



uff Universidade
Federal
Fluminense

Computer Graphics for Engineering



numsim

Numerical simulation
in technical sciences

Geometric Transformations

Luiz Fernando Martha
André Pereira

Graz, Austria
June 2014

Geometric Transformation

$$\begin{aligned} T: \mathbf{R}^n &\rightarrow \mathbf{R}^m \\ \{P\} &\rightarrow \{P'\} = T\{P\} \end{aligned}$$

Linear Transformation:

$$T(\alpha \{P\} + \beta \{Q\}) = \alpha T\{P\} + \beta T\{Q\}$$

$$\begin{aligned} \forall \alpha, \beta \in \mathbf{R} \\ \{P\}, \{Q\} \in \mathbf{R}^n \end{aligned}$$

$$\Rightarrow \begin{cases} T\{0\} = \{0\} \\ \{P'\} = T\{P\} = [M]\{P\} \end{cases}$$

Geometric Transformations

Example: $\mathbb{R}^3 \rightarrow \mathbb{R}^2$

$$T: \mathbb{R}^3 \rightarrow \mathbb{R}^2$$

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} \rightarrow \begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$$

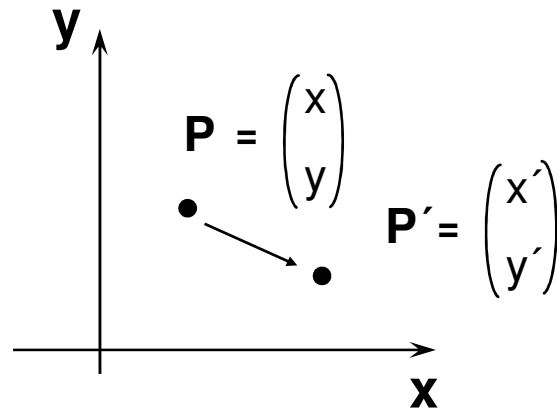
$$T \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} = [M] \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} m_{11} \\ m_{21} \end{Bmatrix}$$

$$T \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} = [M] \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} = \begin{Bmatrix} m_{12} \\ m_{22} \end{Bmatrix}$$

$$T \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} = [M] \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} = \begin{Bmatrix} m_{13} \\ m_{23} \end{Bmatrix}$$

Linear Transformations

$$\mathbb{R}^2 \rightarrow \mathbb{R}^2$$



$$T(a_1 \mathbf{P}_1 + a_2 \mathbf{P}_2) = a_1 T(\mathbf{P}_1) + a_2 T(\mathbf{P}_2)$$

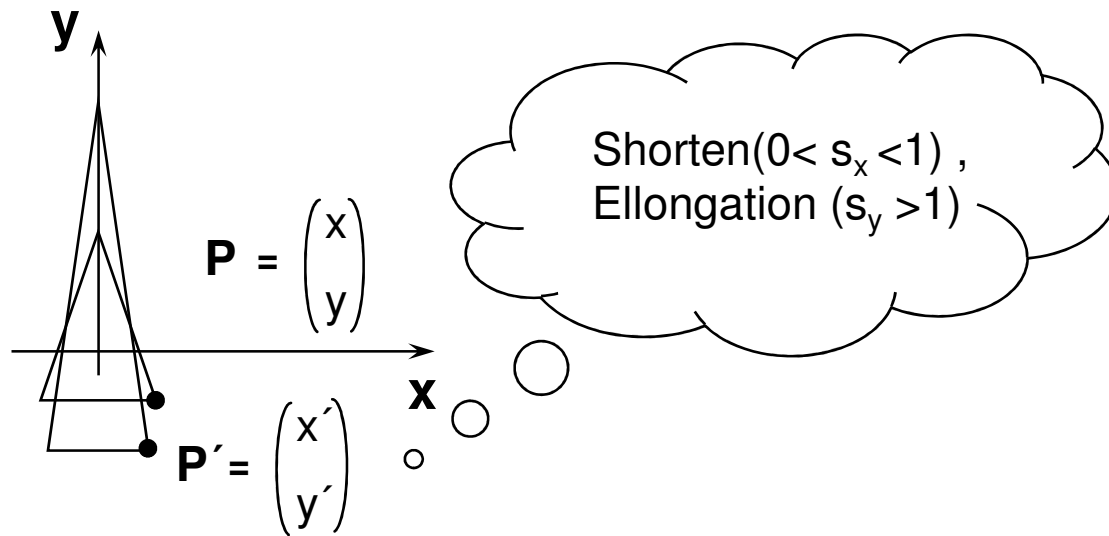
Show that:

A) $T(\mathbf{0}) = \mathbf{0}$

B)

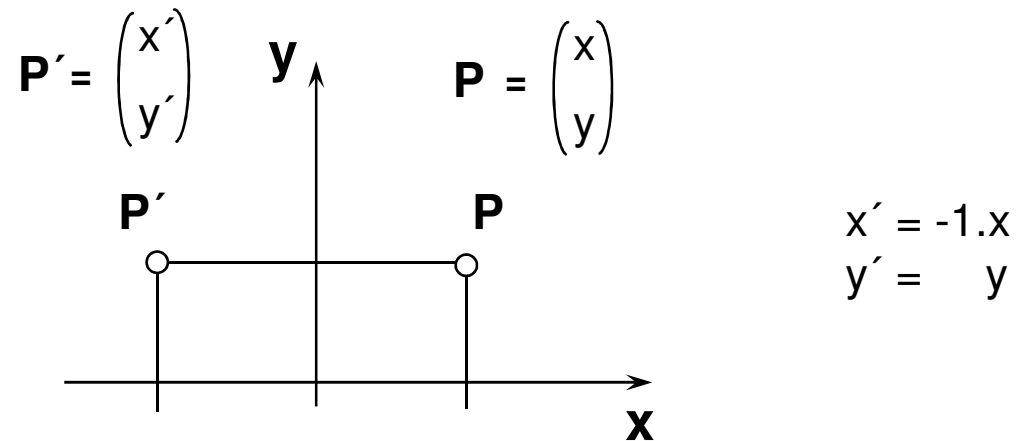
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Linear Transformations (scale with respect to origin)



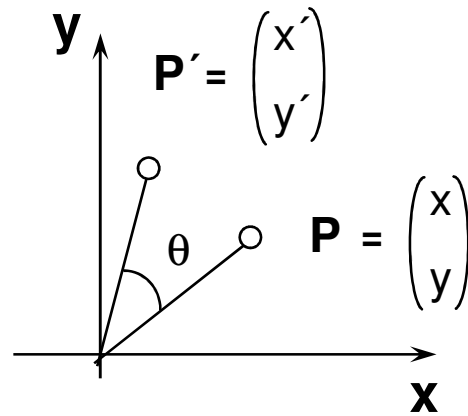
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Linear Transformations (mirror with respect to y axis)



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

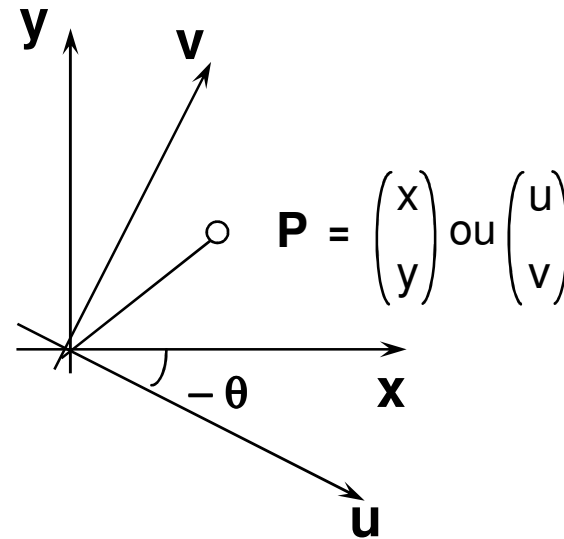
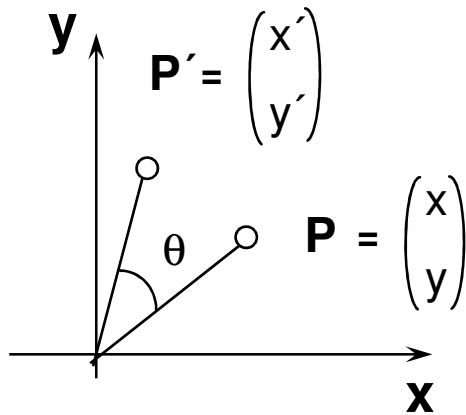
Linear Transformations (rotation with respect to origin)



$$\begin{aligned}x' &= x \cdot \cos \theta - y \cdot \sin \theta \\y' &= x \cdot \sin \theta + y \cdot \cos \theta\end{aligned}$$

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

Linear Transformations (rotation -vs- base change)



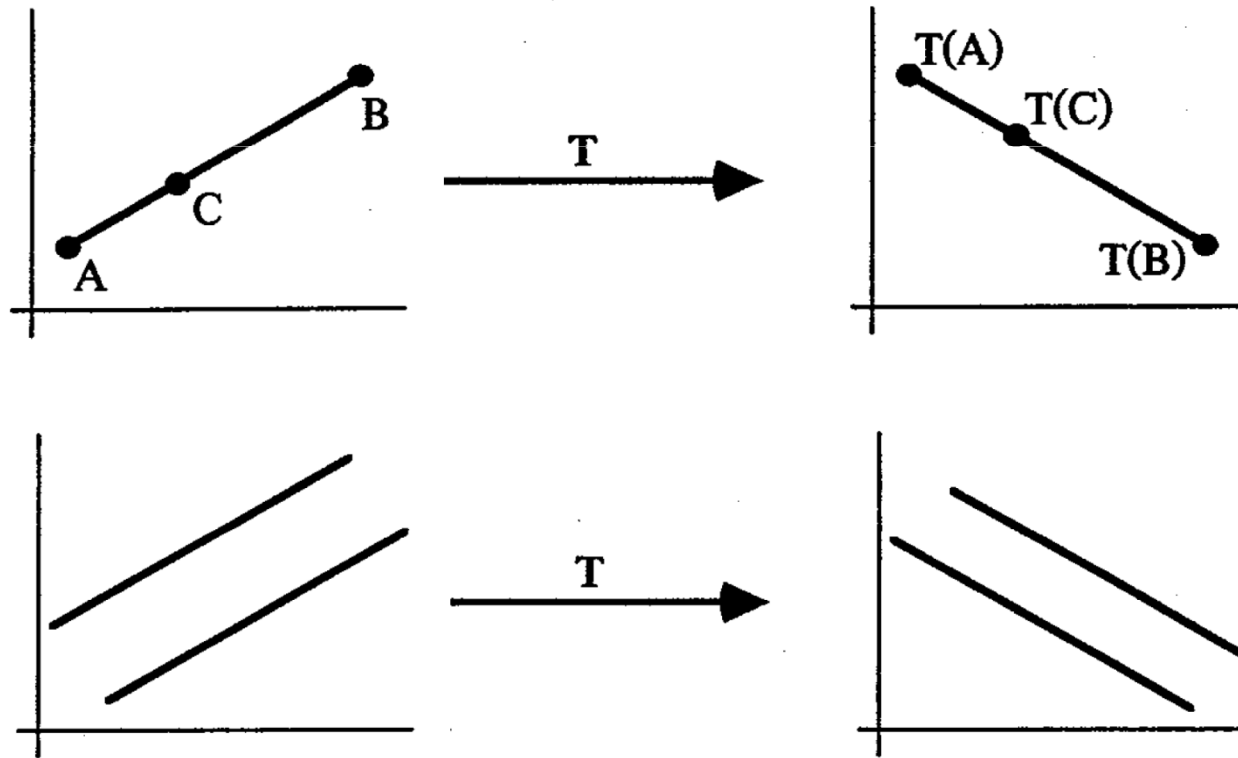
$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

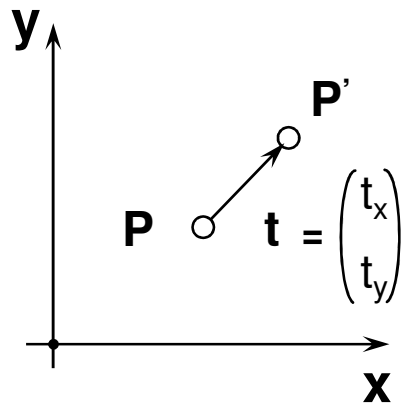
Rotation of a point by an angle θ has the same effect as base axis rotation by an angle $-\theta$.
The matrix that implements the transformation is constructed assembling the coefficients of the new base normalized vectors in its rows.

Linear Transformations Properties

Straight Line \xrightarrow{T} Straight Line
(Very important property in Computer Graphics)



Linear Transformations (translation)



$$\mathbf{P}' = \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

← Cannot be written \boxtimes in this manner

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

← Not good for \boxtimes implementation

Affine Transformations

(linear transformations with translation)

Generalizing:

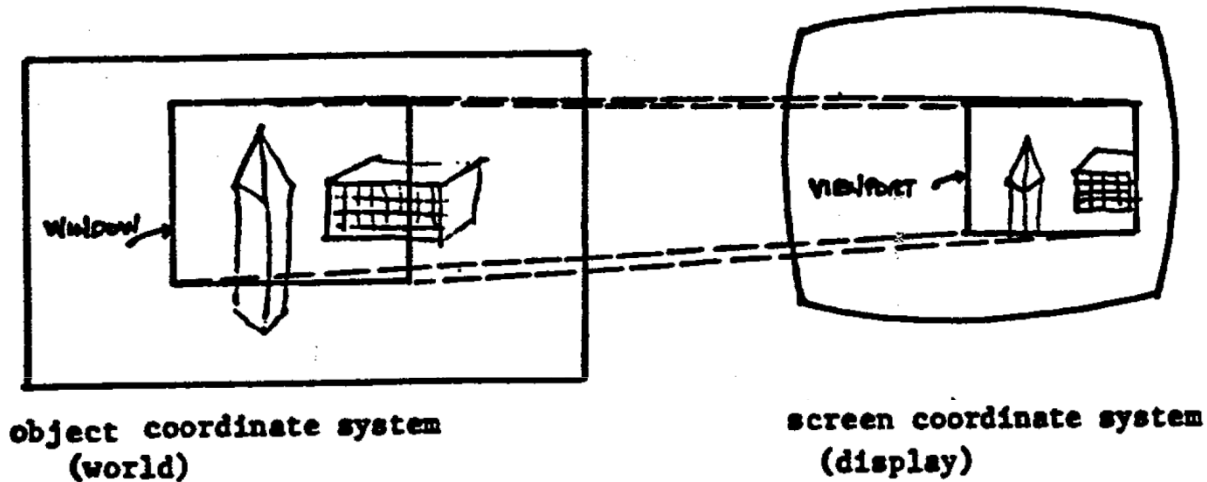
$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} + \begin{Bmatrix} e \\ f \end{Bmatrix}$$

$T\{0\} \neq \{0\} \quad \Rightarrow$ This is not a linear transformation
However, it preserves colinearity,
ratio and parallelism

Transformations of this form are called: Affine Transformations

Transformation

Window-Viewport



A *window* defines a rectangular area in world coordinates. You can define the window to be larger than, the same size as, or smaller than the actual range of data values, depending on whether you want to show all of the data or only part of the data.

A *viewport* defines in normalized coordinates a rectangular area on the display device where the image of the data appears. You can have your graph take up the entire display device or show it in only a portion, say the upper right part.

Definitions

World Coordinate System - This is object space or the space in which the application model is defined.

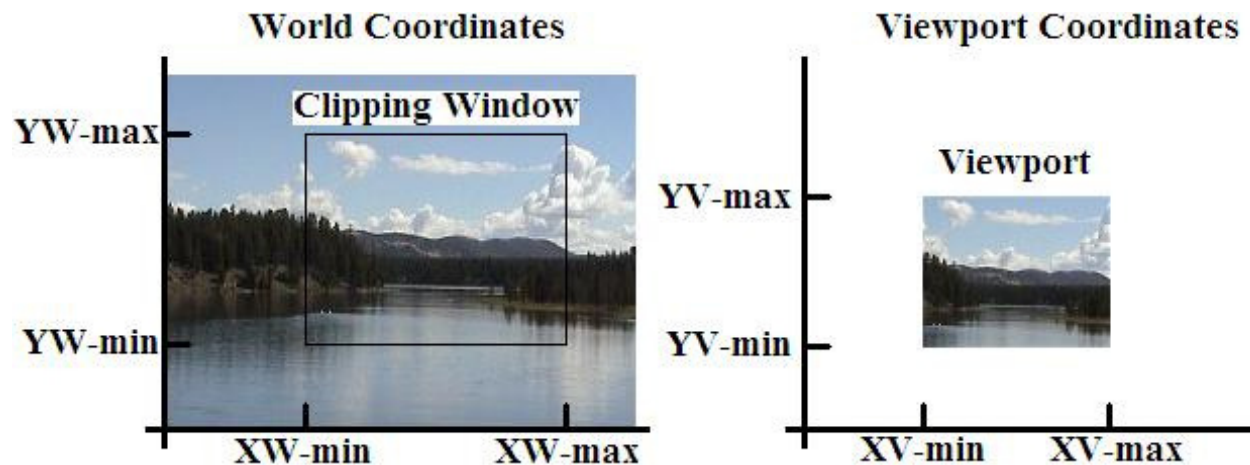
Screen Coordinate System - The space in which the image is displayed.

World Window (or clipping) - This is the rectangle in the world defining the region that is to be displayed.

Interface Window - The window opened on the raster graphics screen in which the image will be displayed.

Viewport - The rectangular portion of the interface window that defines where the image will actually appear (usually the entire interface window but in some cases modified to be a portion of the interface window).

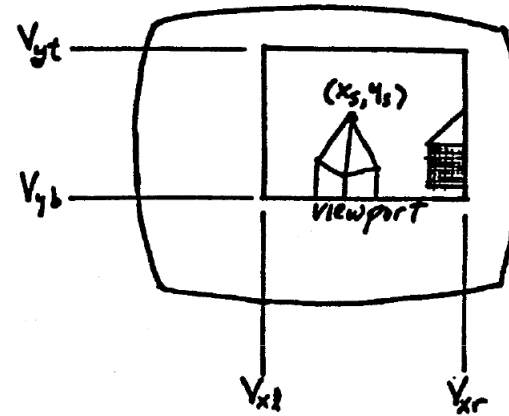
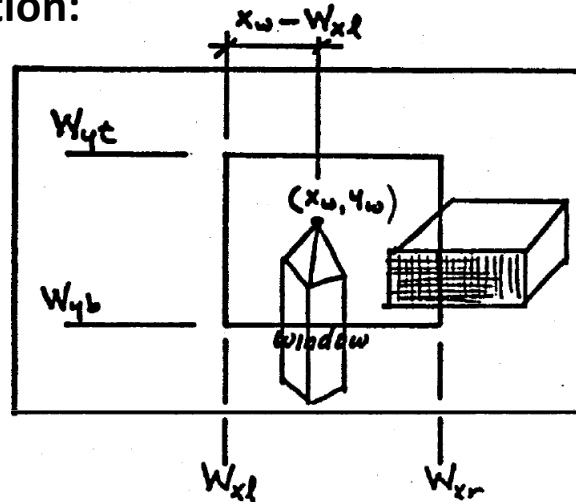
Viewing Transformation - The process of mapping a world window in *World Coordinates* to the *Viewport*.



(Source: Hankins, 2014 – Lecture Notes on Computer Graphics at MTSU)

Transformation Window-Viewport is an Affine Transformation

Proportion:



$$x_s = \underbrace{\frac{(x_w - W_{xl})}{(W_{xr} - W_{xl})}}_{\text{fraction of displacement of full viewport}} \cdot \underbrace{(V_{xr} - V_{xl})}_{\text{viewport width}} + \underbrace{V_{xl}}_{\text{viewport offset}}$$

$$x_s = \frac{(V_{xr} - V_{xl})}{(W_{xr} - W_{xl})} (x_w - W_{xl}) + V_{xl}$$

$$y_s = \frac{(y_w - W_{yb})}{(W_{yt} - W_{yb})} \cdot (V_{yt} - V_{yb}) + V_{yb}$$

$$y_s = \frac{(V_{yt} - V_{yb})}{(W_{yt} - W_{yb})} (y_w - W_{yb}) + V_{yb}$$

$$\begin{Bmatrix} x_s \\ y_s \end{Bmatrix} = \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} \begin{Bmatrix} x_w \\ y_w \end{Bmatrix} + \begin{Bmatrix} e \\ f \end{Bmatrix} = \begin{bmatrix} a & 0 & e \\ 0 & d & f \end{bmatrix} \begin{Bmatrix} x_w \\ y_w \\ 1 \end{Bmatrix}$$