

CEE598 - Visual Sensing for Civil Infrastructure Eng. & Mgmt.

Session 2 – Review of Linear Algebra and Geometric Transformations

Mani Golparvar-Fard

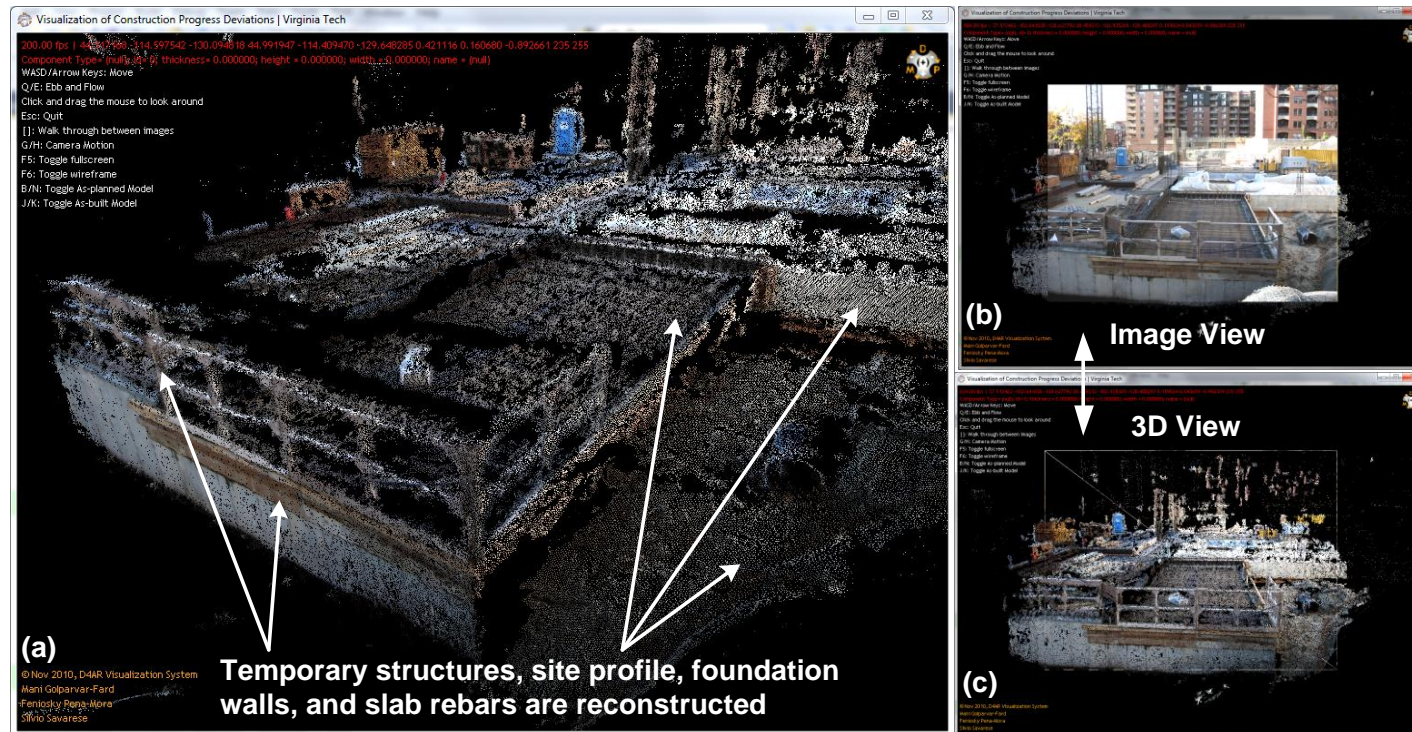
Department of Civil and Environmental Engineering

3129D, Newmark Civil Engineering Lab

e-mail: mgolpar@illinois.edu

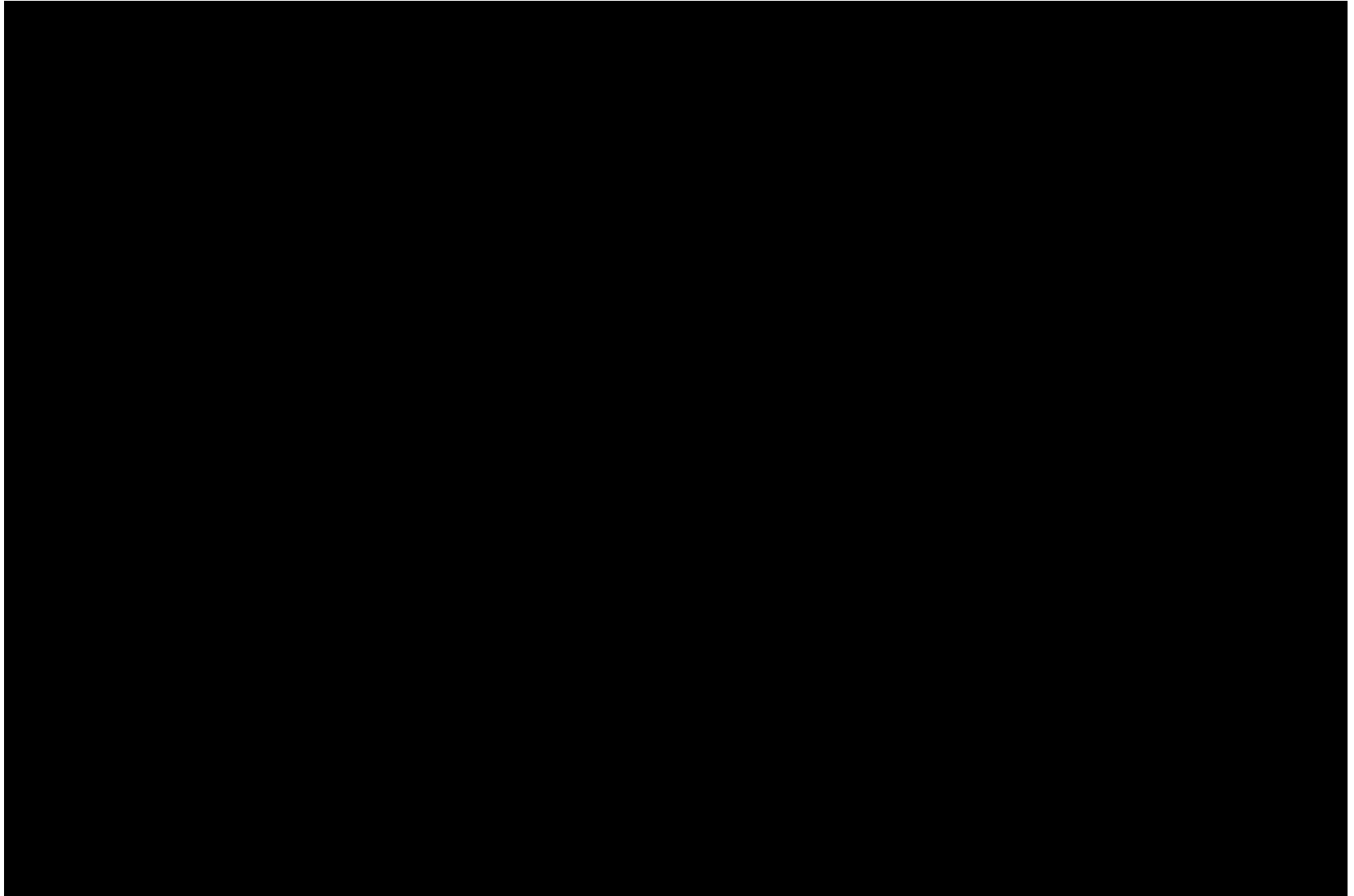
Useful Readings

- Any book on linear algebra!
- [HZ] – chapters 2, 4.

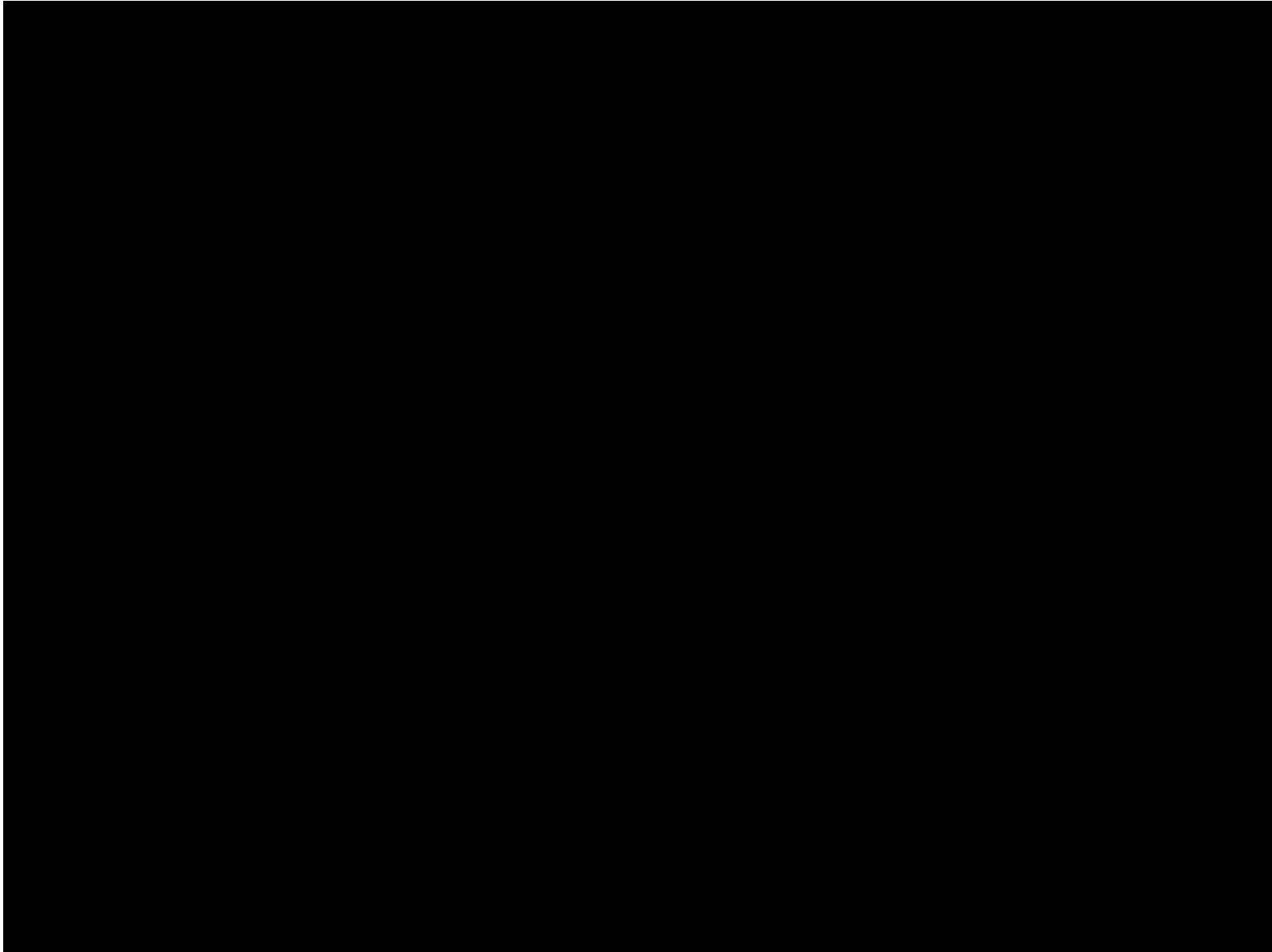


Some of the slides in this lecture are courtesy to Prof. Octavia I. Camps, Penn State University, Prof. Silvio Savarese, University of Michigan and Prof. Noah Snavely, Cornell University.

Application of Traffic Engineering



Google Art Project



CAPTCHA & Re-CAPTCHA

Outline

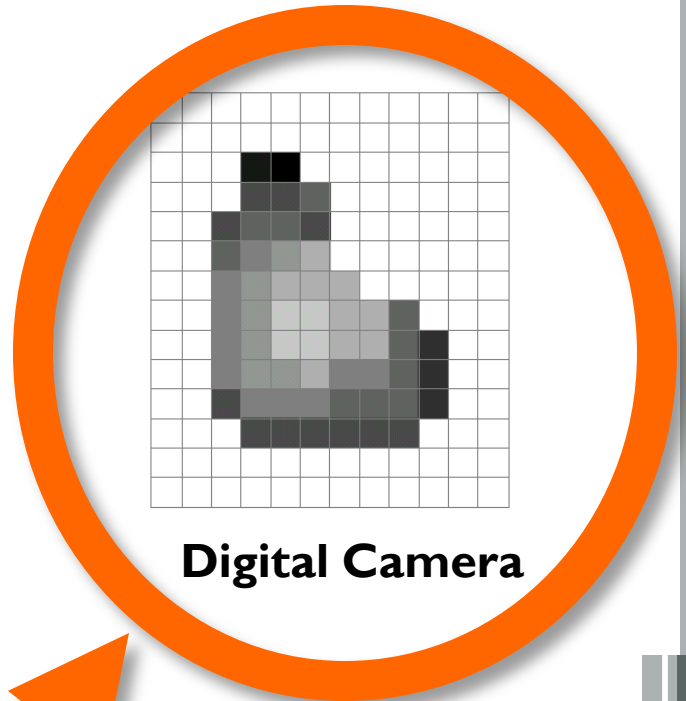
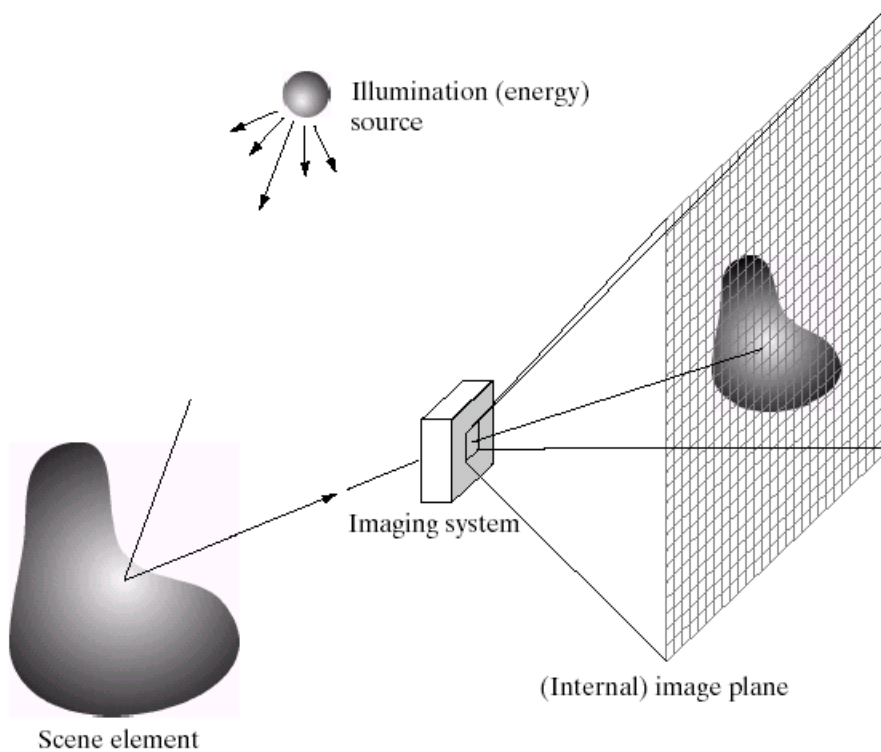
- Image Formation
- Review of Linear Algebra and Geometric Transformation
 - Basics definitions and properties
 - Geometrical transformations
 - Application: removing perspective distortion – the DLT algorithm
- Next Class

What is an image?



<http://media.photobucket.com/image/best%20construction%20image/Joeyjordanfan16/burj%2520dubai%25202/115oi1.jpg>

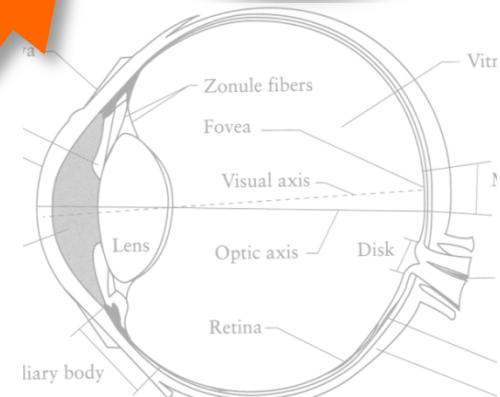
What is an image?



Digital Camera

We'll focus on these in this class

(More on this process later)



The Eye

Images

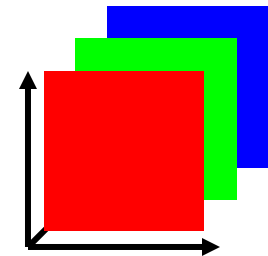
- Discrete representation of a continuous function
 - Each image is a two dimensional array of pixels
 - Pixel: Element of a picture – cell of constant color in a digital image (i.e., numeric value representing a uniform portion of an image)

- Grayscale
 - All pixels represent the intensity of light in an image, be it **red**, **green**, **blue**, or another color
 - Similar to holding a piece of transparent colored plastic over your eyes
 - Intensity of light in a pixel is stored as a number, generally 0..255 inclusive

Images (Cont'd)

■ Color

- Three grayscale images layered on top of each other with each layer indicating the intensity of a specific color light, generally red, green, and blue (RGB)
- Third dimension in a digital image

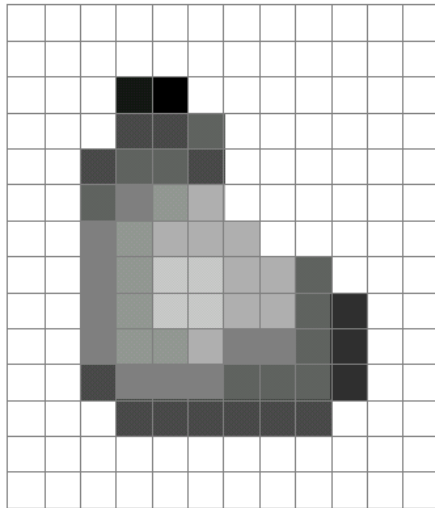


■ Resolution

- Number of pixels across in horizontal
- Number of pixels in the vertical
- Number of layers used for color
 - Often measured in bits per pixel (bpp) where each color uses 8 bits of data
- Ex: 640x480x24bpp

Images (Cont'd)

- A grid (matrix) of intensity values



=

255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	20	0	255	255	255	255	255	255	255
255	255	255	75	75	75	255	255	255	255	255	255
255	255	75	95	95	75	255	255	255	255	255	255
255	255	96	127	145	175	255	255	255	255	255	255
255	255	127	145	175	175	175	255	255	255	255	255
255	255	127	145	200	200	175	175	95	255	255	255
255	255	127	145	200	200	175	175	95	47	255	255
255	255	127	145	145	175	127	127	95	47	255	255
255	255	74	127	127	127	95	95	95	47	255	255
255	255	255	74	74	74	74	74	74	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255

(common to use one byte per value: 0 = black, 255 = white)

Color Image

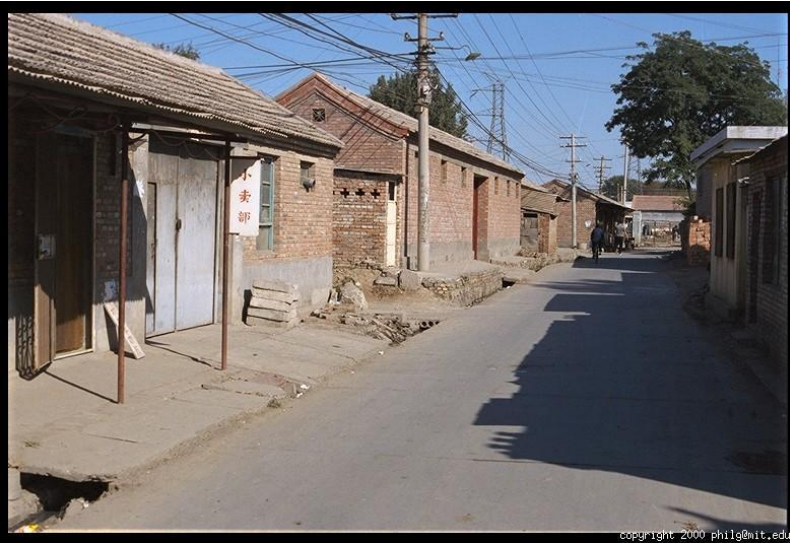
R



G

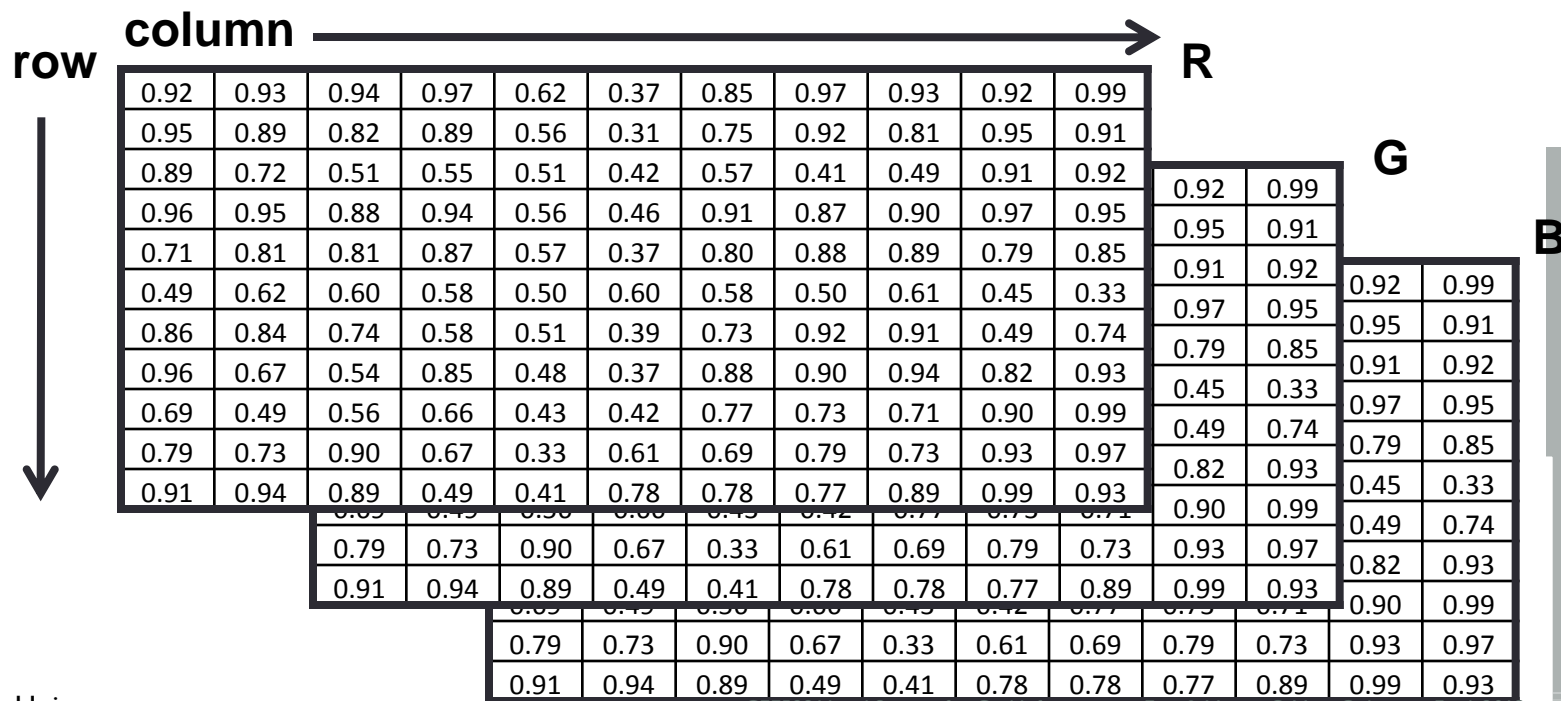


B



Images in Matlab

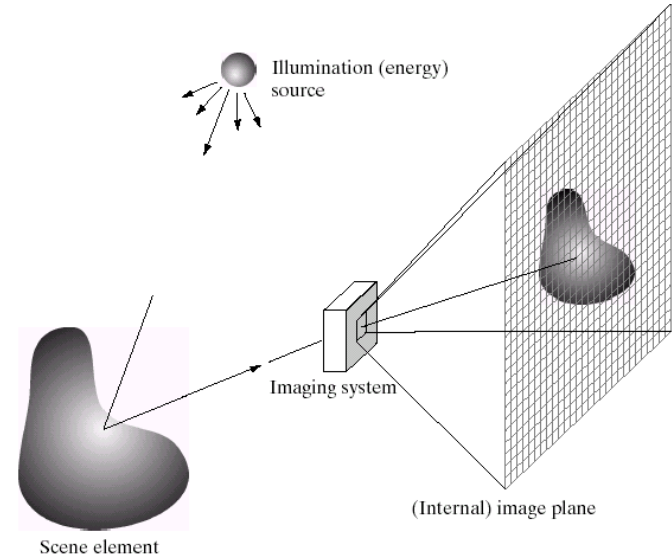
- Images represented as a matrix
- Suppose we have a NxM RGB image called “im”
 - $im(1,1,1)$ = top-left pixel value in R-channel
 - $im(y, x, b)$ = y pixels down, x pixels to right in the bth channel
 - $im(N, M, 3)$ = bottom-right pixel in 3-channel
- **imread(filename)** returns a uint8 image (values 0 to 255)
 - Convert to double format (values 0 to 1) with **im2double**



Why is linear algebra useful in computer vision?

■ Representation

- 3D points in the scene
- 2D points in the image



■ Coordinates will be used to

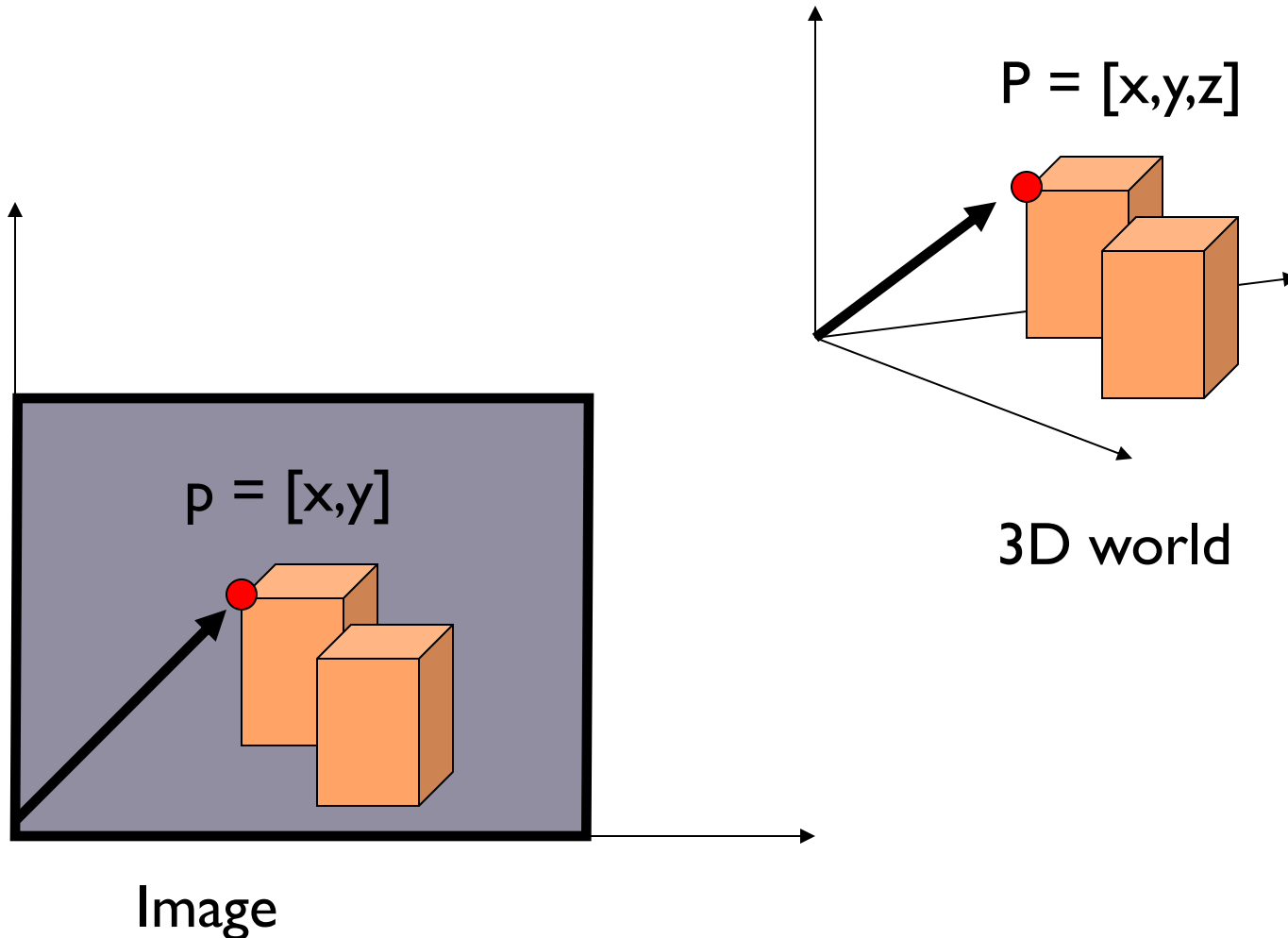
- Perform geometrical transformations
- Associate 3D with 2D points

■ Images are matrices of numbers

- Find properties of these numbers

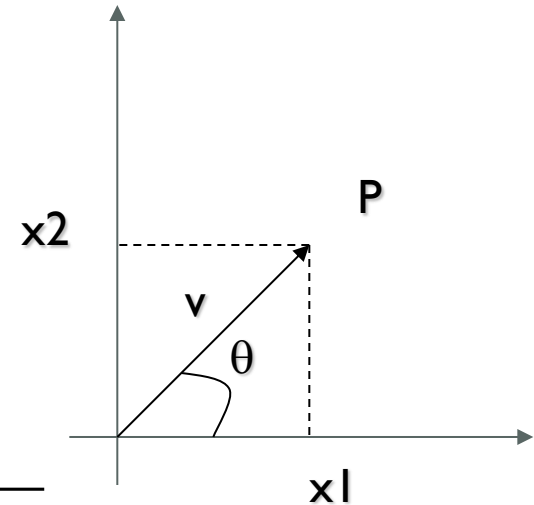
Linear Algebra

Vectors (i.e., 2D or 3D vectors)



Vectors (i.e., 2D vectors)

$$\mathbf{v} = (x_1, x_2)$$



Magnitude: $\|\mathbf{v}\| = \sqrt{x_1^2 + x_2^2}$

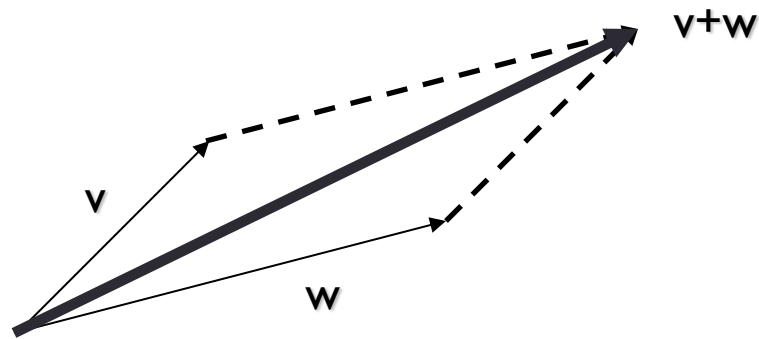
If $\|\mathbf{v}\| = 1$, \mathbf{v} is a UNIT vector

$$\frac{\mathbf{v}}{\|\mathbf{v}\|} = \left(\frac{x_1}{\|\mathbf{v}\|}, \frac{x_2}{\|\mathbf{v}\|} \right) \text{ is a unit vector}$$

Orientation: $\theta = \tan^{-1} \left(\frac{x_2}{x_1} \right)$

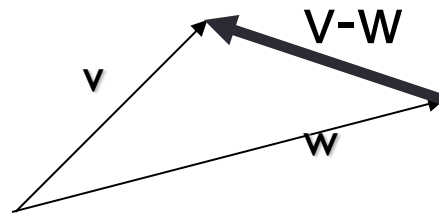
Vector Addition

$$\mathbf{v} + \mathbf{w} = (x_1, x_2) + (y_1, y_2) = (x_1 + y_1, x_2 + y_2)$$



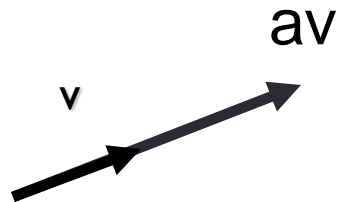
Vector Subtraction

$$\mathbf{v} - \mathbf{w} = (x_1, x_2) - (y_1, y_2) = (x_1 - y_1, x_2 - y_2)$$

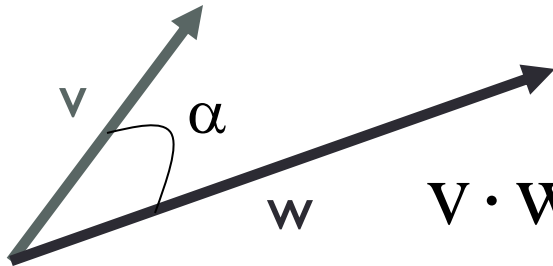


Scalar Product

$$a\mathbf{v} = a(x_1, x_2) = (ax_1, ax_2)$$



Inner (dot) Product



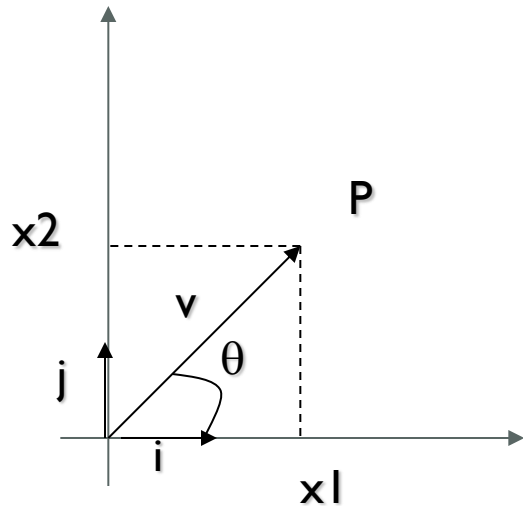
$$v \cdot w = (x_1, x_2) \cdot (y_1, y_2) = x_1 y_1 + x_2 y_2$$

The inner product is a **SCALAR!**

$$v \cdot w = (x_1, x_2) \cdot (y_1, y_2) = \|v\| \cdot \|w\| \cos \alpha$$

$$\text{if } v \perp w, \quad v \cdot w = ? = 0$$

Orthonormal Basis



$$\begin{aligned}\mathbf{i} &= (1,0) & \|\mathbf{i}\| &= 1 & \mathbf{i} \cdot \mathbf{j} &= 0 \\ \mathbf{j} &= (0,1) & \|\mathbf{j}\| &= 1 & & \end{aligned}$$

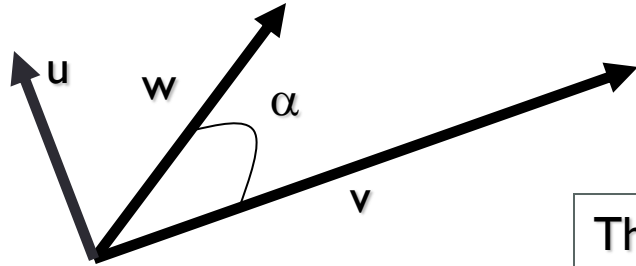
$$\mathbf{v} = (x_1, x_2)$$

$$\mathbf{v} = x_1 \mathbf{i} + x_2 \mathbf{j}$$

$$\mathbf{v} \cdot \mathbf{i} = ? = (x_1 \mathbf{i} + x_2 \mathbf{j}) \cdot \mathbf{i} = x_1 \cdot 1 + x_2 \cdot 0 = x_1$$

$$\mathbf{v} \cdot \mathbf{j} = (x_1 \mathbf{i} + x_2 \mathbf{j}) \cdot \mathbf{j} = x_1 \cdot 0 + x_2 \cdot 1 = x_2$$

Vector (cross) Product



$$u = v \times w$$

The cross product is a **VECTOR!**

Magnitude: $\|u\| = \|v \times w\| = \|v\| \|w\| \sin \alpha$

Orientation: $u \perp v \Rightarrow u \cdot v = (v \times w) \cdot v = 0$
 $u \perp w \Rightarrow u \cdot w = (v \times w) \cdot w = 0$

if $v \parallel w$? $\rightarrow u = 0$

Vector Product Computation

$$\mathbf{i} = (1,0,0) \quad \|\mathbf{i}\| = 1$$

$$\mathbf{j} = (0,1,0) \quad \|\mathbf{j}\| = 1 \quad \mathbf{i} \cdot \mathbf{j} = 0 \quad \mathbf{i} \cdot \mathbf{k} = 0 \quad \mathbf{j} \cdot \mathbf{k} = 0$$

$$\mathbf{k} = (0,0,1) \quad \|\mathbf{k}\| = 1$$

$$\begin{aligned} \mathbf{u} &= \mathbf{v} \times \mathbf{w} = (x_1, x_2, x_3) \times (y_1, y_2, y_3) \\ &= (x_2 y_3 - x_3 y_2) \mathbf{i} + (x_3 y_1 - x_1 y_3) \mathbf{j} + (x_1 y_2 - x_2 y_1) \mathbf{k} \end{aligned}$$

Matrices

$$A_{n \times m} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ a_{31} & a_{32} & \cdots & a_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$



Pixel's intensity value

$$\text{Sum: } C_{n \times m} = A_{n \times m} + B_{n \times m} \quad c_{ij} = a_{ij} + b_{ij}$$

A and B must have the same dimensions!

$$\text{Example: } \begin{bmatrix} 2 & 5 \\ 3 & 1 \end{bmatrix} + \begin{bmatrix} 6 & 2 \\ 1 & 5 \end{bmatrix} = \begin{bmatrix} 8 & 7 \\ 4 & 6 \end{bmatrix}$$

Matrices

$$A_{n \times m} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ a_{31} & a_{32} & \cdots & a_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \quad \mathbf{a}_i$$

$$B_{n \times m} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ b_{31} & b_{32} & \cdots & b_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{bmatrix} \quad \mathbf{b}_j$$

Product:

$$C_{n \times p} = A_{n \times m} B_{m \times p}$$

$$c_{ij} = \mathbf{a}_i \cdot \mathbf{b}_j = \sum_{k=1}^m a_{ik} b_{kj}$$

A and B must have compatible dimensions!

$$A_{n \times n} B_{n \times n} \neq B_{n \times n} A_{n \times n}$$

Matrices

Transpose:

$$C_{m \times n} = A^T_{n \times m}$$

$$c_{ij} = a_{ji}$$

$$(A + B)^T = A^T + B^T$$

$$(AB)^T = B^T A^T$$

If $A^T = A$

A is symmetric

Examples:

$$\begin{bmatrix} 6 & 2 \\ 1 & 5 \end{bmatrix}^T = \begin{bmatrix} 6 & 1 \\ 2 & 5 \end{bmatrix}$$

$$\begin{bmatrix} 6 & 2 \\ 1 & 5 \\ 3 & 8 \end{bmatrix}^T = \begin{bmatrix} 6 & 1 & 3 \\ 2 & 5 & 8 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 2 \\ 1 & 5 \end{bmatrix}$$

Symmetric? **No!**

$$\begin{bmatrix} 3 & 2 \\ 2 & 7 \end{bmatrix}$$

Symmetric? **Yes!**

Matrices

Determinant:

$$\det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

$$\det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

A must be square

Example: $\det \begin{bmatrix} 2 & 5 \\ 3 & 1 \end{bmatrix} = 2 - 15 = -13$

Matrices

Inverse:

A must be square

$$A_{n \times n} A^{-1}_{n \times n} = A^{-1}_{n \times n} A_{n \times n} = I$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{1}{a_{11}a_{22} - a_{21}a_{12}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

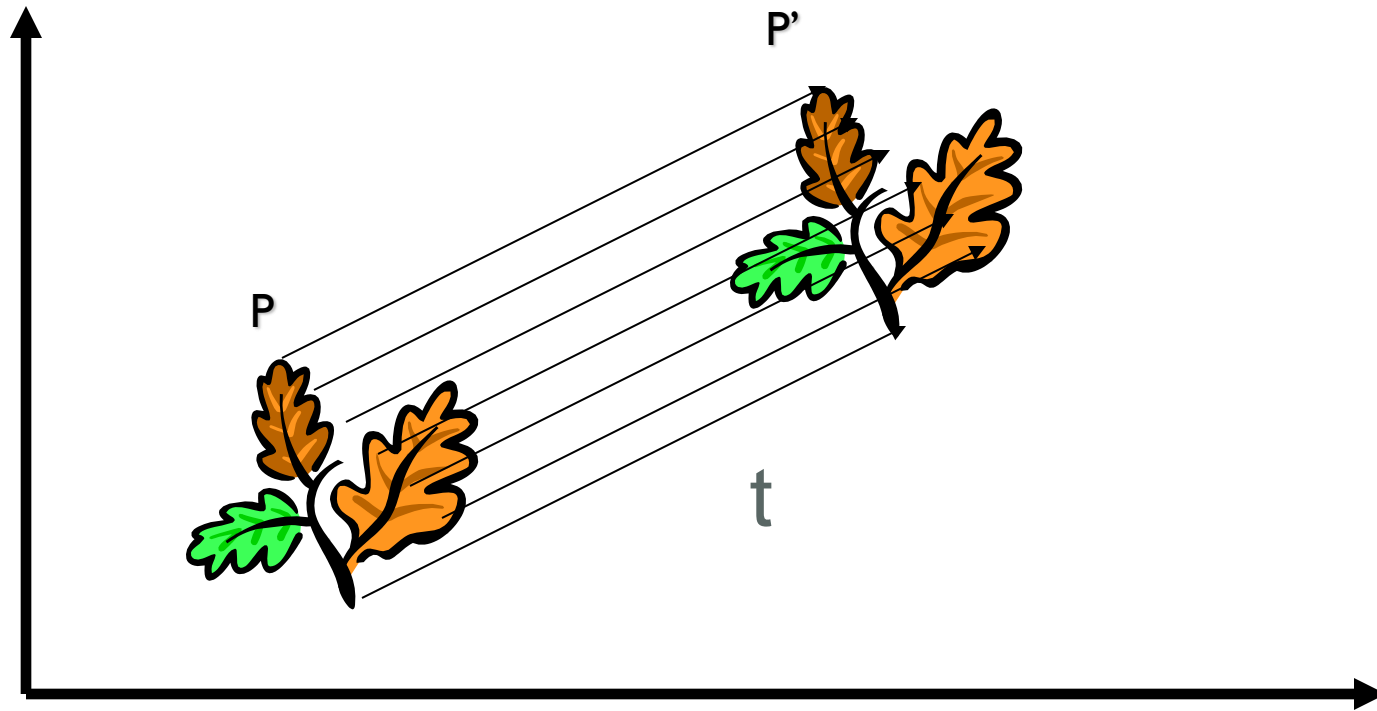
Example:

$$\begin{bmatrix} 6 & 2 \\ 1 & 5 \end{bmatrix}^{-1} = ? = \frac{1}{28} \begin{bmatrix} 5 & -2 \\ -1 & 6 \end{bmatrix}$$

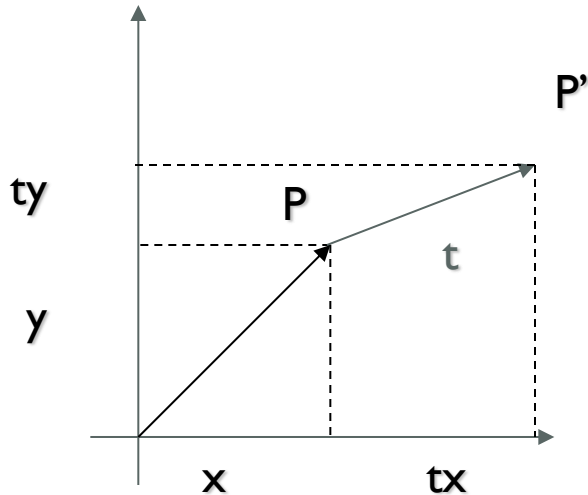
$$\begin{bmatrix} 6 & 2 \\ 1 & 5 \end{bmatrix}^{-1} \begin{bmatrix} 6 & 2 \\ 1 & 5 \end{bmatrix} = \frac{1}{28} \begin{bmatrix} 5 & -2 \\ -1 & 6 \end{bmatrix} \begin{bmatrix} 6 & 2 \\ 1 & 5 \end{bmatrix} = \frac{1}{28} \begin{bmatrix} 28 & 0 \\ 0 & 28 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

2D Geometrical Transformations

2D Translation



2D Translation Equation

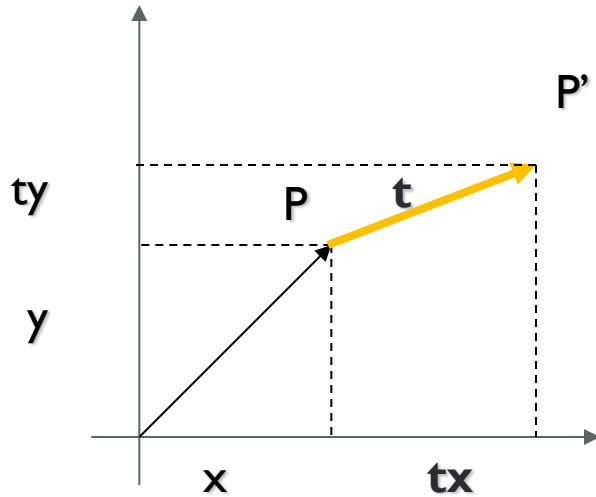


$$\mathbf{P} = (x, y)$$

$$\mathbf{t} = (t_x, t_y)$$

$$\mathbf{P}' = \mathbf{P} + \mathbf{t} = (x + t_x, y + t_y)$$

2D Translation using Matrices



$$\mathbf{P} = (x, y)$$

$$\mathbf{t} = (t_x, t_y)$$

Homogeneous
Coordinates

$$\mathbf{P}' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates

- Multiply the coordinates by a non-zero scalar and add an extra coordinate equal to that scalar. For example,

$$(x, y) \rightarrow (x \cdot z, y \cdot z, z) \quad z \neq 0$$

$$(x, y, z) \rightarrow (x \cdot w, y \cdot w, z \cdot w, w) \quad w \neq 0$$

Back to Cartesian Coordinates

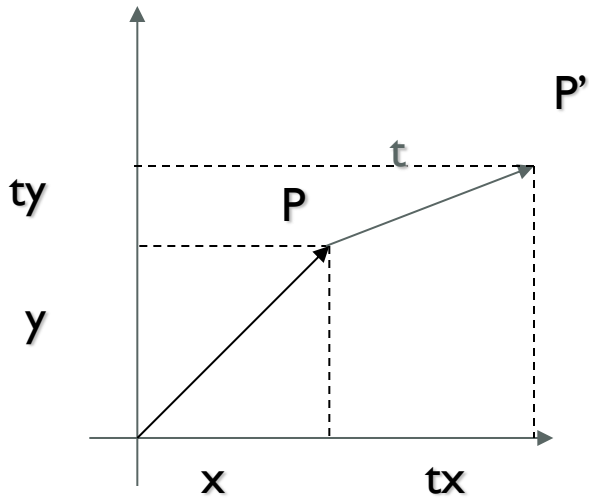
- Divide by the last coordinate and eliminate it. For example,

$$(x, y, z) \quad z \neq 0 \rightarrow (x/z, y/z)$$

$$(x, y, z, w) \quad w \neq 0 \rightarrow (x/w, y/w, z/w)$$

- NOTE: in our example the scalar was 1

2D Translation using Homogeneous Coordinates



$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

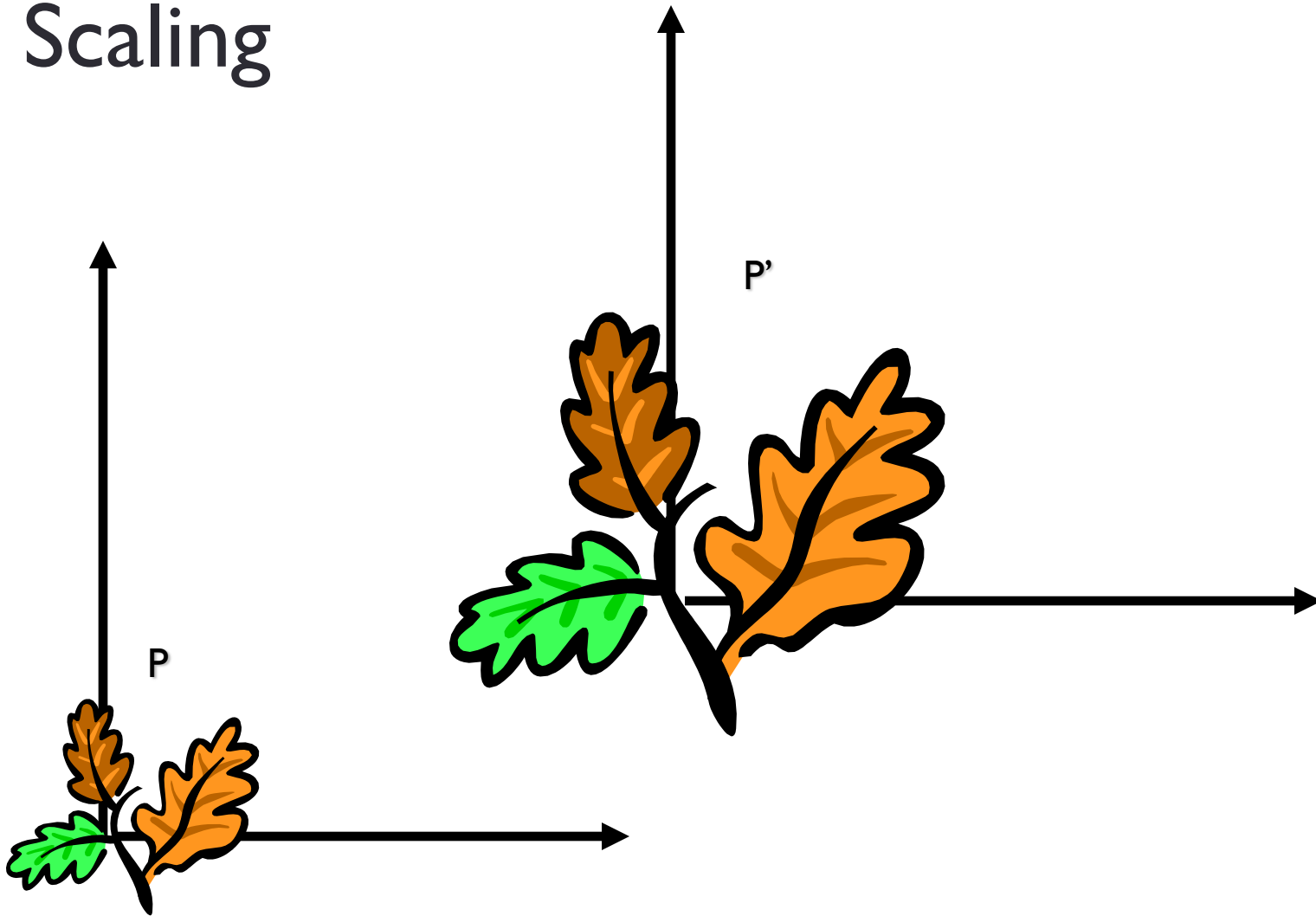
$$\mathbf{t} = (t_x, t_y) \rightarrow (t_x, t_y, 1)$$

$$\mathbf{P}' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

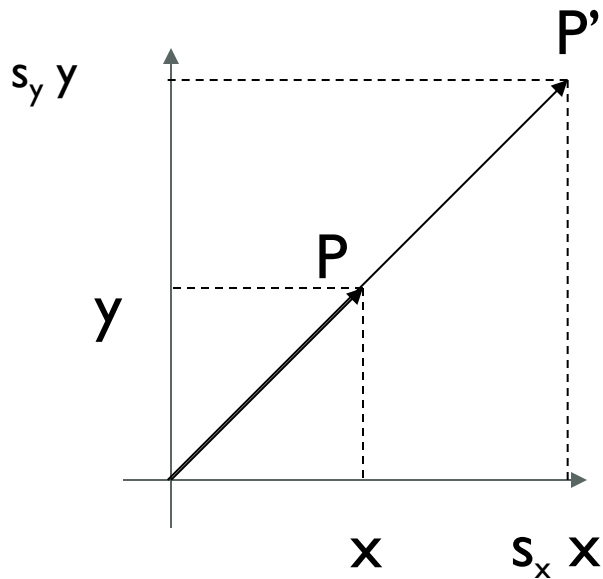
$$= \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \mathbf{P} = \mathbf{T} \cdot \mathbf{P}$$

The diagram shows the matrix multiplication for 2D translation. The translation vector \mathbf{t} is shown as a dashed box around the t_x and t_y elements in the translation matrix. The point \mathbf{P} is shown as a dashed box around the x , y , and 1 elements in the point vector.

Scaling



Scaling Equation



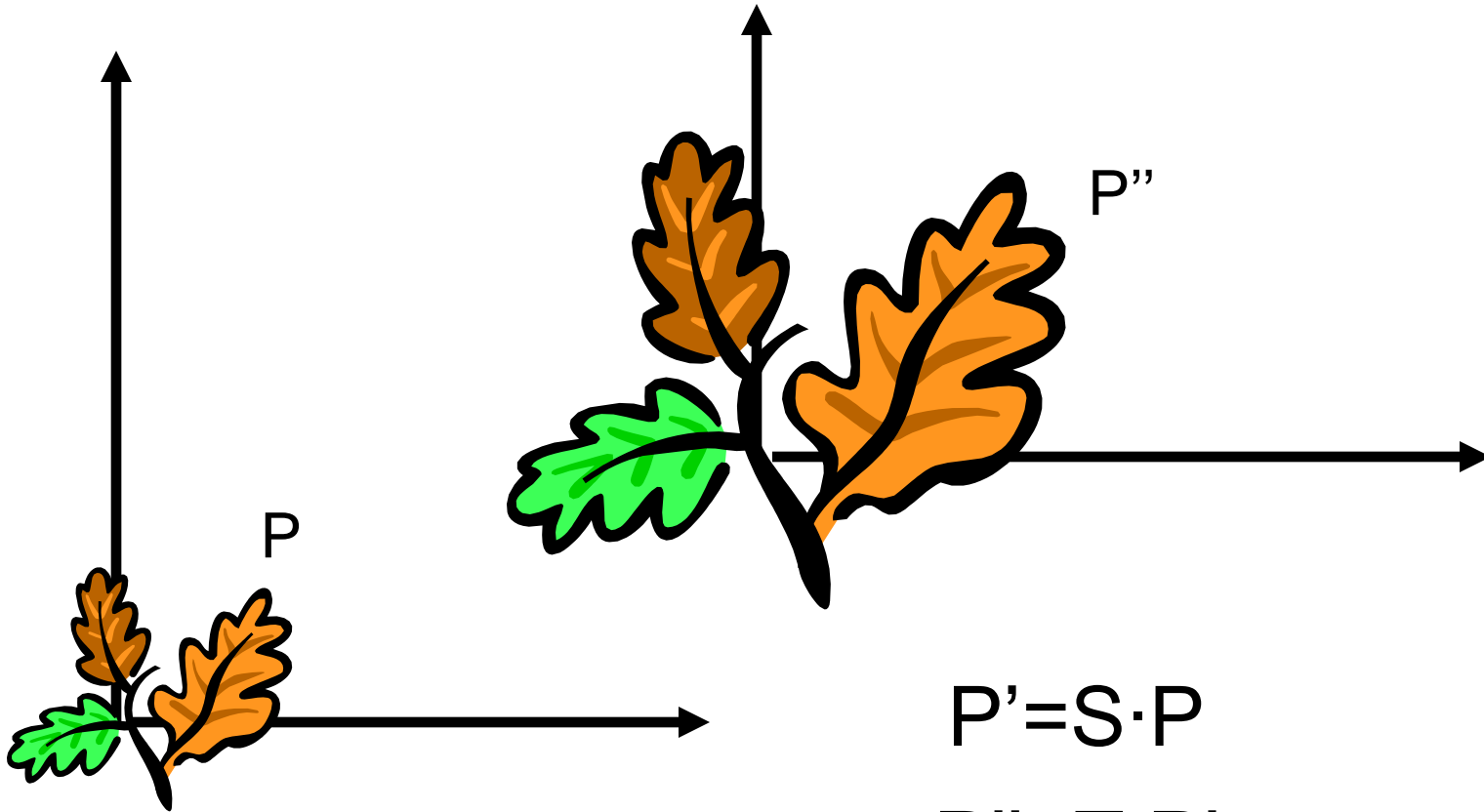
$$\mathbf{P} = (x, y) \rightarrow \mathbf{P}' = (s_x x, s_y y)$$

$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{P}' = (s_x x, s_y y) \rightarrow (s_x x, s_y y, 1)$$

$$\mathbf{P}' \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S}} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{S}' & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \mathbf{P} = \mathbf{S} \cdot \mathbf{P}$$

Scaling & Translating



$$P' = S \cdot P$$

$$P'' = T \cdot P'$$

$$P'' = T \cdot P' = T \cdot (S \cdot P) = (T \cdot S) \cdot P = A \cdot P$$

Scaling & Translating

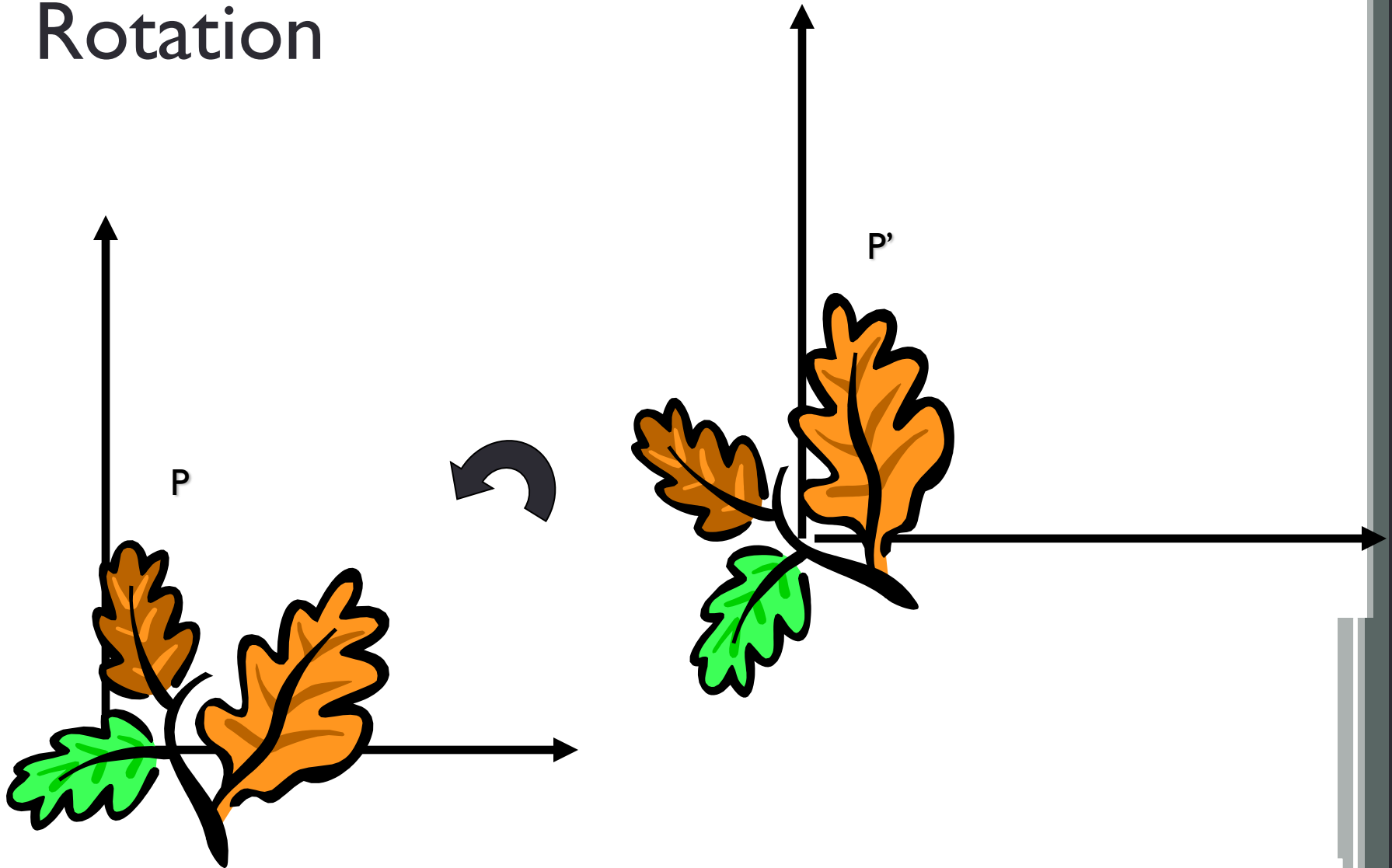
$$\begin{aligned} \mathbf{P}'' = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} &= \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \\ &= \underbrace{\begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}}_A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} S & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix} \end{aligned}$$

Translating & Scaling = Scaling & Translating ?

$$\mathbf{P}''' = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix}$$

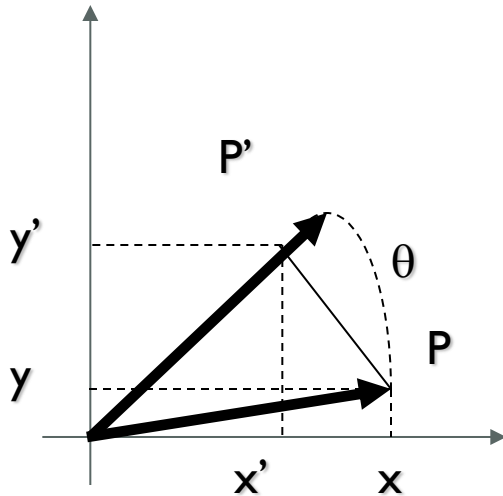
$$\mathbf{P}''' = \mathbf{S} \cdot \mathbf{T} \cdot \mathbf{P} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$
$$= \begin{bmatrix} s_x & 0 & s_x t_x \\ 0 & s_y & s_y t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + s_x t_x \\ s_y y + s_y t_y \\ 1 \end{bmatrix}$$

Rotation



Rotation Equations

Counter-clockwise rotation by an angle θ



$$x' = \cos \theta x - \sin \theta y$$

$$y' = \cos \theta y + \sin \theta x$$

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

$$\mathbf{P}' = \mathbf{R} \mathbf{P}$$

Degrees of Freedom

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

R is 2x2



4 elements

Note: R belongs to the category of *normal* matrices and satisfies many interesting properties:

$$\mathbf{R} \cdot \mathbf{R}^T = \mathbf{R}^T \cdot \mathbf{R} = \mathbf{I}$$

$$\det(\mathbf{R}) = 1$$

Rotation+ Scaling + Translation

$$P' = (T R S) P$$

$$P' = T \cdot R \cdot S \cdot P = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

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

$$= \begin{bmatrix} R' & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \boxed{\begin{bmatrix} R' S & t \\ 0 & 1 \end{bmatrix}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

If $s_x = s_y$, this is a similarity transformation!

Transformation in 2D

- Isometries
- Similarities
- Affinity
- Projective

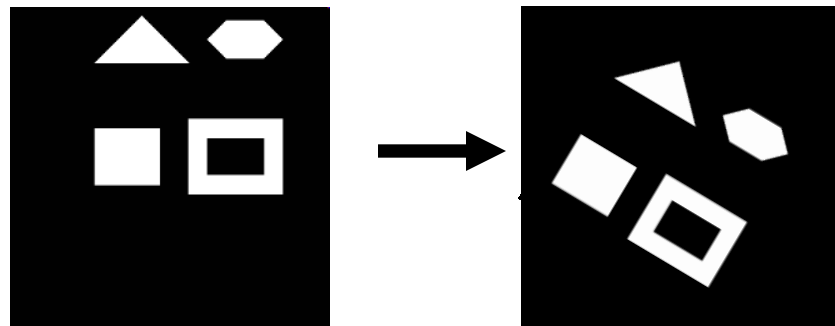
Transformation in 2D

Isometries:

[Euclidean]

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{H}_e \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Preserve distance (areas)
- 3 DOF
- Regulate motion of rigid object

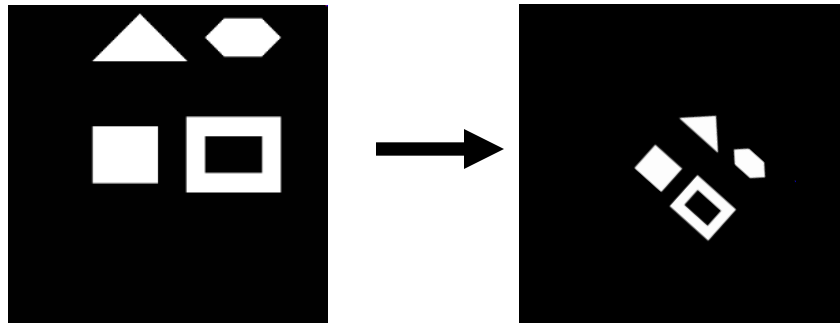


Transformation in 2D

Similarities:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s & \mathbf{R} & \mathbf{t} \\ & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{H}_s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Preserve
 - ratio of lengths
 - angles
- 4 DOF

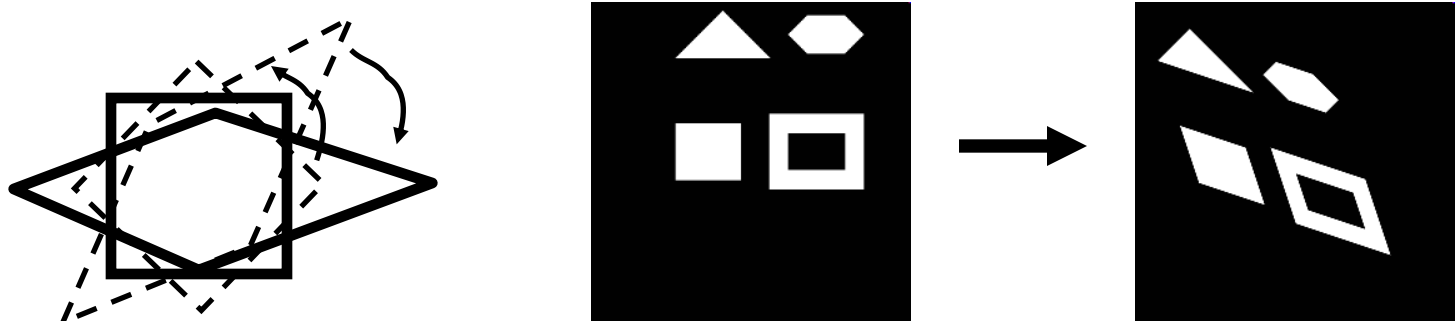


Transformation in 2D

Affinities:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_a \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = R(\theta) \cdot R(-\phi) \cdot D \cdot R(\phi) \quad D = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$



Transformation in 2D

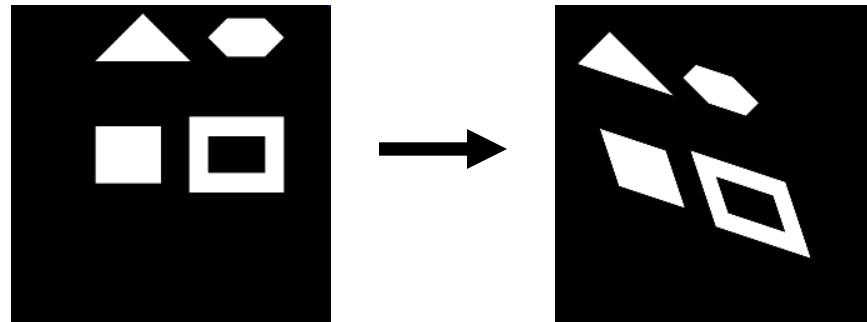
Affinities:
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_a \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = R(\theta) \cdot R(-\phi) \cdot D \cdot R(\phi) \quad D = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

-Preserve:

- Parallel lines
- Ratio of areas
- Ratio of lengths on collinear lines
- others...

- 6 DOF



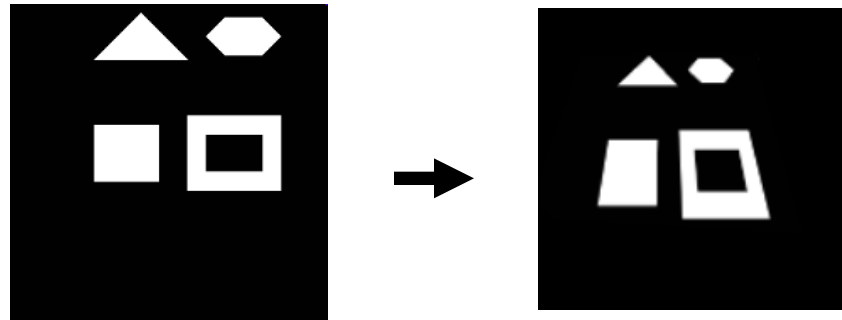
Transformation in 2D

Projective:
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ \mathbf{v} & b \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{H}_p \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

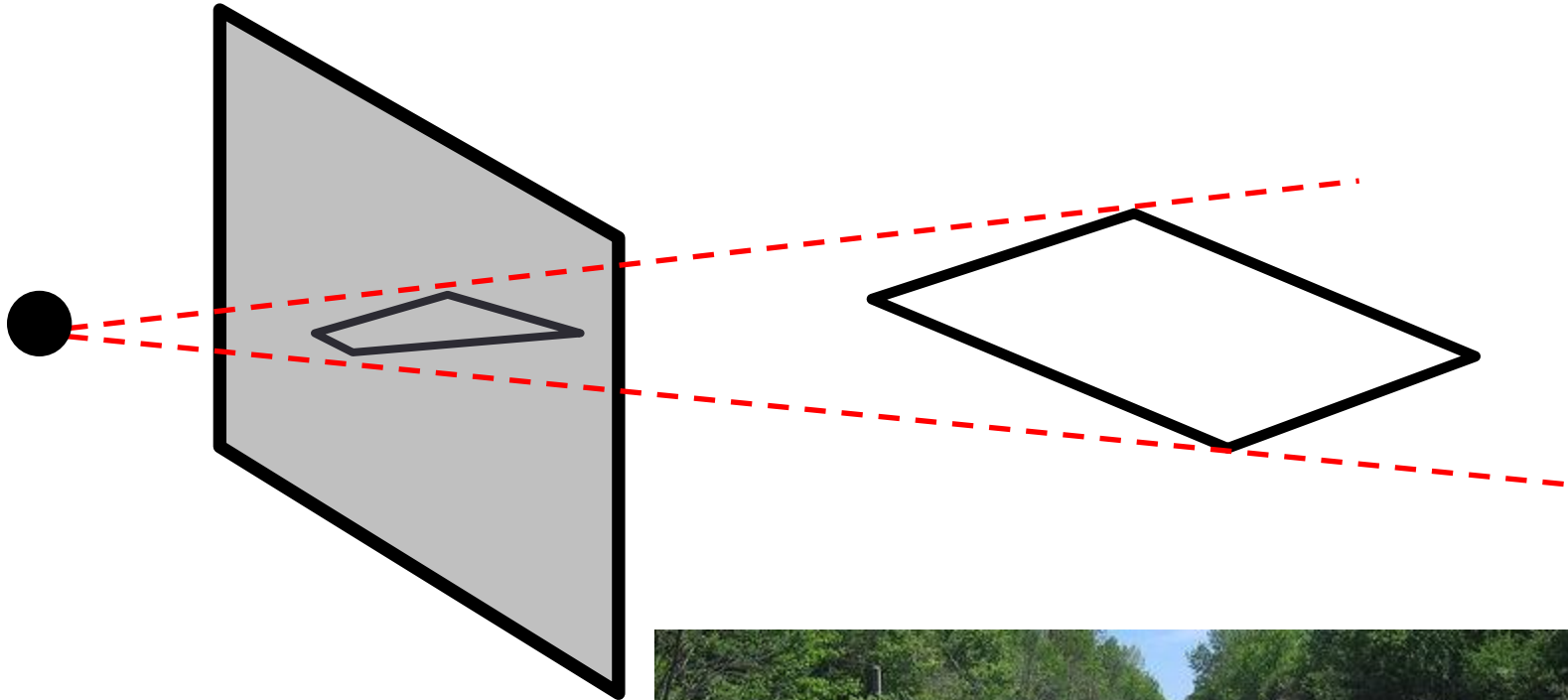
-Preserve:

- cross ratio of 4 collinear points
- collinearity
- and a few others...

- 8 DOF

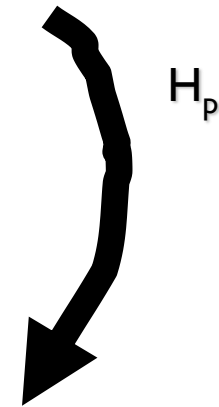


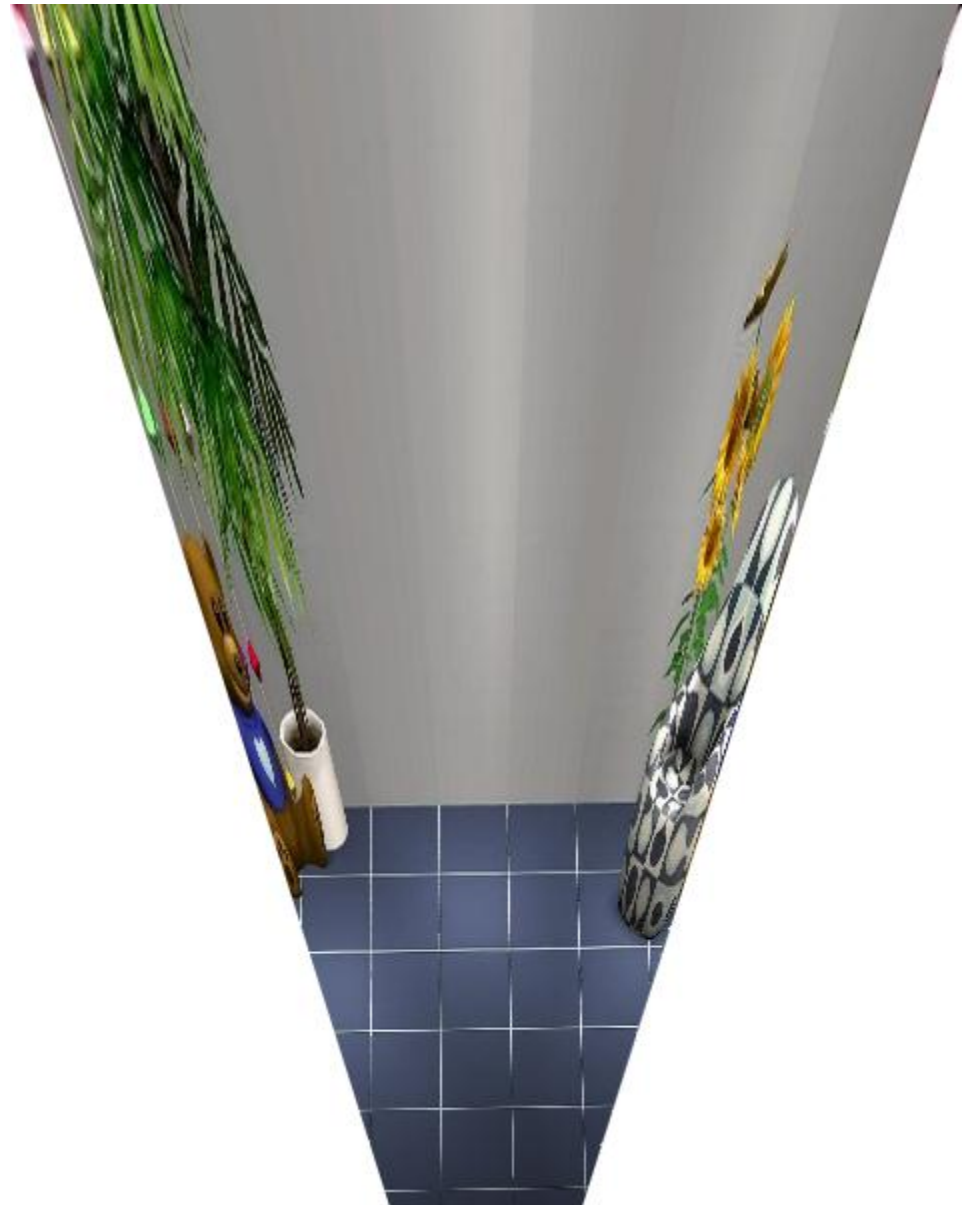
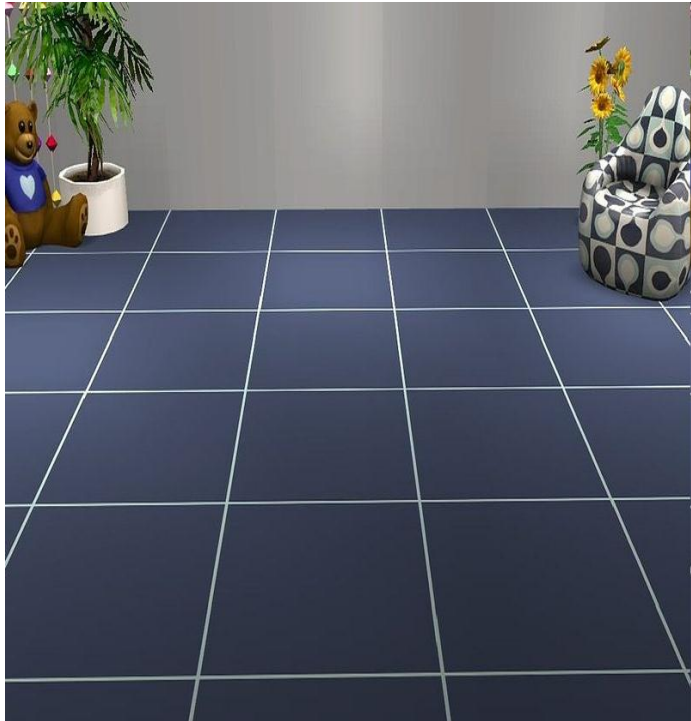
Transformation in 2D



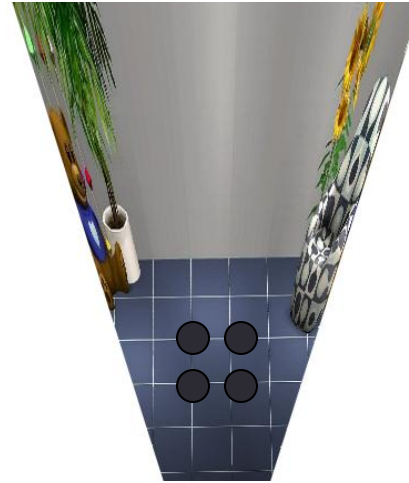
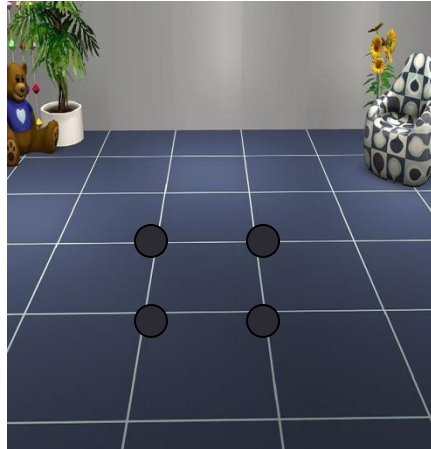
Removing Perspective Distortion

(rectification)





Computing H_p

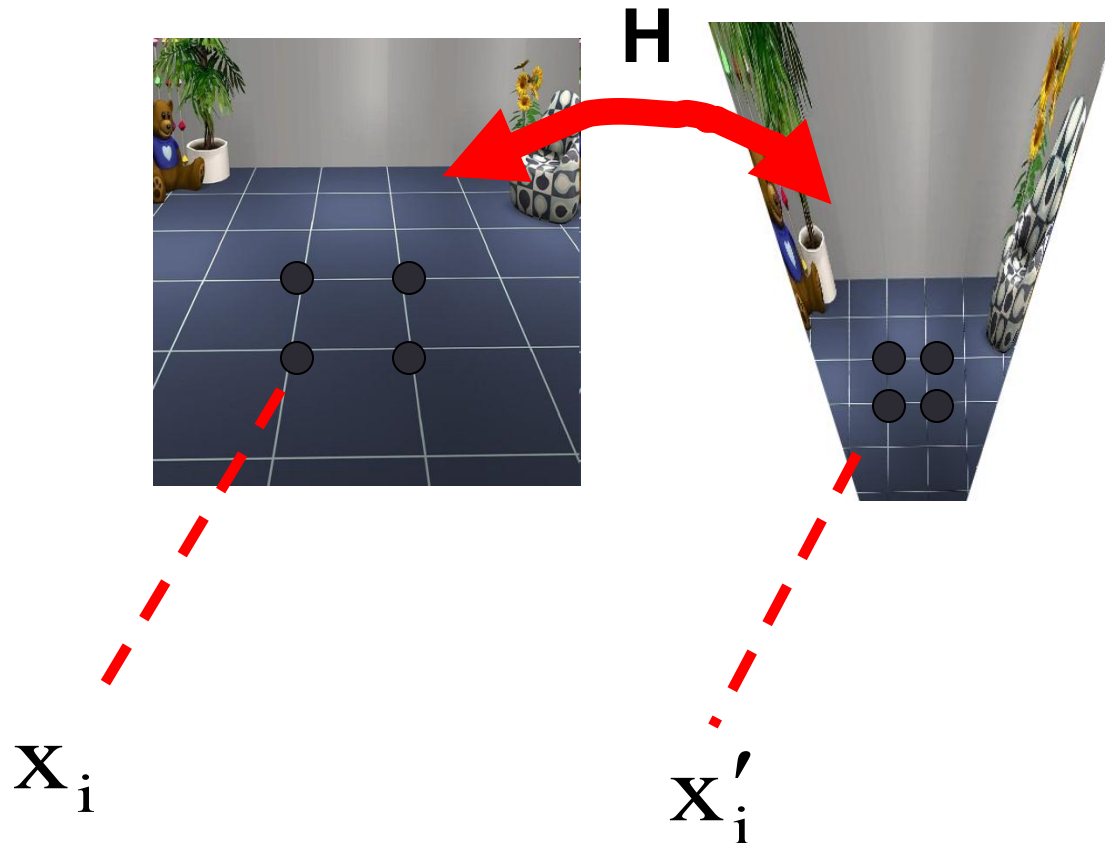


- 8 DOF
- how many points do I need to estimate H_p ?

At least 4 points! (8 equations)

- There are several algorithms...

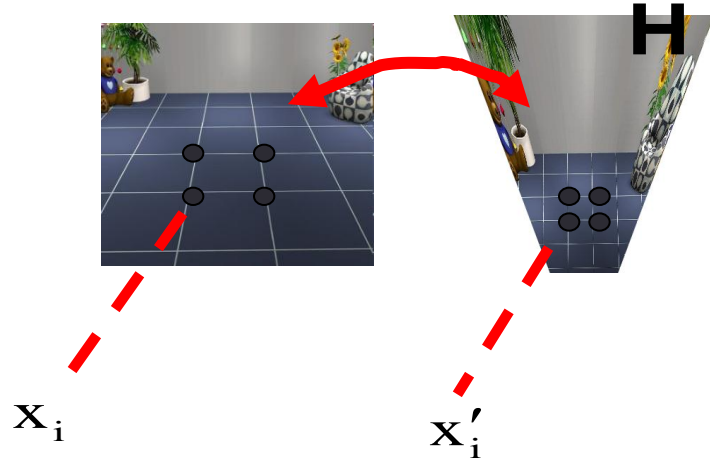
DLT algorithm (Direct Linear Transformation)



$$x'_i = H x_i$$

DLT algorithm (Direct Linear Transformation)

$$\begin{matrix} A_{2 \times 9} & h_{9 \times 1} \\ \boxed{A_i} & \boxed{h} \end{matrix} = 0$$



$$\left\{ \begin{array}{l} A_1 h = 0 \\ A_2 h = 0 \\ \vdots \\ A_N h = 0 \end{array} \right. \rightarrow A_{2N \times 9} h_{9 \times 1} = 0$$

Over determined
Homogenous system

DLT algorithm (Direct Linear Transformation)

How to solve $\mathbf{A}_{2N \times 9} \mathbf{h}_{9 \times 1} = \mathbf{0}$?

Singular Value Decomposition (SVD)!

Eigenvalues and Eigenvectors

- Eigen relation

$$\mathbf{A}\mathbf{u}=\lambda\mathbf{u}$$

- Matrix \mathbf{A} acts on vector \mathbf{u} and produces a scaled version of the vector.
- Eigen is a German word meaning “proper” or “specific”
- \mathbf{u} is the eigenvector while λ is the eigenvalue.

Eigenvalues and Eigenvectors

The eigenvalues of A are the roots of the *characteristic equation*

$$p(\lambda) = \det(\lambda I - A) = 0$$

$$\lambda_1 \cdots \lambda_N \quad S = [\mathbf{v}_1 \quad \mathbf{v}_N]$$

$$S^{-1}AS = \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \cdot & \\ & & & \lambda_N \end{bmatrix} \quad \text{diagonal form of matrix}$$

Eigenvectors of A are columns of S

Singular Value Decomposition

- **Singular values:** Non negative square roots of the eigenvalues of $\mathbf{A}^t\mathbf{A}$. Denoted $\sigma_i, i=1, \dots, n$
- **SVD:** If \mathbf{A} is a real m by n matrix then there exist orthogonal matrices \mathbf{U} ($\in \mathbb{R}^{m \times m}$) and \mathbf{V} ($\in \mathbb{R}^{n \times n}$) such that

$$\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^{-1} \quad \mathbf{U}^{-1}\mathbf{A}\mathbf{V} = \Sigma = \begin{bmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \cdot & & \\ & & & & \sigma_N \end{bmatrix}$$

DLT algorithm (direct Linear Transformation)

How to solve $\mathbf{A}_{2N \times 9} \mathbf{h}_{9 \times 1} = \mathbf{0}$?

Singular Value Decomposition (SVD)!



$$\mathbf{U}_{2n \times 9} \mathbf{D}_{9 \times 9} \mathbf{V}^T_{9 \times 9}$$

Last column of \mathbf{V} gives \mathbf{h} ! $\rightarrow \mathbf{H}$!

DLT algorithm (direct Linear Transformation)

How to solve $\mathbf{A}_{2N \times 9} \mathbf{h}_{9 \times 1} = \mathbf{0}$?

$[\mathbf{U}, \mathbf{D}, \mathbf{V}] = \text{SVD}(\mathbf{A});$
 $\mathbf{X} = \mathbf{V}(:, \text{end});$



Marilia Maschion

Producer & Director

UCSD ICAM undergraduate

Vincent Rabaud

Technical Advisor

UCSD CSE student

Serge Belongie

Project Advisor

UCSD CSE Department

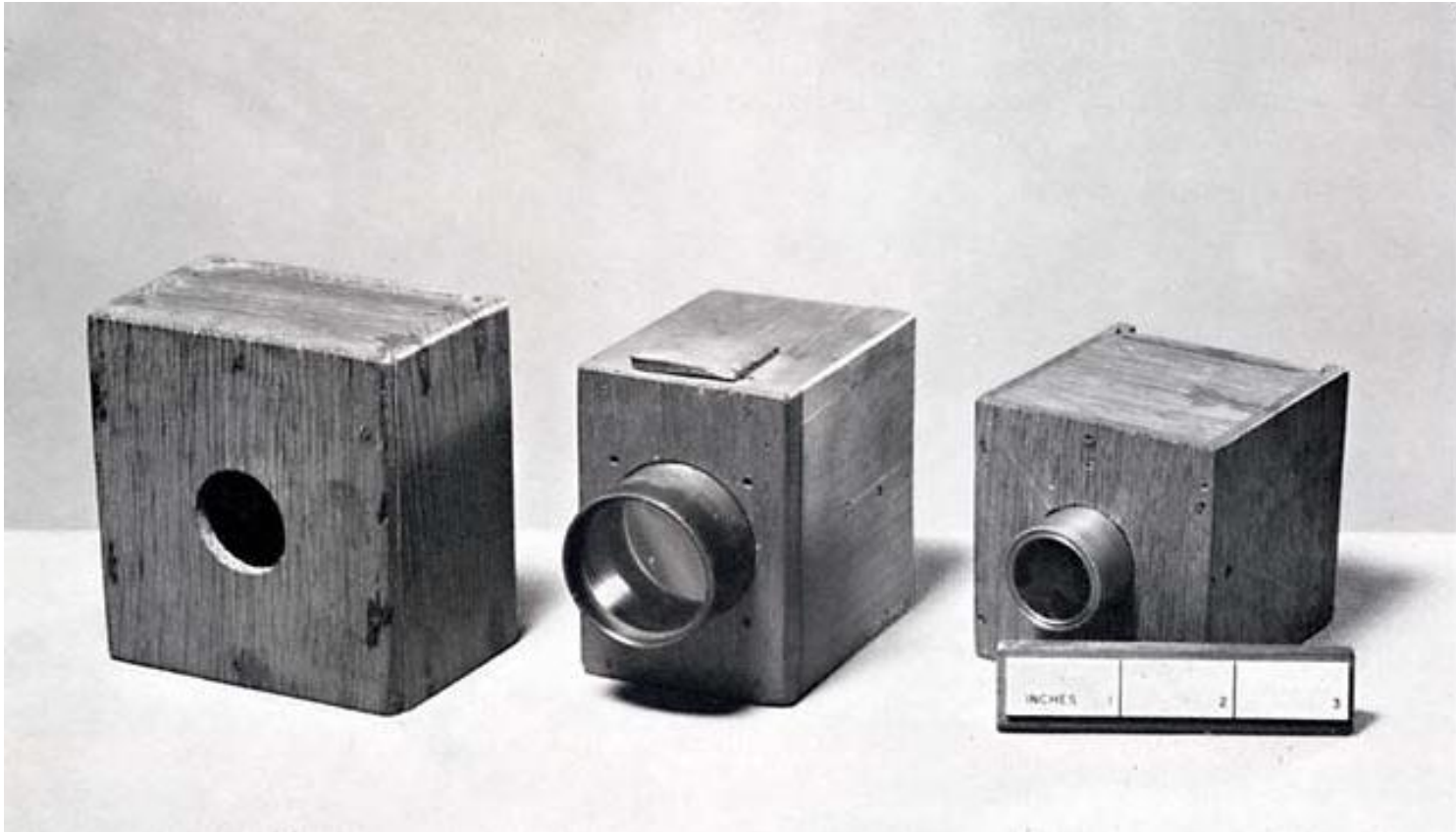
A00: watch

[Illinois Compass2g Site](#)> [Course Contents](#)> [Supplementary Documents](#)> [3DVision.avi](#)

A0: review SVD decomposition

Next lecture

Cameras models



Appendix

- **Properties of SVD**

Properties of the SVD

- Suppose we know the singular values of \mathbf{A} and we know r are non zero

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq \sigma_{r+1} = \dots = \sigma_p = 0$$

- Rank(\mathbf{A}) = r .
- Null(\mathbf{A}) = span $\{\mathbf{v}_{r+1}, \dots, \mathbf{v}_n\}$
- Range(\mathbf{A}) = span $\{\mathbf{u}_1, \dots, \mathbf{u}_r\}$
- $\|\mathbf{A}\|_F^2 = \sigma_1^2 + \sigma_2^2 + \dots + \sigma_p^2$ $\|\mathbf{A}\|_2 = \sigma_1$
- *Numerical rank*: If k singular values of A are larger than a given number ε . Then the ε rank of A is k .
- Distance of a matrix of rank n from being a matrix of rank $k = \sigma_{k+1}$

Why is it useful?

- Square matrix may be singular due to round-off errors.
Can compute a “regularized” solution

–
$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^t)^{-1}\mathbf{b} = \sum_{i=1}^n \frac{\mathbf{u}_i^t \mathbf{b}}{\sigma_i} \mathbf{v}_i$$

- If σ_i is small (vanishes) the solution “blows up”
- Given a tolerance ε we can determine a solution that is “closest” to the solution of the original equation, but that does not “blow up”

$$\mathbf{x}_r = \sum_{i=1}^k \frac{\mathbf{u}_i^t \mathbf{b}}{\sigma_i} \mathbf{v}_i \quad \sigma_k > \varepsilon, \quad \sigma_{k+1} \leq \varepsilon$$

- Least squares solution is the \mathbf{x} that satisfies
$$\mathbf{A}^t \mathbf{A} \mathbf{x} = \mathbf{A}^t \mathbf{b}$$
- can be effectively solved using SVD

Appendix:

DLT algorithm (Direct Linear Transformation)

From:

Multiple View Geometry in Computer Vision,
by R. Hartley and A. Zisserman, Academic Press, 2002

4.1 The Direct Linear Transformation (DLT) algorithm

We begin with a simple linear algorithm for determining H given a set of four 2D to 2D point correspondences, $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. The transformation is given by the equation $\mathbf{x}'_i = H\mathbf{x}_i$. Note that this is an equation involving homogeneous vectors; thus the 3-vectors \mathbf{x}'_i and $H\mathbf{x}_i$ are not equal, they have the same direction but may differ in magnitude by a non-zero scale factor. The equation may be expressed in terms of the vector cross product as $\mathbf{x}'_i \times H\mathbf{x}_i = \mathbf{0}$. This form will enable a simple linear solution for H to be derived.

If the j -th row of the matrix H is denoted by $\mathbf{h}^{j\top}$, then we may write

$$H\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1\top} \mathbf{x}_i \\ \mathbf{h}^{2\top} \mathbf{x}_i \\ \mathbf{h}^{3\top} \mathbf{x}_i \end{pmatrix}.$$

Writing $\mathbf{x}'_i = (x'_i, y'_i, w'_i)^\top$, the cross product may then be given explicitly as

$$\mathbf{x}'_i \times H\mathbf{x}_i = \begin{pmatrix} y'_i \mathbf{h}^{3\top} \mathbf{x}_i - w'_i \mathbf{h}^{2\top} \mathbf{x}_i \\ w'_i \mathbf{h}^{1\top} \mathbf{x}_i - x'_i \mathbf{h}^{3\top} \mathbf{x}_i \\ x'_i \mathbf{h}^{2\top} \mathbf{x}_i - y'_i \mathbf{h}^{1\top} \mathbf{x}_i \end{pmatrix}.$$

Since $\mathbf{h}^{j\top} \mathbf{x}_i = \mathbf{x}'_i{}^\top \mathbf{h}^j$ for $j = 1, \dots, 3$, this gives a set of three equations in the entries of H , which may be written in the form

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i \mathbf{x}_i^\top & y'_i \mathbf{x}_i^\top \\ w'_i \mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i \mathbf{x}_i^\top \\ -y'_i \mathbf{x}_i^\top & x'_i \mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}. \quad (4.1)$$

These equations have the form $A_i \mathbf{h} = \mathbf{0}$, where A_i is a 3×9 matrix, and \mathbf{h} is a 9-vector made up of the entries of the matrix H ,

$$\mathbf{h} = \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix}, \quad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \quad (4.2)$$

with h_i the i -th element of \mathbf{h} . Three remarks regarding these equations are in order here.

- (i) The equation $A_i \mathbf{h} = \mathbf{0}$ is an equation *linear* in the unknown \mathbf{h} . The matrix elements of A_i are quadratic in the known coordinates of the points.
- (ii) Although there are three equations in (4.1), only two of them are linearly independent (since the third row is obtained, up to scale, from the sum of x'_i times the first row and y'_i times the second). Thus each point correspondence gives two equations in the entries of H . It is usual to omit the third equation in solving for H ([Sutherland-63]). Then (for future reference) the set of equations becomes

$$\begin{bmatrix} \mathbf{0}^T & -w'_i \mathbf{x}'_i{}^T & y'_i \mathbf{x}'_i{}^T \\ w'_i \mathbf{x}'_i{}^T & \mathbf{0}^T & -x'_i \mathbf{x}'_i{}^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = \mathbf{0}. \quad (4.3)$$

This will be written

$$A_i \mathbf{h} = \mathbf{0}$$

where A_i is now the 2×9 matrix of (4.3).

- (iii) The equations hold for any homogeneous coordinate representation $(x'_i, y'_i, w'_i)^T$ of the point \mathbf{x}'_i . One may choose $w'_i = 1$, which means that (x'_i, y'_i) are the coordinates measured in the image. Other choices are possible, however, as will be seen later.

Solving for H

Each point correspondence gives rise to two independent equations in the entries of H. Given a set of four such point correspondences, we obtain a set of equations $A\mathbf{h} = \mathbf{0}$, where A is the matrix of equation coefficients built from the matrix rows A_i contributed from each correspondence, and \mathbf{h} is the vector of unknown entries of H. We seek a non-zero solution \mathbf{h} , since the obvious solution $\mathbf{h} = \mathbf{0}$ is of no interest to us. If (4.1) is used then A has dimension 12×9 , and if (4.3) the dimension is 8×9 . In either case A has rank 8, and thus has a 1-dimensional null-space which provides a solution for \mathbf{h} . Such a solution \mathbf{h} can only be determined up to a non-zero scale factor. However, H is in general only determined up to scale, so the solution \mathbf{h} gives the required H. A scale may be arbitrarily chosen for \mathbf{h} by a requirement on its norm such as $\|\mathbf{h}\| = 1$.

4.1.2 Inhomogeneous solution

An alternative to solving for \mathbf{h} directly as a homogeneous vector is to turn the set of equations (4.3) into a inhomogeneous set of linear equations by imposing a condition $h_j = 1$ for some entry of the vector \mathbf{h} . Imposing the condition $h_j = 1$ is justified by the observation that the solution is determined only up to scale, and this scale can be chosen such that $h_j = 1$. For example, if the last element of \mathbf{h} , which corresponds to H_{33} , is chosen as unity then the resulting equations derived from (4.3) are

$$\begin{bmatrix} 0 & 0 & 0 & -x_i w'_i & -y_i w'_i & -w_i w'_i & x_i y'_i & y_i y'_i \\ x_i w'_i & y_i w'_i & w_i w'_i & 0 & 0 & 0 & -x_i x'_i & -y_i x'_i \end{bmatrix} \tilde{\mathbf{h}} = \begin{pmatrix} -w_i y'_i \\ w_i x'_i \end{pmatrix}$$

where $\tilde{\mathbf{h}}$ is an 8-vector consisting of the first 8 components of \mathbf{h} . Concatenating the equations from four correspondences then generates a matrix equation of the form