Course overview

1. Geometry
2. Low & Mid-level vision
3. High level vision

Course overview

1. Geometry
2. Low & Mid-level vision
3. High level vision

- How to extract 3d information?
- Which cues are useful?
- What are the mathematical tools?

Linear Algebra & Geometry

why is linear algebra useful in computer vision?

References:
-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

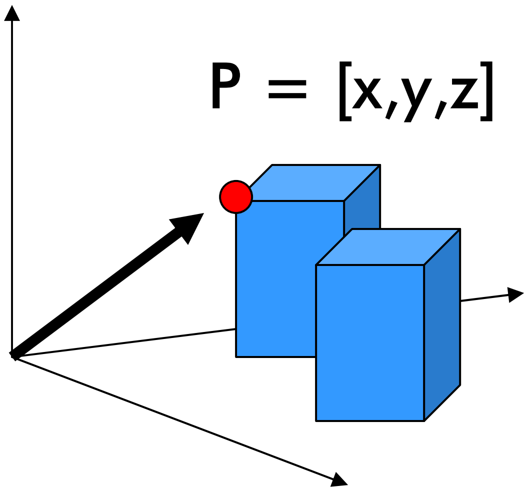
Why is linear algebra useful in computer vision?

- 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

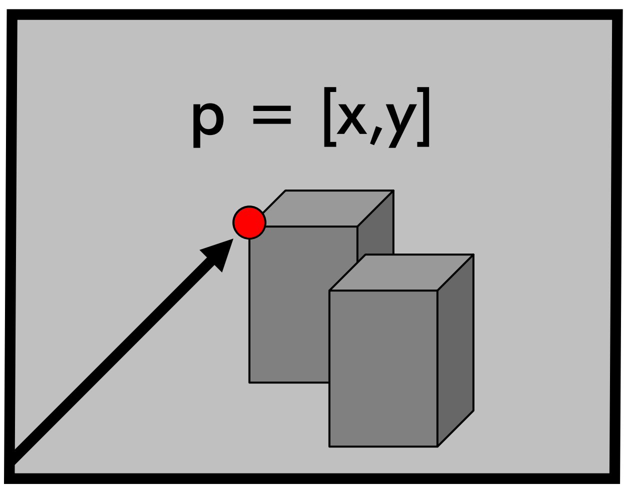
Agenda

1. How did you like the movie? ☺
2. Basics definitions and properties
3. Geometrical transformations
4. Application: removing perspective distortion

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



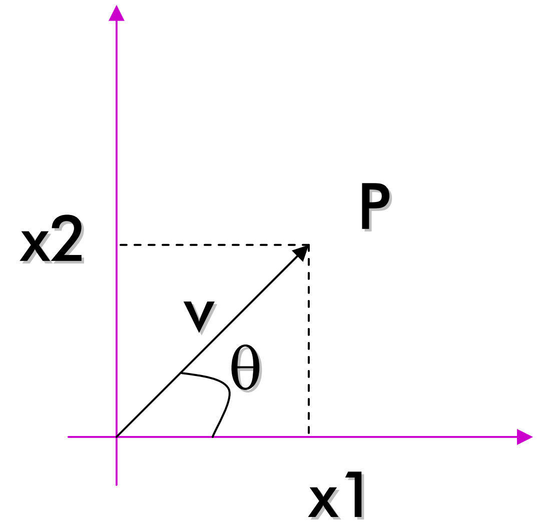
3D world



Image

Vectors (i.e., 2D vectors)

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



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

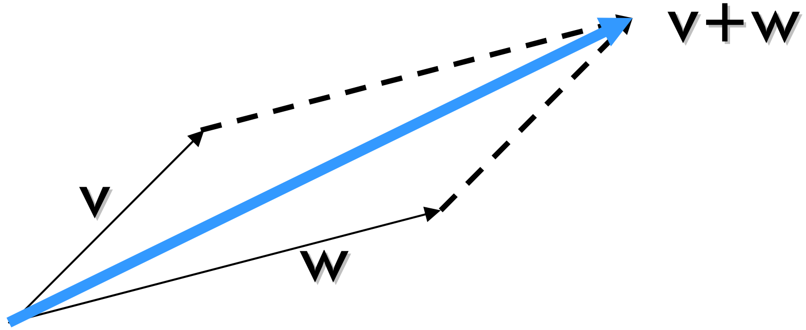
If $\| \mathbf{v} \| = 1$, $\mathbf{v}$ Is a UNIT vector

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

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

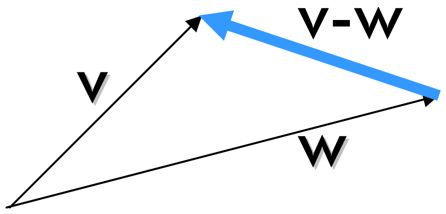
Vector Addition

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



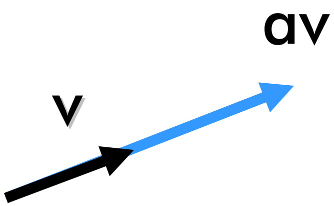
Vector Subtraction

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

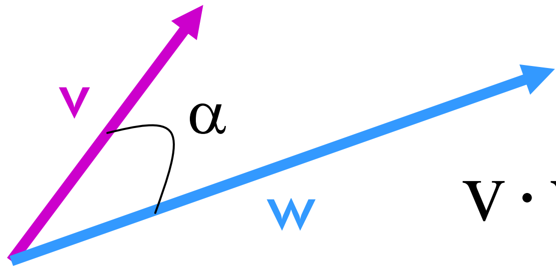


Scalar Product

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



Inner (dot) Product



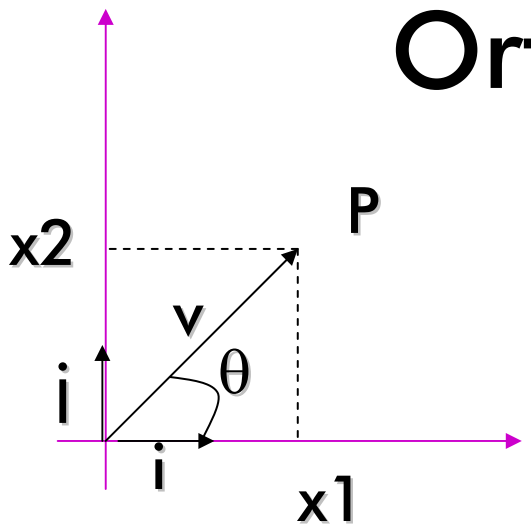
$$v \cdot w = (x_1, x_2) \cdot (y_1, y_2) = x_1 y_1 + x_2 y_2$$

The inner product is a SCALAR!

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

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

Orthonormal Basis



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

$$\mathbf{j} = (0,1) \qquad \|\mathbf{j}\| = 1$$

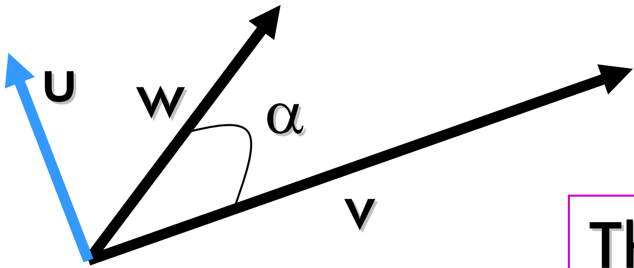
$$\mathbf{i} \cdot \mathbf{j} = 0$$

$$\mathbf{v} = (x_1, x_2) \qquad \mathbf{v} = \mathrm{x}_1\mathbf{i} + \mathrm{x}_2\mathbf{j}$$

$$\mathbf{v} \cdot \mathbf{i} = ? \;\; = (\mathrm{x}_1\mathbf{i} + \mathrm{x}_2\mathbf{j}) \cdot \mathbf{i} = \mathrm{x}_1 1 + \mathrm{x}_2 0 = \mathrm{x}_1$$

$$\mathbf{v} \cdot \mathbf{j} = (\mathrm{x}_1\mathbf{i} + \mathrm{x}_2\mathbf{j}) \cdot \mathbf{j} = \mathrm{x}_1.0 + \mathrm{x}_2.1 = \mathrm{x}_2$$

Vector (cross) Product



$$u = v \times w$$

The cross product is a VECTOR!

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

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 $\quad v // w \,?$ $\qquad \rightarrow u = 0$

Vector Product Computation

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

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

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

$$\mathbf{u} = \mathbf{v} \times \mathbf{w} = (x_1, x_2, x_3) \times (y_1, y_2, y_3)$$

$$= (x_2 y_3 - x_3 y_2)\mathbf{i} + (x_3 y_1 - x_1 y_3)\mathbf{j} + (x_1 y_2 - x_2 y_1)\mathbf{k}$$

Matrices

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

Pixel's intensity value

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!

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

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

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

$$\mathbf{b}_{\mathrm{j}}$$

Product:

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

$$\mathrm{c}_{\mathrm{ij}} = \mathbf{a}_{\mathrm{i}} \cdot \mathbf{b}_{\mathrm{j}} = \sum_{\mathrm{k}=1}^{\mathrm{m}} \mathrm{a}_{\mathrm{ik}} \mathrm{b}_{\mathrm{kj}}$$

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:

$$C_{m\times n} = A^T{}_{n\times m} \qquad (A+B)^T = A^T + B^T$$

$$c_{ij} = a_{ji} \qquad (AB)^T = B^T A^T$$

If $\quad A^T = A \quad$ 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

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

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

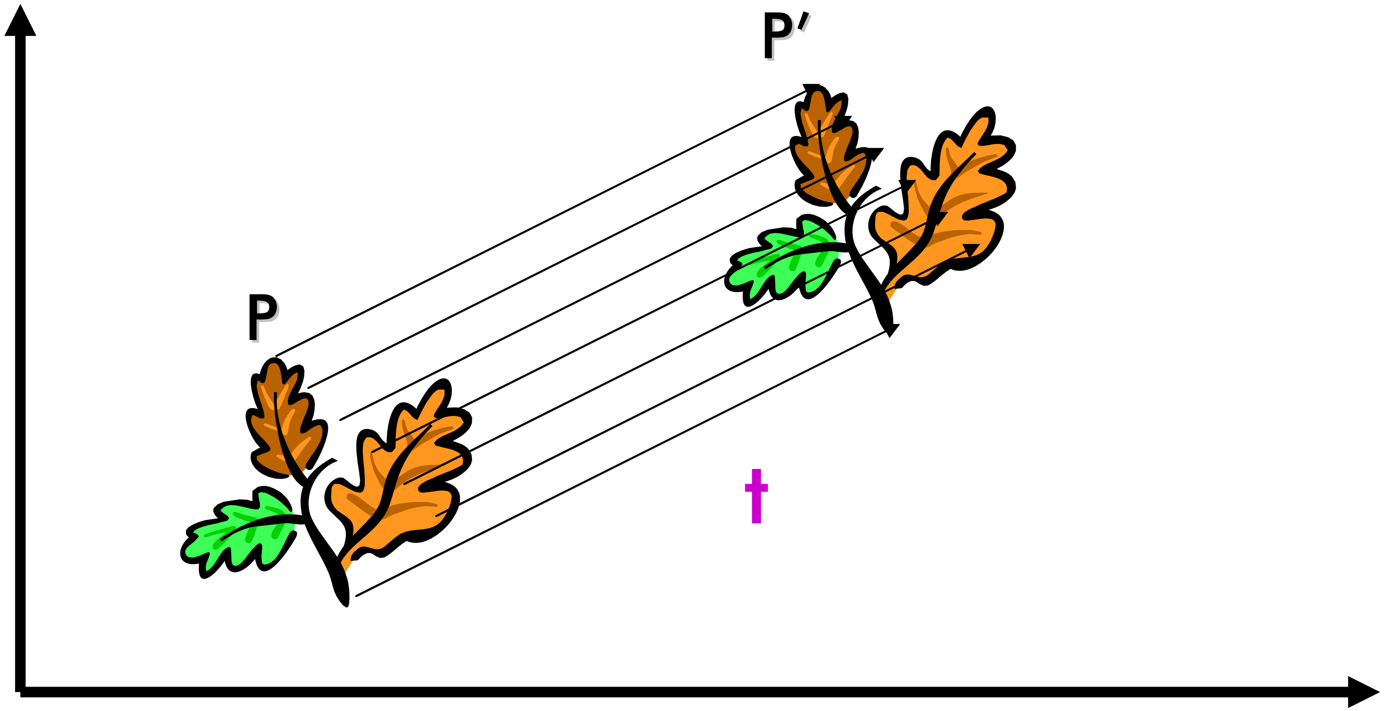
Example:

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

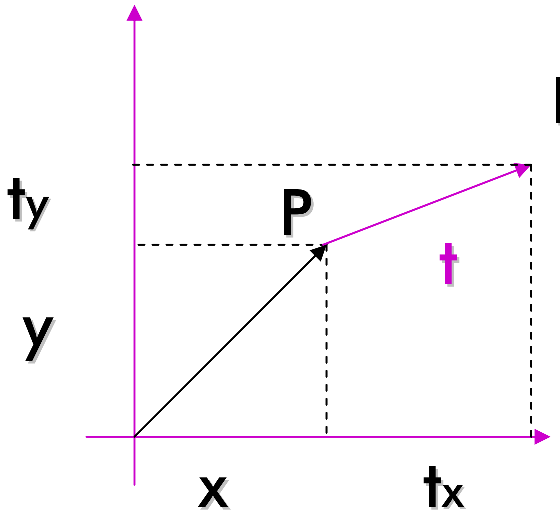
$$\begin{bmatrix} 6 & 2 \\ 1 & 5 \end{bmatrix}^{-1}.\begin{bmatrix} 6 & 2 \\ 1 & 5 \end{bmatrix}=\frac{1}{28}\begin{bmatrix} 5 & -2 \\ -1 & 6 \end{bmatrix}.\begin{bmatrix} 6 & 2 \\ 1 & 5 \end{bmatrix}=\frac{1}{28}\begin{bmatrix} 28 & 0 \\ 0 & 28 \end{bmatrix}=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

2D Geometrical Transformations

2D Translation



2D Translation Equation

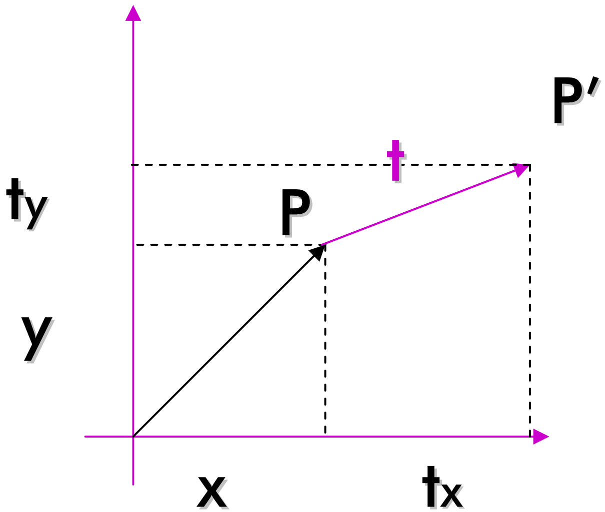


$$\mathbf{P} = (x, y)$$

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

$$\mathbf{P}' = \mathbf{P} + \mathbf{t} = (\mathrm{x} + \mathrm{t}_{\mathrm{x}}, \mathrm{y} + \mathrm{t}_{\mathrm{y}})$$

2D Translation using Matrices



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

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

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

Homogeneous Coordinates

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

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

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

Back to Cartesian Coordinates:

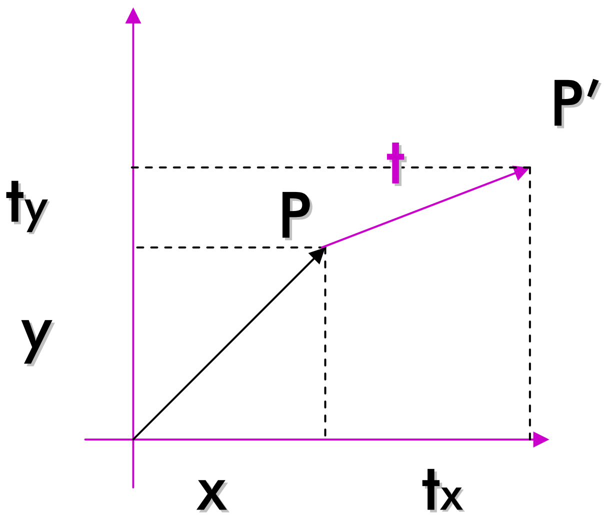
- Divide by the last coordinate and eliminate it. For example,

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

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

- NOTE: in our example the scalar was 1

2D Translation using Homogeneous Coordinates



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

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

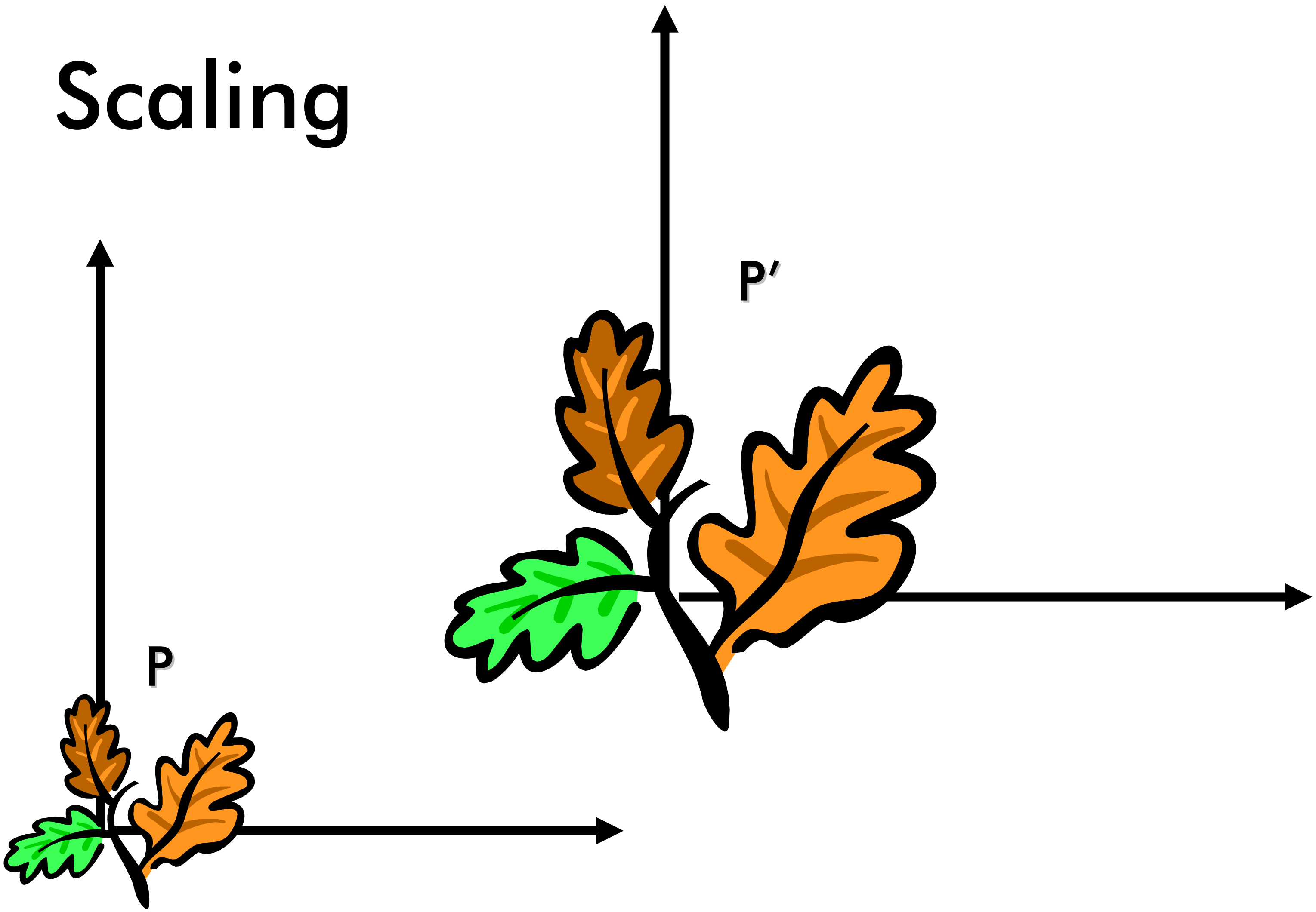
t

P

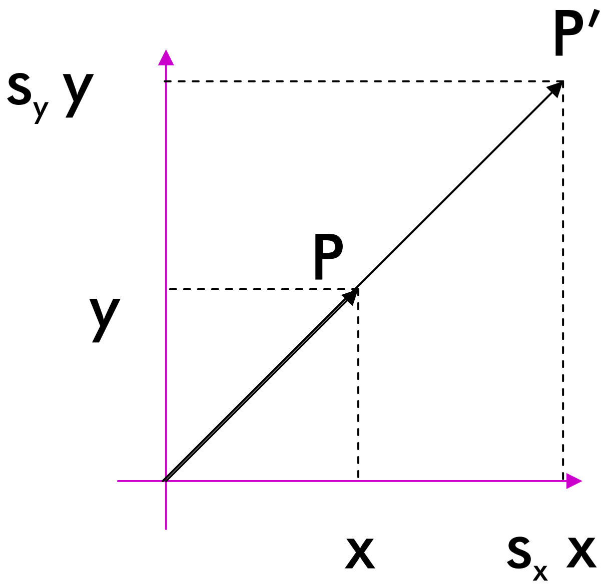
$$\mathbf{P}' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

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

Scaling



Scaling Equation



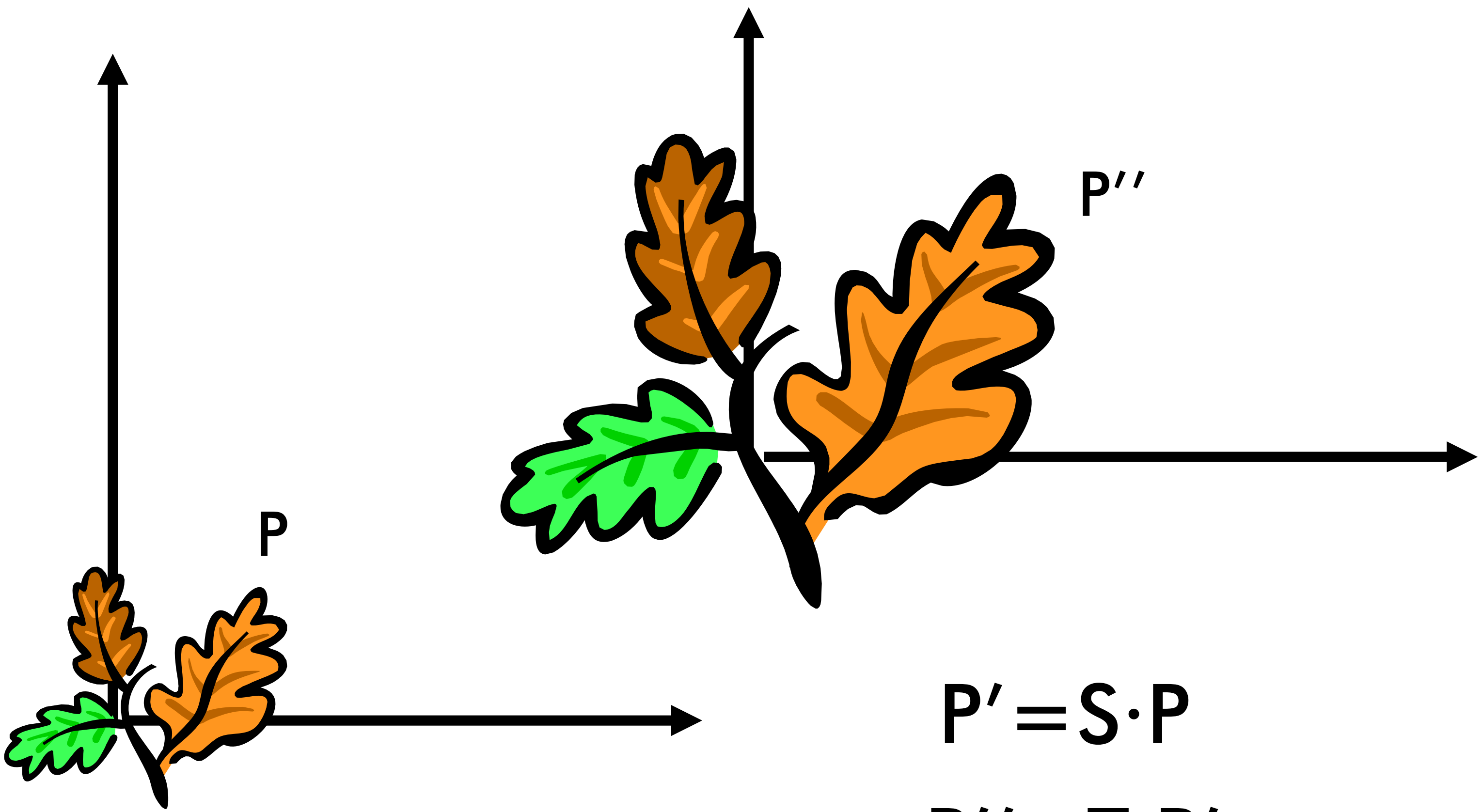
$$\mathbf{P} = (\mathrm{x}, \mathrm{y}) \rightarrow \mathbf{P}' = (\mathrm{s_x x}, \mathrm{s_y y})$$

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

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

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

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

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

$$= \underbrace{\begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathrm{A}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} S & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix}$$

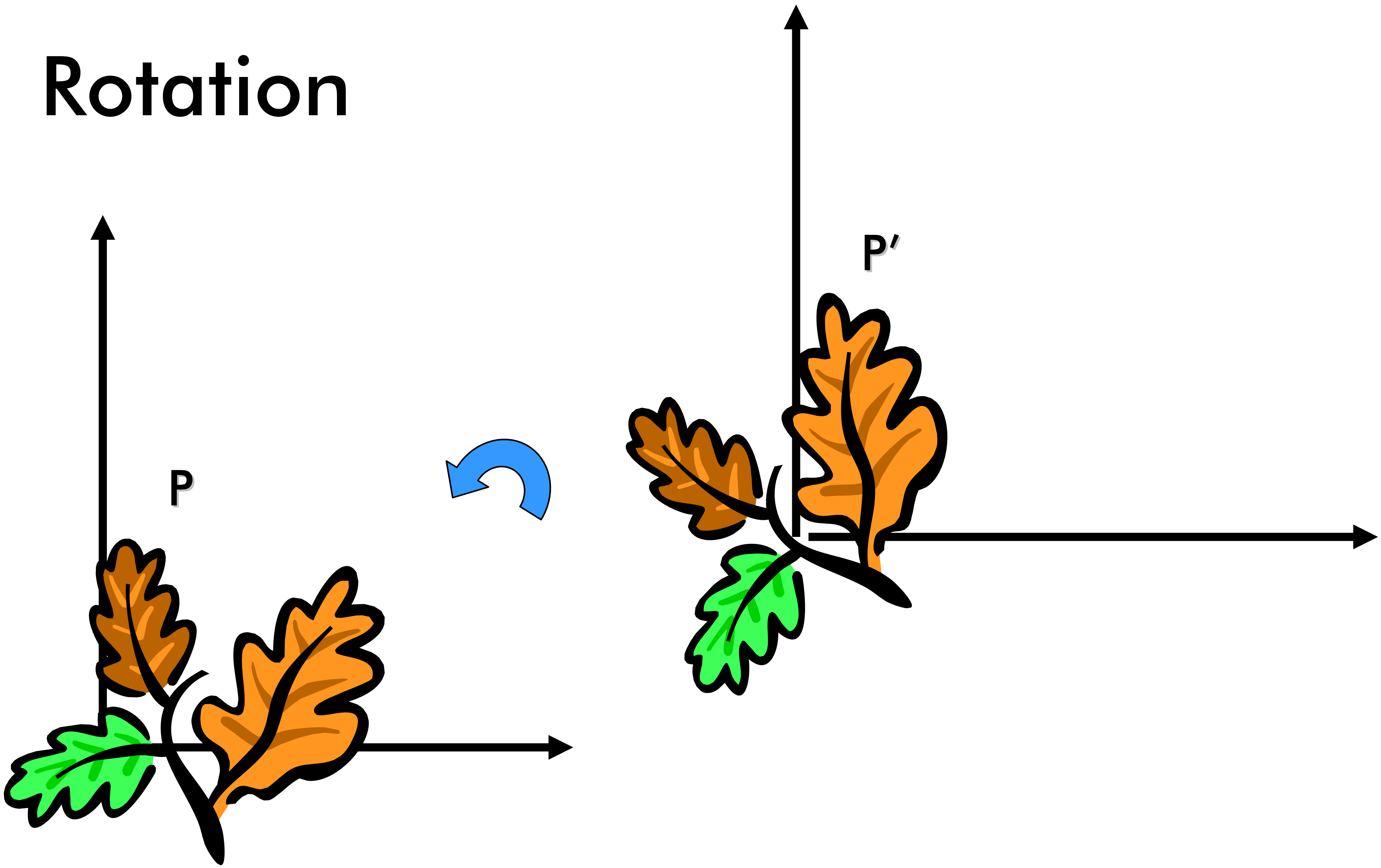
Translating & Scaling = Scaling & Translating ?

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

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

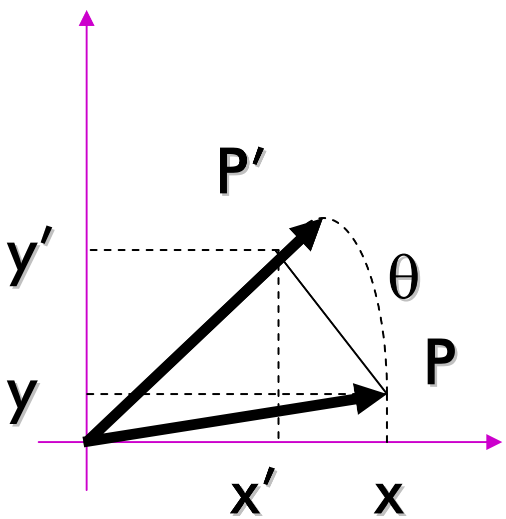
$$= \begin{bmatrix} s_x & 0 & s_x t_x \\ 0 & s_y & s_y t_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + s_x t_x \\ s_y y + s_y t_y \\ 1 \end{bmatrix}$$

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 $\Rightarrow$ 4 elements

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

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

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

Rotation+ Scaling +Translation

P' = (T R S) P

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

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

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

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

Transformation in 2D

-Isometries

-Similarities

-Affinity

-Projective

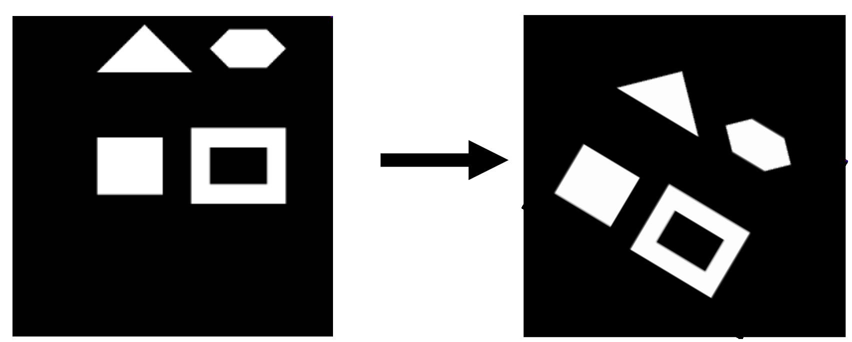
Transformation in 2D

Isometries:

[Euclideans]

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

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

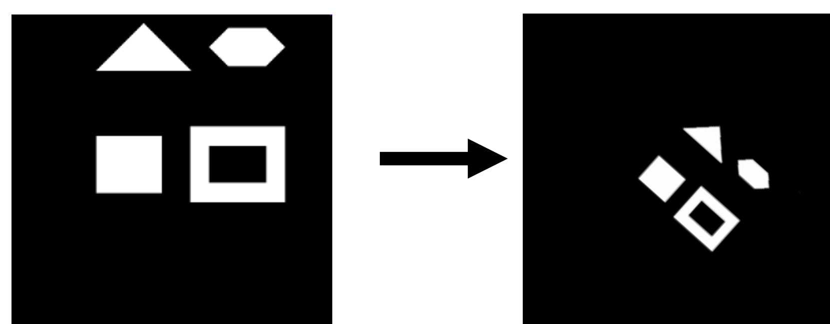


Transformation in 2D

Similarities:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Preserve
  - ratio of lengths
  - angles
- 4 DOF

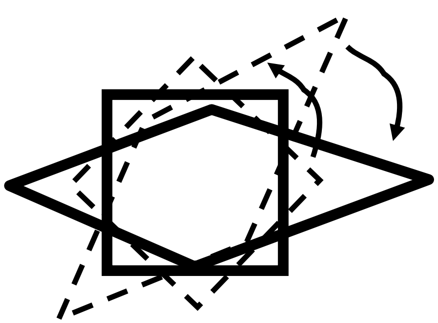


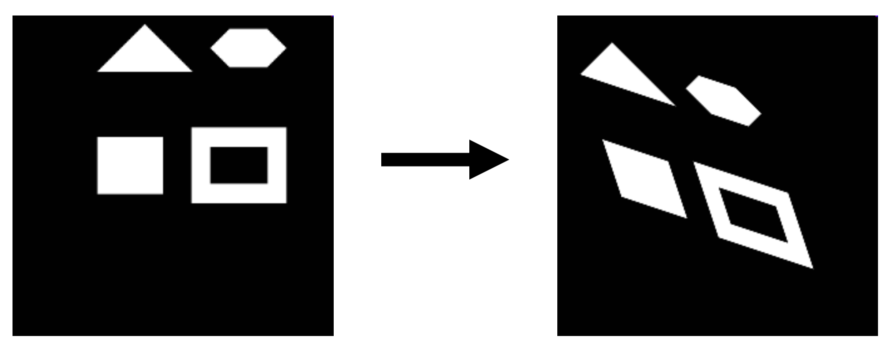
Transformation in 2D

Affinities:

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

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





Transformation in 2D

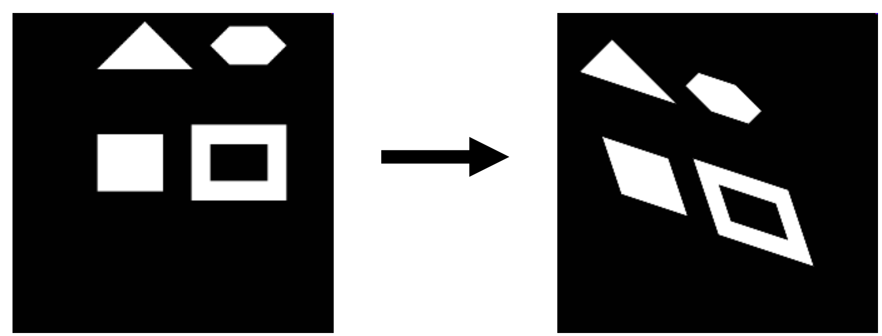
Affinities:

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

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

-Preserve:
- Parallel lines
- Ratio of areas
- Ratio of lengths on collinear lines
- others…

- 6 DOF

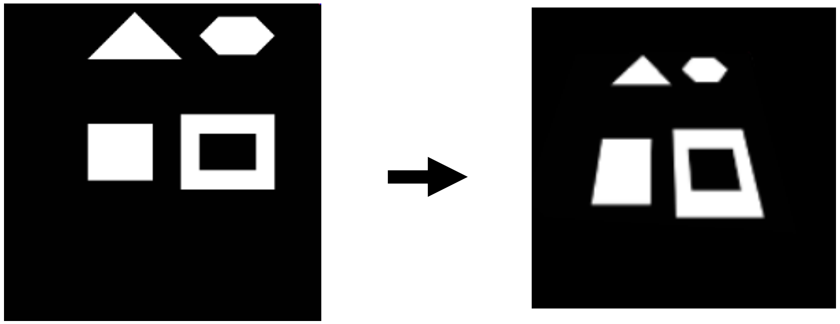


Transformation in 2D

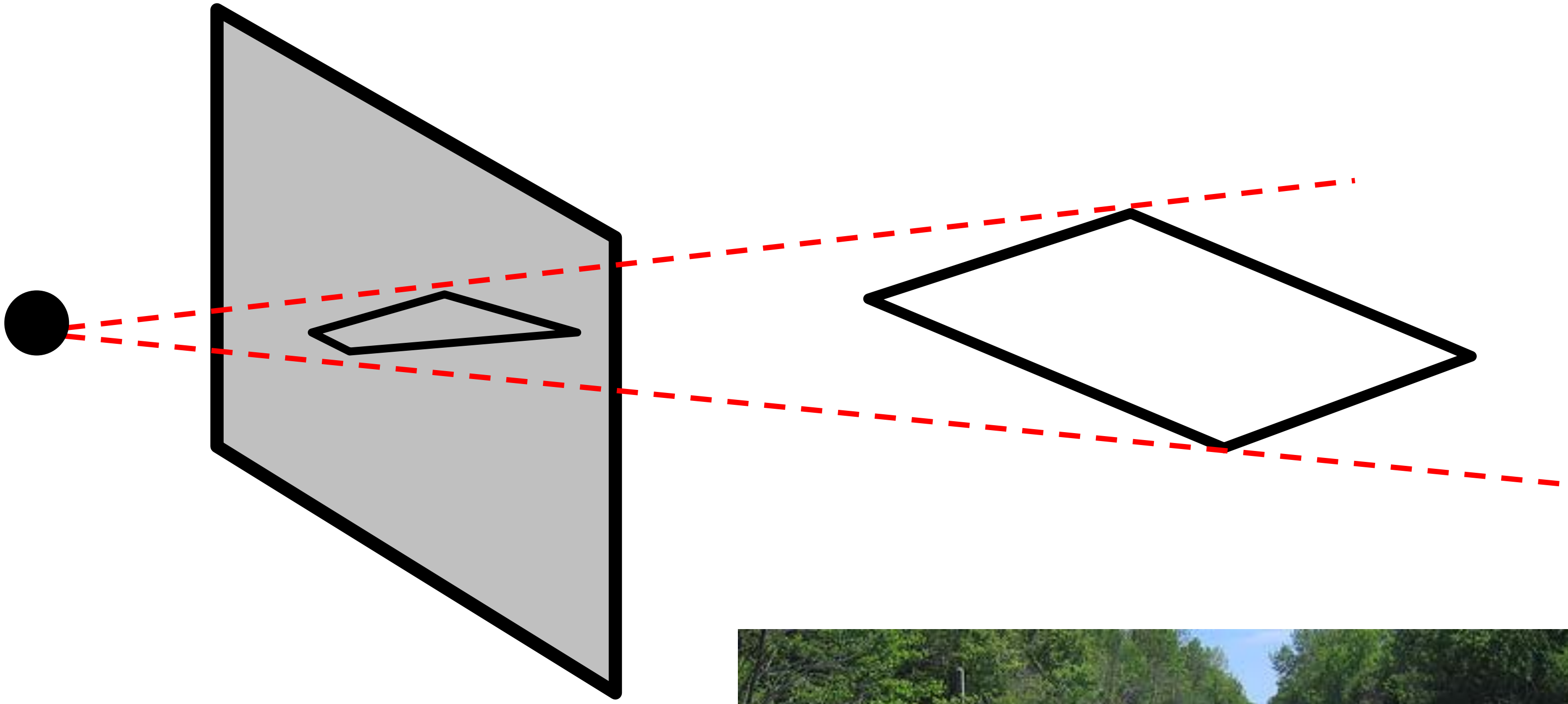
Projective:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ \boxed{v} & b \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_p \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- 8 DOF
- Preserve:
    - cross ratio of 4 collinear points
    - collinearity
    - and a few others…



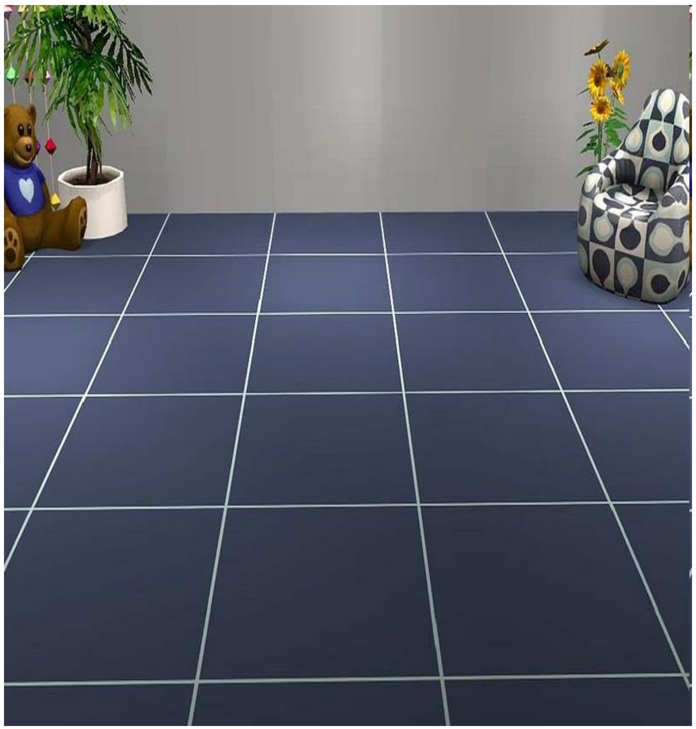
Transformation in 2D

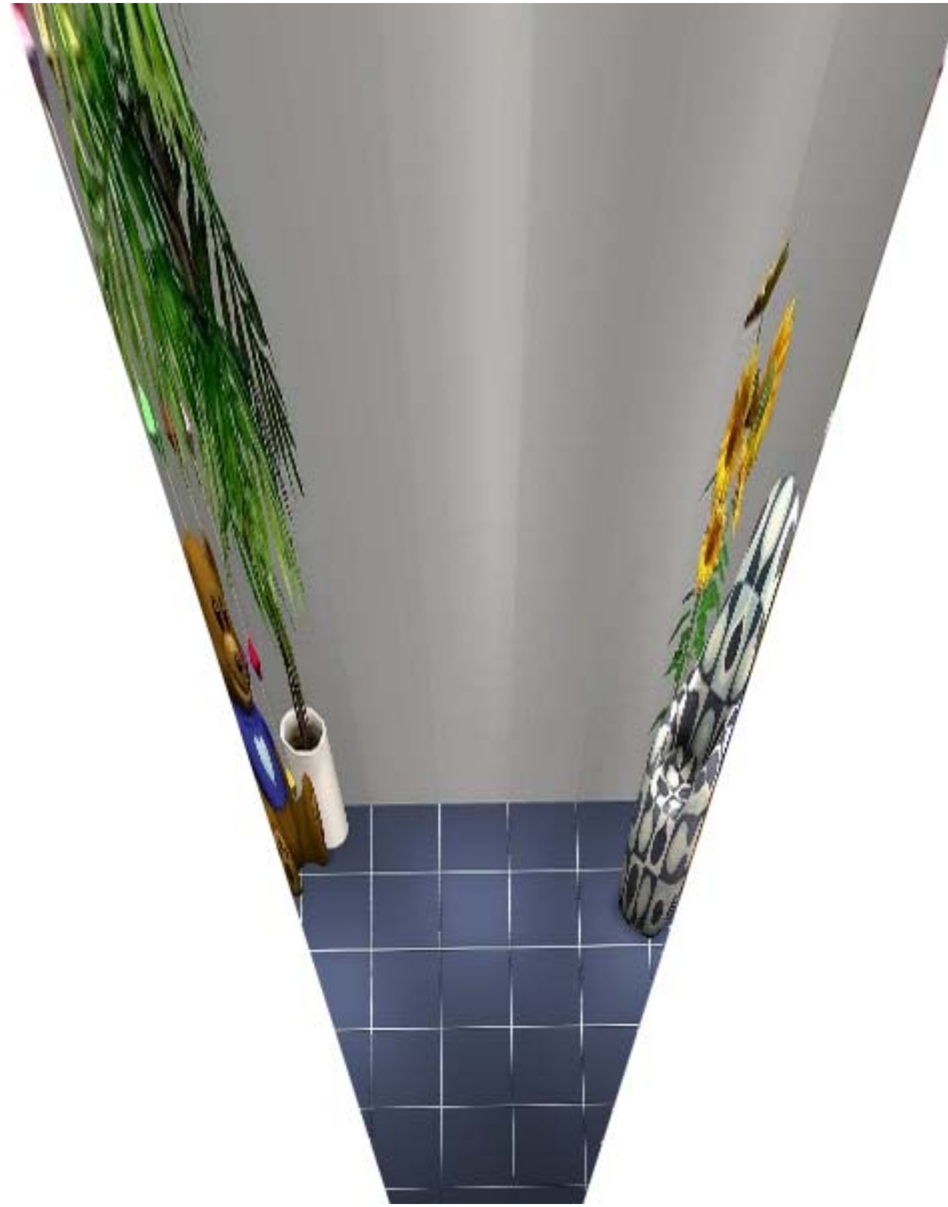


Removing perspective distortion

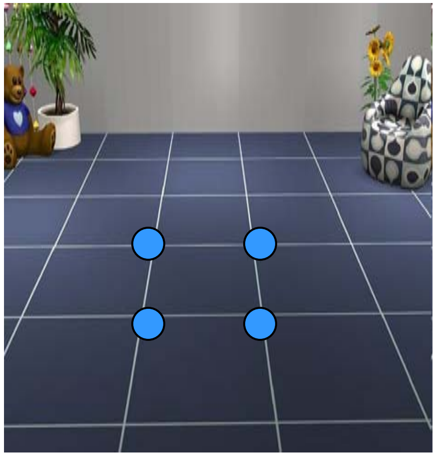
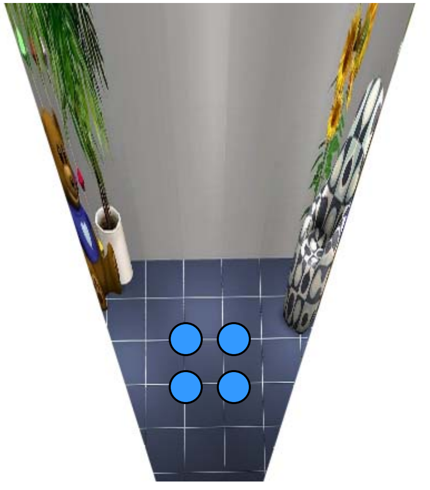
(rectification)

H_p





Computing H_p

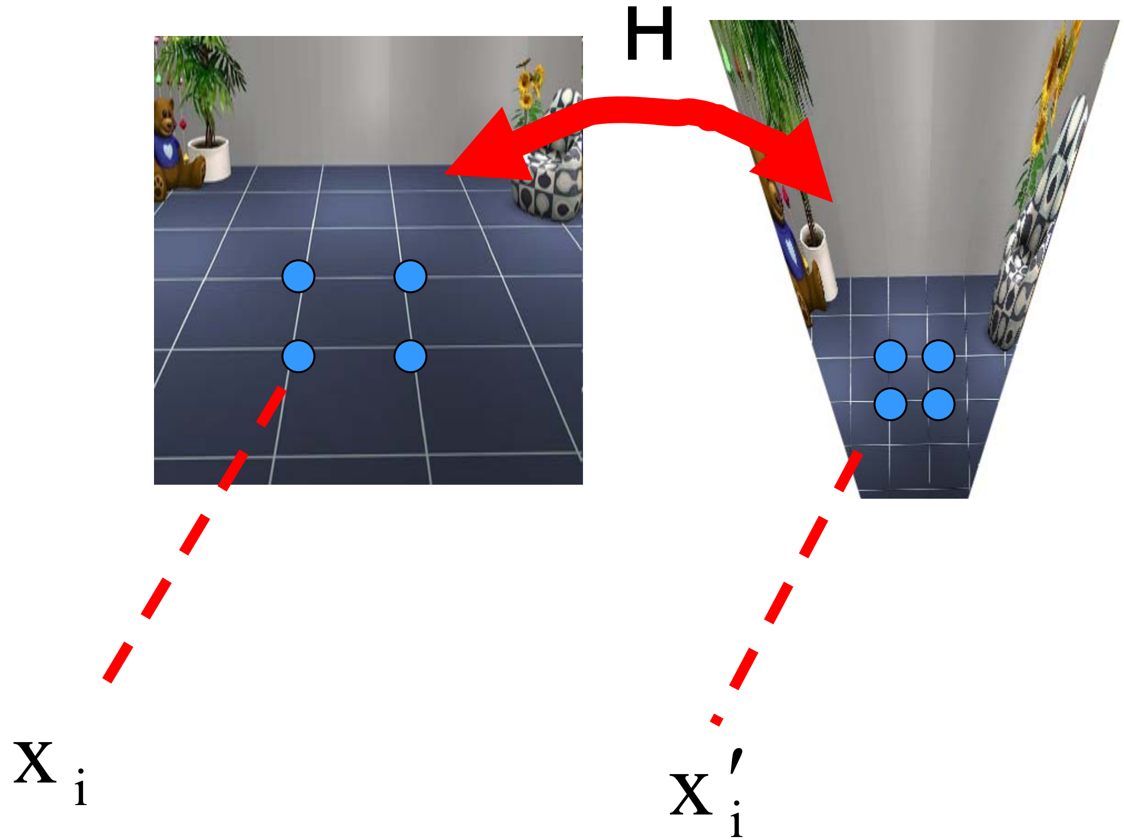
 

- 8 DOF
- how many points do I need to estimate H_p?

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)

$$x_i' \times H\, x_i = 0 \longrightarrow \underbrace{A_i}_{\text{Function of measurements}} \overbrace{\mathbf{h}}^{\text{unknown}} = 0$$

$$\underbrace{\mathbf{h} = \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix}}_{9\text{x}1}, \qquad \mathtt{H} = \begin{array}{c} (h^1)^T \\ \begin{bmatrix} \boxed{h_1 \quad h_2 \quad h_3} \\ h_4 \quad h_5 \quad h_6 \\ h_7 \quad h_8 \quad h_9 \end{bmatrix} \end{array}$$

Homogenous system!

DLT algorithm (direct Linear Transformation)

How to solve $A_i \, h = 0$?

Singular Value Decomposition (SVD)!

Eigenvalues and Eigenvectors

- Eigen relation

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

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

Eigenvalues and Eigenvectors

The eigenvalues of A are the roots of the
characteristic equation

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

$$\mathrm{S}^{-1}\mathrm{AS} = \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & . & \\ & & & \lambda_\mathrm{N} \end{bmatrix}$$ diagonal form of matrix

Eigenvectors of A are columns of S

Singular Value decomposition

- **Singular values**: Non negative square roots of the eigenvalues of $\mathbf{A^tA}$. Denoted σ_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

$$\mathrm{U}^{-1}\mathrm{AV} = \Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \cdot & \\ & & & \sigma_{\mathrm{N}} \end{bmatrix}$$

Properties of the SVD

- Suppose we know the singular values of **A** and we know r are non zero

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

  - Rank(**A**) = r.
  - Null(**A**) = span$\{\mathbf{v}_{r+1},\ldots,\mathbf{v}_n\}$
  - Range(**A**)=span$\{\mathbf{u}_1,\ldots,\mathbf{u}_r\}$
- $||\boldsymbol{A}||_F^2 = \sigma_1^2 + \sigma_2^2 + \ldots + \sigma_p^2$ $\qquad$ $||A||_2 = \sigma_1$
- *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
  - $$\mathbf{x}=\mathbf{A}^{-1}\mathbf{b}=(\mathbf{U}\,\Sigma\,\mathbf{V}^{\mathbf{t}})^{-1}\mathbf{b} = \sum_{i=1}^{n}\frac{\mathbf{u}_i^t\mathbf{b}}{\sigma_i}\mathbf{v}_i$$
- 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" $$\mathbf{x}_r = \sum_{i=1}^{k}\frac{\mathbf{u}_i^t\mathbf{b}}{\sigma_i}\mathbf{v}_i \qquad \sigma_k > \varepsilon,\ \ \sigma_{k+1} \le \varepsilon$$

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

DLT algorithm (direct Linear Transformation)

$$\overset{A_{2\times 9}}{\boxed{A_i}}\ \overset{h_{9\times 1}}{\boxed{h}} = 0$$

How to solve $\boxed{A_i}\,\boxed{h} = 0$?

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

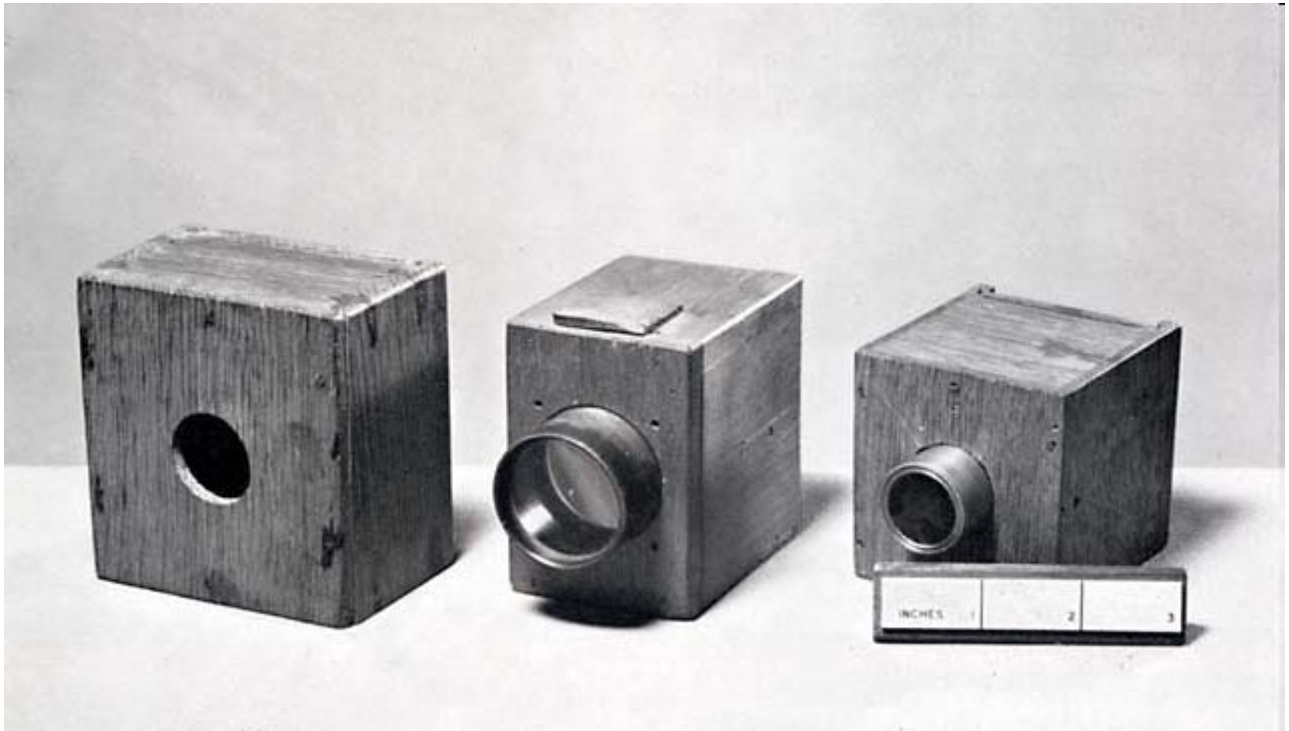
$$\downarrow$$

$$U_{2n\times 9}\, D_{9\times 9}\, \boxed{V^T_{9\times 9}}$$

Last column of V gives h! $\rightarrow$ H!

Next lecture

Cameras models



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 $\mathtt{H}$ given a set of four 2D to 2D point correspondences, $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. The transformation is given by the equation $\mathbf{x}'_i = \mathtt{H}\mathbf{x}_i$. Note that this is an equation involving homogeneous vectors; thus the 3-vectors $\mathbf{x}'_i$ and $\mathtt{H}\mathbf{x}_i$ 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 $\mathbf{x}'_i \times \mathtt{H}\mathbf{x}_i = \mathbf{0}$. This form will enable a simple linear solution for $\mathtt{H}$ to be derived.

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

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

Writing $\mathbf{x}'_i = (x'_i, y'_i, w'_i)^\mathsf{T}$, the cross product may then be given explicitly as

$$\mathbf{x}'_i \times \mathtt{H}\mathbf{x}_i = \begin{pmatrix} y'_i\mathbf{h}^{3\mathsf{T}}\mathbf{x}_i - w'_i\mathbf{h}^{2\mathsf{T}}\mathbf{x}_i \\ w'_i\mathbf{h}^{1\mathsf{T}}\mathbf{x}_i - x'_i\mathbf{h}^{3\mathsf{T}}\mathbf{x}_i \\ x'_i\mathbf{h}^{2\mathsf{T}}\mathbf{x}_i - y'_i\mathbf{h}^{1\mathsf{T}}\mathbf{x}_i \end{pmatrix}.$$

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

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w'_i\mathbf{x}_i^\mathsf{T} & y'_i\mathbf{x}_i^\mathsf{T} \\ w'_i\mathbf{x}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x'_i\mathbf{x}_i^\mathsf{T} \\ -y'_i\mathbf{x}_i^\mathsf{T} & x'_i\mathbf{x}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}. \tag{4.1}$$

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

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

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

(i) The equation $\mathtt{A}_i\mathbf{h} = \mathbf{0}$ is an equation *linear* in the unknown $\mathbf{h}$. The matrix elements of $\mathtt{A}_i$ 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 $\mathtt{H}$. It is usual to omit the third equation in solving for $\mathtt{H}$ ([Sutherland-63]). Then (for future reference) the set of equations becomes

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i'\mathbf{x}_i^\mathsf{T} & y_i'\mathbf{x}_i^\mathsf{T} \\ w_i'\mathbf{x}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i'\mathbf{x}_i^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}. \tag{4.3}$$

This will be written

$$\mathtt{A}_i\mathbf{h} = \mathbf{0}$$

where $\mathtt{A}_i$ is now the 2×9 matrix of (4.3).

(iii) The equations hold for any homogeneous coordinate representation $(x_i', y_i', w_i')^\mathsf{T}$ 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 $\mathtt{H}$

Each point correspondence gives rise to two independent equations in the entries of $\mathtt{H}$. Given a set of four such point correspondences, we obtain a set of equations $\mathtt{A}\mathbf{h} = \mathbf{0}$, where $\mathtt{A}$ is the matrix of equation coefficients built from the matrix rows $\mathtt{A}_i$ contributed from each correspondence, and $\mathbf{h}$ is the vector of unknown entries of $\mathtt{H}$. 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 $\mathtt{A}$ has dimension 12×9, and if (4.3) the dimension is 8×9. In either case $\mathtt{A}$ 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, $\mathtt{H}$ is in general only determined up to scale, so the solution $\mathbf{h}$ gives the required $\mathtt{H}$. 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