Basics

CS 554 – Computer Vision
Pinar Duygulu
Bilkent University

Outline

- Image Representation
- Review some basics of linear algebra and geometrical transformations
  - Slides adapted from Octavia Camps, Penn State and Stefan Roth, Brown University
- Introduction to Matlab
- Handling Images in Matlab

Image Representation

- Digital Images are 2D arrays (matrices) of numbers
- Each pixel is a measure of the brightness (intensity of light)
  - that falls on an area of an sensor (typically a CCD chip)

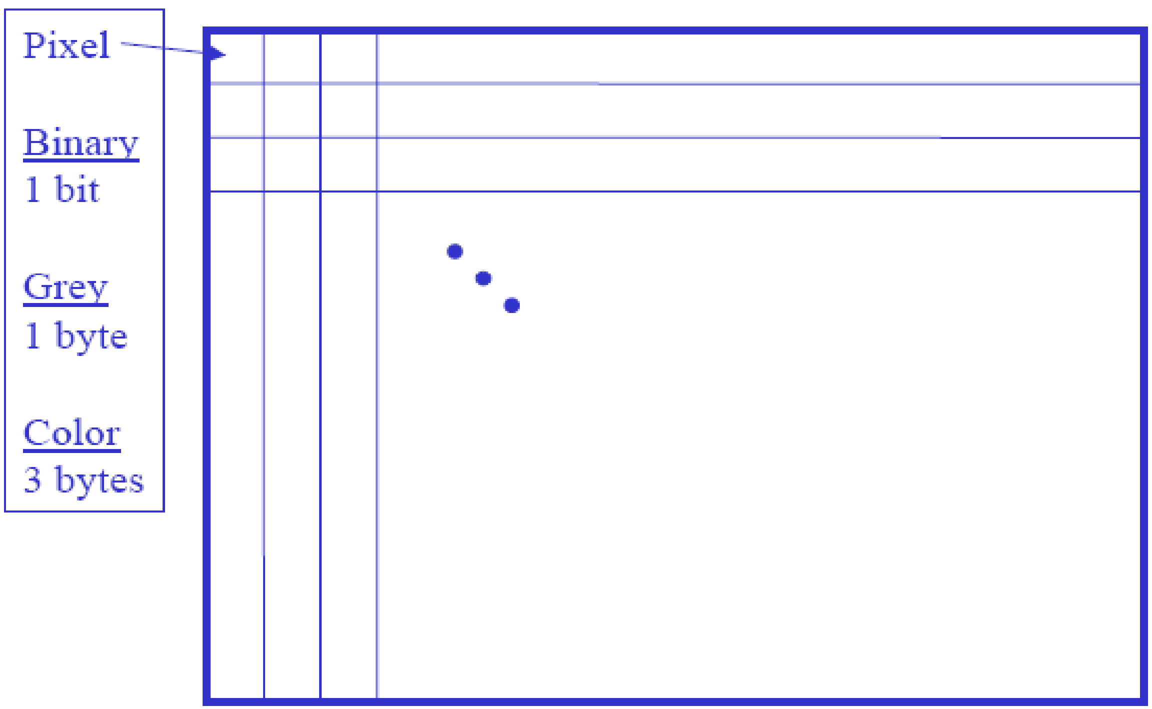
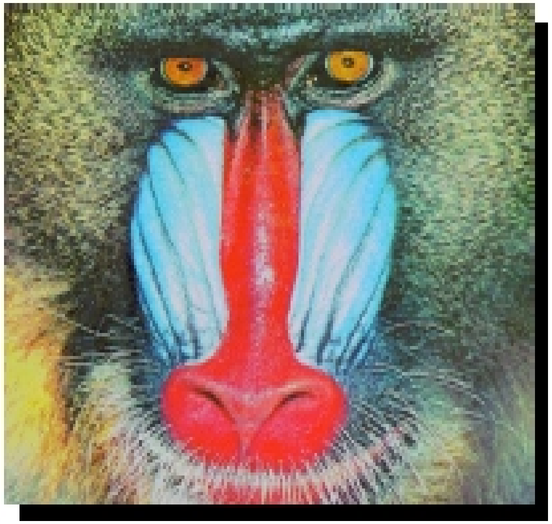
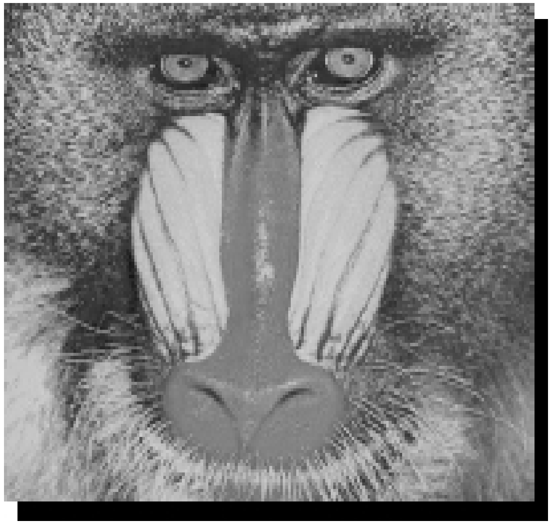


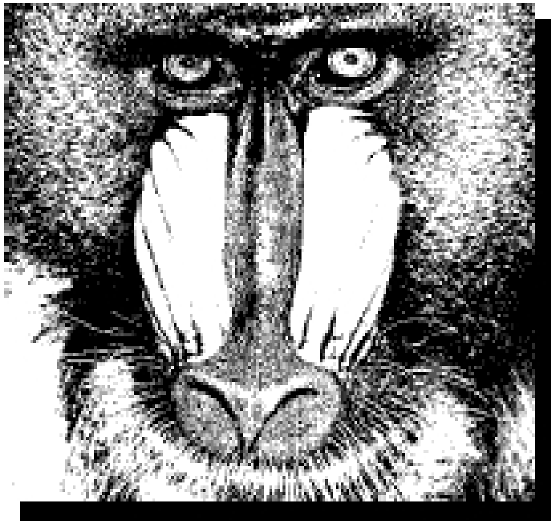
Image Representation



RGB



Greyscale



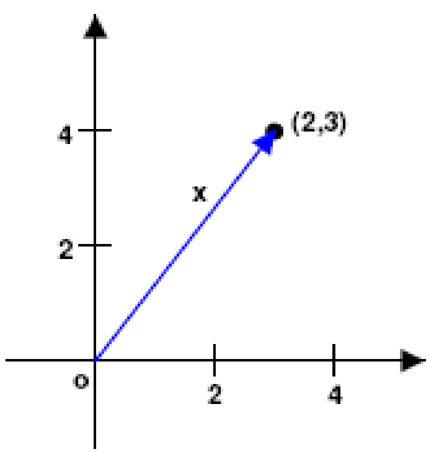
Binary

Vectors

Ordered set of numbers

$$\mathbf{x} \in \mathbb{R}^n, \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \qquad \mathbf{x} = (x_1, x_2, \ldots, x_n)^{\mathsf{T}}$$

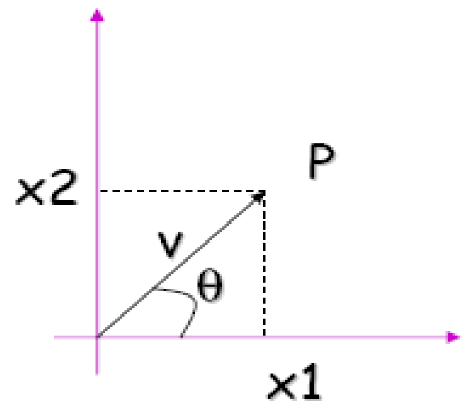
Example: coordinates of point



Vectors

2D Vector

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



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

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

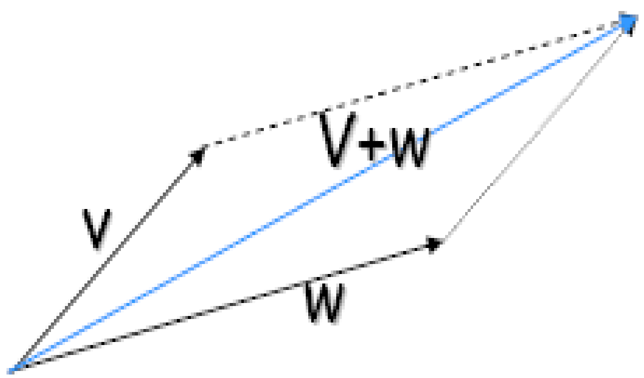
$$\frac{\mathbf{v}}{\|\mathbf{v}\|} = \left(\frac{x_1}{\|\mathbf{v}\|}, \frac{x_2}{\|\mathbf{v}\|} \right)$$ Is a unit vector

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

Vectors

Vector addition:

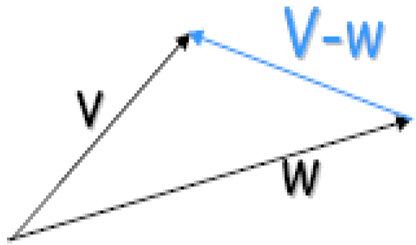
$$\mathbf{v} + \mathbf{w} = (x_1, x_2) + (y_1, y_2) = (x_1 + y_1, x_2 + y_2)$$



Vectors

Vector subtraction:

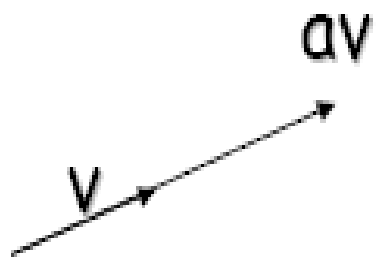
$$\mathbf{v} - \mathbf{w} = (x_1, x_2) - (y_1, y_2) = (x_1 - y_1, x_2 - y_2)$$



Vectors

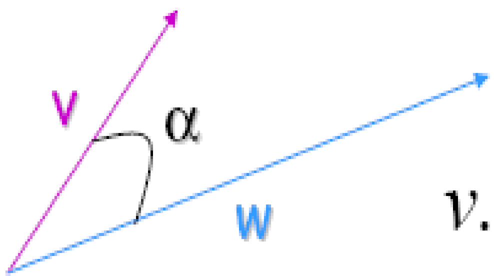
Scalar Product:

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



Vectors

Inner (dot) Product:



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

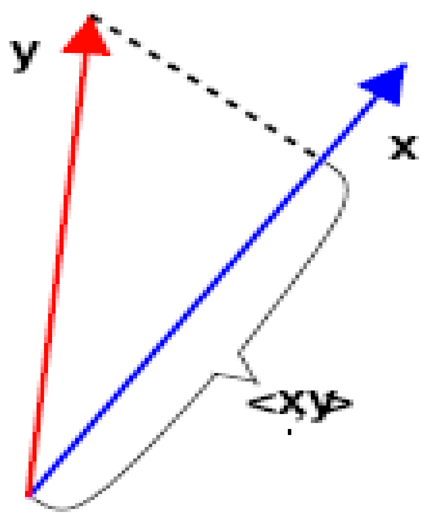
The inner product is a SCALAR!

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

$$v.w = 0 \Leftrightarrow v \perp w$$

Vectors

Projection of one point onto the other



Different notations

$$\langle x, y \rangle$$

$$x^{T} y$$

$$x\ y \text{ or } x \cdot y$$

The shown segment has length <x,y>, if x and y are unit vectors

Linear Dependence

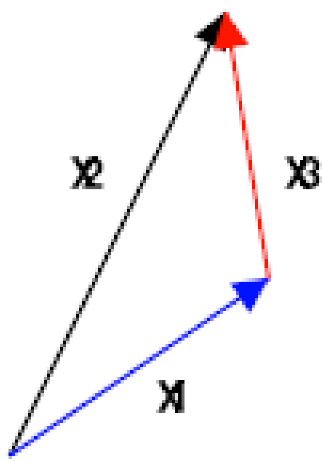
Linear combination of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$:

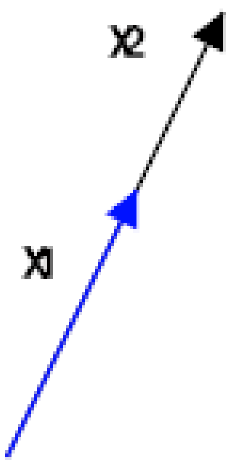
$$\alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_n \mathbf{x}_n$$

A set of vectors $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is linearly dependant if $\mathbf{x}_i \in X$ can be written as a linear combination of the rest, i.e., $X \setminus \{\mathbf{x}_i\}$.

In $\mathbb{R}^n$ it holds that

- a set of 2 to n vectors *can* be linearly dependant
- sets of $n+1$ or more vectors are *always* linearly dependant





Basis

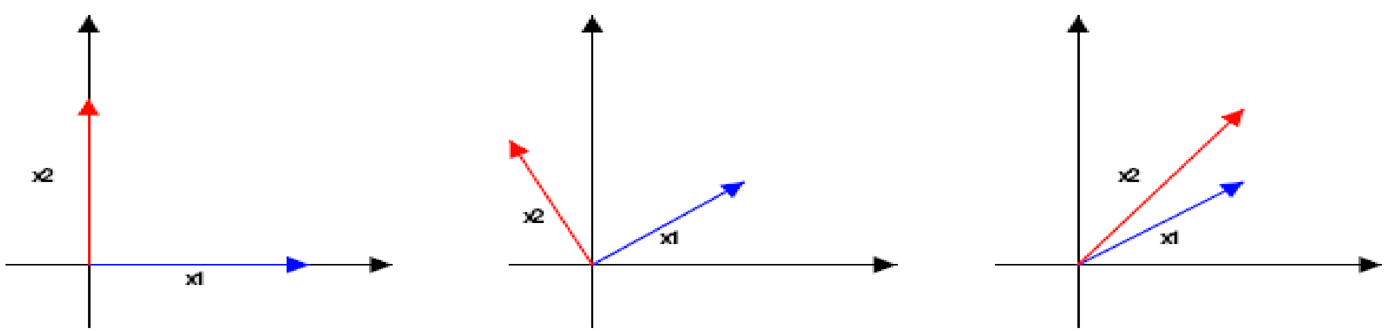
A basis is a linearly independent set of vectors that spans the "whole space". I. e., we can write every vector in our space as linear combination of vectors in that set.

Every set of n linearly independent vectors in $\mathbb{R}^n$ is a basis of $\mathbb{R}^n$.

Orthogonality: Two non-zero vectors $\mathbf{x}$ and $\mathbf{y}$ are orthogonal if $\langle \mathbf{x}, \mathbf{y} \rangle = 0$.

A basis is called

- *orthogonal*, if every basis vector is orthogonal to all other basis vectors
- *orthonormal*, if additionally all basis vectors have length 1.



Bases

Standard basis in $\mathbb{R}^n$ (also called unit vectors):

$$\{\mathbf{e}_i \in \mathbb{R}^n \ : \ \mathbf{e}_i = (\underbrace{0, \ldots, 0}_{i-1 \text{ times}}, 1, \underbrace{0, \ldots, 0}_{n-i-1 \text{ times}})\}$$

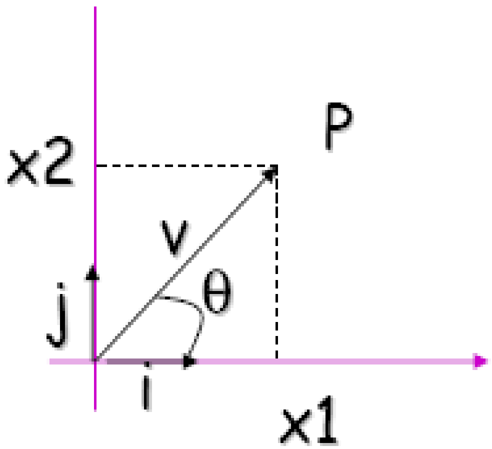
We can write a vector in terms of its standard basis, e.g.,

$$\begin{pmatrix} 4 \\ 7 \\ -3 \end{pmatrix} = 4 \cdot \mathbf{e}_1 + 7 \cdot \mathbf{e}_2 - 3 \cdot \mathbf{e}_3$$

Important observation: $x_i = \langle \mathbf{e}_i, \mathbf{x} \rangle$, i.e., to find the coefficient for a particular basis vector, we project our vector onto it.

Bases

Orthonormal basis:



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

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

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

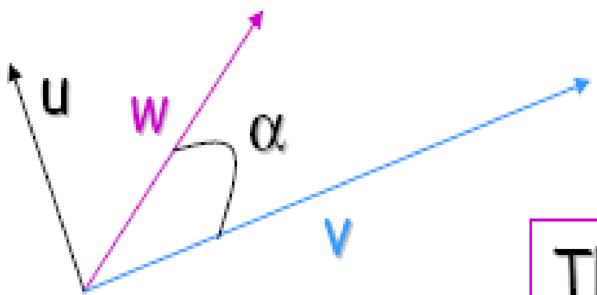
$$\mathbf{v} = (x_1, x_2) \qquad \mathbf{v} = x_1.\mathbf{i} + x_2.\mathbf{j}$$

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

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

Vectors

Vector (cross) product:



$$u = v \times w$$

The cross product is a VECTOR!

Magnitude: $\| \mathrm{u} \| = \| v.w \| = \| v \| \| w \| \sin \alpha$

Orientation:

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

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

Note that vXw is not equal to wXv

Vectors

Vector product computation:

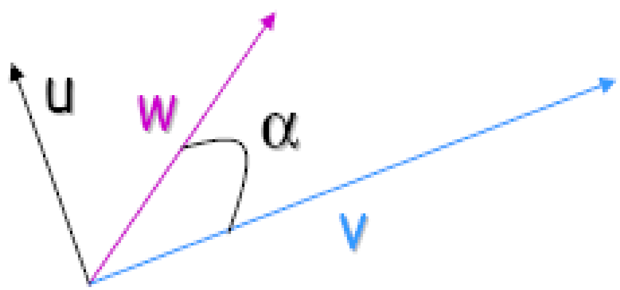
$$\mathbf{i} = (1,0,0) \qquad \|\mathbf{i}\| = 1$$

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

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

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

$$\mathbf{u} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}$$



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

Matrices

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

Sum:

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

$$c_{ij} = a_{ij} + b_{ij}$$

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

Product:

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

A and B must have compatible dimensions

$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj}$$

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

Examples:

$$\begin{bmatrix} 2 & 5 \\ 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 6 & 2 \\ 1 & 5 \end{bmatrix} = \begin{bmatrix} 17 & 29 \\ 19 & 11 \end{bmatrix}$$

$$\begin{bmatrix} 6 & 2 \\ 1 & 5 \end{bmatrix} \cdot \begin{bmatrix} 2 & 5 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 18 & 32 \\ 17 & 10 \end{bmatrix}$$

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:

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

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

Matrices

Determinant: A must be square

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

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

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:

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

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

Matrices

The rank of a matrix is the number of linearly independent rows or columns.

Examples: $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ has rank 2, but $\begin{pmatrix} 2 & 1 \\ 4 & 2 \end{pmatrix}$ only has rank 1.

Equivalent to the dimension of the range of the linear transformation.

A matrix with full rank is called *non-singular*, otherwise it is singular.

For singular matrices determinant is 0

Eigenvalues and Eigenvectors

All non-zero vectors $\mathbf{x}$ for which there is a $\lambda \in \mathbb{R}$ so that

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

are called *eigenvectors* of $\mathbf{A}$. λ are the associated *eigenvalues*.

If $\mathbf{e}$ is an eigenvector of $\mathbf{A}$, then also $c \cdot \mathbf{e}$ with $c \neq 0$.

Label eigenvalues $\lambda_1 \geq \lambda_1 \geq \cdots \geq \lambda_n$ with their eigenvectors $\mathbf{e}_1, \ldots, \mathbf{e}_n$ (assumed to be unit vectors).

Eigenvalues and Eigenvectors

Rewrite the definition as

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = 0$$

$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$ has to hold, because $\mathbf{A} - \lambda\mathbf{I}$ cannot have full rank.

This gives a polynomial in λ, the so-called *characteristic polynomial*.

Find the eigenvalues by finding the roots of that polynomial.

Find the associated eigenvector by solving linear equation system.

Eigendecomposition

Every real, square, symmetric matrix $\mathbf{A}$ can be decomposed as:

$$\mathrm{A} = \mathrm{V}\mathrm{D}\mathrm{V}^{\mathsf{T}},$$

where $\mathbf{V}$ is an orthonormal matrix of $\mathbf{A}$'s eigenvectors and $\mathbf{D}$ is a diagonal matrix of the associated eigenvalues.

The eigendecomposition is essentially a restricted variant of the Singular Value Decomposition.

Singular Value Decomposition (SVD)

Suppose $\mathbf{A} \in \mathbb{R}^{m \times n}$. Then a $\lambda \geq 0$ is called a *singular value* of $\mathbf{A}$, if there exist $\mathbf{u} \in \mathbb{R}^{m}$ and $\mathbf{v} \in \mathbb{R}^{n}$ such that

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{u} \qquad \text{and} \qquad \mathbf{A}^{\mathsf{T}}\mathbf{u} = \lambda \mathbf{v}$$

We can decompose *any* matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ as

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^{\mathsf{T}},$$

where $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are orthonormal and Σ is a diagonal matrix of the singular values.

Singular Value Decomposition

The determinant of a square matrix is the product of its eigenvalues: $\det(\mathbf{A}) = \lambda_1 \cdot \ldots \cdot \lambda_n$.

A square matrix is singular if it has some eigenvalues of value 0.

A square matrix $\mathbf{A}$ is called positive (semi-)definite if all of its eigenvalues are positive (non-negative). Equivalent criterion:

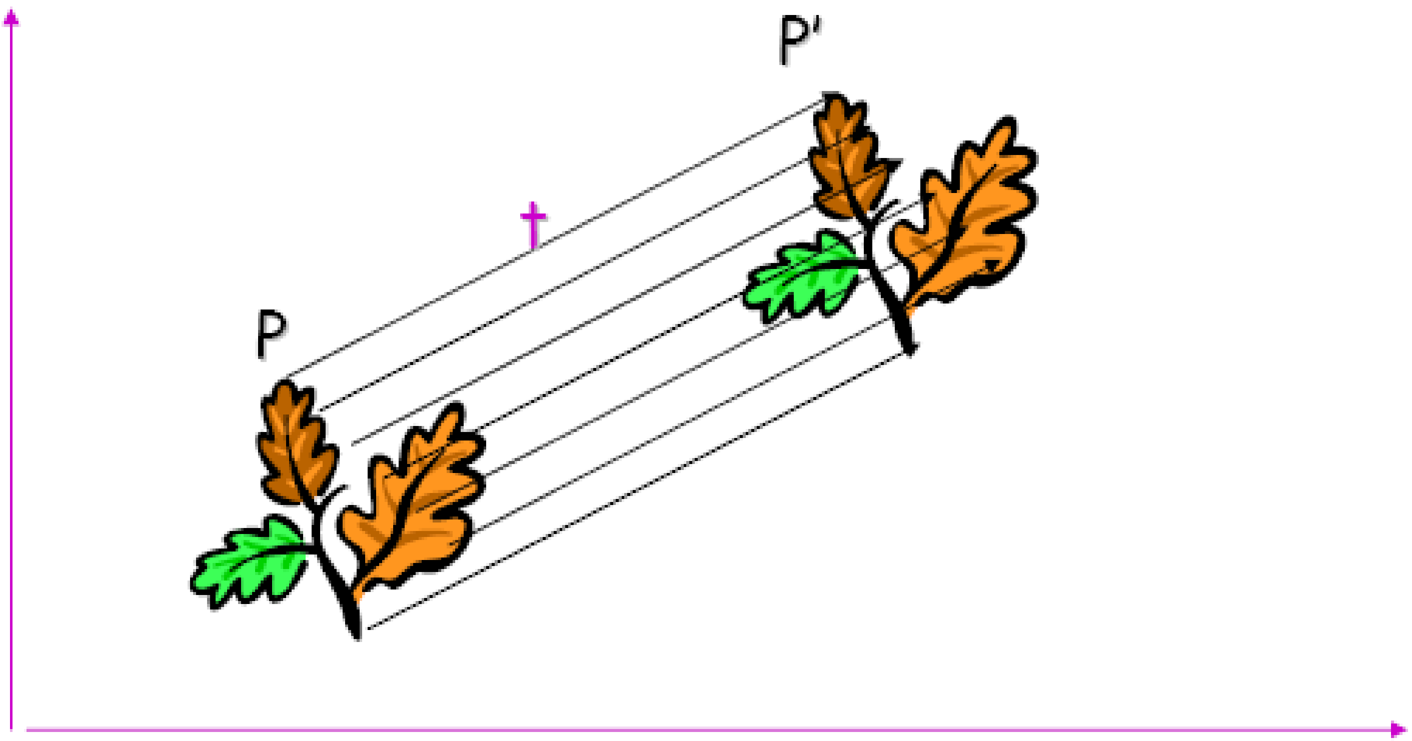
$\mathbf{x}^\top \mathbf{A} \mathbf{x} > 0 \quad \forall \mathbf{x}$ $\quad$ ($\geq$ if semi-definite).

Singular Value Decomposition (SVD)

The columns of U are the eigenvectors of AA^T and the (non-zero) singular values of A are the square roots of the (non-zero) eigenvalues of AA^T

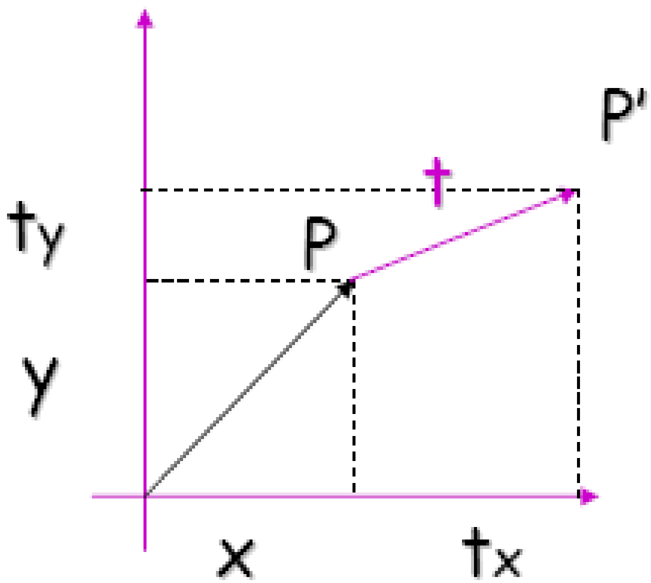
Geometrical Transformations

2D Translation:



Geometrical Transformations

2D Translation Equation:



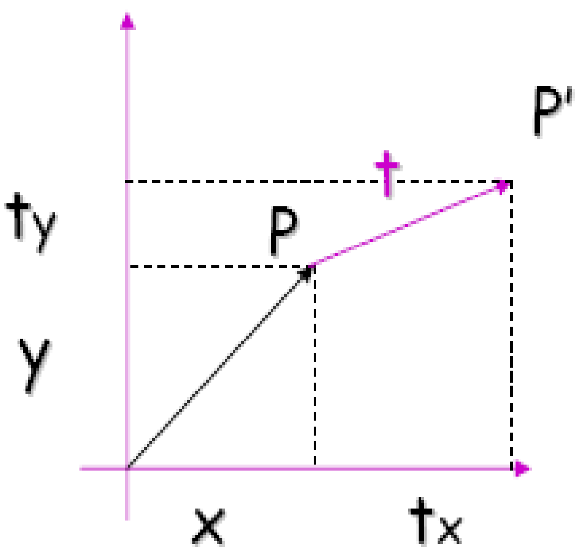
$$\mathbf{P} = (x, y)$$

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

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

Geometrical Transformations

2D Translation using matrices:



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

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

t P

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

Geometrical Transformations

Homogeneous Coordinates

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

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

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

NOTE: If the scalar is 1, there is no need for the multiplication!

Geometrical Transformations

Back to Cartesian Coordinates

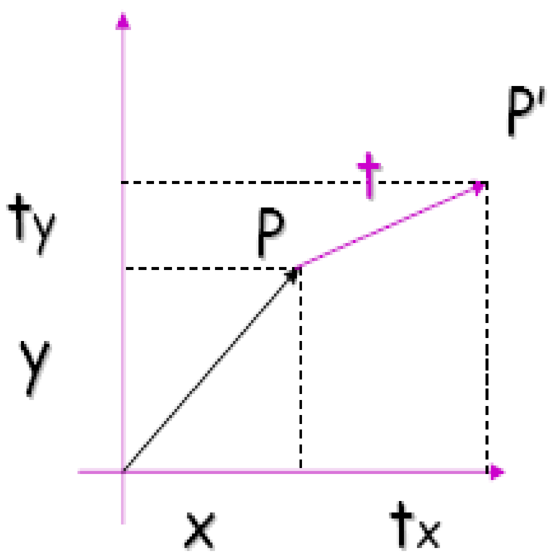
Divide by the last coordinate and eliminate it. For example,

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

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

Geometrical Transformations

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

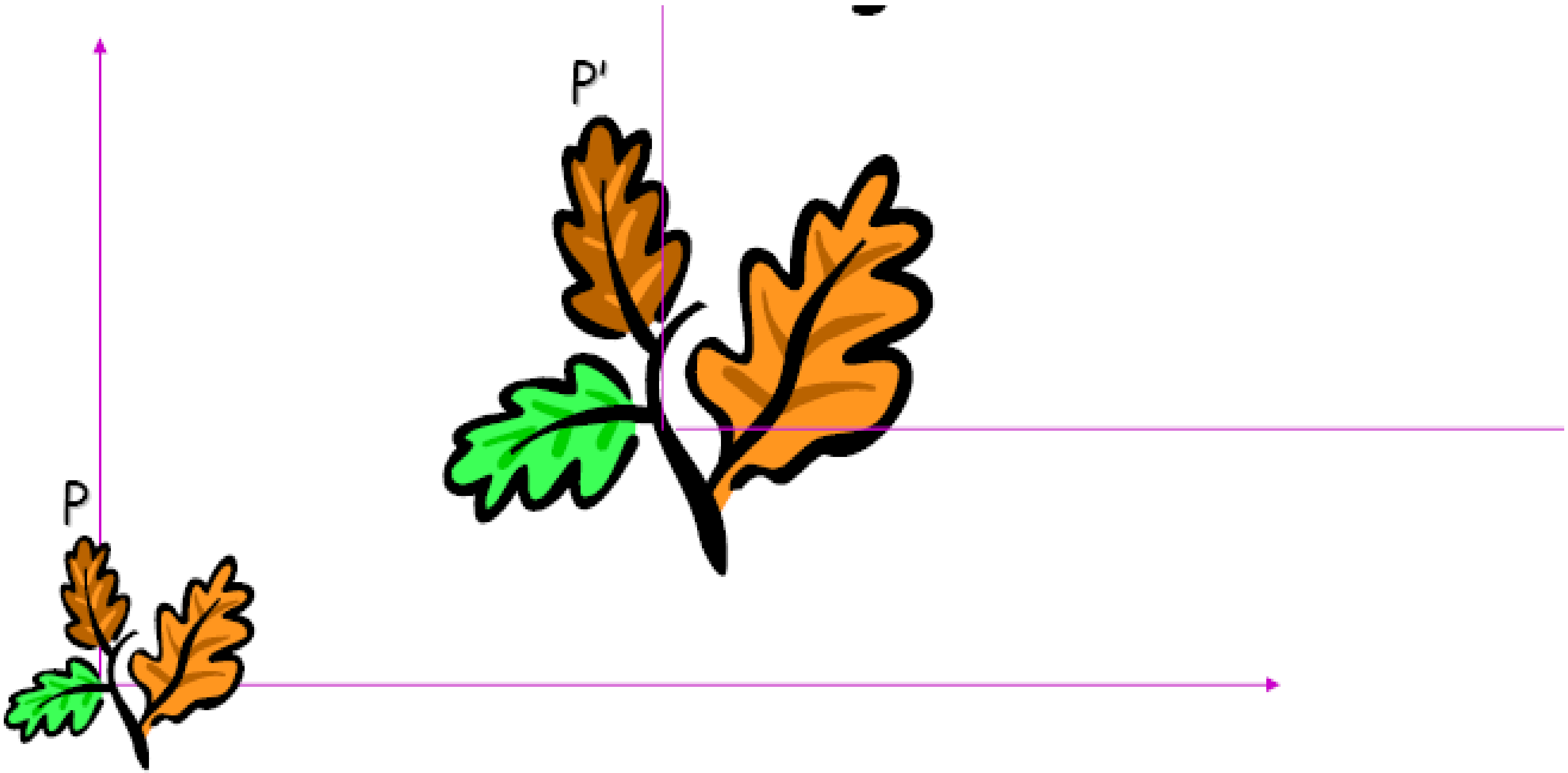
P

$$\mathbf{P'} \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\mathbf{P'} = \mathbf{T} \cdot \mathbf{P}$$

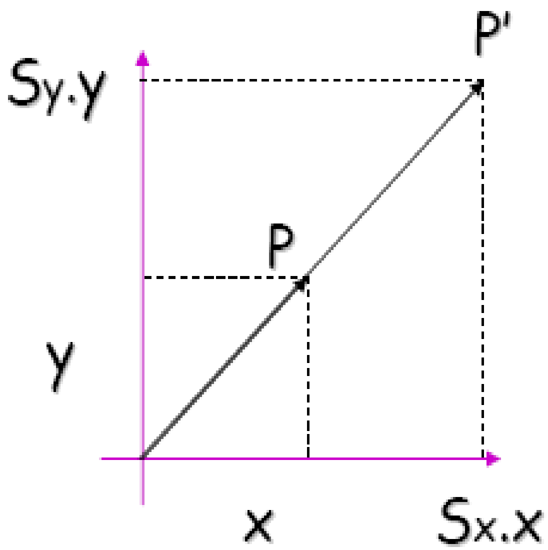
Geometrical Transformations

Scaling



Geometrical Transformations

Scaling Equation



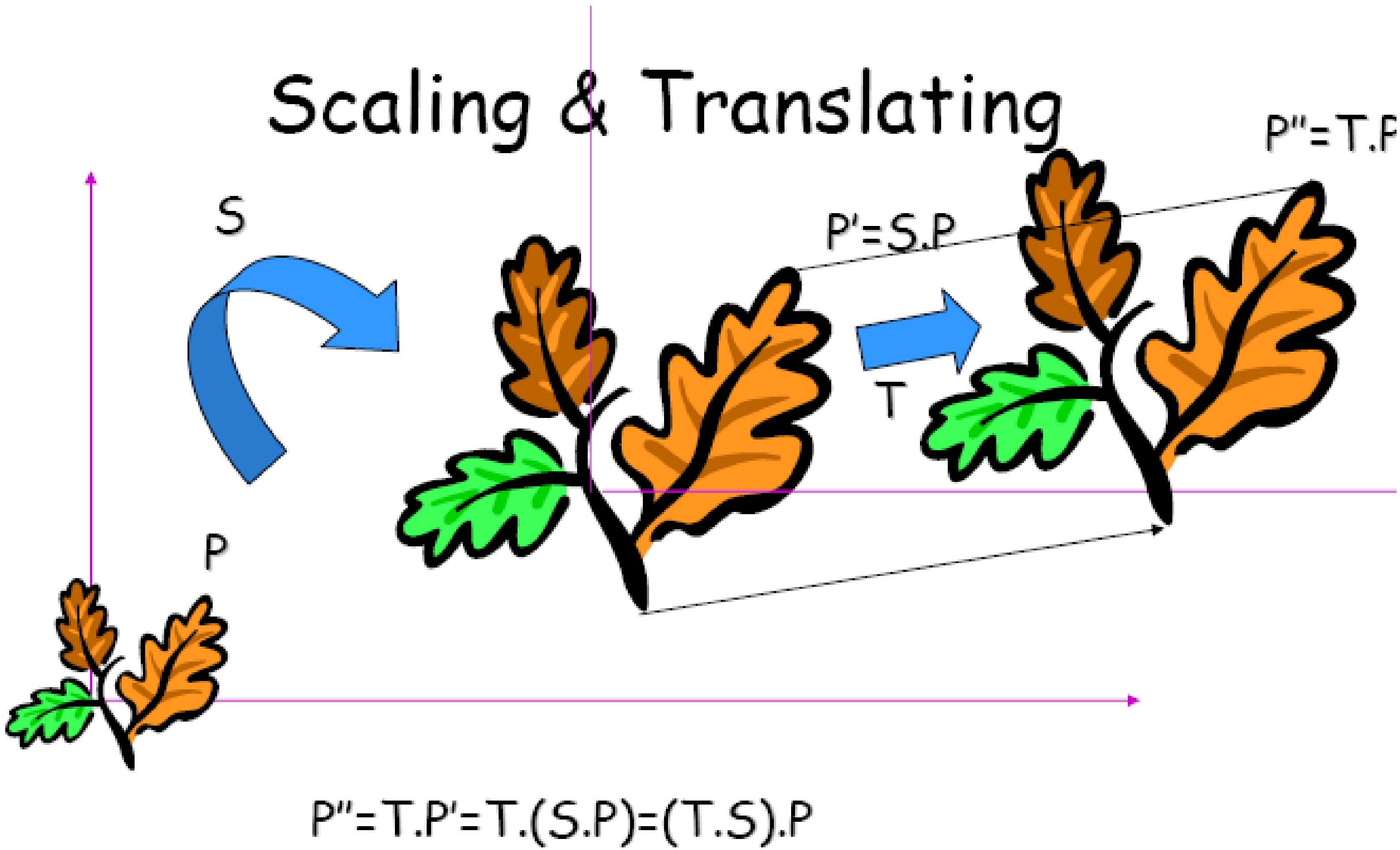
$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

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

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

$$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P}$$

Geometrical Transformations



Geometrical Transformations

Scaling & Translating

P''=T.P'=T.(S.P)=(T.S).P

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

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

Geometrical Transformations

Translating & Scaling
≠ Scaling & Translating

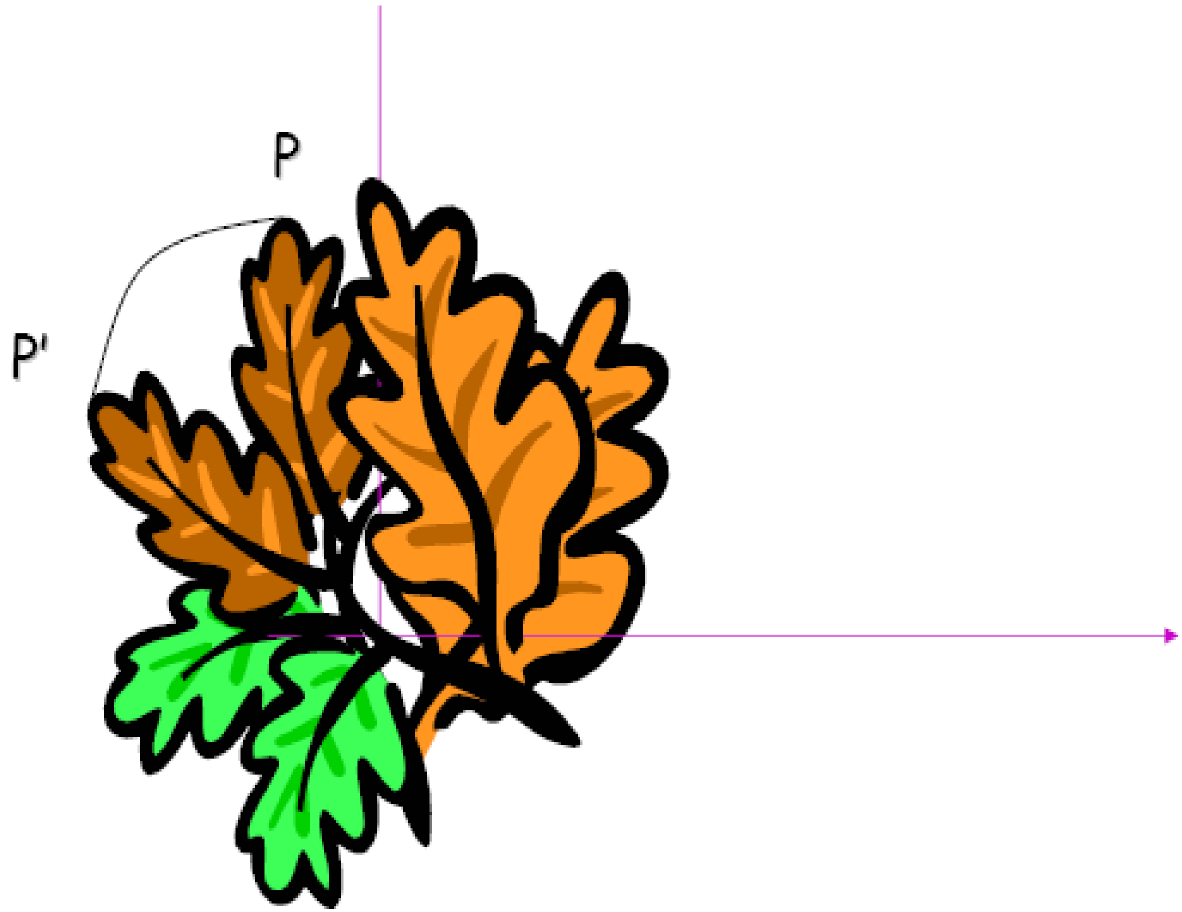
$$P''=S.P'=S.(T.P)=(S.T).P$$

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

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

Geometrical Transformations

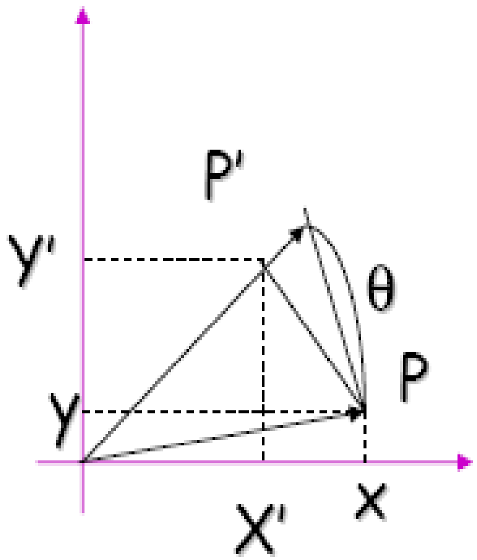
Rotation



Geometrical Transformations

Rotation Equations:

Counter-clockwise rotation by an angle θ

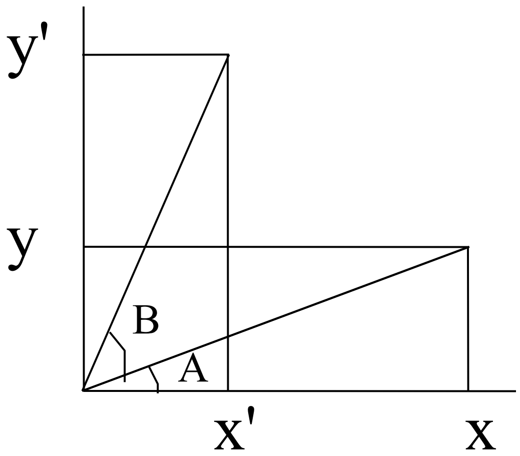


$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

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

Geometrical Transformations

Rotation Equations:



$x = r.\cos A$
$y = r.\sin A$

$x' = r.\cos(A+B) = r.\cos A.\cos B - r.\sin A.\sin B$
$y' = r.\sin(A+B) = r.\sin A.\cos B + r.\cos A.\sin B$

$x' = x.\cos B - y.\sin B$
$y' = y.\cos B + x.\sin B$

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

$\sin(A + B) = \sin A \cos B + \cos A \sin B$
$\cos(A + B) = \cos A \cos B - \sin A \sin B$

Geometrical Transformations

Scaling, Translating & Rotating:

Order matters!

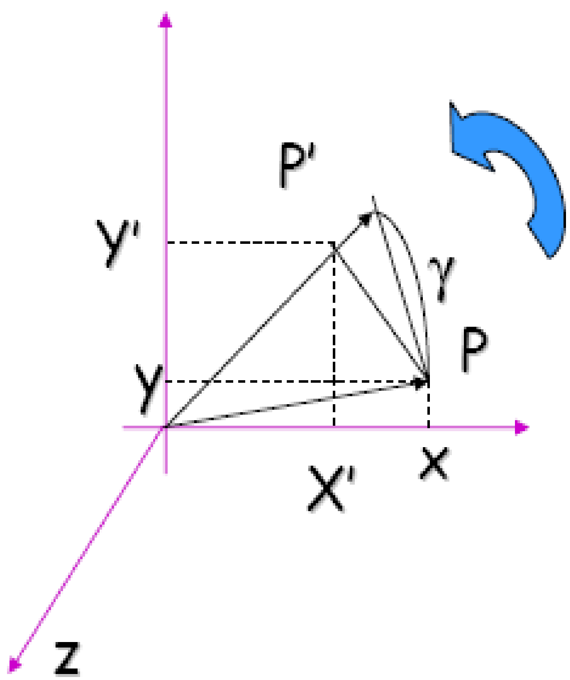
$$P' = S.P$$
$$P'' = T.P' = (T.S).P$$
$$P''' = R.P'' = R.(T.S).P = (R.T.S).P$$

$$R.T.S \neq R.S.T \neq T.S.R \ \dots$$

Geometrical Transformations

3D Rotation of Points:

Rotation around the coordinate axes, counter-clockwise:



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

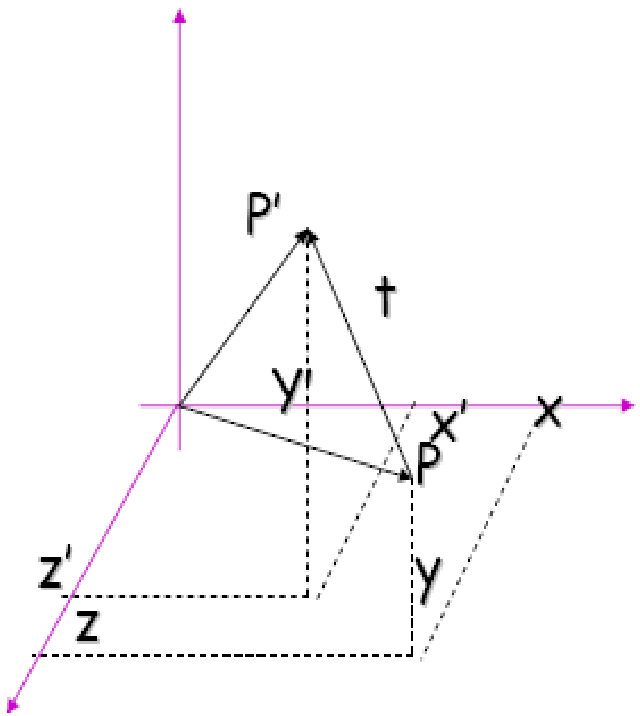
$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

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

Geometrical Transformations

3D Translation of Points:

Translate by a vector $t=(t_x,t_y,t_x)^T$:



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

Introduction to Matlab

One kind of object – a rectangular numerical matrix
Scalars : 1x1 matrices
Vectors : matrices with only one row or one column

a 3x3 matrix
A = [1 2 3; 4 5 6; 7 8 9]
is equal to
A = [1 2 3
4 5 6
7 8 9]

Introduction to Matlab

```
A = [1 2; 3 4]    % creates a 2x2 matrix
N = 5             % a scalar
v = [1 0 0]       % a row vector
V = [1; 2; 3]     % a column vector
v = v'            % transpose of a vector
v = 1:2:7         % [start:stepsize:end] v = [1 3 5 7]
v = []            % empty vector
```

Introduction to Matlab

```
m = zeros(2,3)  % creates a 2x3 matrix of zeros
v = ones(1,3)   % creates a 1x3 matrix (row vector) of ones
m = eye(3)      % identity matrix
v = rand(3,1)   % randomly filled matrix
m = zeros(3)        % 3x3 matrix of zeros

d = diag(a)     % diagonal of matrix a
```

Introduction to Matlab

```
v = [1 2 3]
v(3)                % access a vector element

m = [1 2 3 4; 5 6 7 8; 9 10 11 12; 13 14 15 16]
m(1,3)     % access a matrix element (row #, column #)
m(2,:)     % access a whole matrix row
m(:,1)     % access a whole matrix column

m(1,1:3)  % access elements 1 through 3 of 1st row
m(2:end, 3)
```

Introduction to Matlab

```
m = [1 2 3; 4 5 6]
size(m)                    % returns the size of a matrix

m1 = zeros(size(m))

a = [1 2 3 4]'
2 * a
a / 4
b = [5 6 7 8]'
a + b
a – b
a .^2
```

Introduction to Matlab

```
a = [1 4 6 3]
sum(a)          % sum of vector elements
mean(a)         % mean
var(a)          % variance
std(a)          % standard deviation
max(a)          % maximum
min(a)          % minimum

a = [1 2 3; 4 5 6]
mean(a)         % mean of each column
mean(a,2)       % mean of each row
max(max(a))     % maximum value of the matrix
```

Introduction to Matlab

```
a = [1 2 3; 4 5 6; 7 8 9]
inv(a)      % matrix inverse
eig(a)      % vector of eigenvalues of a
[V, D] = eig(a)
% D:eigenvalues on diagonal
%V :matrix of eigenvectors

[U,S,V] = svd(a)    % singular value decomposition of a
% a = U * S * V'
```

Introduction to Matlab

```
B = zeros(m,n)
for i=1:m
    for j=1:n
        if A(i,j) > 0
            B(i,j) = A(i,j)
        end
    end
end
```

```
B = zeros(m,n)
ind = find(A>0)
B(ind) = A(ind)
```

Introduction to Matlab

```
x = [0 1 2 3 4]
plot(x, 2*x)
xlabel('x')
ylabel('2*x')
title('dummy')
bar(x)
```

Introduction to Matlab

myfunction.m

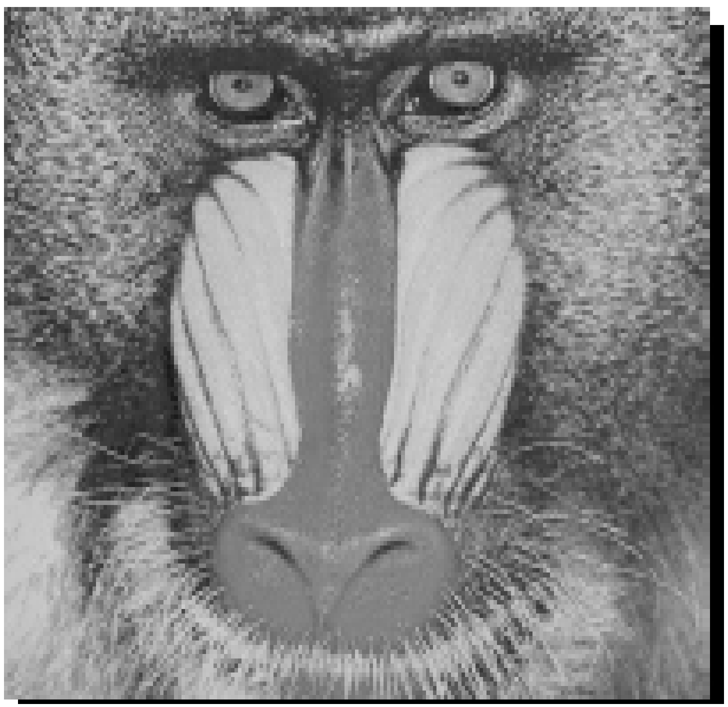
```
function y = myfunction(x)

a = [-2 -1 0 1]
y = a + x;
```

Handling Images in Matlab

```
I = imread('img.jpg');      % read a jpg image
imshow(I)                   % shows an image
imwrite(I, filename)        % writes an image to  file
```

Handling Images in Matlab

```
figure
imagesc(I)                    % display it as gray level image
colormap gray;
```

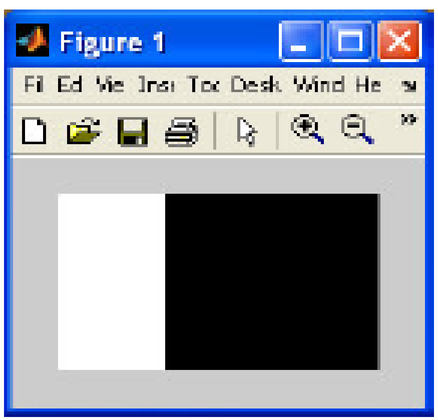


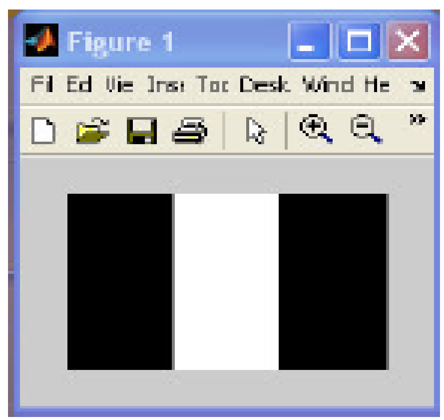
Handling Images in Matlab

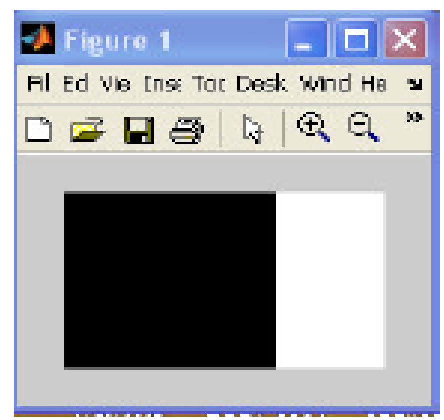
size(img)
90 150 3

Handling Images in Matlab

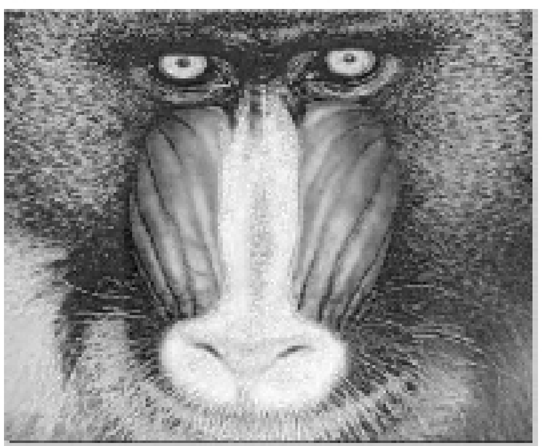
```
R = img(:,:,1)
G = img(:,:,2)
B = img(:,:,3)
imshow(R)
imshow(G)
imshow(B)
```

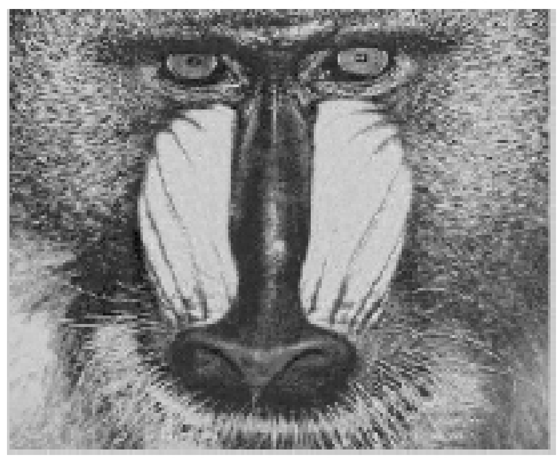


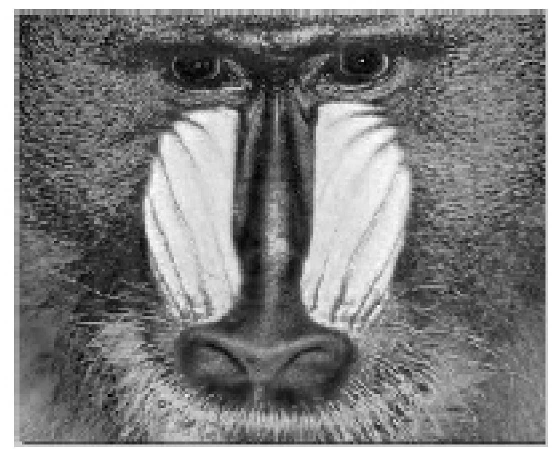




Handling Images in Matlab

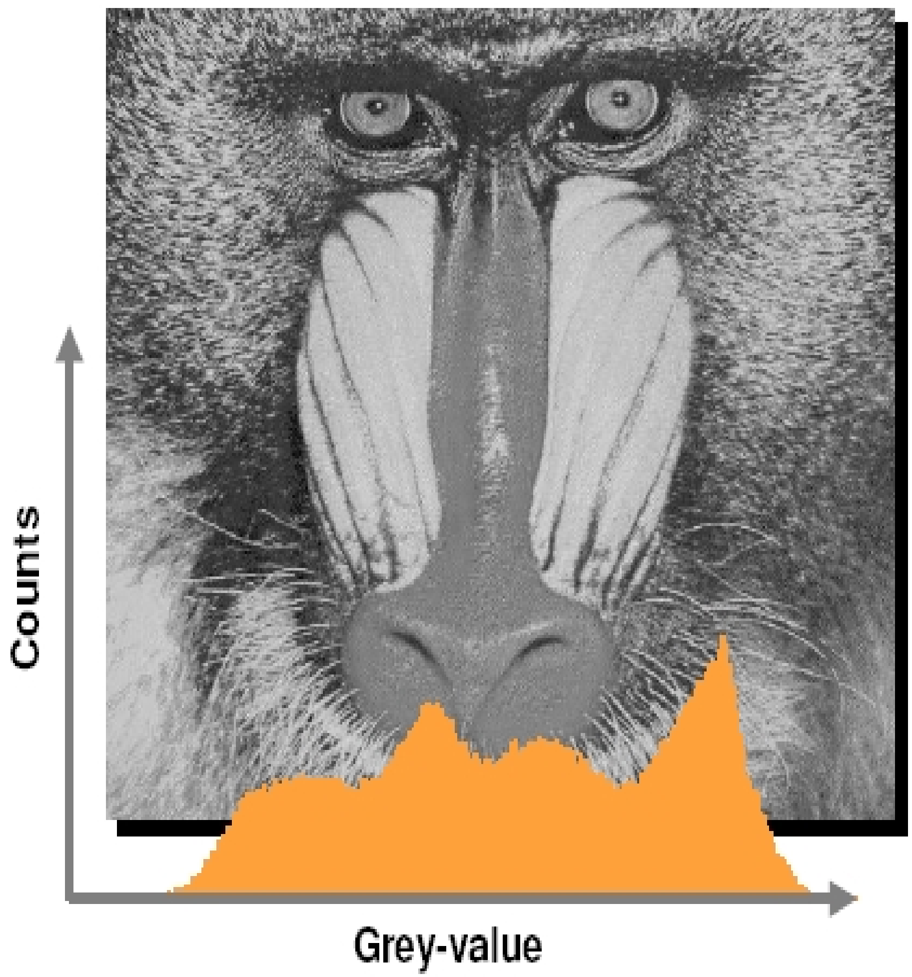






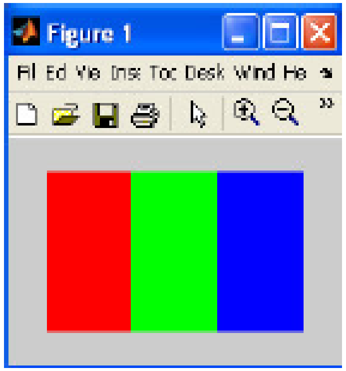
Handling Images in Matlab

Histograms



Handling Images in Matlab

Histograms



h = imhist(R)

bar(h)