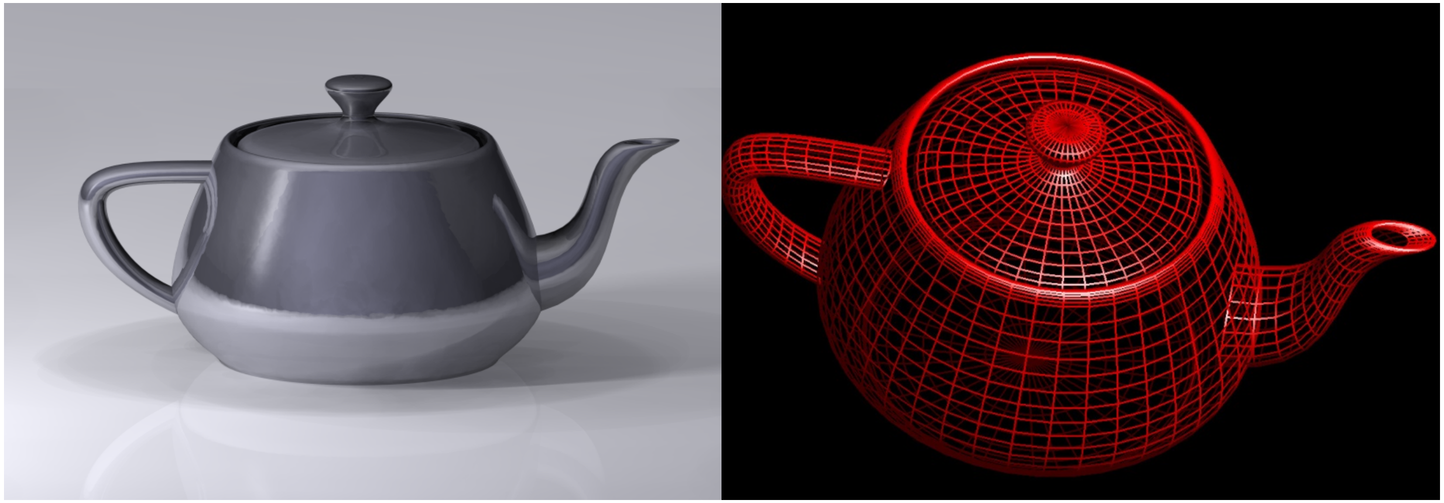


Foundations of Computer Graphics

Saurabh Ray

Reading

Reading for Lectures 6 and 7 : Sections 4.3, 4.7-4.10.

Review of Basic Linear Algebra

When do we say that a set of vectors are linearly independent?

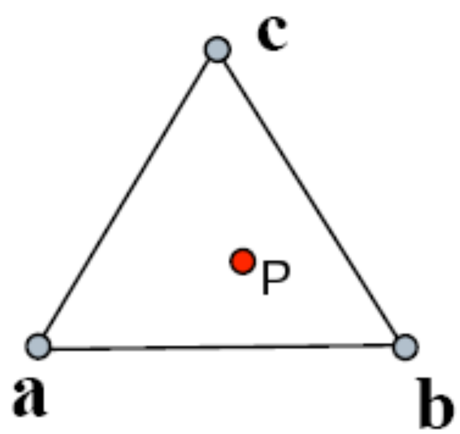
What is a basis?

What is an orthonormal basis?

What is the difference between a coordinate system
and a coordinate frame?

What is the difference between affine combination
and convex combination of points?

Barycentric Coordinates

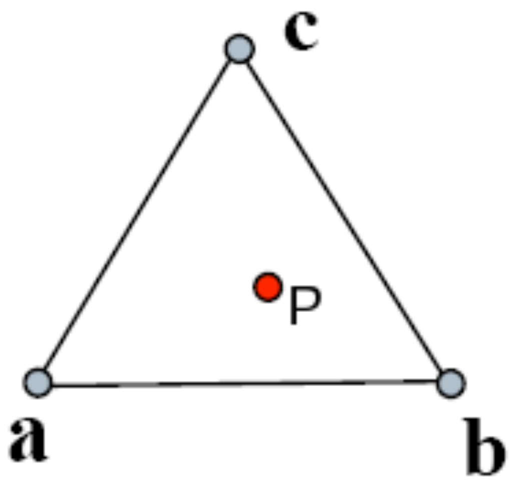


We are given a triangle whose corners are $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$.
P is some point in the triangle.

Claim: P can be written uniquely as $\alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}$,
where $\alpha, \beta, \gamma \geq 0$ and $\alpha + \beta + \gamma = 1$.

(α, β, γ) is the barycentric coordinate of P in the triangle.

Barycentric Coordinates



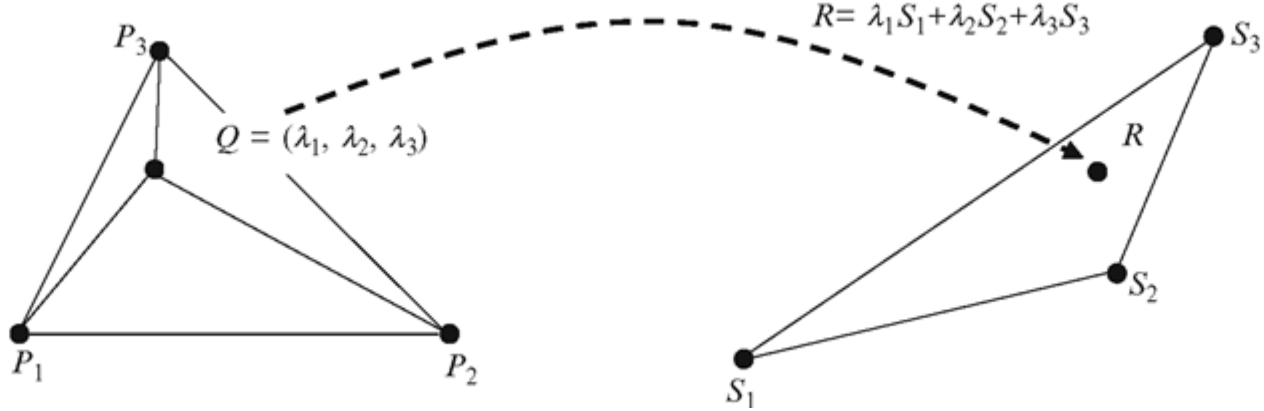
Barycentric Coordinates of P:

$$(\alpha, \beta, \gamma)$$

During a transformation, if a, b and c are mapped to a', b' and c', then P is mapped to $\alpha a' + \beta b' + \gamma c'$.

Barycentric Coordinates

Our meshes consist of a lot of triangles, whose vertices we transform. This is how we know where the points in the interior of the triangles go.



Not only positions but also **colors** and **normals** are interpolated this way via *varying* variables.

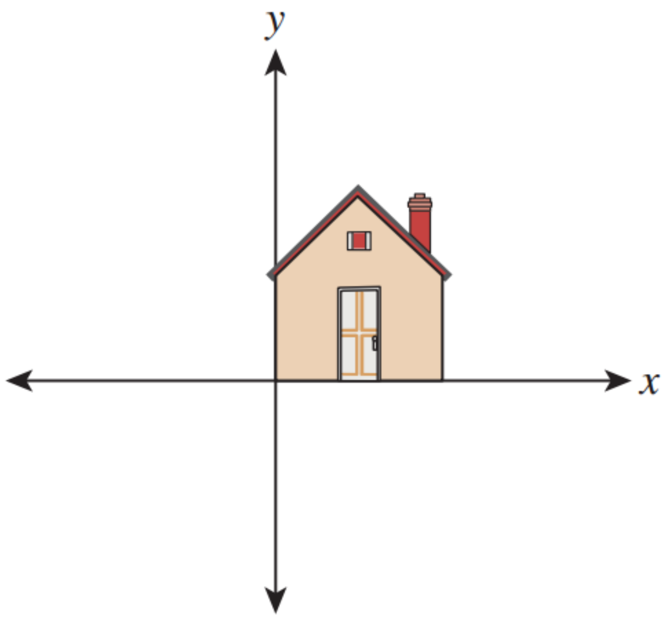
Transformations

Typical operations required to put a geometric model in a scene:

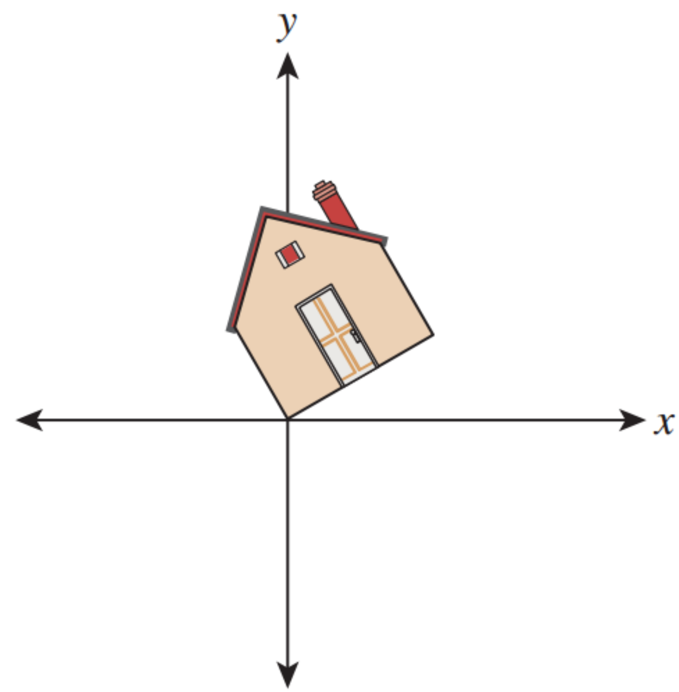
- Translation
- Rotation
- Scaling

In doing these operations, we will treat points and vectors as elements of $\mathbb{R}^d$, i.e., as tuples of numbers.

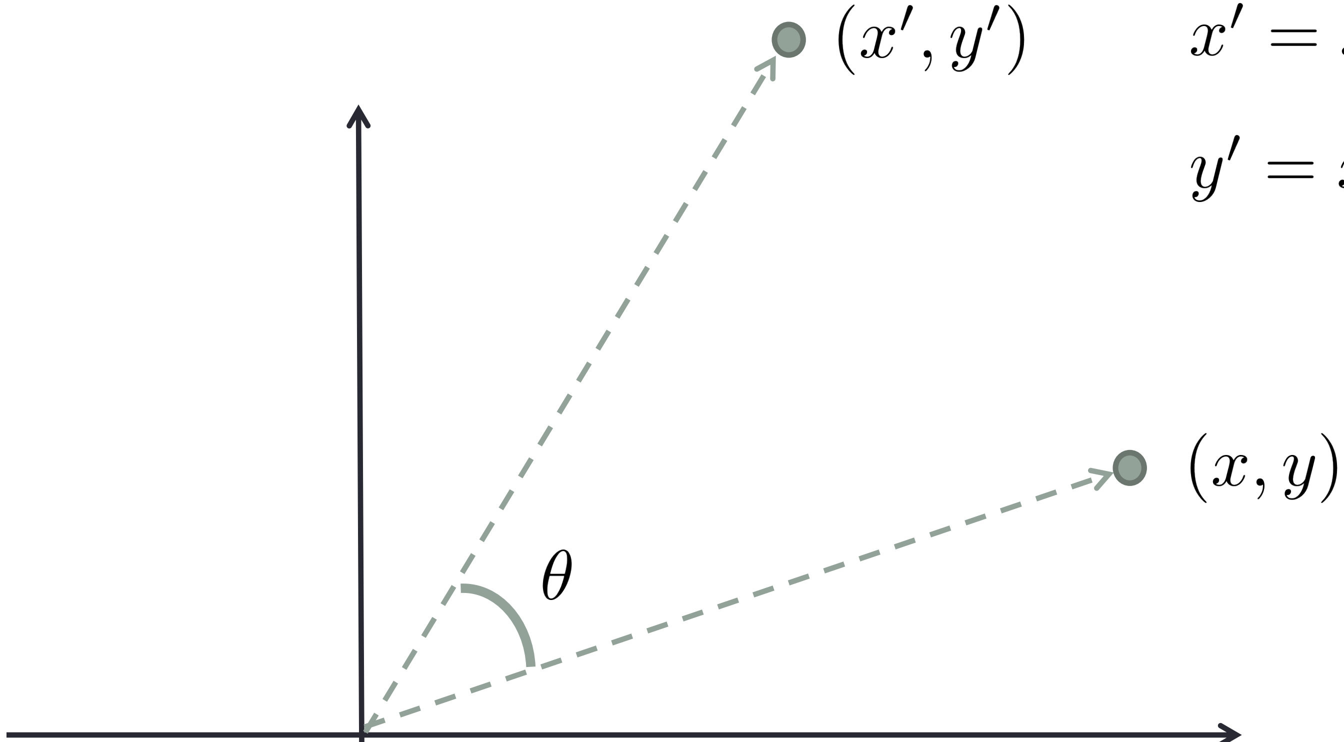
Rotation



Rotation
by 30°



Rotation by θ (counter-clockwise)



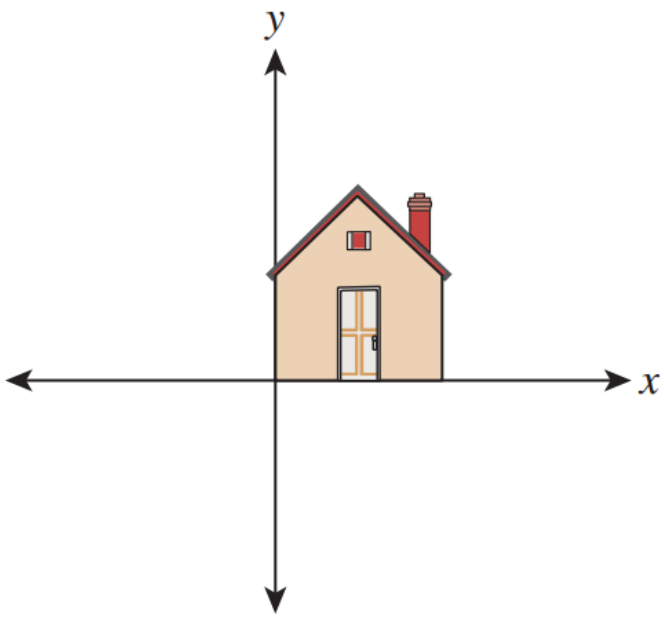
$$x' = x\ cos\theta - y \sin\ \theta$$

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

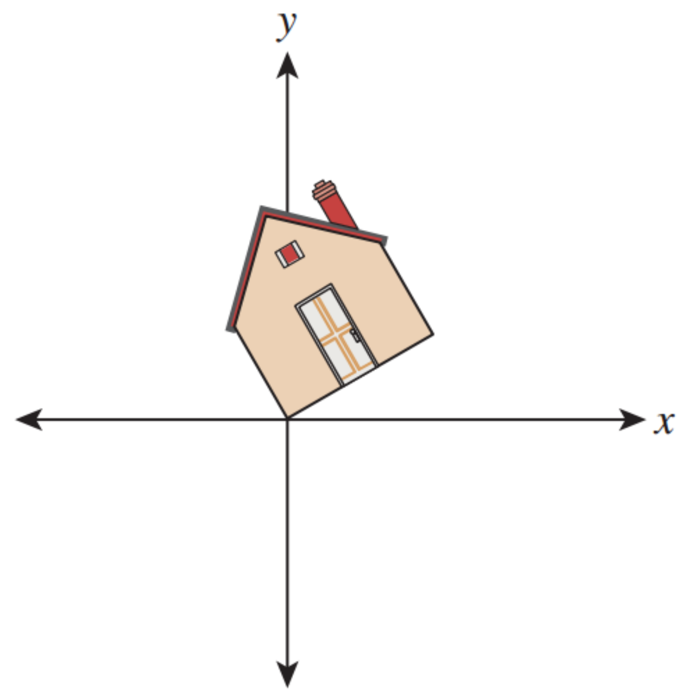
Matrix Form:

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

Rotation



Rotation by 30 degrees

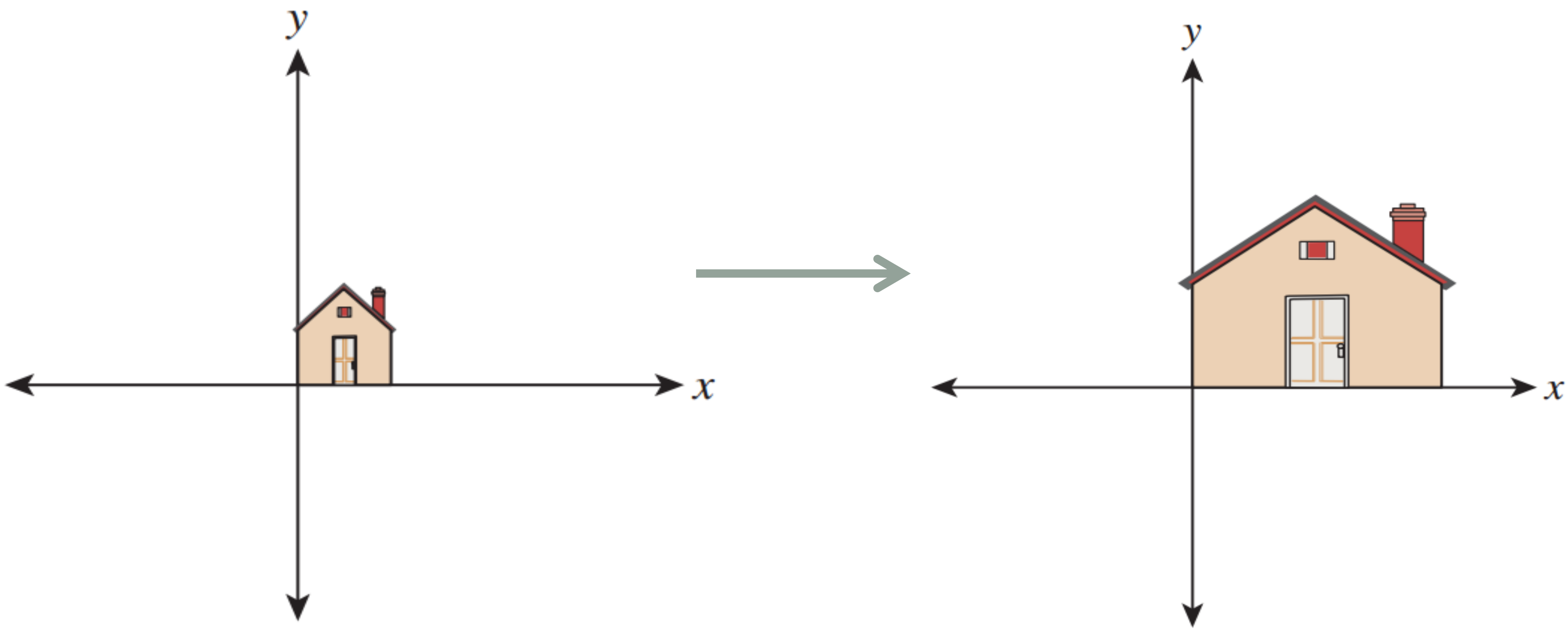


$$x' = x \cos 30^\circ - y \sin 30^\circ$$
$$y' = x \sin 30^\circ + y \cos 30^\circ$$

Matrix Form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos 30^\circ & -\sin 30^\circ \\ \sin 30^\circ & \cos 30^\circ \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Scaling



Streching by factor 3 *along* x*-axis and factor* 2 *along* y*-axis*

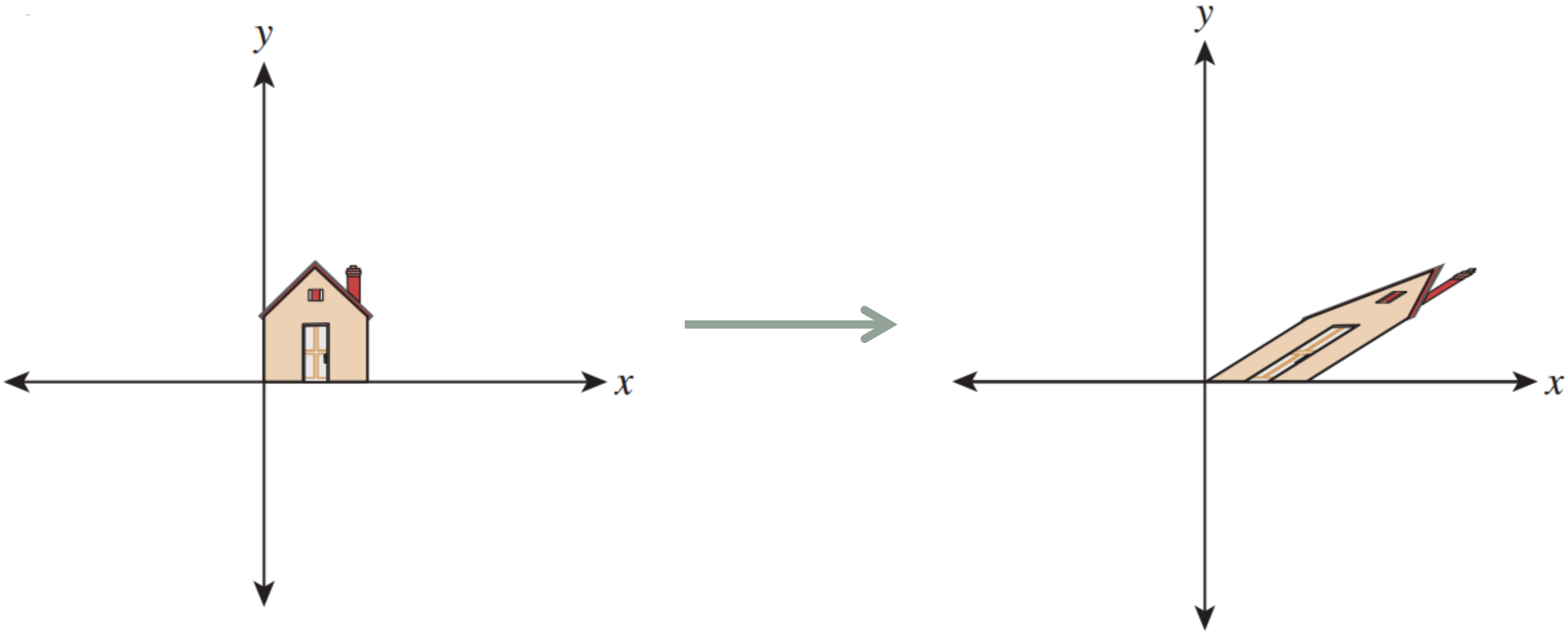
$$x' = 3x$$

$$y' = 2y$$

Matrix Form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Shearing



The y-coordinate of any point p remains the same.

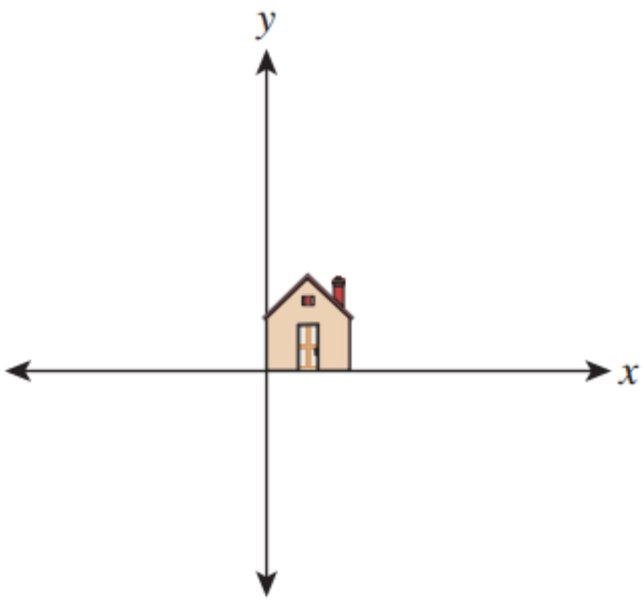
The x-coordinate increases by an amount proportional to the y-coordinate.

$$x' = x + 2y$$
$$y' = y$$

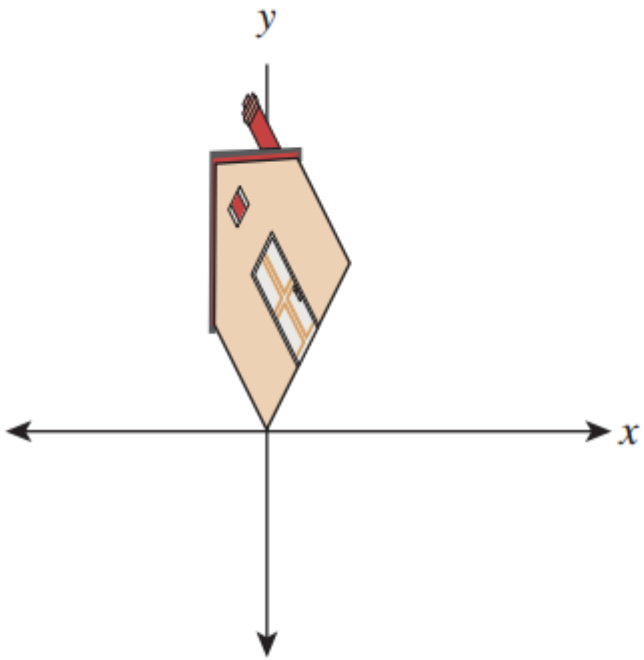
Matrix Form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Multiplication by an arbitrary matrix



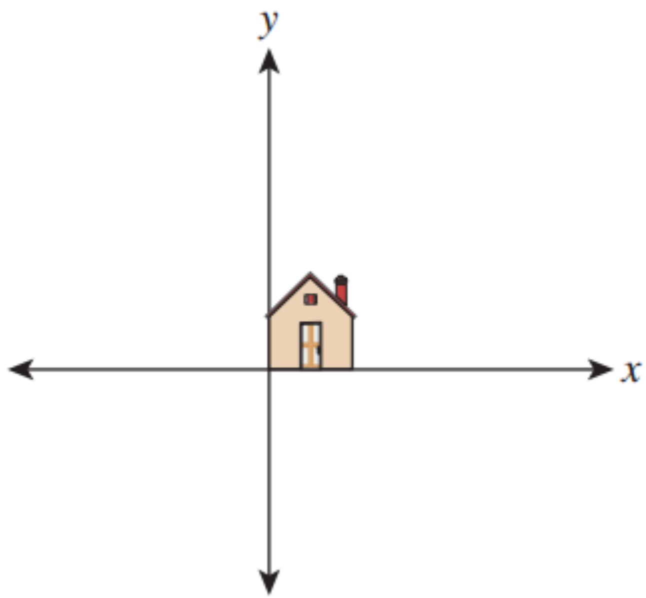
$$\begin{bmatrix} 1 & -1 \\ 2 & 2 \end{bmatrix}$$



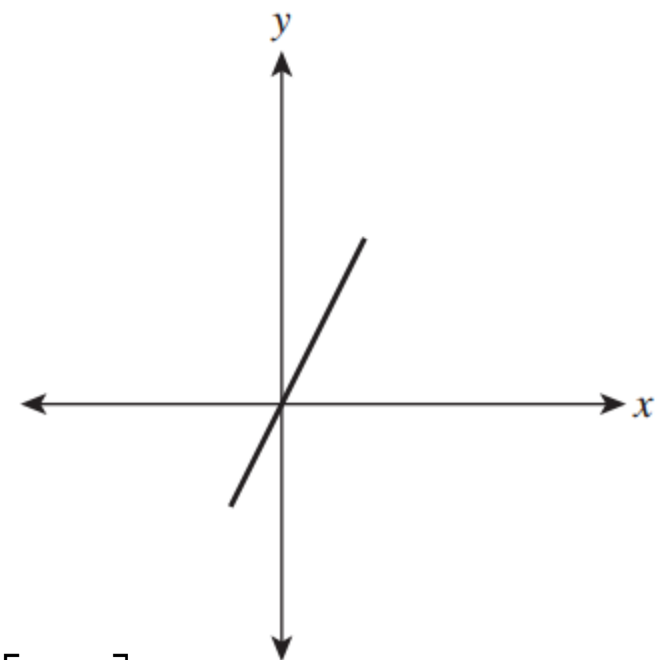
This is not rotation, scaling or shearing.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Multiplication by a singular matrix



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



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 2 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

This transformation is degenerate. No one-to-one correspondence of points.

This happened because the matrix is singular. Rows are linearly dependent.

Linear Transformations

A transformation T is called **linear** if

- $T(\alpha x) = \alpha T(x)$ for any scalar α
- $T(x + y) = T(x) + T(y)$

(Combined: $T(\alpha x + y) = \alpha T(x) + T(y)$ for any scalar α)

Multiplication by a matrix is a linear transformation.

$T_M : x \mapsto Mx$

- $M(\alpha x) = \alpha Mx$ for any scalar α
- $M(x + y) = Mx + My$

Exercise: All linear transformations are multiplication by matrix.

Function Composition and Matrix Multiplication

$T_M : x \mapsto Mx \qquad T_L : x \mapsto Lx$

Then $T_M(T_L(x)) = T_M(Lx) = MLx$ *Order matters!*

Therefore, $T_M \circ T_L = T_{ML}$ *composition of linear transformations corresponds to multiplication of the corresponding matrices*

If M is an invertible matrix and $B = M^{-1}$, i.e. $BM = MB = I$,

- $T_B(T_M(x)) = x$
- $T_M(T_B(x)) = x$

Translation

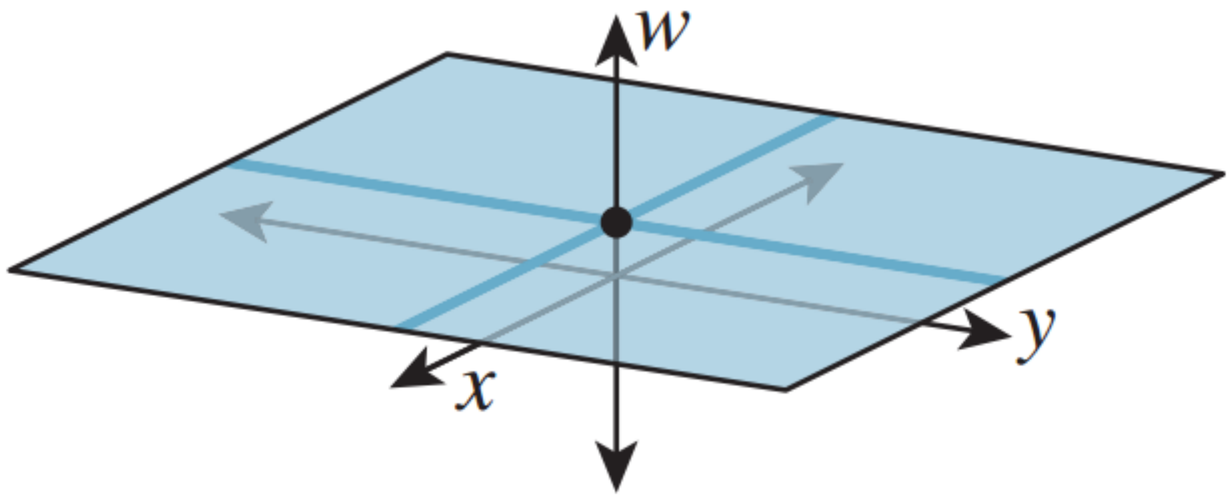
Suppose that we want to move a point (x, y) 3 units right and 2 units up.

$$x' = x + 3,\; y' = y + 2$$

Is this a linear transformation?

No! *Anything that moves the origin is not linear.*

Homogeneous Coordinates



Idea: Consider the $w = 1$ plane in 3D as our euclidean plane.

Any point (x, y) is now represented by $(x, y, 1)$.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ p & q & r \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Want this to be 1.

$px + qy + r = 1$ for any x, y

$\implies \; p, q = 0, \; r = 1$

Homogeneous Coordinates

Our transformations are of the form:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Special Case:

$$\begin{bmatrix} 1 & 0 & c \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + c \\ y + f \\ 1 \end{bmatrix}$$

Translation!

Homogeneous Coordinates

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

We can also perform the other operations we had seen earlier.

- Set $c = f = 0$. This makes the 3^{rd} coordinate irrelevant.
- Set the upper left 2×2 matrix as before

The matrix essentially implements a *linear transformation* followed by a *translation*. (Not the other way around)

Points and Vectors

Points in $\mathbb{R}^2$: elements of $\mathbb{R}^3$ with third coordinate 1

Vectors in $\mathbb{R}^2$: difference of points in $\mathbb{R}^2$

elements of $\mathbb{R}^3$ with third coordinate 0

This convention makes the distinction between points and vectors clear.

Homogeneous Coordinates in 3D

Just like in two dimensions, we add an additional coordinate.

A point with coordinates (x, y, z) is represented as $(x, y, z, 1)$.

A vector with coordinates (x, y, z) is represented as $(x, y, z, 0)$.

Transformation matrices are 4×4 matrices.

$$\begin{array}{c} \textit{Linear} \\ \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \longleftarrow \textit{Translation}$$

This matches the coordinates in a coordinate frame.

Recall: *Coordinate Frame = Coordinate System + Origin*

Given a Coordinate Frame $(\vec{v}_1, \vec{v}_2, \vec{v}_3, P_0)$:

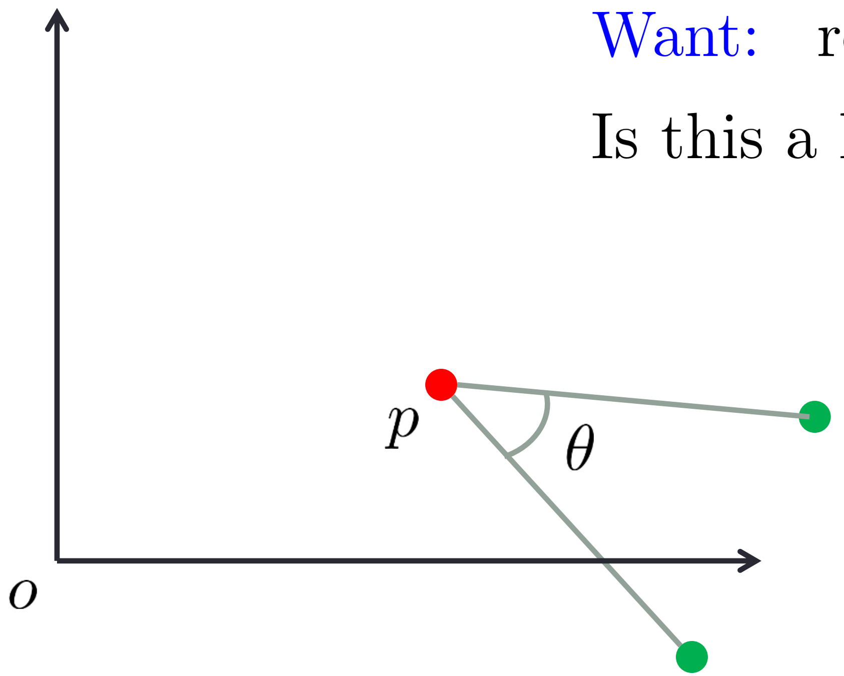
- Vectors are written as $\vec{v} = \alpha_1\vec{v}_1 + \alpha_2\vec{v}_2 + \alpha_3\vec{v}_3$

- Points are written as $\vec{v} = \beta_1\vec{v}_1 + \beta_2\vec{v}_2 + \beta_3\vec{v}_3 + P_0$

Representation of points and vectors are thus four dimensional.

The fourth coordinate is 0 for vectors and 1 for points.

Rotation about an arbitrary point



Want: rotate around p by some angle θ.

Is this a linear transformation? No!

We can still represent this as a matrix using homogeneous coordinates.

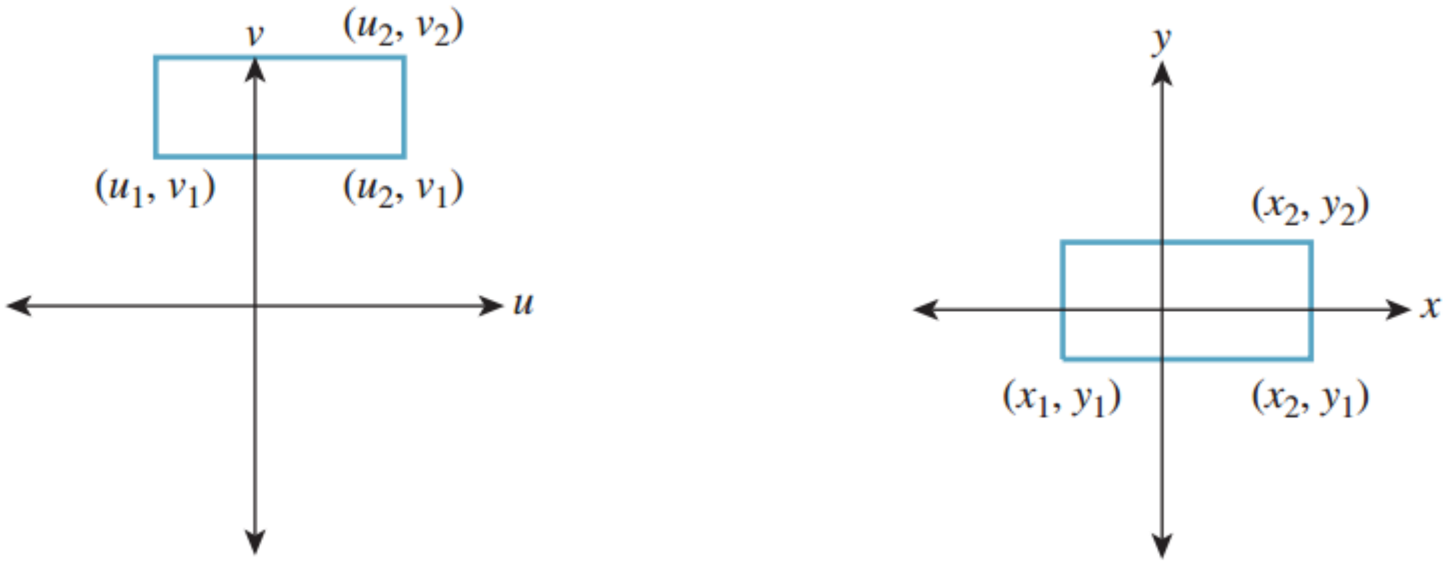
Idea: Translate the coordinate frame so that the origin moves to p

Do rotation. Then, move back the coordinate frame.

Let $\vec{p} = p - o$. 1) Translate by $-\vec{p}$, 2) $Rotate(\theta)$, 3) Translate by $\vec{p}$.

$$M = Translate(\vec{p}).Rotate(\theta).Translate(-\vec{p}).$$

Windowing Transformations



Want: to map rectangle on left to rectangle on right

Consider the first coordinate: send u_1 to x_1 and u_2 to x_2

Need to scale by $\frac{x_2-x_1}{u_2-u_1}$ and translate by some amount δ.

$$\frac{x_2-x_1}{u_2-u_1}u_1 + \delta = x_1 \implies \delta = x_1 - \frac{x_2-x_1}{u_2-u_1}u_1$$

Windowing Transformations

$$u' = \frac{x_2 - x_1}{u_2 - u_1} u + x_1 - \frac{x_2 - x_1}{u_2 - u_1} u_1$$

$$= \frac{x_2 - x_1}{u_2 - u_1} u + \frac{x_1 u_2 - x_2 u_1}{u_2 - u_1}$$

Similarly, $$v' = \frac{y_2 - y_1}{v_2 - v_1} v + \frac{y_1 v_2 - y_2 v_1}{v_2 - v_1}$$

So, we can write this transformation as $T(x) = Mx$, where

$$\mathbf{M} = \begin{bmatrix} \frac{x_2 - x_1}{u_2 - u_1} & 0 & \frac{x_1 u_2 - x_2 u_1}{u_2 - u_1} \\ 0 & \frac{y_2 - y_1}{v_2 - v_1} & \frac{y_1 v_2 - y_2 v_1}{v_2 - v_1} \\ 0 & 0 & 1 \end{bmatrix}$$

Rotations in 3D

(without homogeneous coordinates)

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

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

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

Arbitrary rotations in 3D

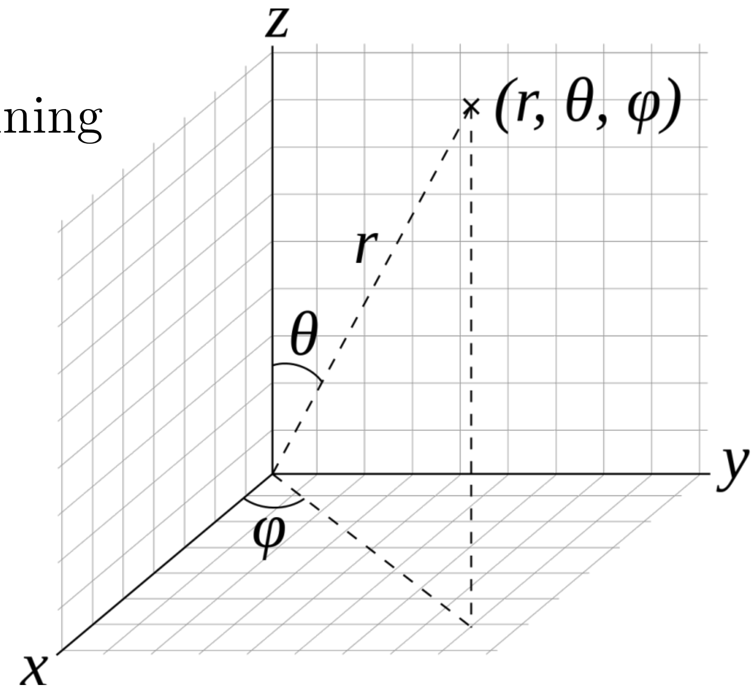
Rotation: Any *rigid transformation* that keeps the origin fixed.

Rigid transformation: any transformation that preserves distances between points.

Euler's rotation theorem: Any rotation or composition of rotations in 3D is equivalent to rotation about a line through the origin.

Rotation about an arbitrary axis in 3D

Suppose that we want rotate about an axis joining the origin o to the point $p = (r, \theta, \phi)$ (spherical coordinates).



We want counterclockwise rotation by ψ as seen from p.

Idea:

1. Do rotations so that p lies on z-axis. — $Rotate_z(-\phi)$ followed by $Rotate_y(-\theta)$

2. Do rotation about the z-axis. — $Rotate_z(\psi)$

3. Do reverse of Step 1. — $Rotate_y(\theta)$ followed by $Rotate_z(\phi)$

$$Rotate_{\vec{p}}(\psi) = Rotate_z(\phi) \cdot Rotate_y(\theta) \cdot Rotate_z(\psi) \cdot Rotate_y(-\theta) \cdot Rotate_z(-\phi)$$