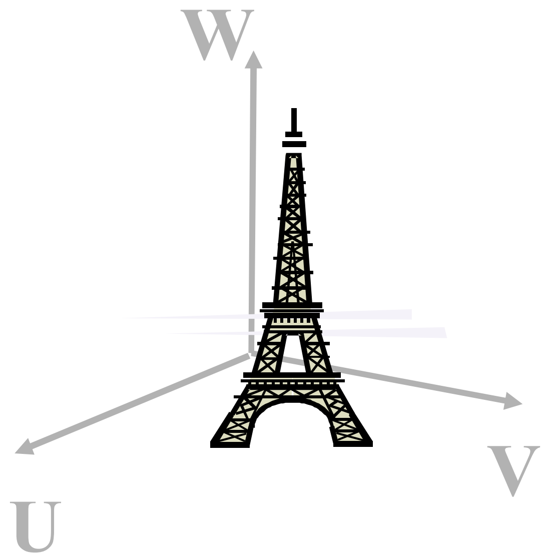
Lecture 13:
Camera Projection II

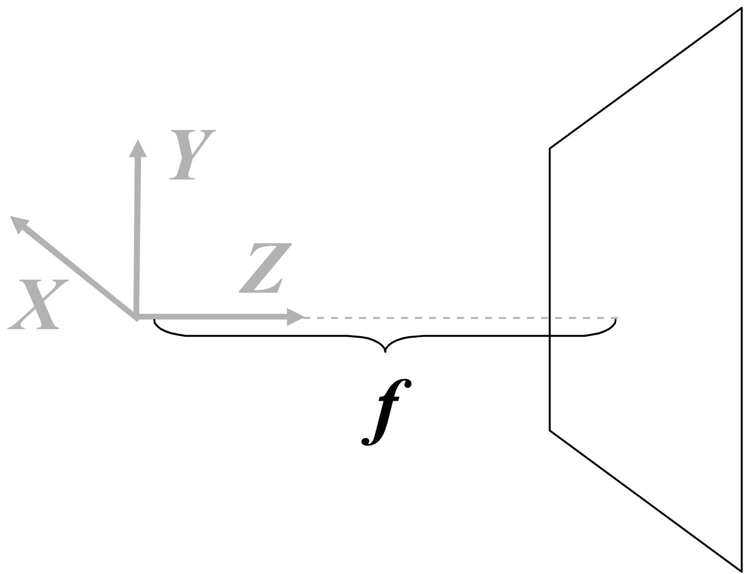
Reading: T&V Section 2.4

Recall: Imaging Geometry

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



Imaging Geometry

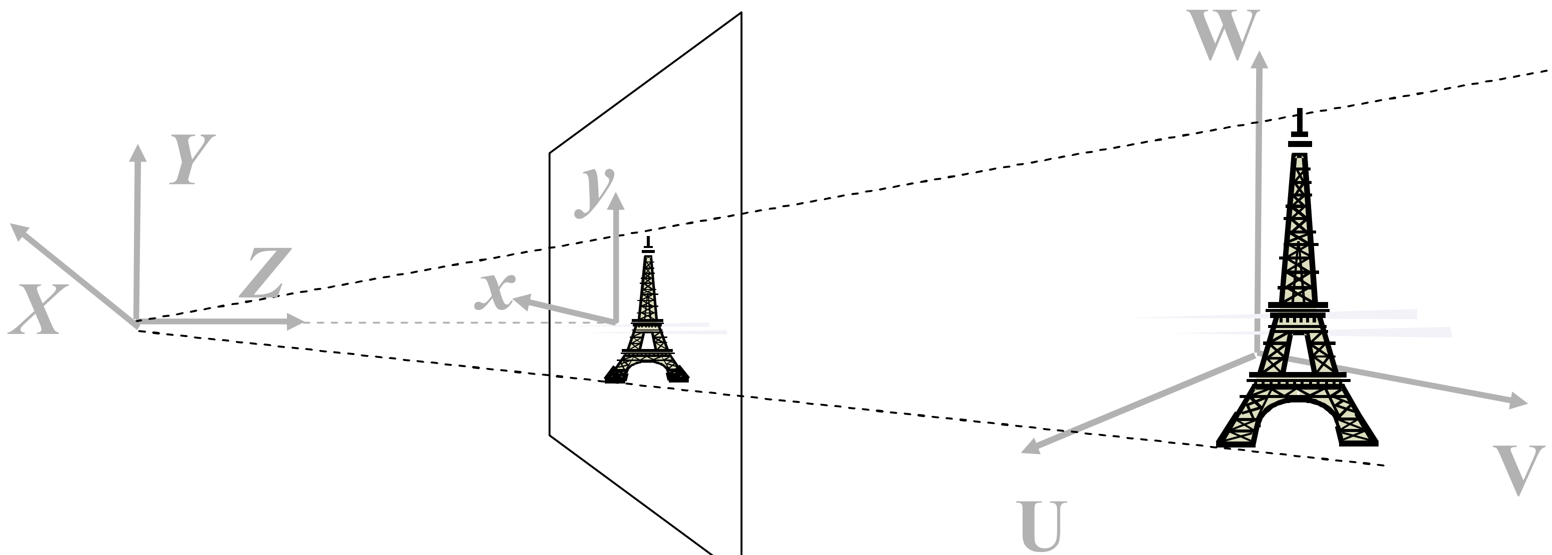


Camera Coordinate System (X,Y,Z).

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

Robert Collins
CSE486, Penn State

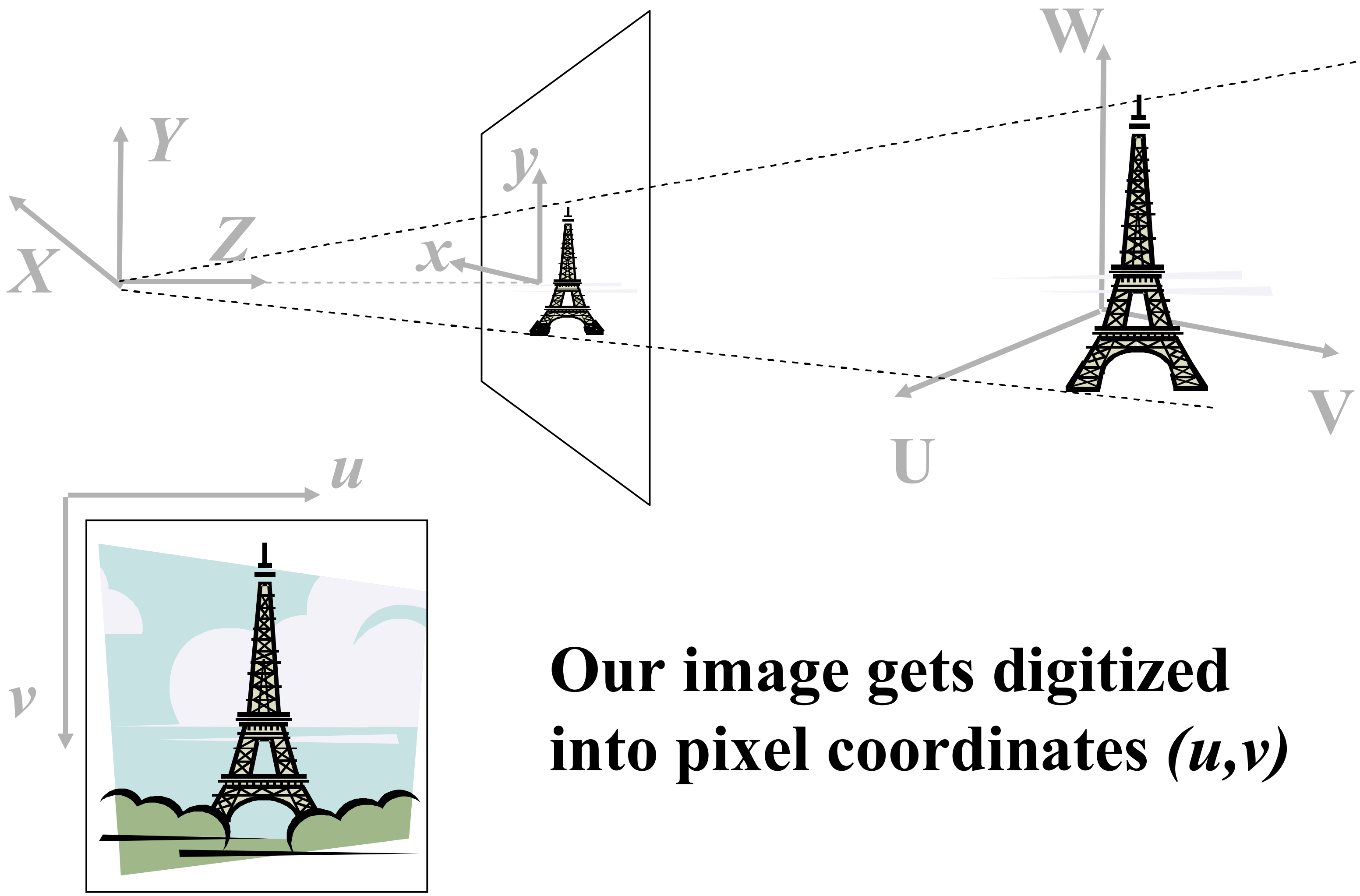
Imaging Geometry



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

Robert Collins
CSE486, Penn State

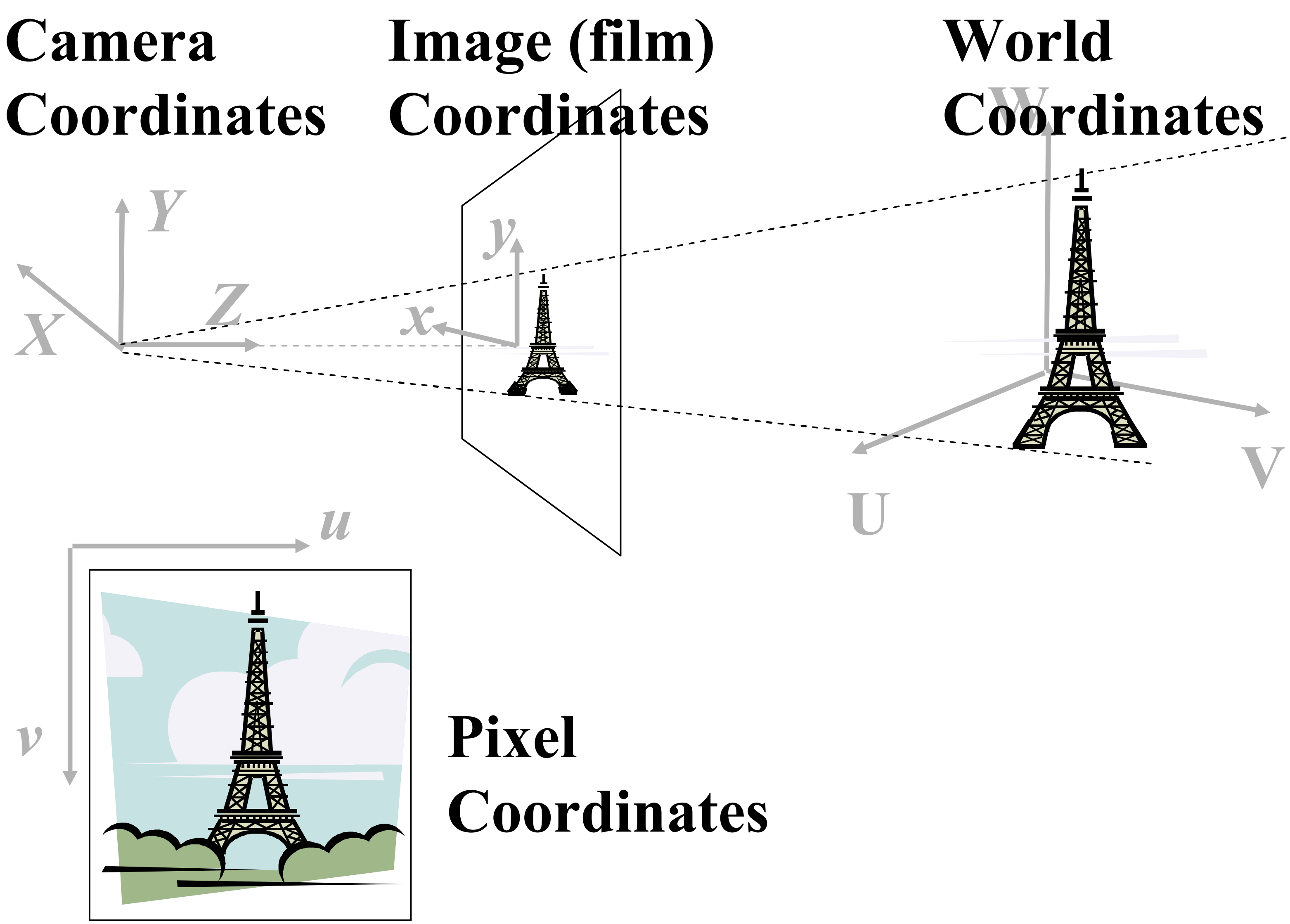
Imaging Geometry



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

Robert Collins
CSE486, Penn State

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!

Intrinsic Camera Parameters

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

Affine Transformation

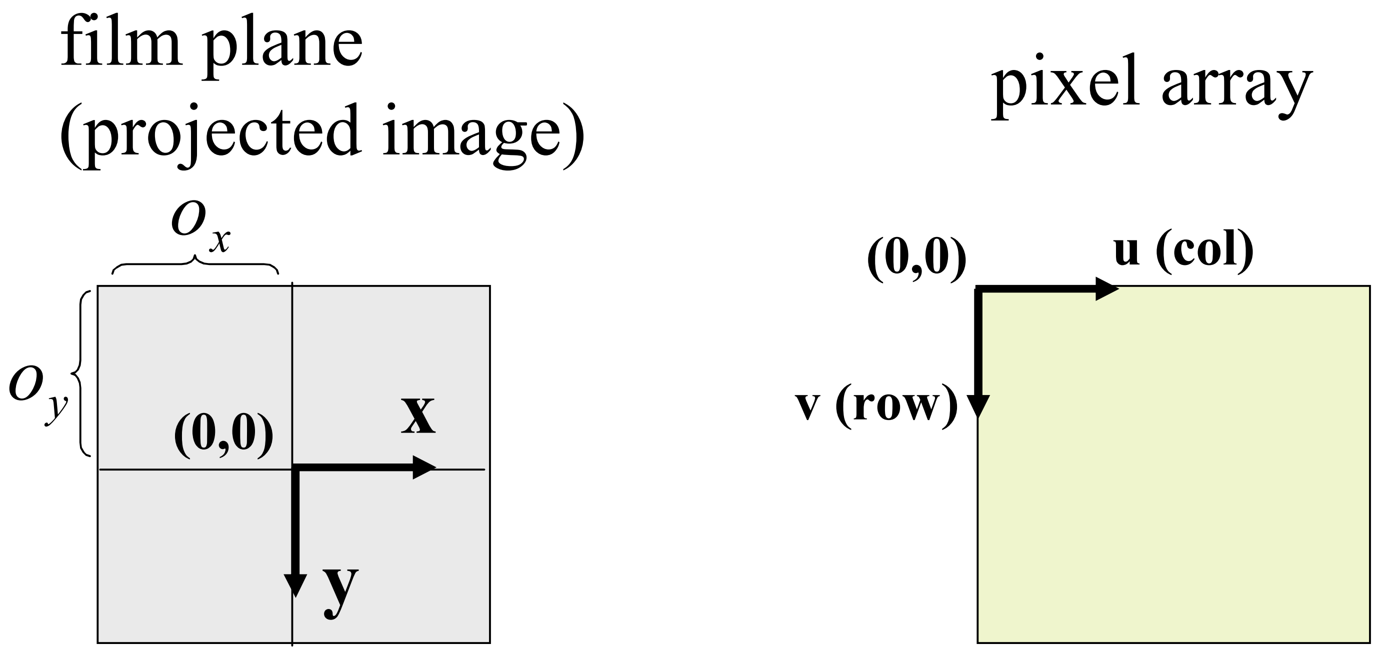
Intrinsic parameters

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

still in T&V section 2.4

Robert Collins
CSE486, Penn State

Intrinsic parameters (offsets)



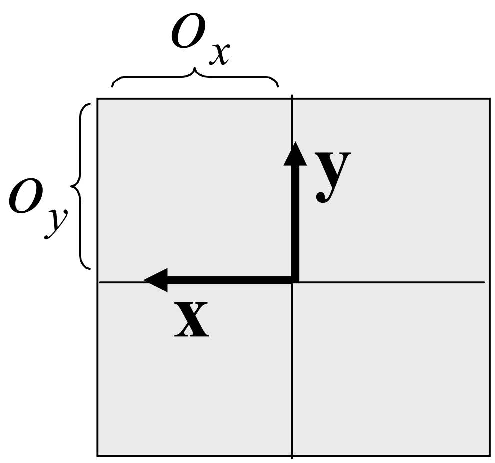
$$u = f\frac{X}{Z} + o_x \qquad v = f\frac{Y}{Z} + o_y$$

o_x and o_y called image center or principle point

Intrinsic parameters

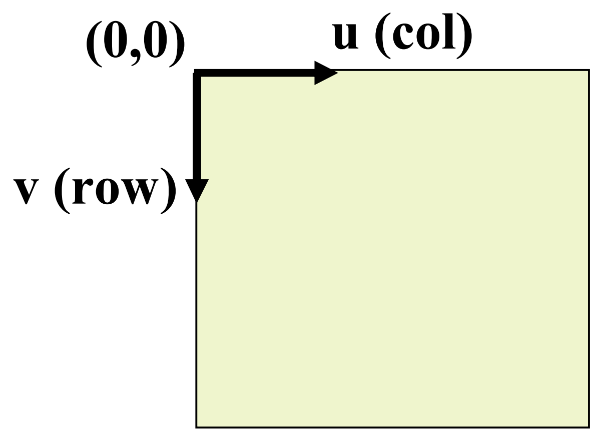
sometimes one or more coordinate axes are flipped (e.g. T&V section 2.4)

film plane



(0,0)

pixel array

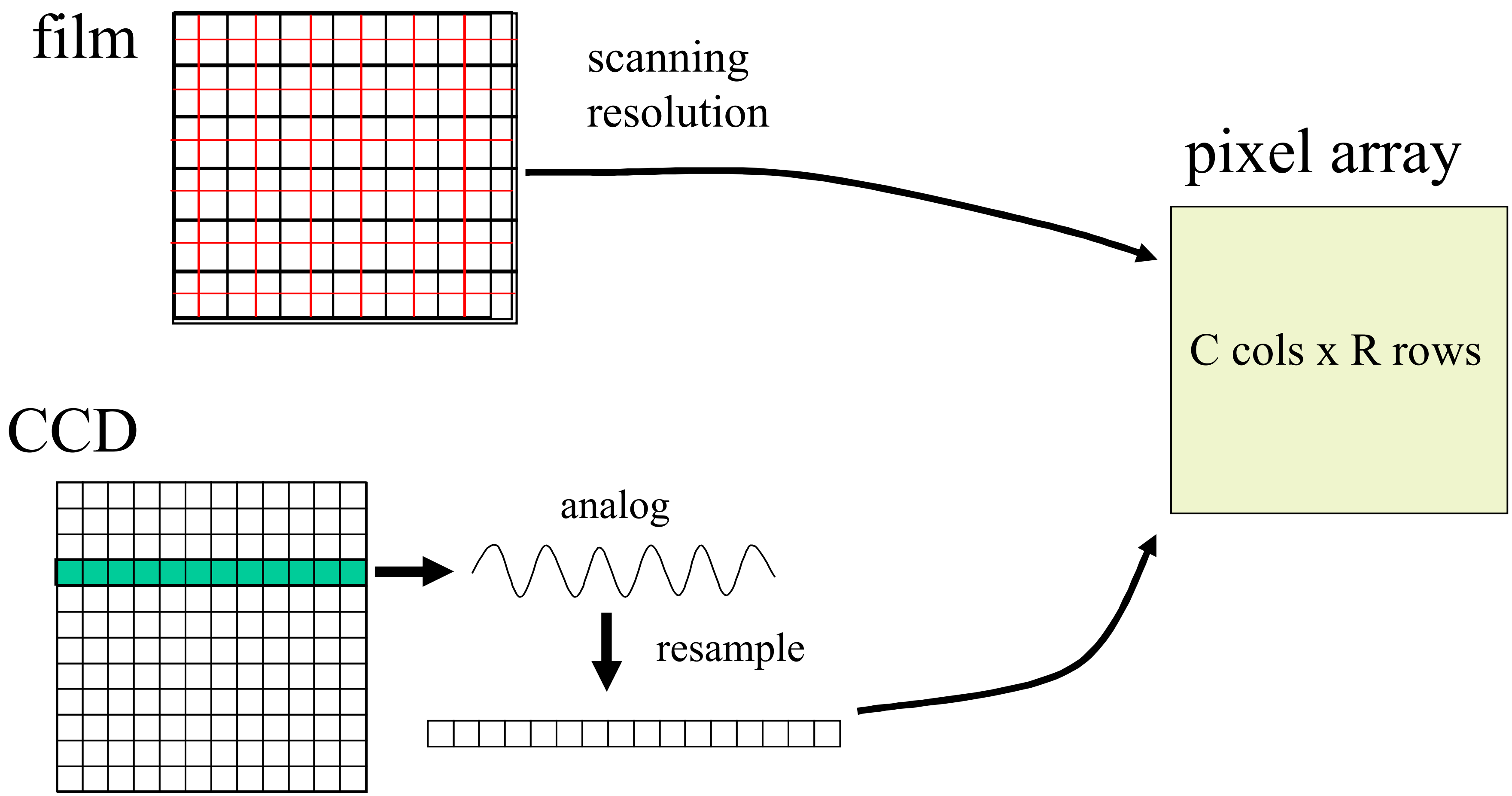


$$u = -f\frac{X}{Z} + o_x \qquad v = -f\frac{Y}{Z} + o_y$$

Robert Collins
CSE486, Penn State

Intrinsic parameters (scales)

sampling determines how many rows/cols in the image



Effective Scales: s_x and s_y

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

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

Perspective projection matrix

Adding the intrinsic parameters into the perspective projection matrix:

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

To verify:

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

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

Robert Collins
CSE486, Penn State

Note:

Sometimes, the image and the camera coordinate systems have opposite orientations: [the book does it this way]

$$f\frac{X}{Z} = -(u - o_x)s_x$$

$$f\frac{Y}{Z} = -(v - o_y)s_y$$

$$\Rightarrow$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} -f/s_x & 0 & +o_x & 0 \\ 0 & -f/s_y & +o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Robert Collins
CSE486, Penn State

Note 2

In general, I like to think of the conversion as a separate 2D affine transformation from film coords (x,y) to pixel coordinates (u,v):

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

$$\mathbf{u = M_{int}\, P_C = M_{aff}\, M_{proj}\, P_C}$$

Huh?

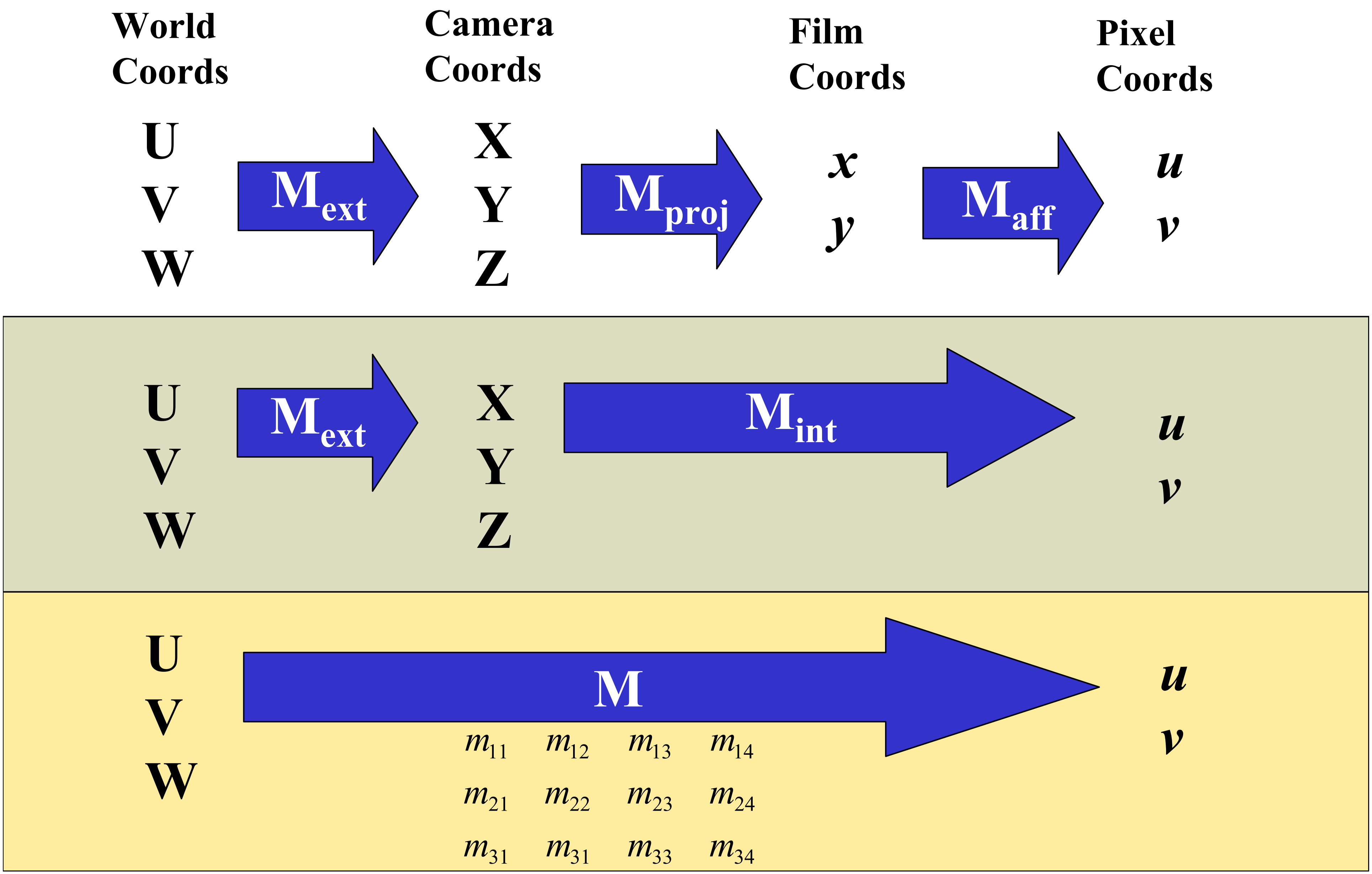
Did he just say it was "a fine" transformation?

No, it was "affine" transformation, a type of 2D to 2D mapping defined by 6 parameters.

More on this in a moment...

Robert Collins
CSE486, Penn State

Summary : Forward Projection



Summary: Projection Equation

$$
\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim
\underbrace{\underbrace{\overbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}}^{\text{Film plane to pixels}}}_{\mathbf{M_{aff}}}
\underbrace{\overbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}^{\text{Perspective projection}}}_{\mathbf{M_{proj}}}}_{\mathbf{M_{int}}}
\underbrace{\overbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}^{\text{World to camera}}}_{\mathbf{M_{ext}}}
\begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix}
$$

$$\mathbf{M} = \mathbf{M_{int}}\,\mathbf{M_{ext}}$$

Lecture 13/14: Intro to Image Mappings

Image Mappings Overview

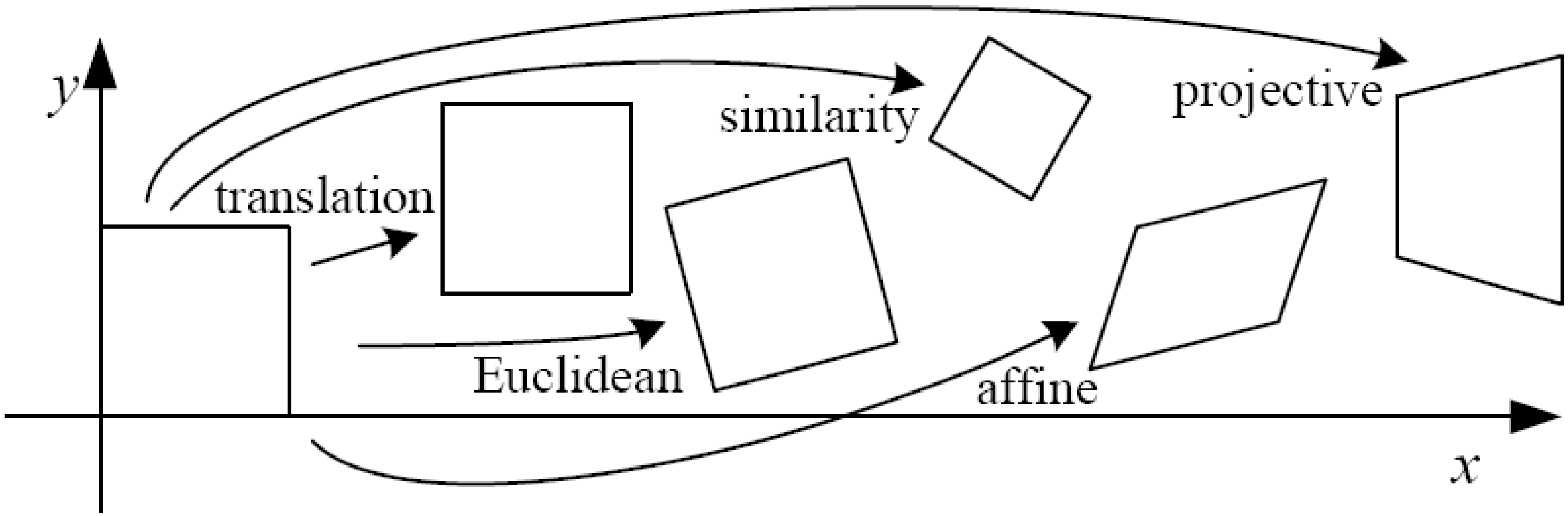
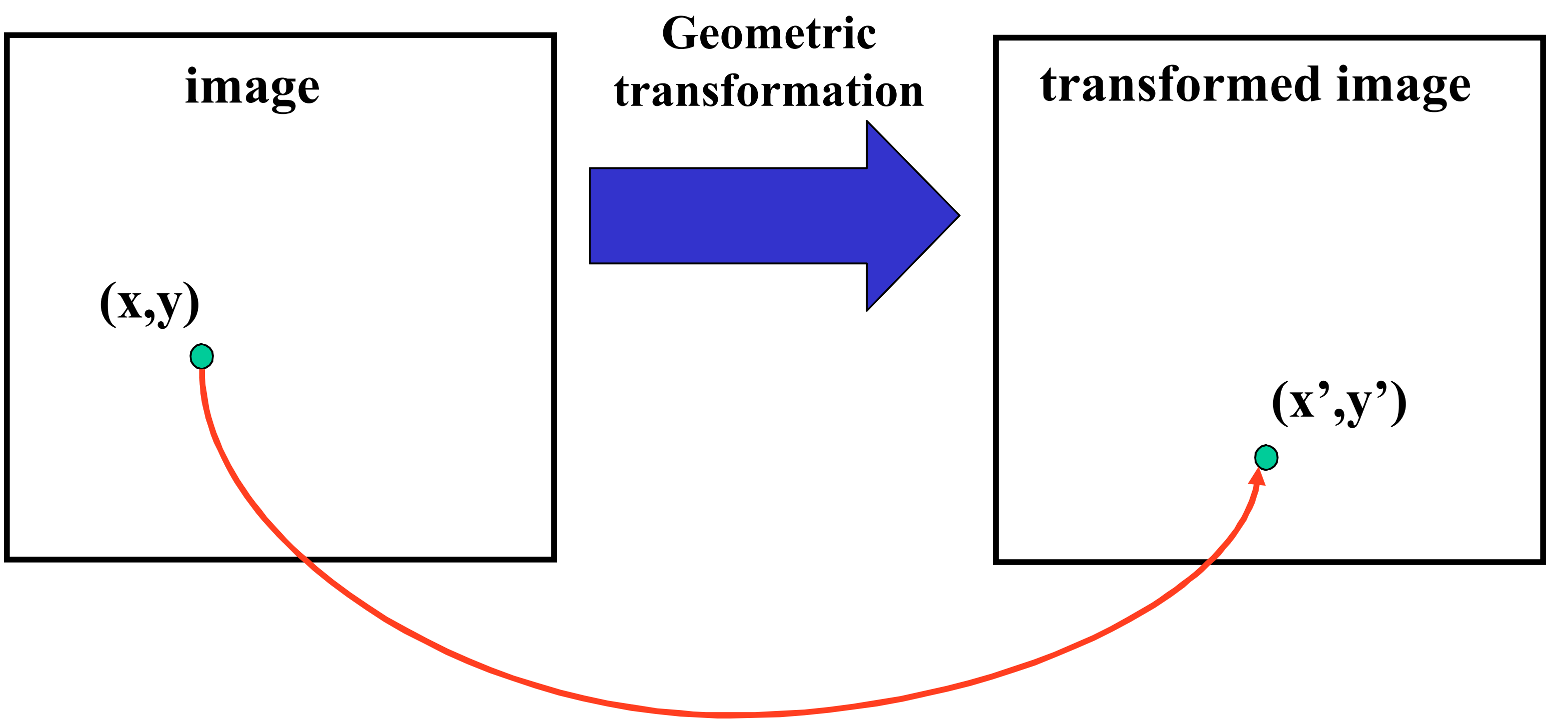


FIGURE 1. Basic set of 2D planar transformations

from R.Szeliski

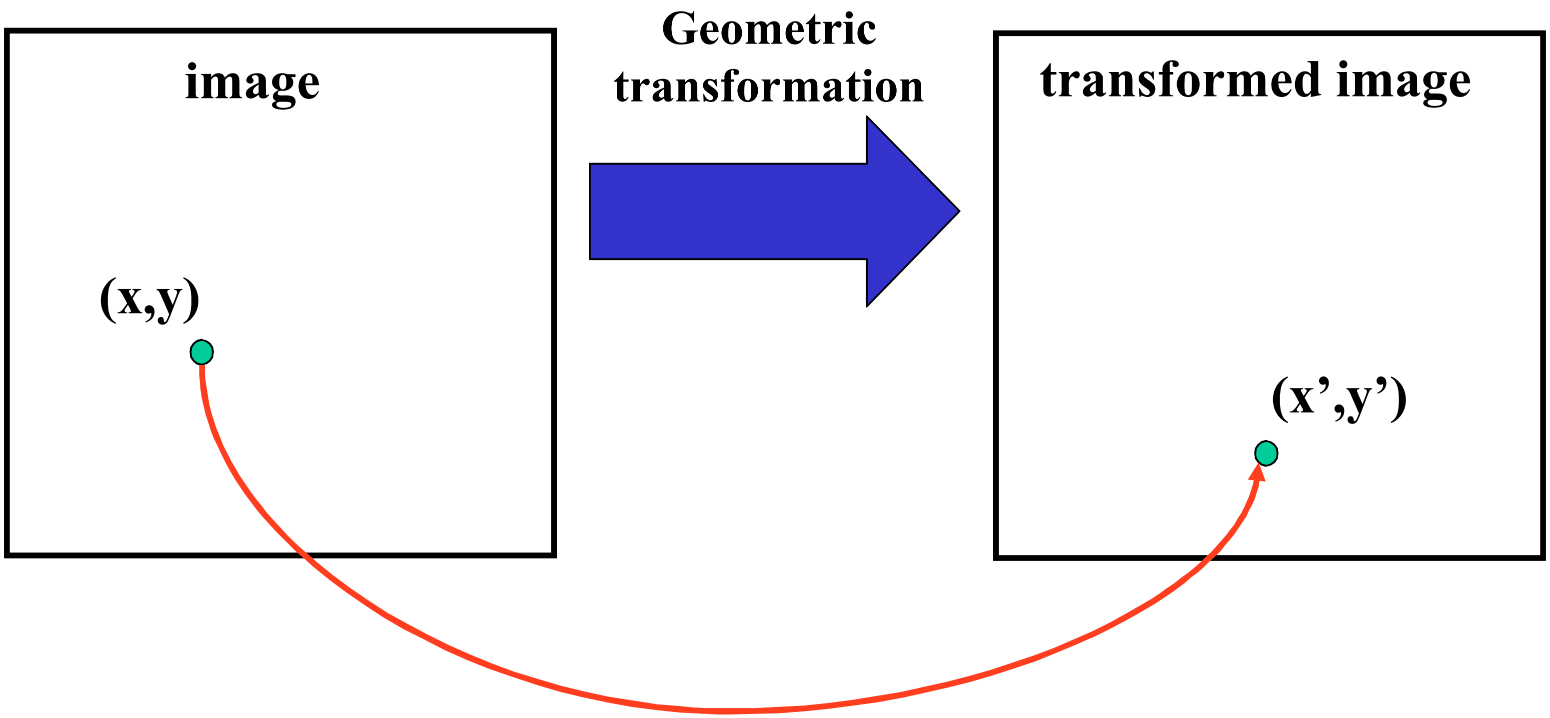
Robert Collins
CSE486, Penn State

Geometric Image Mappings



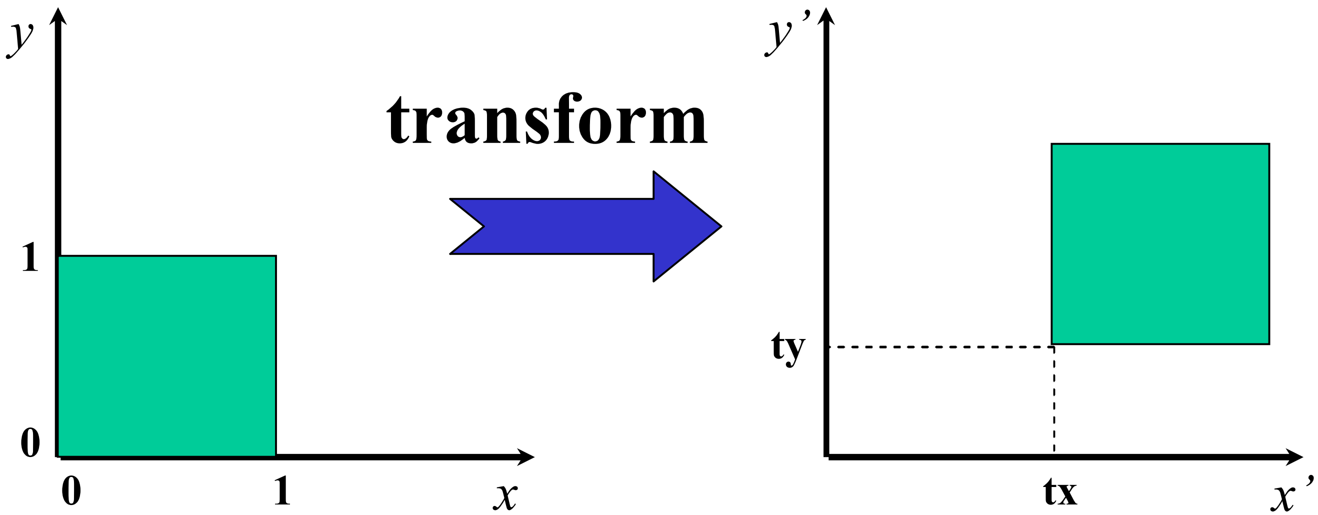
$$x' = f(x, y, \{\text{parameters}\})$$
$$y' = g(x, y, \{\text{parameters}\})$$

Linear Transformations
(Can be written as matrices)



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{pmatrix} \quad M(\text{params}) \quad \end{pmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translation



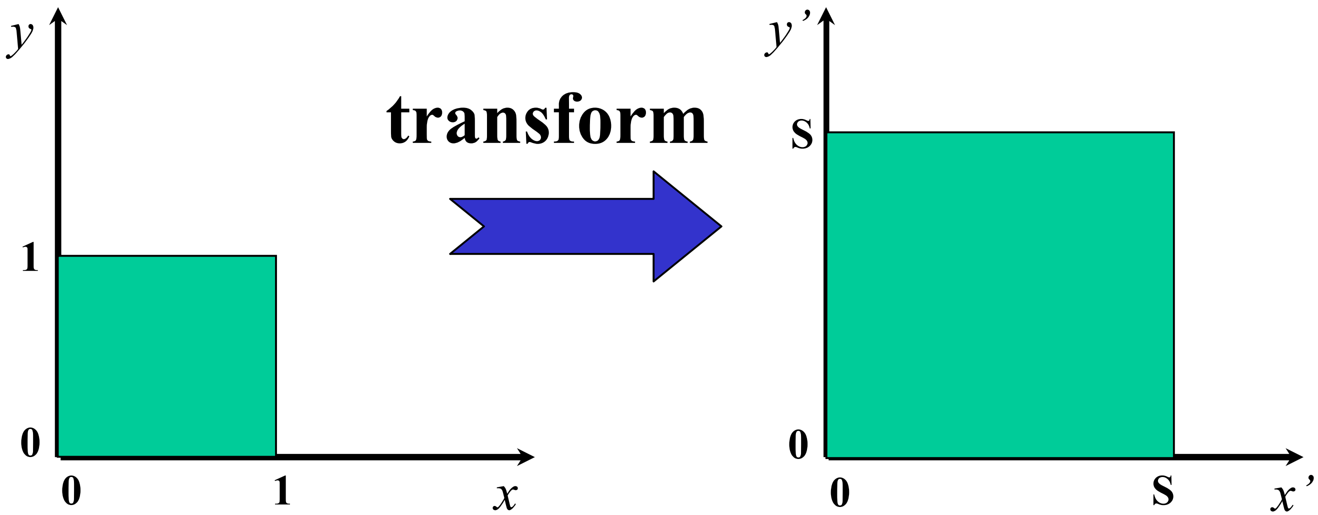
$$x' = x + t_x$$
$$y' = y + t_y$$

equations

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

matrix form

Scale



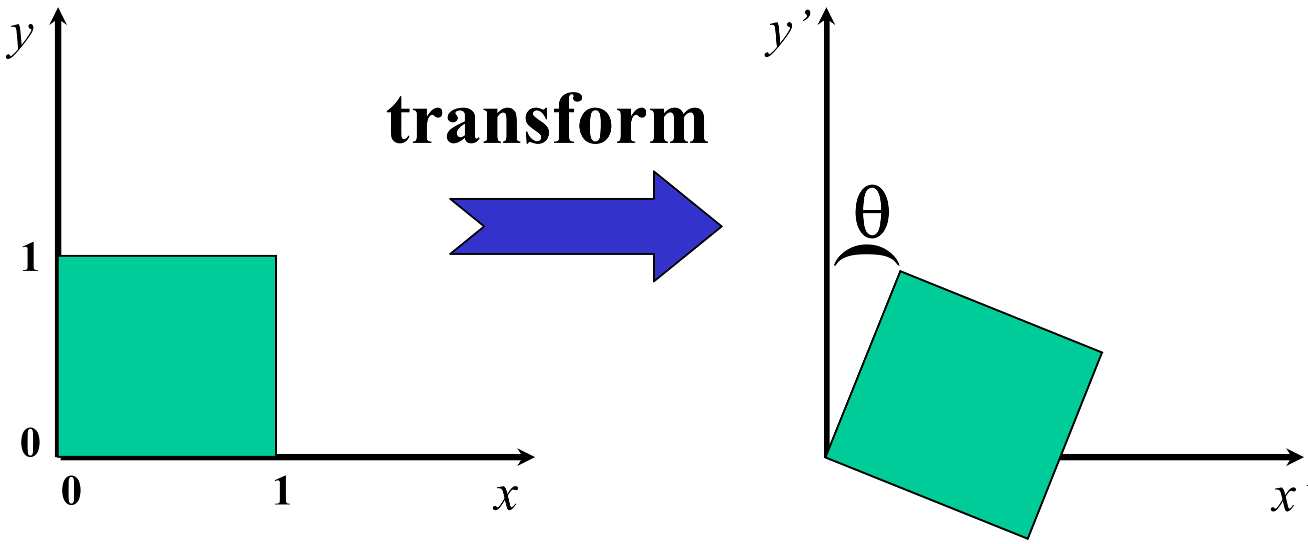
$$x' = s\,x_i$$
$$y' = s\,y_i$$

equations

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

matrix form

Rotation



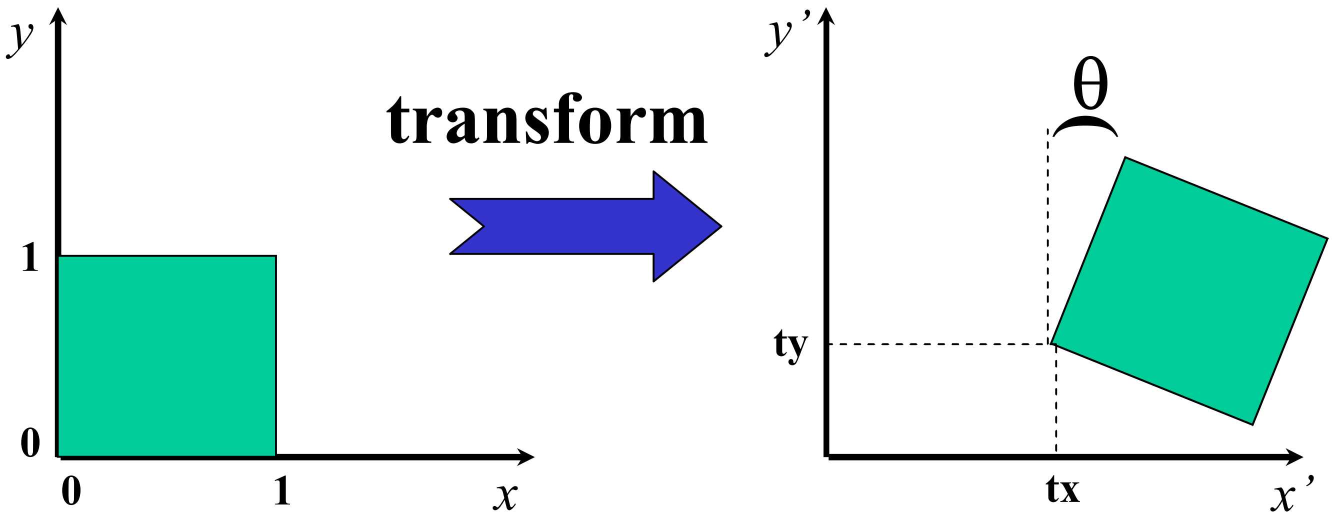
$$x' = x_i \cos\theta - y_i \sin\theta$$
$$y' = x_i \sin\theta + y_i \cos\theta$$

equations

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

matrix form

Euclidean (Rigid)



$$x' = x_i \cos\theta - y_i \sin\theta + t_x$$
$$y' = x_i \sin\theta + y_i \cos\theta + t_y$$

equations

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

matrix form

Robert Collins
CSE486, Penn State

Partitioned Matrices

A *partitioned matrix*, or a *block matrix*, is a matrix M that has been constructed from other smaller matrices. These smaller matrices are called *blocks* or *sub-matrices* of M.

For instance, if we partition the below 5×5 matrix as follows

$$L = \left(\begin{array}{cc|ccc} 1 & 0 & 1 & 2 & 3 \\ 0 & 1 & 1 & 2 & 3 \\ \hline 2 & 3 & 9 & 9 & 9 \\ 2 & 3 & 9 & 9 & 9 \\ 2 & 3 & 9 & 9 & 9 \end{array}\right),$$

then we can define the matrices

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, C = \begin{pmatrix} 2 & 3 \\ 2 & 3 \\ 2 & 3 \end{pmatrix}, D = \begin{pmatrix} 9 & 9 & 9 \\ 9 & 9 & 9 \\ 9 & 9 & 9 \end{pmatrix}$$

and write L as

$$L = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \quad \text{or} \quad L = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right).$$

http://planetmath.org/encyclopedia/PartitionedMatrix.html

Partitioned Matrices

$$\begin{bmatrix} x' \\ y' \\ \hline 1 \end{bmatrix} = \left[\begin{array}{cc|c} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ \hline 0 & 0 & 1 \end{array}\right] \begin{bmatrix} x \\ y \\ \hline 1 \end{bmatrix}$$

$$\begin{bmatrix} \overset{\text{2x1}}{p'} \\ \overset{\text{1x1}}{1} \end{bmatrix} = \begin{bmatrix} \overset{\text{2x2}}{R} & \overset{\text{2x1}}{t} \\ \overset{\text{1x2}}{0} & \overset{\text{1x1}}{1} \end{bmatrix} \begin{bmatrix} \overset{\text{2x1}}{p} \\ \overset{\text{1x1}}{1} \end{bmatrix}$$

matrix form

$$p' = Rp + t$$

equation form

Another Example (from last time)

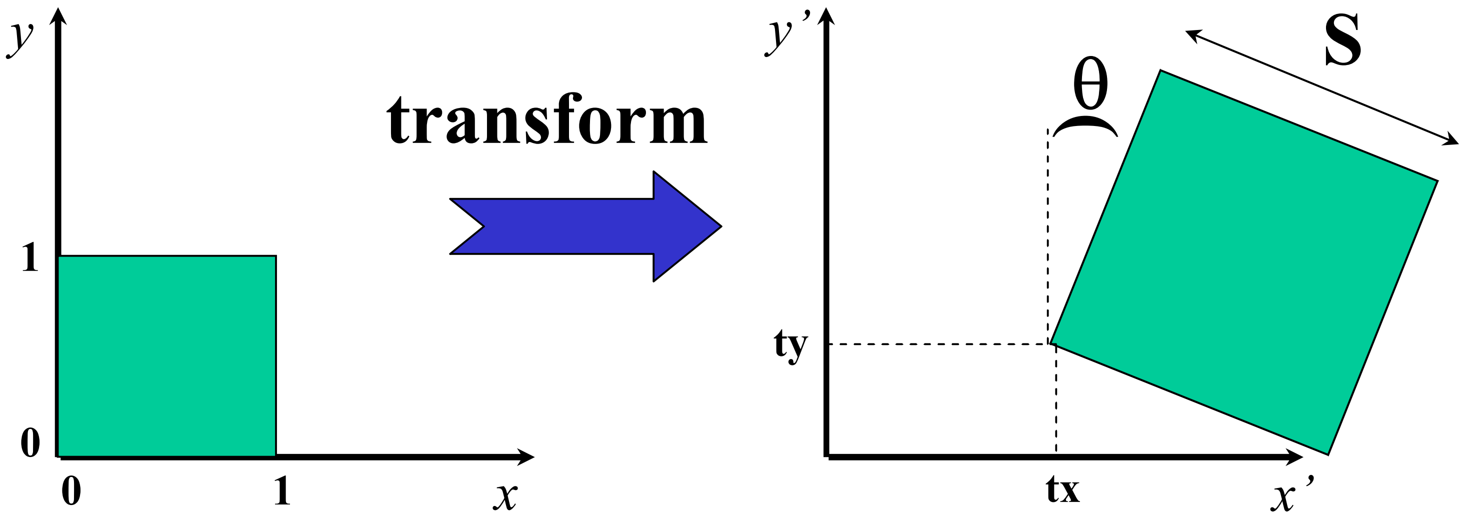
$$\begin{pmatrix} X \\ Y \\ Z \\ \hline 1 \end{pmatrix} = \left(\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \begin{pmatrix} U \\ V \\ W \\ \hline 1 \end{pmatrix}$$

$$\begin{pmatrix} \overset{3\text{x}1}{\mathbf{P}_{\mathbf{C}}} \\ \overset{1\text{x}1}{1} \end{pmatrix} = \begin{pmatrix} \overset{3\text{x}3}{\mathbf{R}} & \overset{3\text{x}1}{\mathbf{T}} \\ \overset{1\text{x}3}{0} & \overset{1\text{x}1}{1} \end{pmatrix} \begin{pmatrix} \overset{3\text{x}1}{\mathbf{P}_{\mathbf{W}}} \\ \overset{1\text{x}1}{1} \end{pmatrix}$$

$$\mathbf{P}_{\mathbf{C}} = \mathbf{R}\,\mathbf{P}_{\mathbf{W}} + \mathbf{T}$$

Robert Collins
CSE486, Penn State

Similarity (scaled Euclidean)



$$p' = sRp + t$$

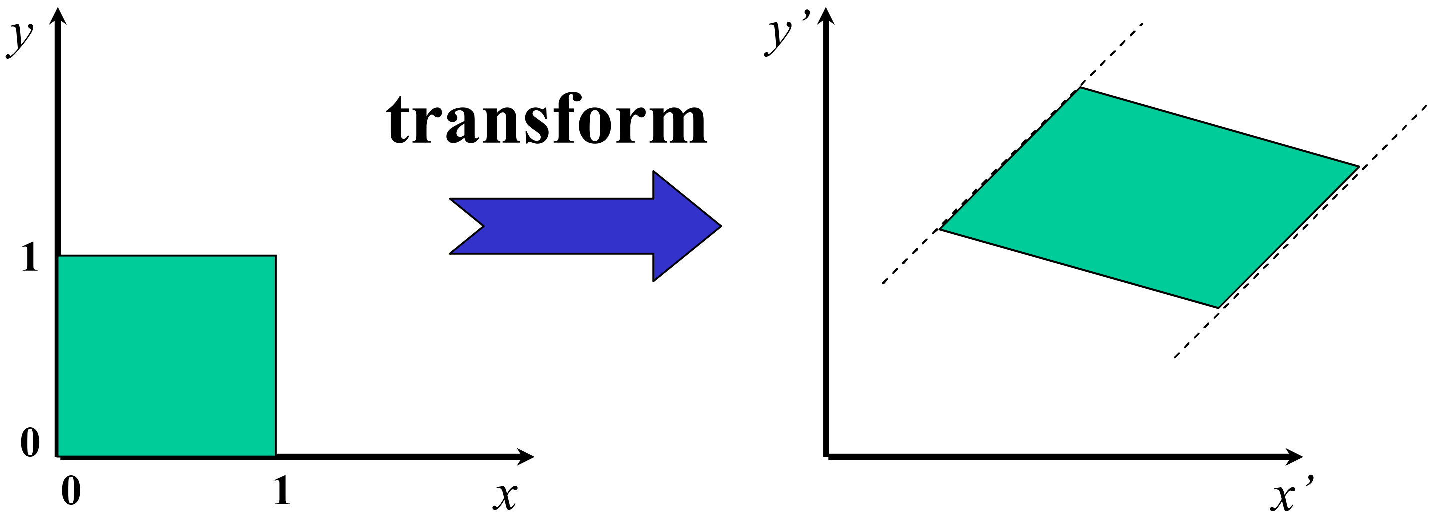
equations

$$\begin{bmatrix} p' \\ 1 \end{bmatrix} = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix}$$

matrix form

Robert Collins
CSE486, Penn State

Affine



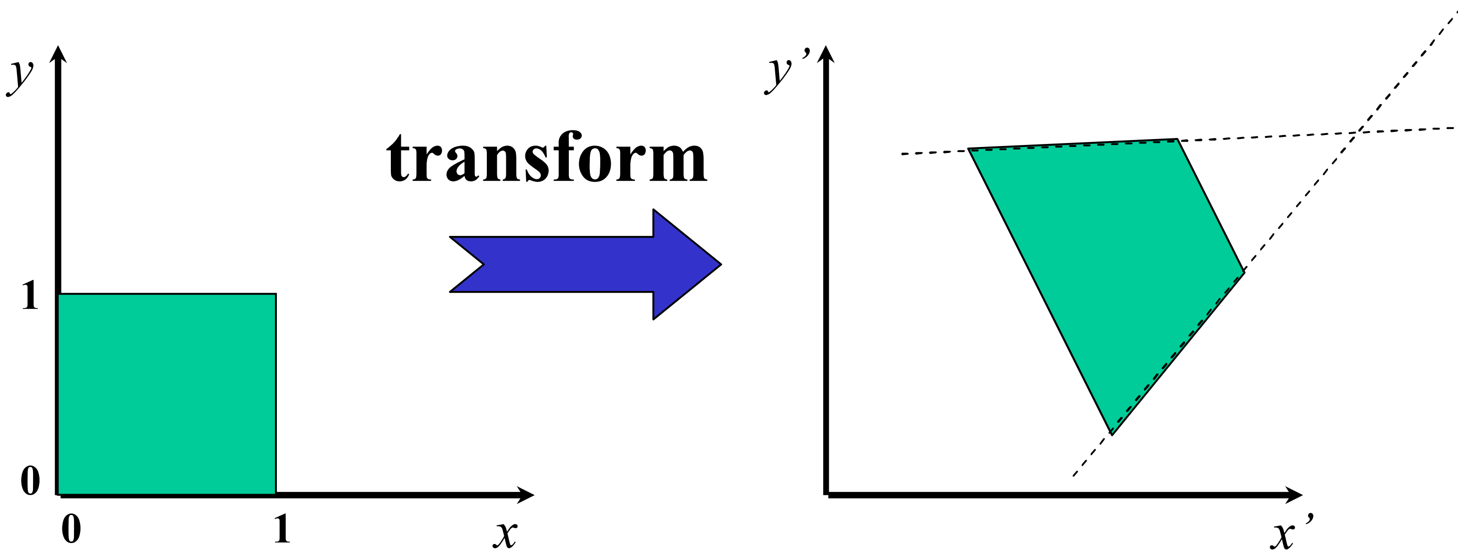
$$p' = Ap + b$$

equations

$$\begin{bmatrix} p' \\ 1 \end{bmatrix} = \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix}$$

matrix form

Projective



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

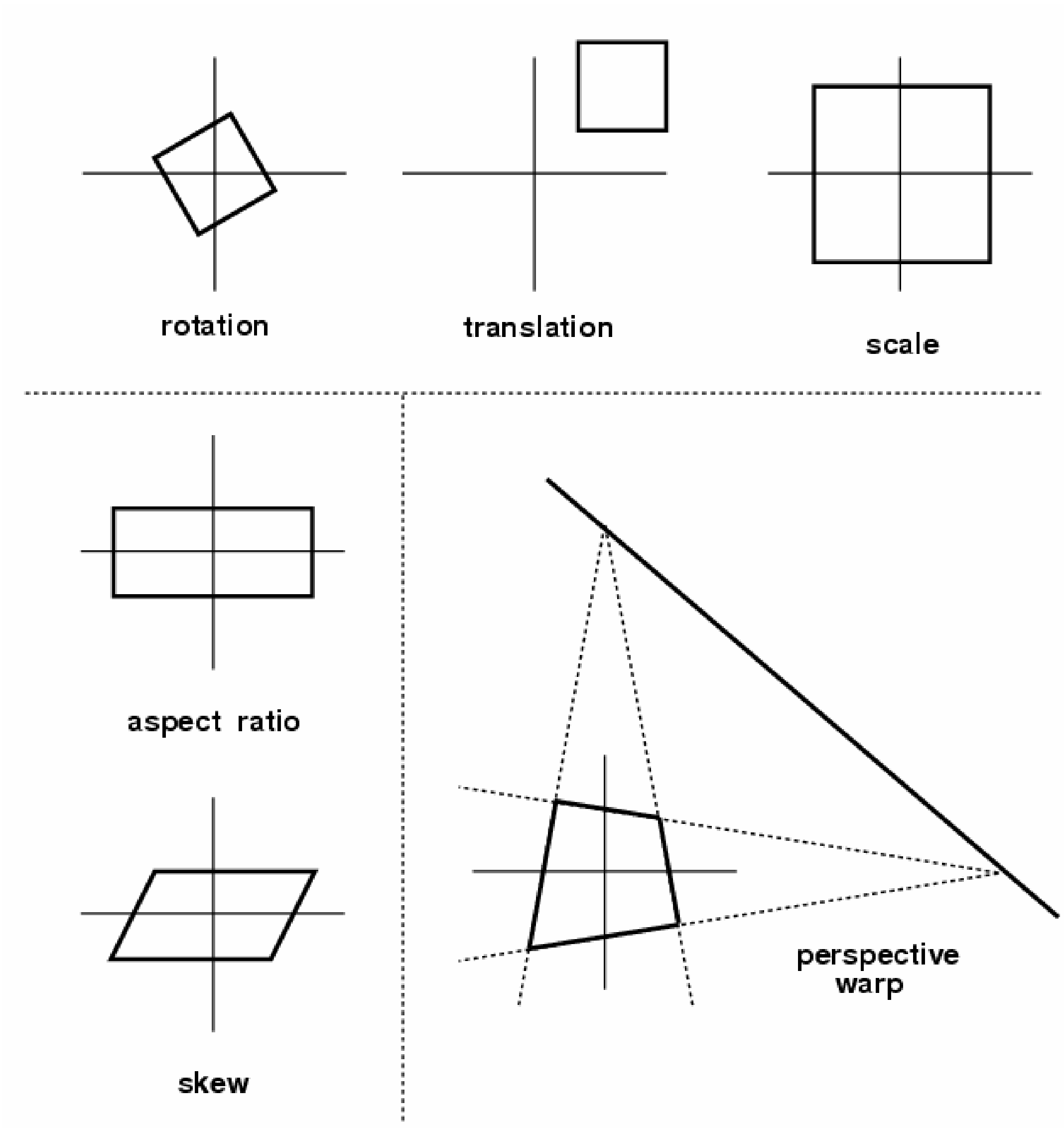
equations

Note!

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

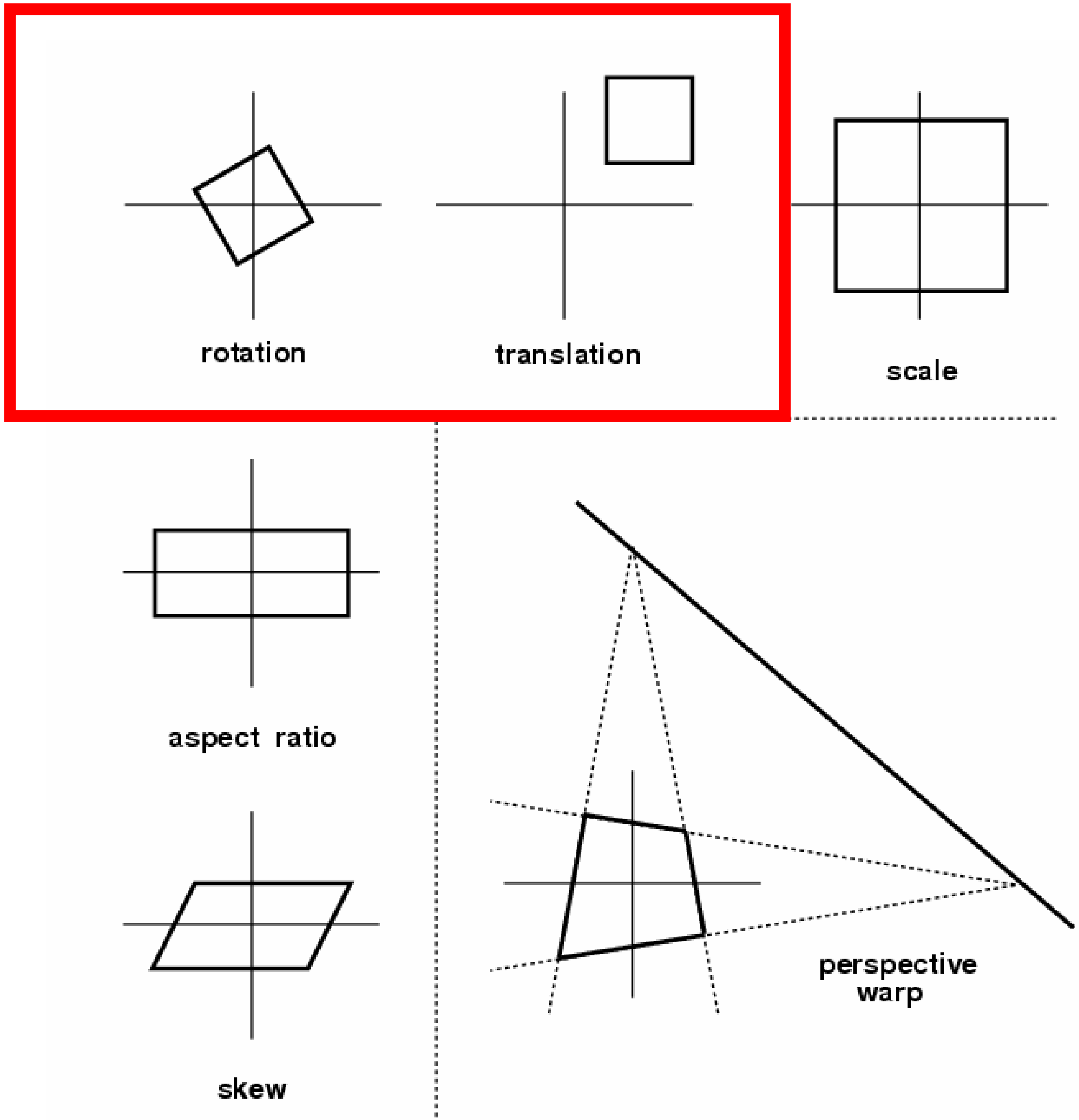
matrix form

Summary of 2D Transformations



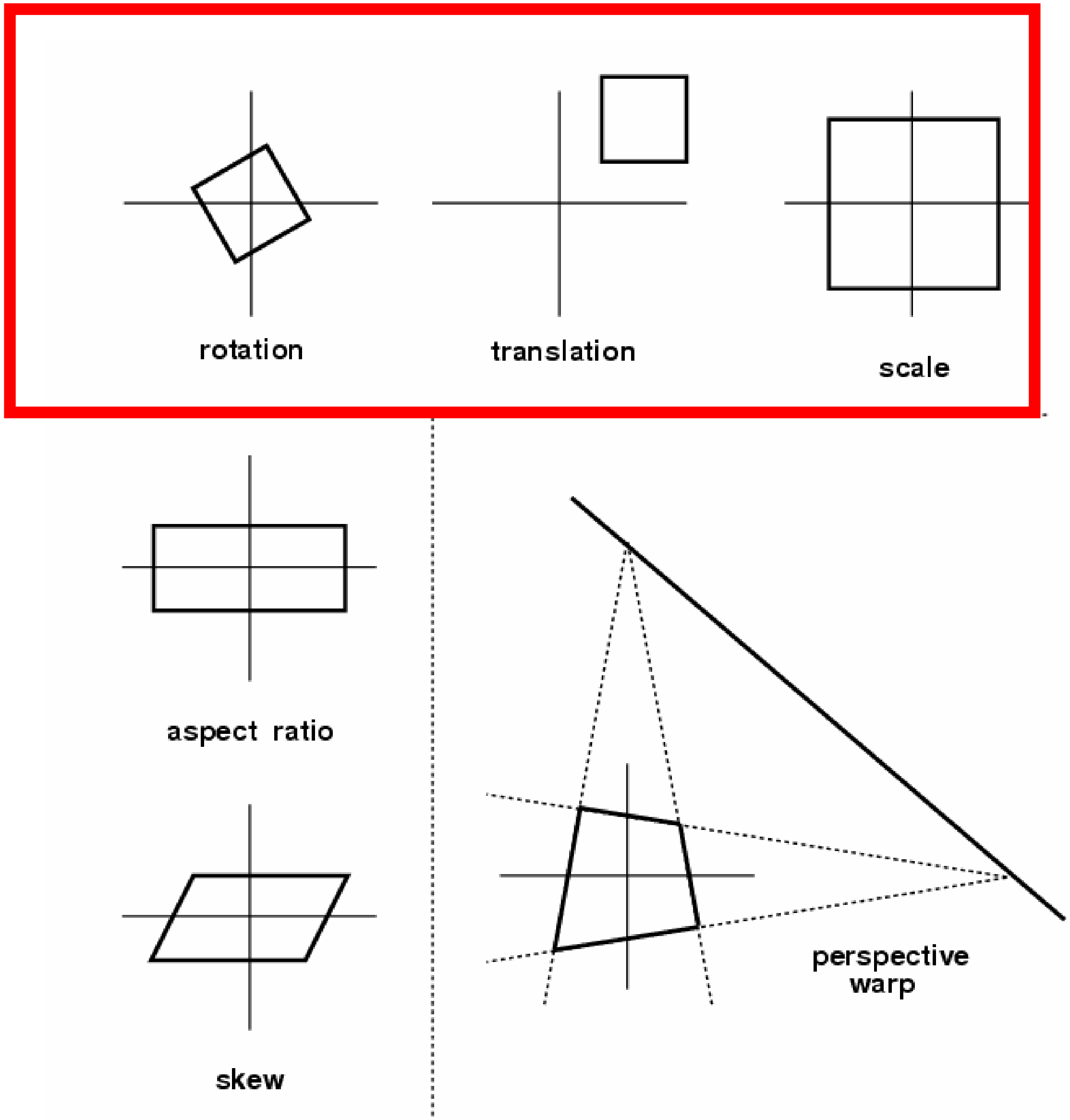
Summary of 2D Transformations

Euclidean



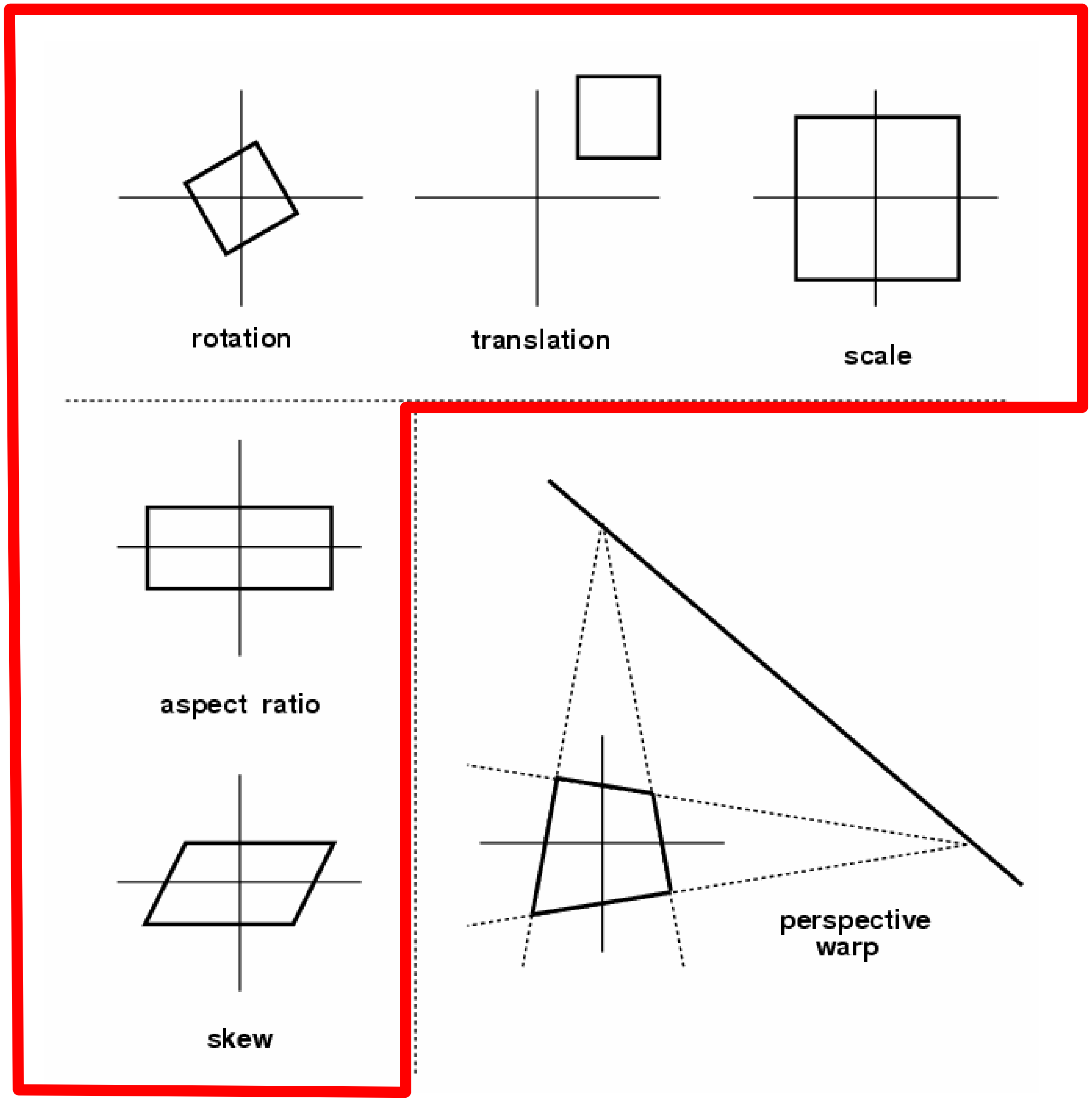
Summary of 2D Transformations

Similarity



Summary of 2D Transformations

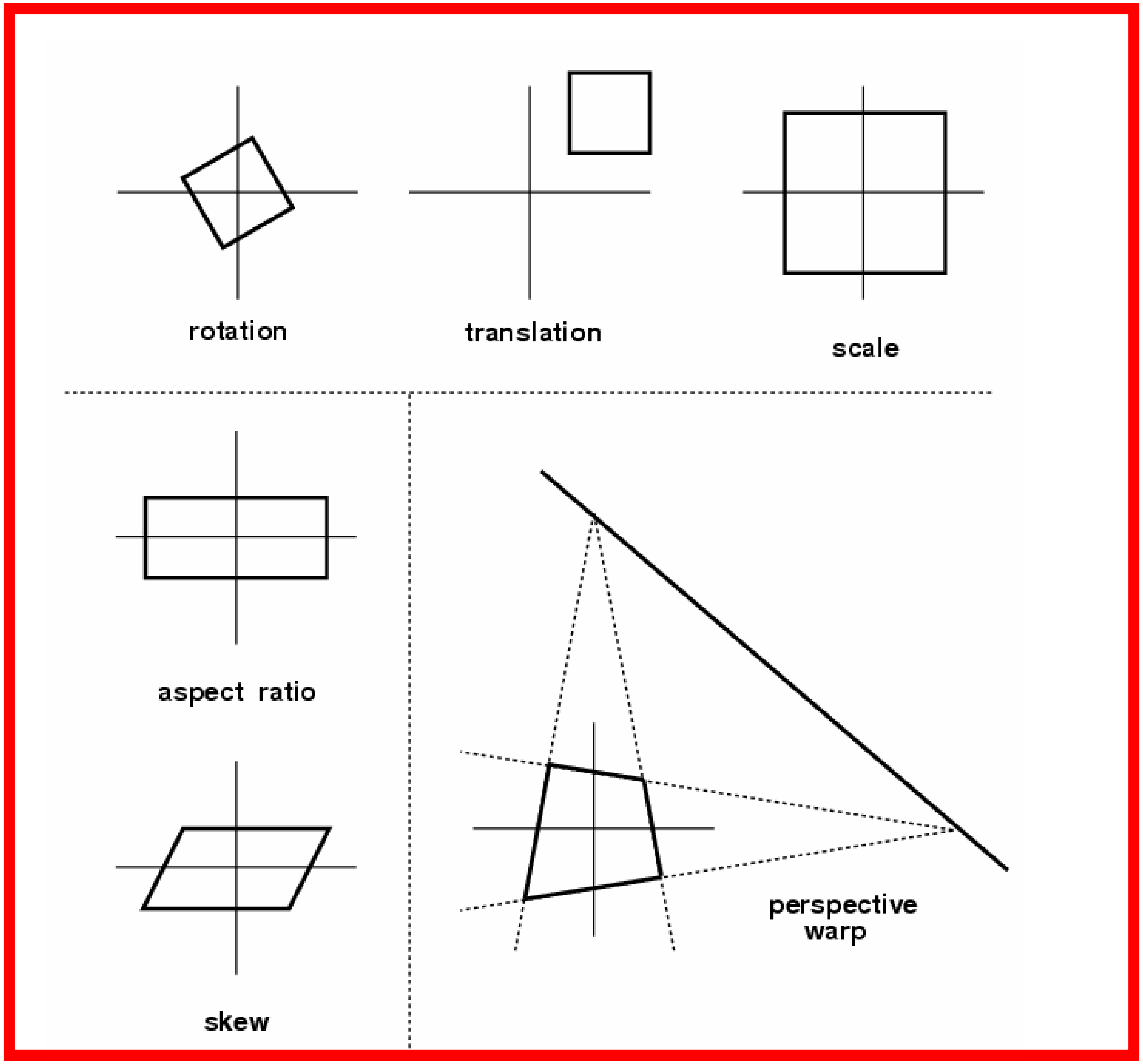
Affine



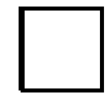
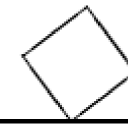
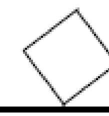
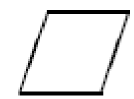
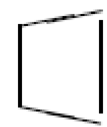
Robert Collins
CSE486, Penn State

Summary of 2D Transformations

Projective



Summary of 2D Transformations

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

from R.Szeliski