

CSE251

Basics of Computer Graphics

Module: Geometry

Avinash Sharma
Spring 2017

Overview

Preliminary Concepts

Translations and Rotations

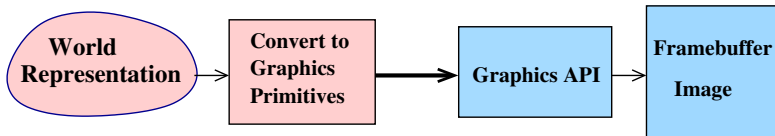
Other Transforms

Composite Transformations

Transformations About A Point

Points and Frames

Graphics Process



- ▶ **Model** the desired world in your head.
- ▶ **Represent** it using natural structures in the program. Convert to standard primitives supported by the API
- ▶ **Processing** is done by the API. Converts the primitives in stages and forms an image in the framebuffer
- ▶ The image is displayed automatically on the device

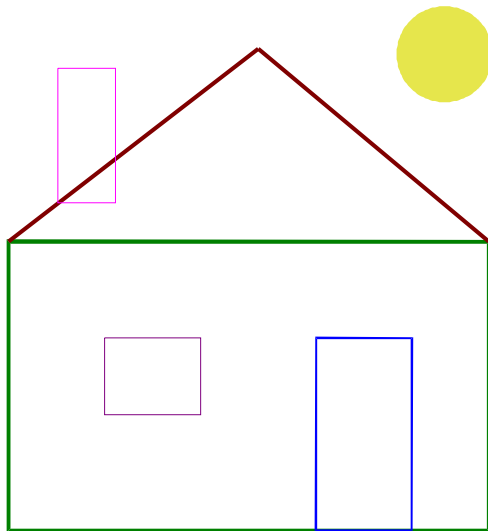
How to Draw A House?

- ▶ Compose out of basic shapes

<code>drawRectangle(v1, v2, v3, v4);</code>	<code>// Main part</code>
<code>drawTriangle(v2, v3, v5);</code>	<code>// Roof</code>
<code>drawRectangle(...);</code>	<code>// Door</code>
<code>drawRectangle(...);</code>	<code>// Window</code>
<code>drawRectangle(...);</code>	<code>// Chimney</code>
<code>drawCircle(...);</code>	<code>// Sun</code>

- ▶ That's all, really!

Resulting House



Graphics Primitives

- ▶ Points: 2D or 3D. (x, y) or (x, y, z) .
- ▶ Lines: specified using end-points
- ▶ Triangles/Polygons: specified using vertices
- ▶ Why not **circles, ellipses, hyperbolas**?

Graphics Attributes

- ▶ Colour, Point width.
- ▶ Line width, Line style.
- ▶ Fill, Fill Pattern.

Point Representation

- ▶ A point is represented using 2 or 3 numbers $(x, y, [z])$ that are the projections on to the respective coordinate axes.
- ▶ Fundamental shape-defining primitive in most Graphics APIs. Everything else is built from it!
- ▶ Represented using **byte, short, int, float, double**, etc.
- ▶ The scale and unit are application dependent. Could be angstroms or lightyears!
- ▶ Points undergo transformations:
Translations, Rotations, Scaling, Shearing.

3D Coordinates

► Cartesian: (x, y, z) .

► Polar: (ρ, θ, ϕ)

► $z =$

$y =$

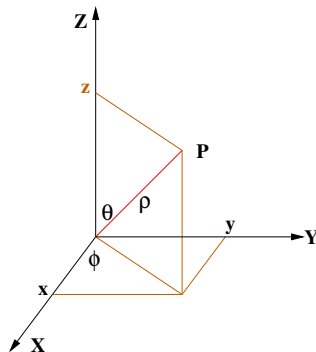
$x =$

► $\rho =$

$\phi =$

$\theta =$

► Elevation: θ , Azimuthal: ϕ



3D Coordinates

► Cartesian: (x, y, z) .

► Polar: (ρ, θ, ϕ)

► $z = \rho \cos \theta$

$y = \rho \sin \theta \sin \phi$

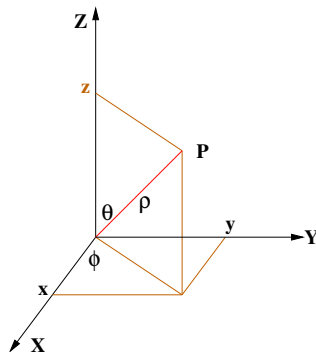
$x = \rho \sin \theta \cos \phi$

► $\rho^2 = x^2 + y^2 + z^2$

$\phi = \tan^{-1}(y/x)$

$\theta = \tan^{-1}(\sqrt{x^2 + y^2}/z)$

► Elevation: θ , Azimuthal: ϕ



Translation

- ▶ Translate a point $P = (x, y, [z])$ by $(a, b, [c])$.
- ▶ Points coordinates become $P' = (?, ?, ?)$.
- ▶ In vector form, $P' = ?$.

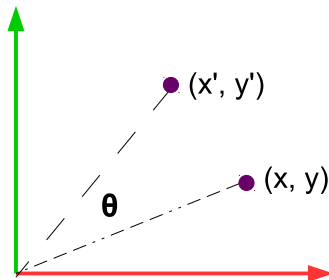
Translation

- ▶ Translate a point $P = (x, y, [z])$ by $(a, b, [c])$.
- ▶ Points coordinates become $P' = (x + a, y + b, [z + c])$.
- ▶ In vector form, $P' = P + T$, where $T = (a, b, [c])$.
- ▶ Distances, angles, parallelism are all maintained.

2D Rotation

- ▶ Rotate about origin CCW by θ .
- ▶ $x' = ?$, $y' = ?$
- ▶ Matrix notation: $P' = R P$

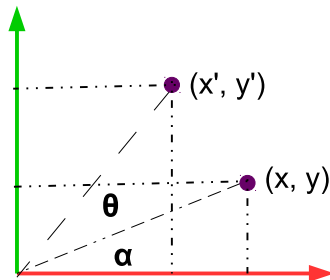
$$\begin{bmatrix} x \\ y \end{bmatrix}' = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



2D Rotation

- ▶ Rotate about origin CCW by θ .
- ▶ $x' = ?$, $y' = ?$
- ▶ Matrix notation: $P' = R P$

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

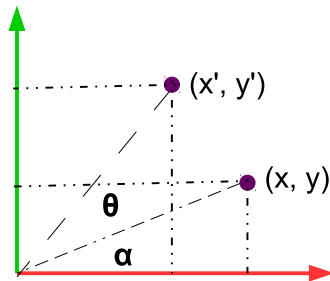


2D Rotation

- ▶ Rotate about origin CCW by θ .
- ▶ $x' = x \cos \theta - y \sin \theta$,
 $y' = x \sin \theta + y \cos \theta$.

- ▶ Matrix notation: $P' = R P$

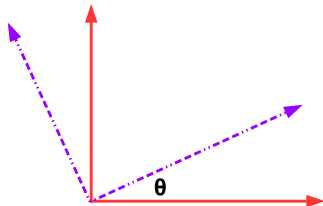
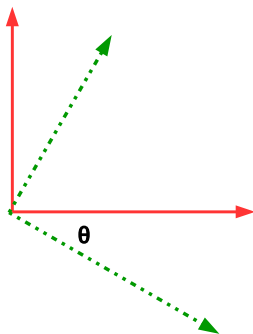
$$\begin{bmatrix} x \\ y \end{bmatrix}' = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



2D Rotation: Observations

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

- ▶ Orthonormal: $R^{-1} = R^T$
- ▶ Rows: vectors that **rotate to** coordinate axes
- ▶ Cols: vectors coordinate axes **rotate to**
- ▶ Invariants: distances, angles, parallelism.



3D Rotations

- ▶ Rotation could be about any axis in 3D!
- ▶ About Z-axis: Just like 2D rotation case. Z-coordinates don't change anyway.
- ▶ X-Y coordinates change exactly the same way as in 2D.
- ▶ CCW +ve, when looking into the **arrowhead**

$$R_z(\theta) = ??$$

3D Rotations

- ▶ Rotation could be about any axis in 3D!
- ▶ About Z-axis: Z-coordinates don't change anyway

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

- ▶ CCW +ve; orthonormal; length preserving
- ▶ Rows: vectors that rotate onto axes; columns: vectors that axes rotate into....

3D Rotations

$$R_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

- ▶ CCW +ve; orthonormal
- ▶ Rows: vectors that rotate onto axes; columns: vectors that axes rotate into....
- ▶ Rotation about an arbitrary axis, for example, $[1, 1, 1]^T$??
Coming soon

Non-uniform Scaling

- ▶ Scale along X, Y, Z directions by s , u , and t .
- ▶ $x' = s x$, $y' = u y$, $z' = t z$.
- ▶ We are more comfortable with $P' = S P$ or

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}' = \begin{bmatrix} s & 0 & 0 \\ 0 & u & 0 \\ 0 & 0 & t \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- ▶ Invariants: parallelism, ratios of lengths in any direction (Angles also for uniform scaling.)

Shearing

- Add a little bit of x to y or other combinations

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}' = \begin{bmatrix} 1 & x_y & x_z \\ y_x & 1 & y_z \\ z_x & z_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- One of $x_y, x_z, y_x, y_z, z_x, z_y \neq 0$. Rectangles can become parallelograms.
- Invariants: parallelism, ratios of lengths in any direction.

Reflection

- ▶ Negative entries in a matrix indicate reflection.

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

- ▶ Reflection needn't be about a coordinate axis alone

General Transformation

- ▶ Rotation, scaling, shearing, and reflection: **Matrix-vector** product. Vectors get transformed into other vectors
- ▶ Translation alone is a **vector-vector** addition
- ▶ Sequence of: Translation, rotation, scaling, translation and rotation: $\mathbf{P}' = \mathbf{R}_2 [\mathbf{S} \mathbf{R}_1 (\mathbf{P} + \mathbf{t}_1) + \mathbf{t}_2]$
- ▶ If translation is also a matrix-vector product, we can combine above transformations into a single matrix:
 $\mathbf{P}' = \mathbf{R}_2 \mathbf{T}_2 \mathbf{S} \mathbf{R}_1 \mathbf{T}_1 \mathbf{P} = \mathbf{M} \mathbf{P}$
- ▶ How? Answer: **homogeneous coordinates**!

Homogeneous Coordinates

- ▶ Add a *non-zero scale factor* w to each coordinate.
A 2D point is represented by a vector $[x \ y \ w]^T$
- ▶ $[x \ y \ w]^T \equiv (x/w, y/w)$.
- ▶ Translate $[x \ y]^T$ by $[a \ b]^T$ to get $[x + a \ y + b]^T$

$$\begin{bmatrix} x + a \\ y + b \\ 1 \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates

- ▶ Add a *non-zero scale factor* w to each coordinate.

A 2D point is represented by a vector $[x \ y \ w]^T$

- ▶ $[x \ y \ w]^T \equiv (x/w, y/w)$

- ▶ Translate $[x \ y]^T$ by $[a \ b]^T$ to get $[x + a \ y + b]^T$

$$\begin{bmatrix} x + a \\ y + b \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- ▶ Now, translation is also: $\mathbf{P}' = \mathbf{T} \mathbf{P}$, a matrix-vector product and a linear operation.

Homogeneous Coordinates

- ▶ Add a *non-zero scale factor* w to each coordinate.
A 2D point is represented by a vector $[x \ y \ w]^T$
- ▶ $[x \ y \ w]^T \equiv (x/w, y/w)$.
- ▶ Now, translation is also: $\mathbf{P}' = \mathbf{T} \mathbf{P}$
- ▶ For a point: Rotation followed by translation followed by scaling, followed by another rotation: $\mathbf{P}' = \mathbf{R}_2 \mathbf{S} \mathbf{T} \mathbf{R}_1 \mathbf{P}$.
- ▶ Similarly for 3D. Points represented by: $[x \ y \ z \ w]^T$.
- ▶ All matrices are 3×3 in 2D. Last row is $[0 \ 0 \ 1]$.
- ▶ All matrices are 4×4 in 3D. Last row is $[0 \ 0 \ 0 \ 1]$.

Homogeneous Representation

- ▶ Convert to a 4-vector with a scale factor as fourth.
 $(x, y, z) \equiv [kx \ ky \ kz \ k]^T$ for any $k \neq 0$.
- ▶ Translation, rotation, scaling, shearing, etc. become linear operations represented by 4×4 matrices.
- ▶ What does $[x \ y \ z \ 0]^T$ mean?
- ▶ $[a \ b \ c \ d]^T$ could be a point or a plane. Lines are specified using two such vectors, either as join of two points or intersection of two planes!

Transformation Matrices: Rotations

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

$$R_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_z = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- CCW +ve; orthonormal; length preserving; rows give direction vectors that rotate onto each axis; columns

Translation, Scaling, Composite

$$T(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S(a, b, c) = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- ▶ A sequence of transforms can be represented using a composite matrix: $\mathbf{M} = \mathbf{R}_x \mathbf{T} \mathbf{R}_y \mathbf{S} \mathbf{T} \dots$
- ▶ Operations are not commutative, but are associative.
- ▶ \mathbf{RT} and \mathbf{TR} ??

Rotation and Translation

► $T_{4 \times 4} = \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$ and

$$R_{4 \times 4} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$$

► $T R = \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} = ?$

► $R T = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = ?$

Rotation and Translation

$$\blacktriangleright T_{4 \times 4} = \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad \text{and} \quad R_{4 \times 4} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$$

$$\blacktriangleright T R = \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$$

$$\blacktriangleright R T = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{Rt} \\ \mathbf{0} & 1 \end{bmatrix}$$

$$\blacktriangleright TR = RT \text{ if: (a) } \mathbf{R} = \mathbf{I} \text{ or (b) } \mathbf{t} = \mathbf{0} \text{ or (c) } \mathbf{Rt} = ?$$

Rotation and Translation

- ▶ $T_{4 \times 4} = \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$ and $R_{4 \times 4} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$
- ▶ $TR = \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$
- ▶ $RT = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{Rt} \\ \mathbf{0} & 1 \end{bmatrix}$
- ▶ $TR = RT$ if: (a) $\mathbf{R} = \mathbf{I}$ or (b) $\mathbf{t} = \mathbf{0}$ or (c) $\mathbf{Rt} = \mathbf{t}$
- ▶ When is $\mathbf{Rt} = \mathbf{t}$? \mathbf{t} is an eigenvector of \mathbf{R}

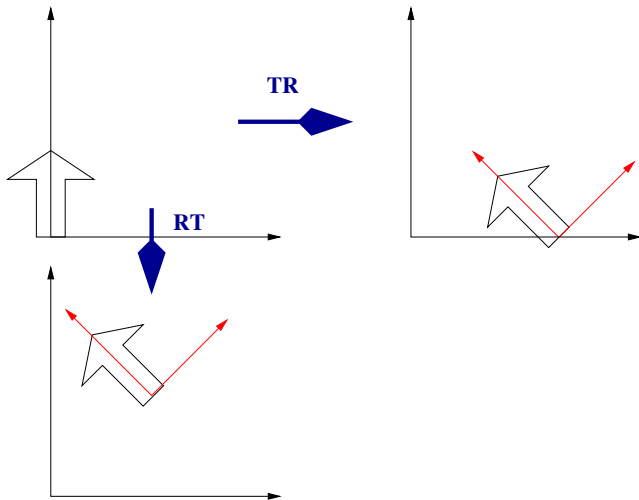
Commutativity

- ▶ Translations are commutative: $\mathbf{T}_1\mathbf{T}_2 = \mathbf{T}_2\mathbf{T}_1$
- ▶ Scaling is commutative: $\mathbf{S}_1\mathbf{S}_2 = \mathbf{S}_2\mathbf{S}_1$
- ▶ Are rotations commutative? $\mathbf{R}_1\mathbf{R}_2 \stackrel{?}{=} \mathbf{R}_2\mathbf{R}_1$
- ▶ What would be an example?
Consider the effect on Z-axis of:

Commutativity

- ▶ Translations are commutative: $\mathbf{T}_1\mathbf{T}_2 = \mathbf{T}_2\mathbf{T}_1$
- ▶ Scaling is commutative: $\mathbf{S}_1\mathbf{S}_2 = \mathbf{S}_2\mathbf{S}_1$
- ▶ Are rotations commutative? $\mathbf{R}_1\mathbf{R}_2 \neq \mathbf{R}_2\mathbf{R}_1$
- ▶ Consider the effect on Z-axis of $\mathbf{R}_x(90)\mathbf{R}_y(90)$ and $\mathbf{R}_y(90)\mathbf{R}_x(90)$
- ▶ $\mathbf{RT} \neq \mathbf{TR}$. (If translation is not parallel to rotation axis)
- ▶ Consider: $\mathbf{R}(\pi/4)$ and $T(5, 0)$.
Where does the origin $(0, 0)$ go in \mathbf{TR} and \mathbf{RT} ?

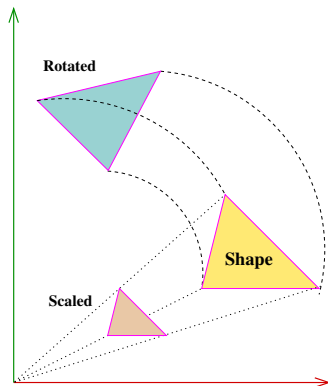
TR and RT



TR keeps it on X axis to $(5, 0)$. **RT** takes it to $(\frac{5}{\sqrt{2}}, \frac{5}{\sqrt{2}})$.

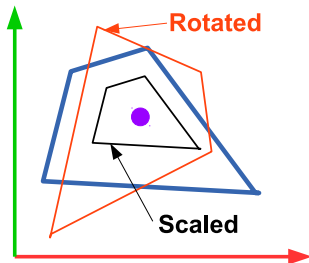
Objects Away from Origin

- ▶ Object “**translates**” when rotated or scaled!!
- ▶ Default: Perform these **about the origin**
- ▶ How do we transform points “**in place**”?
- ▶ Rotate or scale about the centroid of the object. Or about an arbitrary point
- ▶ How?



Transformations About A Point

- ▶ Rotating about point P
 - Align P with origin
 - Rotate/scale about origin
 - Translate back
- ▶ Rotation:
 $\mathbf{R}_C(\theta) = \mathbf{T}(C) \mathbf{R} \mathbf{T}(-C)$
- ▶ Scaling:
 $\mathbf{S}_C() = \mathbf{T}(C) \mathbf{S}() \mathbf{T}(-C)$
- ▶ A transformation \mathbf{M} :
 $\mathbf{M}_C = \mathbf{T}(C) \mathbf{M} \mathbf{T}(-C)$



R, T Operations on Points



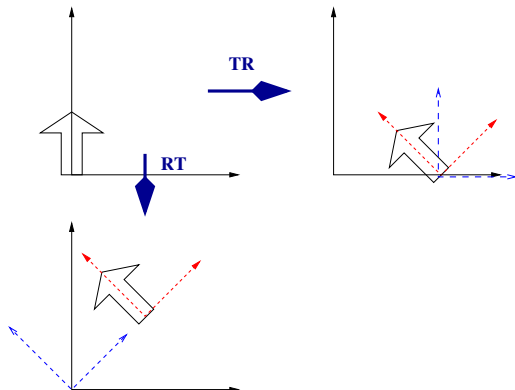
- ▶ **T(5,0) R($\pi/4$)**: Impact on a point:
 - ▶ R($\pi/4$): (Point stays at (0, 0))
 - ▶ T(5, 0) : (Point goes to (5, 0))
- ▶ **R($\pi/4$) T(5,0)**: Impact on the point:
 - ▶ T(5, 0): (Point moves to (5, 0))
 - ▶ R($\pi/4$). (Point rotates about origin)
- ▶ All points on the shape undergo the same motions and get new coordinates

R, T Operations on Frames

- ▶ **T(5,0) R($\pi/4$)**: Impact on coordinate frame:
 - ▶ T(5, 0): (Origin shifted to (5, 0))
 - ▶ R($\pi/4$). (Axes rotated at new origin)
- ▶ **R($\pi/4$) T(5,0)**: Impact on coordinate frame:
 - ▶ R($\pi/4$): (Axes rotate by 45 degrees))
 - ▶ T(5, 0). (Point moves to (5, 0) in new axes)
- ▶ Frames move around, giving new coordinates to points on objects!!

Relating Coordinate Frames

- ▶ $T(5, 0)$ and $R(\pi/4)$
- ▶ Start: Black axes
- Next: Blue axes
- Last: Red axes



Points and Frames in General

- ▶ Points go through changes in a common coordinate frame when a sequence of transformations is **viewed from right to left**
- ▶ Coordinate system goes through the same transformations when the sequence is **viewed from left to right**
- ▶ Composite transformations $P' = M_1 M_2 M_3 P$ relates the coordinates in successive coordinate frames as we go from left to right, starting with $X'Y'$ coordinate frame to finally the XY frame.

Transforming the World Reference

- ▶ Consider $P_4 = \mathbf{M}_4\mathbf{M}_3\mathbf{M}_2\mathbf{M}_1 P_0$
- ▶ Point P_0 undergoes 4 operations and get coordinates P_4
- ▶ Imagine the point having coordinates P_1, P_2, P_3 after operations $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$
- ▶ We can also visualize coordinate frames $\Pi_4, \Pi_3, \Pi_2, \Pi_1, \Pi_0$ in which a point has coordinates P_4 to P_0 respectively
- ▶ Operation \mathbf{M}_i represents a change in coordinates from Π_i to Π_{i-1} , resulting in new labels for the point.

Let us look at Ourselves

- ▶ Model IIIT Campus as a whole. Campus is our “world”
- ▶ Global coordinate frame Π_G for the campus: at the Gate
- ▶ Buildings: Himalaya, Vindhya, Bakul, Parul, ..., Palash. Each has a natural coordinate frame. Π_H is Himalaya's
- ▶ Himalaya has several rooms: B105, B204, B205, B304, etc., with own coordinate frames. Π_C is of B205 (our class)
- ▶ B205 has 55 desks, with coord frames Π_{Di} for desk i
- ▶ Desks are identical in geometry; the coord frame Π_{Di} places each in its location.

Consider a Desk

- ▶ Consider a corner point P of desk 37, with coordinates $(200, 30, 100)$ in Π_{D37} . That is: $P_{D37} = (200, 30, 100)$
- ▶ Since our world is the campus, we have to ultimately describe everything in the coordinate frame Π_G

$$P_G = M_{GH} M_{HC} M_{CD37} P_{D37}$$

- ▶ M_{GH} aligns Π_G to Π_H . M_{HC} aligns Π_H to Π_C .
 M_{CD37} aligns Π_C to Π_{D37}

- ▶ $P_G = M_{GH} \overset{P_H}{\mid} M_{HC} \overset{P_C}{\mid} M_{CD37} \overset{P_{D37}}{\mid} P$ (for any point P on Desk37)

- ▶ We can place a given desk in any **building, room, place!**