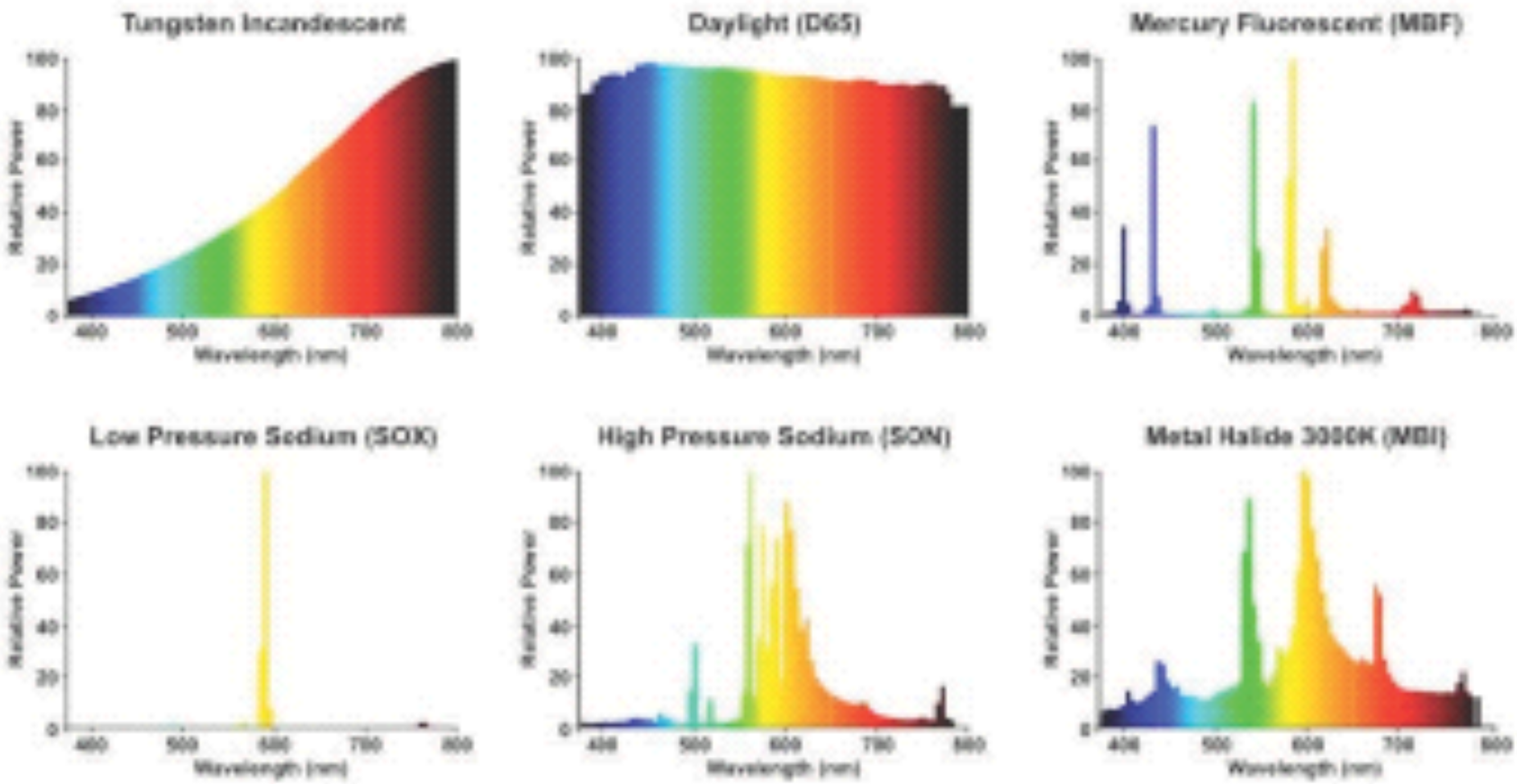
Light Transport



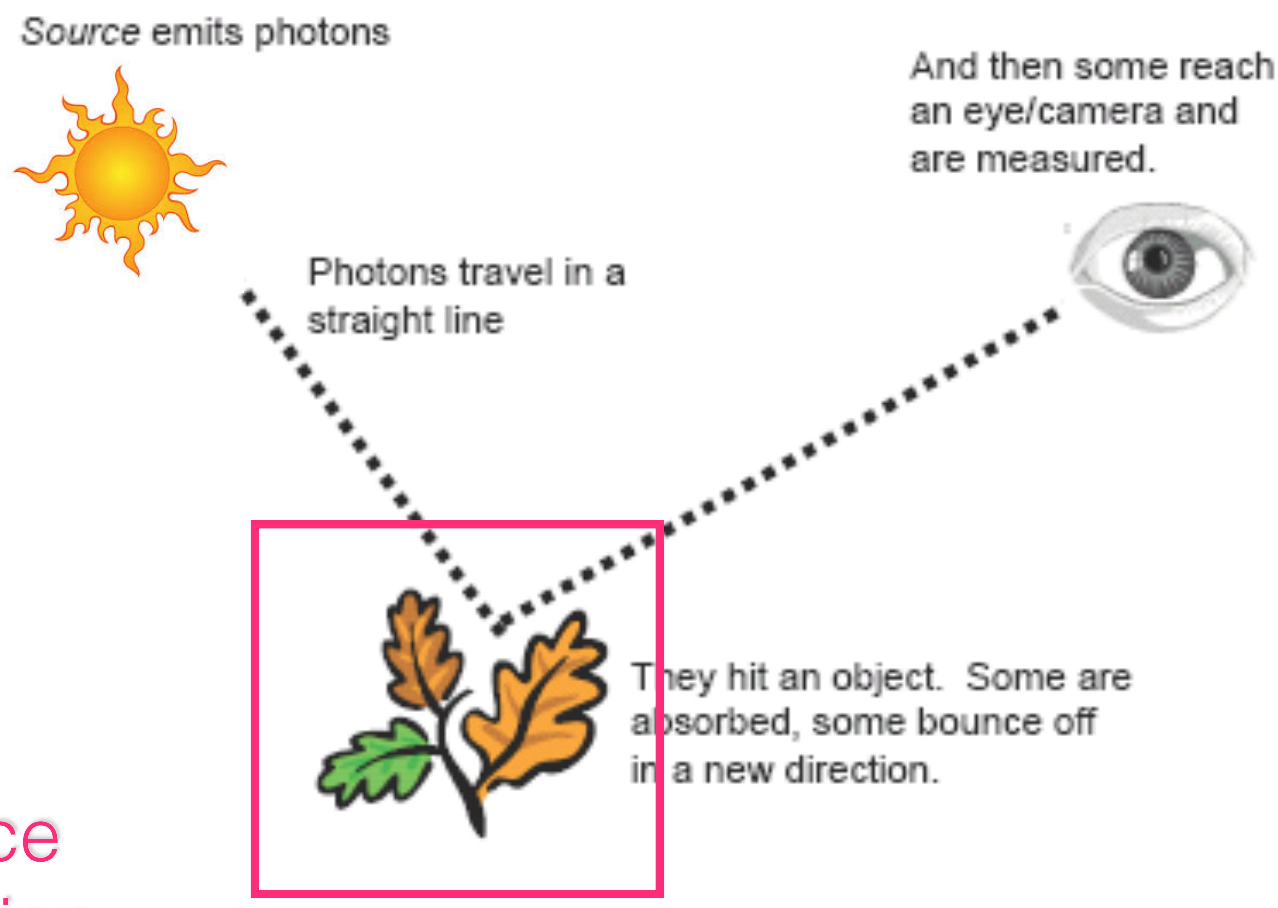
Color of Light Source



Some Light Source SPDs

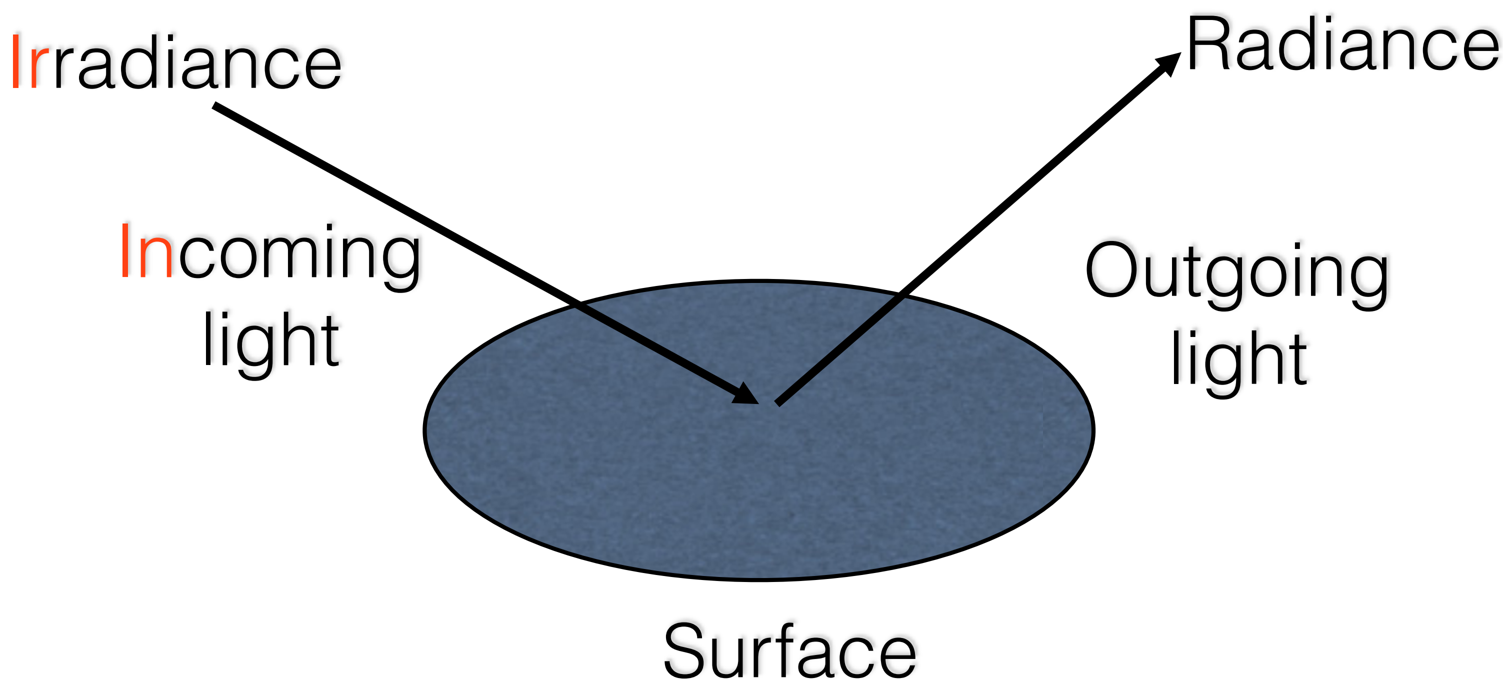


Light Transport



Surface Reflection

(Ir)radiance

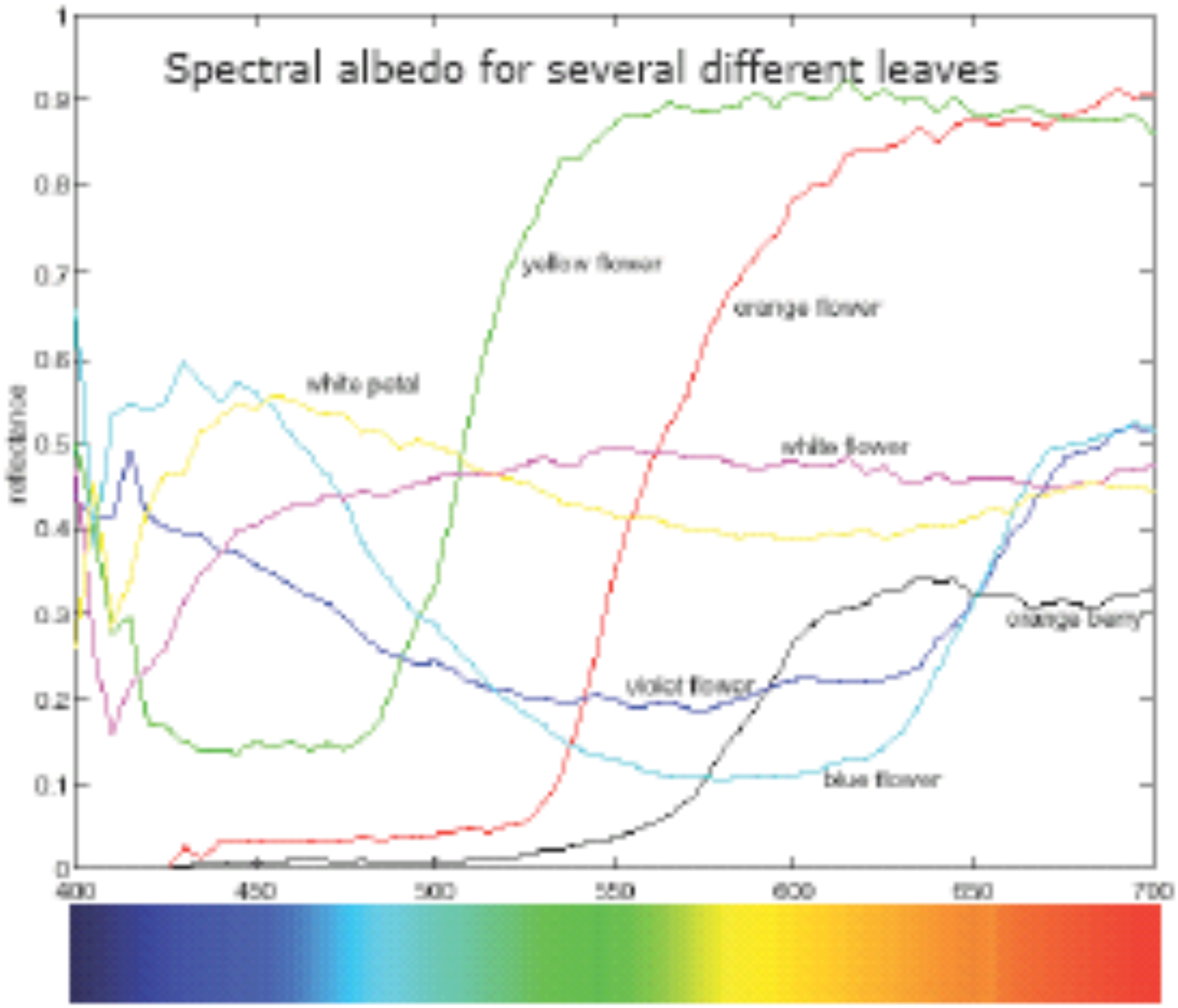


$$f_r(\theta_i, \phi_i, \theta_r, \phi_r; \lambda)$$

BRDF: bidirectional reflectance distribution function

Spectral Albedo

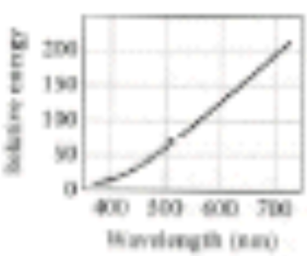
Ratio of incoming to outgoing radiation at different wavelengths.



Spectral Radiance

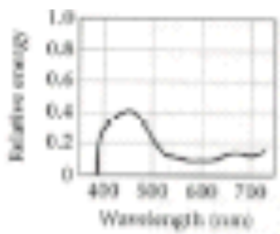
Often are more interested in relative spectral composition than in overall intensity, so the spectral BRDF computation simplifies to a wavelength-by-wavelength multiplication of relative energies

Spectral Irradiance



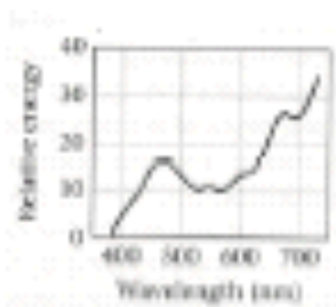
*

Spectral Albedo



=

Spectral Radiance



Light Transport

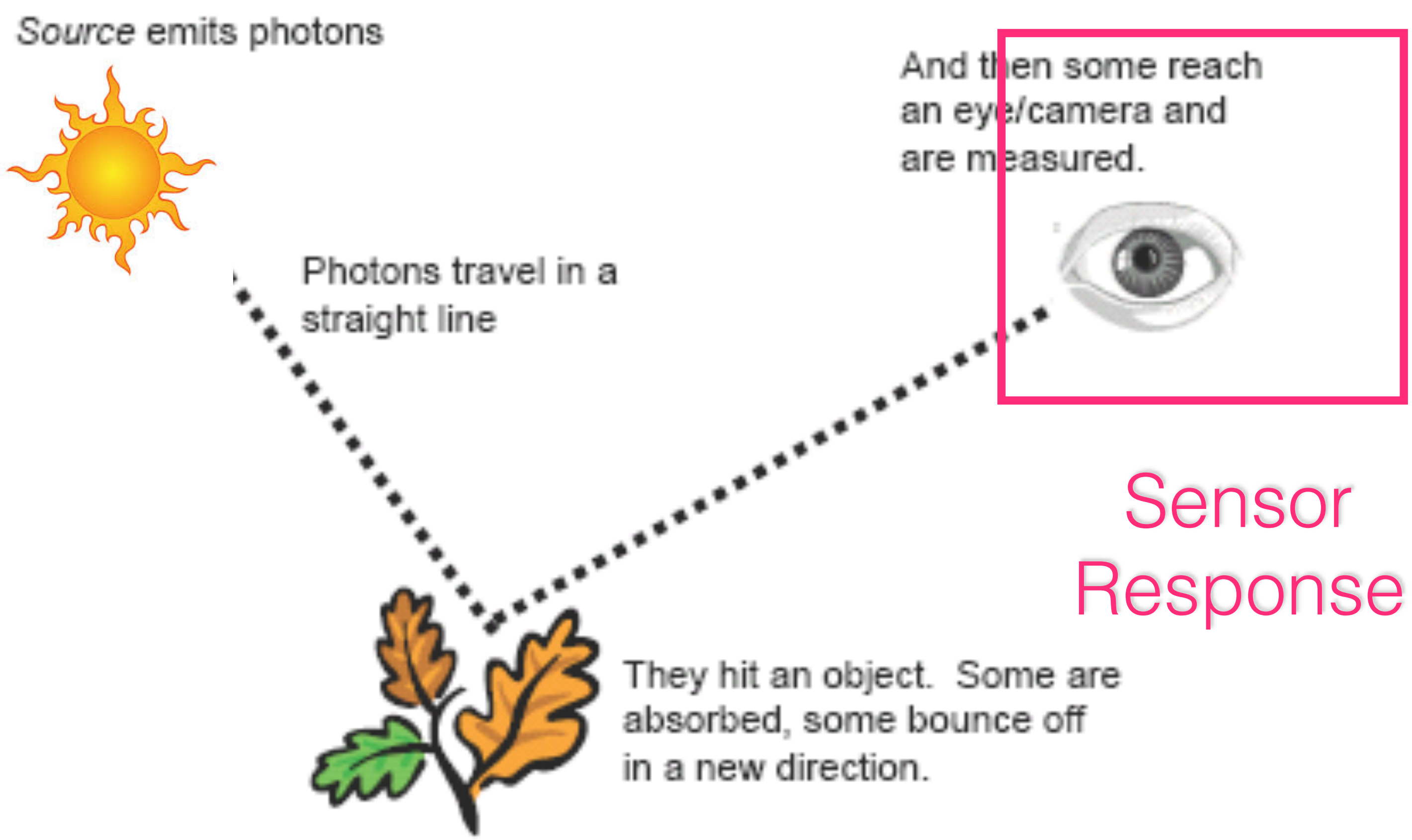
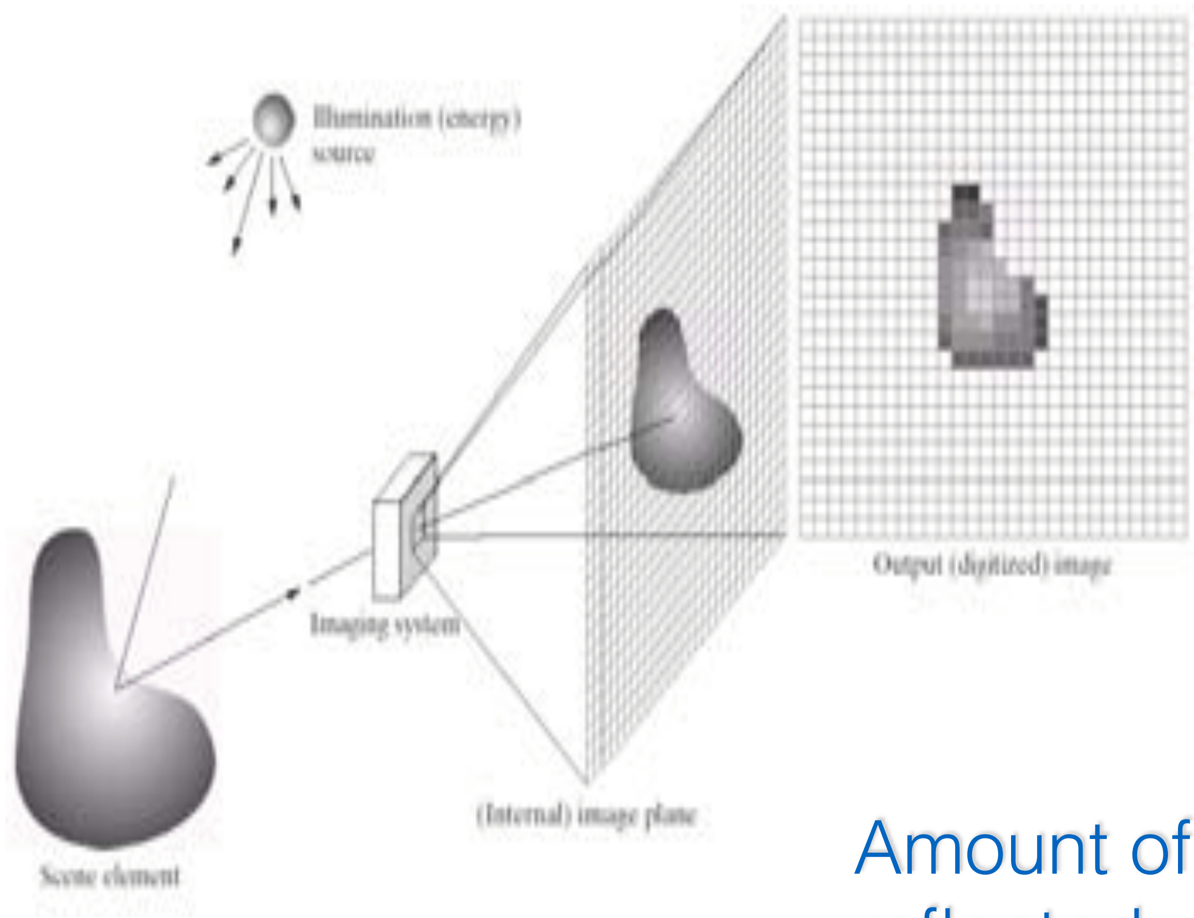


Image Formation Model



Amount of illumination reflected

$$f(x,y) = i(x,y)r(x,y)$$

Gray level

illumination