Geometric Objects and Transformations

Thumrongsak Kosiyatrakul
tkosiyat@cs.pitt.edu

Geometric Objects

- Point: $P, Q, R, \ldots$
  - A location in a space
- Scalar: $\alpha, \beta, \gamma, \ldots$
  - Quantity (Distance between points)
  - Follows rules of arithmetic (addition and multiplication)
- Vector: $v, u, w, \ldots$
  - Direction and Magnitude
  - Scalar-vector multiplication

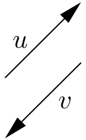
$$v = \alpha u$$

  - Vector-vector addition

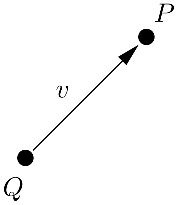
$$v = u + w$$

Geometric Objects

- Zero vector: $\mathbf{0}$
  - zero magnitude and the direction is undefined
- If $v + u = \mathbf{0}$, u is the **inverse** of v
  - $u = -v$

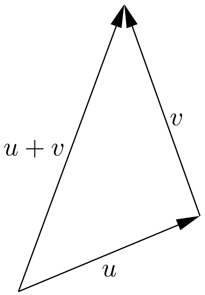


- Point-vector addition: $P = Q + v$
- Point-point subtraction $v = P - Q$

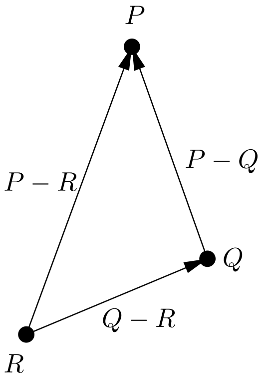


Geometric Objects

- Vector-vector addition and Point-point subtraction



Vector-vector addition



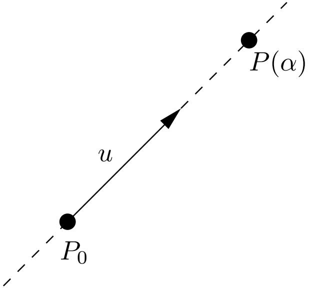
Point-point subtraction

Lines

- Consider the formula:

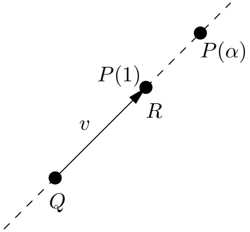
$$P(\alpha) = P_0 + \alpha u$$

  where
  - P_0 is an arbitrary point,
  - u is an arbitrary vector, and
  - α is a scalar.
- $P(\alpha)$ yields a point (point-vector addition)
- If we fix P_0 and u, and vary α, we get a line



Affine Sums

- There is no point-point addition.



- Consider a point Q, vector v, and positive scalar α

$$\begin{aligned} P &= Q + \alpha v \\ v &= R - Q \\ P &= Q + \alpha(R - Q) \\ &= Q + \alpha R - \alpha Q \\ &= \alpha R + (1 - \alpha) Q \end{aligned}$$

- In other words, $P = \alpha_1 R + \alpha_2 Q$ where $\alpha_1 + \alpha_2 = 1$ (point-point addition)

Vector

- A vector u can be represented by a sequence of real (or complex) numbers:

$$u = (x_1, x_2, \ldots, x_n)$$

or in a column matrix form:

$$u = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

but sometimes we use the transpose of a row matrix for simplicity:

$$u = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix}^T$$

Vector Norm

- The norm (length, size, magnitude, etc) of a vector u is given by
$$|u| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$
- Properties:
  - $|u| > 0$ when $u \neq \mathbf{0}$ and $|u| = 0$ iff $u = \mathbf{0}$
  - $|\alpha u| = |\alpha||u|$
  - $|u + v| \leq |u| + |v|$

Vector Inner Product (Dot Product)

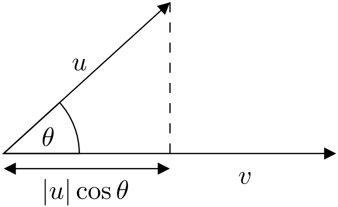
- The dot product of two vectors u and v is defined by

$$u \cdot v = |u||v| \cos \theta$$

  where θ is the angle between u and v when two vectors are placed so that their tails coincide
- If u and v are orthogonal $(\theta = \pi/2)$, $u \cdot v = 0$
- If u and v are codirectional $(\theta = 0)$, $u \cdot v = |u||v|$
- Thus $u \cdot u = |u||u| = |u|^2$
- In other words, $|u| = \sqrt{u \cdot u}$.

Vector Inner Product (Dot Product)

- Consider the following vectors u and v

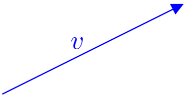


- $|u|\cos\theta$ is the length of the orthogonal projection of u onto v

$$|u|\cos\theta = \frac{u \cdot v}{|v|}$$

- Geometrically, the dot product expresses the length of the projection of u onto the unit vector v when their tails coincide.

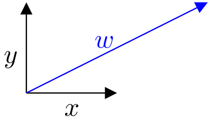
- Consider the following vector v:



- What is the magnitude of v?
    - 23 inches
    - 58 centimeters
- What is the direction of v?
- To be able to answer these questions, we need be able to represent the above vector based on some references

Coordinate Systems and Frames

- In a two-dimensional space, vectors v_1 and v_2 are considered basis vectors if for any vector in this two-dimensional space, it can be represented by linear combination of v_1 and v_2.
- x-axis and y-axis (unit length) are a familiar basis vectors



$$w = 2x + y$$

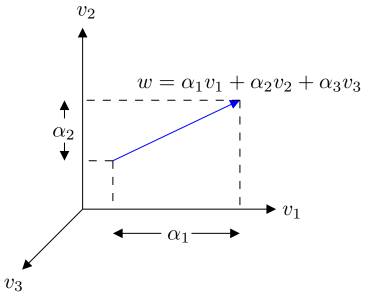
- Note that
  - two basis vectors do not have to be orthogonal to each other, and
  - they do not have to have the same magnitude

Coordinate Systems and Frames

- In three-dimensional vector space, given three basis vectors v_1, v_2, and v_3 a vector w can be represented as:

$$w = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3$$

where α_i are scalar components of w with respect to basis v_1, v_2, and v_3.



- For simplicity, imagine that v_1, v_2, and v_3 are x, y, and z (unit vectors) axes.

- The vector w can be represented as a column matrix as:

$$\mathbf{a} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

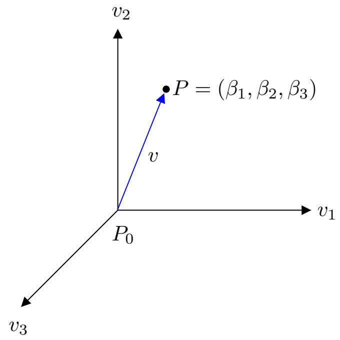
- Note that the textbook use the boldface to denote a representation in a specific basis.
- Note

$$\begin{aligned} w &= \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 \\ &= \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \\ &= \mathbf{a}^T \mathbf{v} \end{aligned}$$

where $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$.

Representing a Point

- Location of a point needs a reference



- Give basis v_1, v_2, v_3, and an origin P_0:

$$P = P_0 + \beta_1 v_1 + \beta_2 v_2 + \beta_3 v_3 = P_0 + \mathbf{b}^T \mathbf{v}$$

Representations and N-Tuples

- As discussed earlier, a vector v can be represented as

$$v = \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3$$

  where e_1, e_2, and e_3 form a basis.

- Since e_i themselves are vectors. They have their own representations:

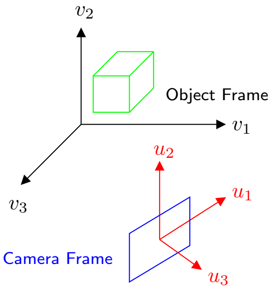
$$\mathbf{e}_1 = \begin{bmatrix} e_{1_1} \\ e_{1_2} \\ e_{1_3} \end{bmatrix} \qquad \mathbf{e}_2 = \begin{bmatrix} e_{2_1} \\ e_{2_2} \\ e_{2_3} \end{bmatrix} \qquad \mathbf{e}_3 = \begin{bmatrix} e_{3_1} \\ e_{3_2} \\ e_{3_3} \end{bmatrix}$$

- Recall that the representation of v can be

$$\mathbf{a} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} \qquad \text{or} \qquad \alpha_1 \mathbf{e}_1 + \alpha_2 \mathbf{e}_2 + \alpha_3 \mathbf{e}_3$$

Change of Coordinate System

- Generally we draw objects in a frame called object frame
  - Object frame is a frame defined by basis vectors v_1, v_2, and v_3
- Imagine that a camera is put in the object frame and point to a direction
- What camera sees generally called a camera frame
  - Camera frame is a frame defined by basis vectors u_1, u_2, and u_3



- To generate an image on the camera frame, we need to know location of vertices of objects in camera frame

Change of Coordinate System

- Consider two sets of basis vectors $\{v_1, v_2, v_3\}$ and $\{u_1, u_2, u_3\}$
- Our goal is to find a representation in the basis $\{u_1, u_2, u_3\}$ of a vector or a point defined in the basis $\{v_1, v_2, v_3\}$
- Recall that any vector w can be represented in terms of basis:

$$w = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3$$

- Since u_1, u_2, and u_3, are vectors, we can represent them as:

$$\begin{aligned} u_1 &= \gamma_{11} v_1 + \gamma_{12} v_2 + \gamma_{13} v_3 \\ u_2 &= \gamma_{21} v_1 + \gamma_{22} v_2 + \gamma_{23} v_3 \\ u_3 &= \gamma_{31} v_1 + \gamma_{32} v_2 + \gamma_{33} v_3 \end{aligned}$$

- Using matrix multiplication, the above equations are the same as

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Change of Coordinate System

- Let $\mathbf{M} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix}$, $\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$, and $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$, we have

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \mathbf{M} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad \text{or } \mathbf{u} = \mathbf{M}\mathbf{v}$$

- $\mathbf{M}$ contains information about how to represent a vector in one basis in another basis.

Change of Coordinate System

- Consider a vector w with respect to $\mathbf{v}$ (an object frame):

$$w = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \mathbf{a}^T \mathbf{v}$$

where $\mathbf{a} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$ a representation of w with respect to $\mathbf{v}$

- Consider the same vector w with respect to $\mathbf{u}$ (a camera frame):

$$w = \beta_1 u_1 + \beta_2 u_2 + \beta_3 u_3 = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \mathbf{b}^T \mathbf{u}$$

where $\mathbf{b} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}$ a representation of w with respect to $\mathbf{u}$

Change of Coordinate System

- Since they are the same vector and $\mathbf{u} = \mathbf{M}\mathbf{v}$, we have:

$$
\begin{aligned}
w = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 &= \beta_1 u_1 + \beta_2 u_2 + \beta_3 u_3 \\
\mathbf{a}^T \mathbf{v} &= \mathbf{b}^T \mathbf{u} \\
&= \mathbf{b}^T \mathbf{M}\mathbf{v}
\end{aligned}
$$

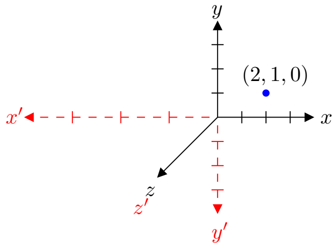
- Thus $\mathbf{a}^T = \mathbf{b}^T \mathbf{M}$ or $\mathbf{a} = \mathbf{M}^T \mathbf{b}$
- Note that

$$
\begin{aligned}
\mathbf{a} &= \mathbf{M}^T \mathbf{b} \\
(\mathbf{M}^T)^{-1}\mathbf{a} &= (\mathbf{M}^T)^{-1}\mathbf{M}^T \mathbf{b} \\
&= \mathbf{I}\mathbf{b} \\
&= \mathbf{b}
\end{aligned}
$$

The matrix $(\mathbf{M}^T)^{-1}$ takes a representation **a**, with respect to **v** (an object frame), to another representation **b** with respect to **u** (a camera frame).

Change of Coordinate System (Example)

- Consider the following three-dimensional space:



There is a point $(2, 1, 0)$ in a familiar three-dimensional space defined by basis x, y, and z.

- What is the coordinate of the same point in a new basis x', y', and z' where

$$x' = (-2)x + (0)y + (0)z$$
$$y' = (0)x + (-1)y + (0)z$$
$$z' = (0)x + (0)y + (1)z$$

Change of Coordinate System (Example)

- Our matrix $\mathbf{M}$ is as follows:

$$\mathbf{M} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Note that $\mathbf{M}$ converts a representation in x, y, z to x', y', and z'.
- $\mathbf{M}^T$ is identical to $\mathbf{M}$ and

$$(\mathbf{M}^T)^{-1} = \begin{bmatrix} -1/2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{T}$$

- Thus, in x', y', and z' space, the point is

$$\begin{bmatrix} -1/2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix} \text{ or } (-1, -1, 0)$$

Homogeneous Coordinates

- Given a column matrix $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$, we have no idea whether it represents a point

$$P = P_0 + xv_1 + yv_2 + zv_3$$

  or a vector

$$w = xv_1 + yv_2 + zv_3$$

- Homogeneous Coordinate use four-dimensional representation for both points and vectors in three-dimensional space
- A frame is defined by four elements (v_1, v_2, v_3, P_0) where P_0 is the origin of the frame and multiplication is defined as follows:
  - $0 \times P = \mathbf{0}$
  - $1 \times P = P$

Homogeneous Coordinates

- A point P is represented by $\mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

$$P = xv_1 + yv_2 + zv_3 + P_0 = \mathbf{p}^T \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix}$$

- A vector w is represented by $\mathbf{w} = \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$

$$w = xv_1 + yv_2 + zv_3 = \mathbf{w}^T \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix} = \begin{bmatrix} x & y & z & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix}$$

- Now, consider two frames (v_1, v_2, v_3, P_0) and (u_1, u_2, u_3, Q_0), vectors u_1, u_2, and u_3 and the point Q_0 can be represented by the first frame as

$$\begin{aligned}
u_1 &= \gamma_{11} v_1 + \gamma_{12} v_2 + \gamma_{13} v_3 \\
u_2 &= \gamma_{21} v_1 + \gamma_{22} v_2 + \gamma_{23} v_3 \\
u_3 &= \gamma_{31} v_1 + \gamma_{32} v_2 + \gamma_{33} v_3 \\
Q_0 &= \gamma_{41} v_1 + \gamma_{42} v_2 + \gamma_{43} v_3 + P_0
\end{aligned}$$

- Let $\mathbf{M}$ be the matrix

$$\begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} & 0 \\ \gamma_{21} & \gamma_{22} & \gamma_{23} & 0 \\ \gamma_{31} & \gamma_{32} & \gamma_{33} & 0 \\ \gamma_{41} & \gamma_{42} & \gamma_{43} & 1 \end{bmatrix}$$

we have

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ Q_0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix}$$

Homogeneous Coordinates

- Let $\mathbf{a}$ and $\mathbf{b}$ represent either same point or same vector in frames (v_1, v_2, v_3, P_0) and (u_1, u_2, u_3, Q_0), respectively.
- We have

$$\mathbf{b}^T \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ Q_0 \end{bmatrix} = \mathbf{b}^T \mathbf{M} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix} = \mathbf{a}^T \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix}$$

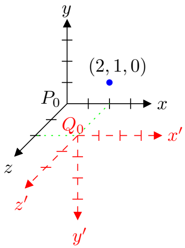
- Thus $\mathbf{a} = \mathbf{M}^T \mathbf{b}$ where

$$\mathbf{M}^T = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- As usual, $(\mathbf{M}^T)^{-1}\mathbf{a} = (\mathbf{M}^T)^{-1}\mathbf{M}^T\mathbf{b} = \mathbf{I}\mathbf{b} = \mathbf{b}$. In other words, $(\mathbf{M}^T)^{-1}$ takes us from $\mathbf{a}$ to $\mathbf{b}$.

Change of Coordinate System (Example)

- Consider the following three-dimensional space:



  There is a point $(2, 1, 0)$ in a familiar three-dimensional space defined by basis x, y, and z.
- What is the coordinate of the same point in a new basis x', y', and z' where

$$
\begin{aligned}
x' &= (1)x + (0)y + (0)z \\
y' &= (0)x + (-1)y + (0)z \\
z' &= (0)x + (0)y + (1)z \\
Q_0 &= (2)x + (0)y + (2)z + P_0
\end{aligned}
$$

Change of Coordinate System (Example)

- From previous slide, we have

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & 0 & 2 & 1 \end{bmatrix} \text{ and } \mathbf{M}^T = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Thus

$$(\mathbf{M}^T)^{-1} = \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$(\mathbf{M}^T)^{-1}\mathbf{a} = \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ -2 \\ 1 \end{bmatrix}$$

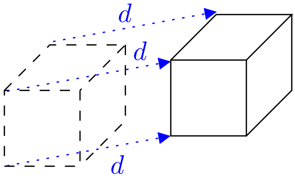
Translation, Rotation, and Scaling

- An object can be drawn directly at its desired size, orientation, and location
  - This requires a lot of calculation and work with unfamiliar imagination
  - Thinking about drawing a cube center at (1.4, 3.5, -2.1), each side is 1.5 by 1.5, and tilt at 32.9 degree
- Generally we draw an object at a familiar location and size
  - A cube center a the origin, each side is 1 by 1, and parallel to an axis, and
  - resize it,
  - rotate it, and
  - move the object to its desired location

Translating

- A **translation** moves points to a new location by a fix distance and direction
- Recall **point-vector** addition:
  - A point P can be moved to a new point P' in a direction of a vector d with the distance $|d|$

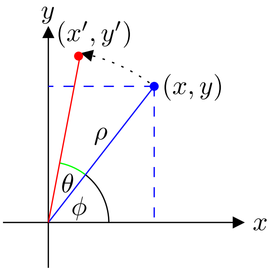
$$P' = P + d$$



- Note that we do not need to know where is the origin. Thus, no reference to a frame.

Rotation

- To rotate a point, we need a reference point
- The following show a rotation of point (x, y) about the origin by θ degree



- Suppose the distance from the origin to the point (x, y) is ρ
  - $x = \rho \cos \phi$
  - $y = \rho \sin \phi$
  - $x' = \rho \cos(\theta + \phi)$
  - $y' = \rho \sin(\theta + \phi)$

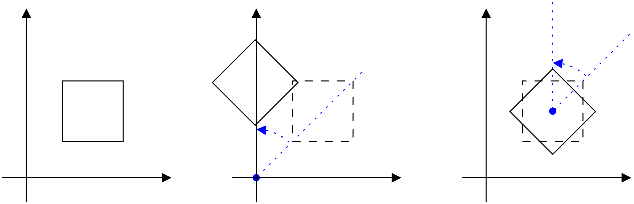
- Recall that
  - $\cos(\theta + \phi) = \cos\theta\cos\phi - \sin\theta\sin\phi$
  - $\sin(\theta + \phi) = \sin\theta\cos\phi + \cos\theta\sin\phi$
- Thus

$$\begin{aligned} x' &= \rho\cos(\theta + \phi) \\ &= \rho(\cos\theta\cos\phi - \sin\theta\sin\phi) \\ &= \rho\cos\theta\cos\phi - \rho\sin\theta\sin\phi \\ &= x\cos\theta - y\sin\theta \\ y' &= \rho\sin(\theta + \phi) \\ &= \rho(\sin\theta\cos\phi + \cos\theta\sin\phi) \\ &= \rho\sin\theta\cos\phi + \rho\cos\theta\sin\phi \\ &= x\sin\theta + y\cos\theta \end{aligned}$$

which can be viewed as

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

- From previous example, the origin is unchanged. This is called the **fixed point**



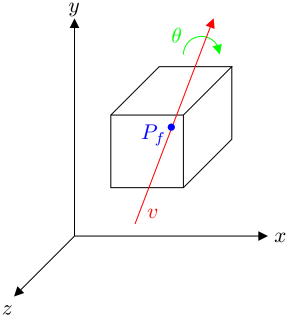
About Origin vs About Center of Mass

- Rotation about the origin in 2D is the same as rotation about the z-axis in 3D.
- We use right-handed system:
  - index finger for x-axis, middle finger for y-axis, and thumb for z-axis
  - Positive is counter clockwise

Rotation

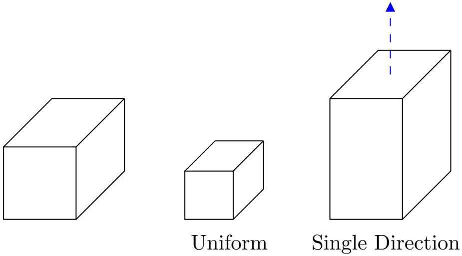
- To rotate an object in three-dimensional frame we need
  - a fixed point (P_f),
  - a rotation angle (θ), and
  - a vector (v) about which to rotate

  as shown below:



- Rotation does not change the shape or volume of an object (**rigid-body transformation**)

- Scaling does change the shape and volume of an object (**non-rigid-body transformation**)



- Can be uniform in all direction or in a specific direction
- To scale an object, we need the following:
  - a fixed-point (P_f),
  - a direction (v) to scale, and
  - a scale factor (α)
    - $\alpha > 1$, longer in a specific direction,
    - $0 < \alpha < 1$, smaller in a specific direction, and
    - $\alpha < 0$, **reflection**.

Transformations in Homogeneous Coordinate

- Recall that a point $P = (x, y, z)$ in the basis v_1, v_2, and v_3 is

$$P = xv_1 + yv_2 + zv_3$$

and its representation in homogeneous coordinate is a column vector

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Any transformation in homogeneous coordinate of the point P should result in a column matrix

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

- An affine transformation is represented by a 4×4 matrix of the form

$$\mathbf{A} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Note that $\mathbf{AP}$ results in a column matrix that we want

$$\mathbf{AP} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

where

$$\begin{aligned} x' &= \alpha_{11}x + \alpha_{12}y + \alpha_{13}z + \alpha_{14}(1) \\ y' &= \alpha_{21}x + \alpha_{22}y + \alpha_{23}z + \alpha_{24}(1) \\ z' &= \alpha_{31}x + \alpha_{32}y + \alpha_{33}z + \alpha_{34}(1) \end{aligned}$$

Translation in Homogeneous Coordinate

- Recall that moving a point P by a distance d to a point P' is a point-vector addition $P = P + d$ or

$$\mathbf{P}' = \mathbf{P} + \mathbf{d}$$

in homogeneous coordinate where

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}, \text{ and } \mathbf{d} = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \\ 0 \end{bmatrix}$$

- Note that

$$\mathbf{P}' = \mathbf{P} + \mathbf{d} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} + \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \\ 0 \end{bmatrix} = \begin{bmatrix} x + \alpha_x \\ y + \alpha_y \\ z + \alpha_z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Translation in Homogeneous Coordinate

- In other words,

$$
\begin{aligned}
x' &= x + \alpha_x \\
y' &= y + \alpha_y \\
z' &= z + \alpha_z
\end{aligned}
$$

- Note that we can obtain the same point $\mathbf{P}'$ using matrix multiplication as follows:

$$
\begin{bmatrix} 1 & 0 & 0 & \alpha_x \\ 0 & 1 & 0 & \alpha_y \\ 0 & 0 & 1 & \alpha_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + \alpha_x \\ y + \alpha_y \\ z + \alpha_z \\ 1 \end{bmatrix}
$$

Translation in Homogeneous Coordinate

- We can achieve the same result using the matrix

$$\mathbf{T}(\alpha_x, \alpha_y, \alpha_z) = \begin{bmatrix} 1 & 0 & 0 & \alpha_x \\ 0 & 1 & 0 & \alpha_y \\ 0 & 0 & 1 & \alpha_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Note that $\mathbf{P}' = \mathbf{T}(\alpha_x, \alpha_y, \alpha_z)\mathbf{P}$.
- The matrix $\mathbf{T}(\alpha_x, \alpha_y, \alpha_z)$ is called the **translation matrix**.
- To move $\mathbf{P}'$ back to $\mathbf{P}$ we need $\mathbf{T}^{-1}$ since

$$\mathbf{T}^{-1}\mathbf{P}' = \mathbf{T}^{-1}\mathbf{T}\mathbf{P} = \mathbf{I}\mathbf{P} = \mathbf{P}$$

- But we can think of this as moving the point $\mathbf{P}'$ in the opposite direction. Thus

$$\mathbf{T}^{-1}(\alpha_x, \alpha_y, \alpha_z) = \mathbf{T}(-\alpha_x, -\alpha_y, -\alpha_z) = \begin{bmatrix} 1 & 0 & 0 & -\alpha_x \\ 0 & 1 & 0 & -\alpha_y \\ 0 & 0 & 1 & -\alpha_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translation Example

- Suppose we have a line from $(-1, 2, 1)$ to $(1, 5, 0)$ and we want to translate this line in the same direction and magnitude of a vector represented by (1,2,-3)
- The translation matrix $\mathbf{T}(1, 2, -3)$ is as follows:

$$\mathbf{T}(1,2,3) = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- The point $(-1, 2, 1)$ is translated to the point $(0, 4, -2)$:

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} -1 \\ 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \\ -2 \\ 1 \end{bmatrix}$$

- The point $(1, 5, 0)$ is translated to the point $(2, 7, -3)$:

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 5 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 7 \\ -3 \\ 1 \end{bmatrix}$$

Scaling in Homogeneous Coordinate

- Consider the fixed point of a scaling is at the origin
- The value of x, y, and z of a point is simply changed by its factor:

$$\begin{aligned} x' &= \beta_x x, \\ y' &= \beta_y y, \text{ and} \\ z' &= \beta_z z. \end{aligned}$$

- This is in the form of

$$\mathbf{p}' = \mathbf{S}\mathbf{p}$$

where

$$\mathbf{S} = \mathbf{S}(\beta_x, \beta_y, \beta_z) = \begin{bmatrix} \beta_x & 0 & 0 & 0 \\ 0 & \beta_y & 0 & 0 \\ 0 & 0 & \beta_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- The matrix $\mathbf{S}(\beta_1, \beta_2, \beta_3)$ is called the **scaling matrix**.

Scaling in Homogeneous Coordinate

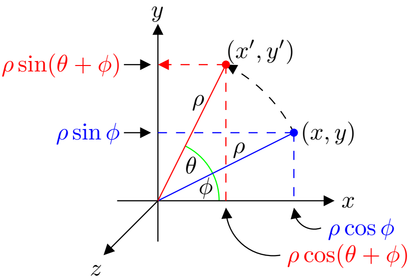
- As usual, $\mathbf{S}^{-1}(\beta_x, \beta_y, \beta_z)$ can be obtained by calculating the inverse of $\mathbf{S}(\beta_x, \beta_y, \beta_z)$.
- Note that

$$\begin{aligned}
\frac{1}{\beta_x}x' &= \frac{1}{\beta_x}\beta_x x = x \\
\frac{1}{\beta_y}y' &= \frac{1}{\beta_y}\beta_y y = y \\
\frac{1}{\beta_z}z' &= \frac{1}{\beta_z}\beta_z z = z
\end{aligned}$$

- In other words, simply scale back with 1 over the factor.

$$\mathbf{S}^{-1}(\beta_x, \beta_y, \beta_z) = \mathbf{S}(\frac{1}{\beta_x}, \frac{1}{\beta_y}, \frac{1}{\beta_z}) = \begin{bmatrix} \frac{1}{\beta_x} & 0 & 0 & 0 \\ 0 & \frac{1}{\beta_y} & 0 & 0 \\ 0 & 0 & \frac{1}{\beta_z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Recall rotating a point about the origin in 2D is the same as rotating a point is 3D about z-axis



$$
\begin{aligned}
x &= \rho\cos\phi \\
y &= \rho\sin\phi \\
x' &= \rho\cos(\theta+\phi) \\
&= \rho(\cos\theta\cos\phi - \sin\theta\sin\phi) \\
&= \rho\cos\theta\cos\phi - \rho\sin\theta\sin\phi \\
&= x\cos\theta - y\sin\theta \\
y' &= \rho\sin(\theta+\phi) \\
&= \rho(\sin\theta\cos\phi + \cos\theta\sin\phi) \\
&= \rho\sin\theta\cos\phi + \rho\cos\theta\sin\phi \\
&= x\sin\theta + y\cos\theta \\
z' &= z
\end{aligned}
$$

Rotation in Homogeneous Coordinate (About z-axis)

- In homogeneous coordinate, it is equivalent to

$$\mathbf{p}' = \mathbf{R}_z\mathbf{p}$$

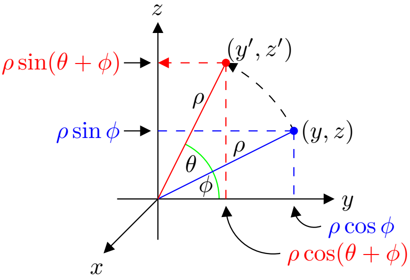
where

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

- Let check with a point $P = (x, y, z)$

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

- Recall rotating a point about the origin in 2D is the same as rotating a point is 3D about x-axis



$$
\begin{aligned}
y &= \rho \cos \phi \\
z &= \rho \sin \phi \\
x' &= x \\
y' &= \rho \cos(\theta + \phi) \\
&= \rho(\cos\theta \cos\phi - \sin\theta \sin\phi) \\
&= \rho \cos\theta \cos\phi - \rho \sin\theta \sin\phi \\
&= y \cos\theta - z \sin\theta \\
z' &= \rho \sin(\theta + \phi) \\
&= \rho(\sin\theta \cos\phi + \cos\theta \sin\phi) \\
&= \rho \sin\theta \cos\phi + \rho \cos\theta \sin\phi \\
&= y \sin\theta + z \cos\theta
\end{aligned}
$$

Rotation in Homogeneous Coordinate (About x-axis)

- In homogeneous coordinate, it is equivalent to

$$\mathbf{p}' = \mathbf{R}_x \mathbf{p}$$

where

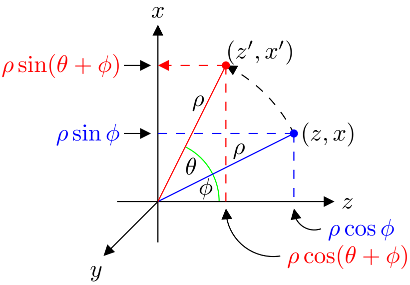
$$\mathbf{R}_x = \mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Let check with a point $P = (x, y, z)$

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

Rotation in Homogeneous Coordinate (About y-axis)

- Recall rotating a point about the origin in 2D is the same as rotating a point is 3D about y-axis



$$
\begin{aligned}
z &= \rho\cos\phi \\
x &= \rho\sin\phi \\
z' &= \rho\cos(\theta+\phi) \\
&= \rho(\cos\theta\cos\phi - \sin\theta\sin\phi) \\
&= \rho\cos\theta\cos\phi - \rho\sin\theta\sin\phi \\
&= z\cos\theta - x\sin\theta \\
y' &= y \\
x' &= \rho\sin(\theta+\phi) \\
&= \rho(\sin\theta\cos\phi + \cos\theta\sin\phi) \\
&= \rho\sin\theta\cos\phi + \rho\cos\theta\sin\phi \\
&= z\sin\theta + x\cos\theta
\end{aligned}
$$

Rotation in Homogeneous Coordinate (About y-axis)

- In homogeneous coordinate, it is equivalent to

$$\mathbf{p}' = \mathbf{R}_y \mathbf{p}$$

where

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

- Let check with a point $P = (x, y, z)$

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

Rotation in Homogeneous Coordinate

- Let $\mathbf{R}$ be any of the three rotation matrices ($\mathbf{R}_x$, $\mathbf{R}_y$, or $\mathbf{R}_z$)
- As usual, we can rotate a point back by simply rotate it by $-\theta$. Thus

$$\mathbf{R}^{-1}(\theta) = \mathbf{R}(-\theta)$$

- Since $\cos(-\theta) = \cos\theta$ and $\sin(-\theta) = -\sin\theta$, we have

$$\mathbf{R}^{-1}(\theta) = \mathbf{R}^T(\theta)$$

- Note that a rotation does not have to be about an axis but it can be achieve by rotating one axis at a time

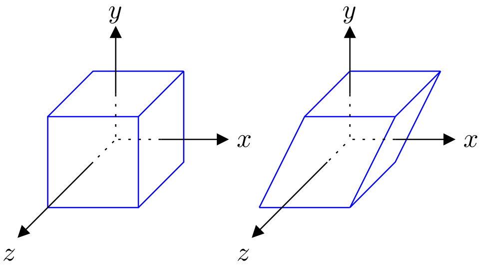
$$\mathbf{R} = \mathbf{R}_z\mathbf{R}_y\mathbf{R}_x$$

and

$$\mathbf{R}^{-1} = \mathbf{R}^T$$

Shear

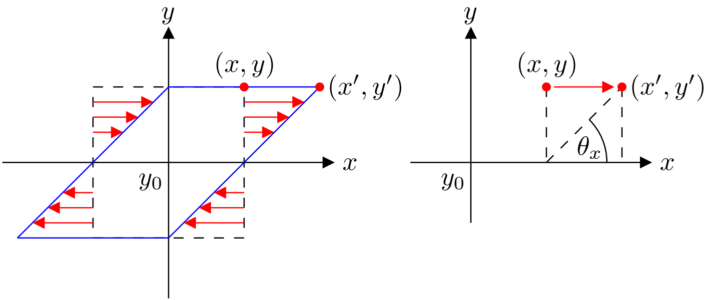
- An example of a shear where the top get pulled to the right and the bottom get pulled to the left is shown below:



- The cube is sheared in the x-direction (y and z are unchanged)
  - If $y > 0$, the larger the y, the more x is pulled to the right.
  - If $y < 0$, the smaller the y, the more x is pulled to the left

Shear

- Let's focus on a cross section where the object is sheared about the y-axis and the center of shearing is at y_0 (origin in this case):



- A point (x, y) is moved to a new point (x', y') where $y' = y$
- From the above picture on the right, we have

$$\tan\theta_x = \frac{y - y_0}{x' - x}$$

Shear

- From previous slide, we have

$$\begin{aligned} \tan\theta_x &= \frac{y-y_0}{x'-x} \\ x'-x &= \frac{y-y_0}{\tan\theta_x} \\ x' &= x+(y-y_0)\cot\theta_x \\ &= x+y\cot\theta_x \quad \text{when } y_0=0 \end{aligned}$$

- If we also shear z by the angle ϕ_z, then we have

$$\begin{aligned} z' &= z+(y-y_0)\cot\phi_z \\ &= z+y\cot\phi_z \quad \text{when } y_0=0 \end{aligned}$$

- Shearing about y-axis where the center of shearing is at the origin $(y_0 = 0)$ can be viewed as a **shearing matrix**

$$\mathbf{H}_y(\theta_x,\phi_z) = \begin{bmatrix} 1 & \cot\theta_x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \cot\phi_z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shear

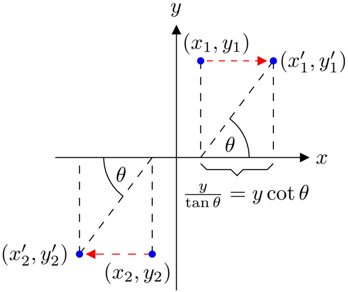
- Note that if θ_x is 90 degree, no shearing about x-axis
- Similarly, if ϕ_z is 90 degree, no shearing about z-axis
- But if θ_x is a multiple of 180 degree, $\tan\theta_x = 0$.
  - $\cot\theta_x$ is undefined
  - To prevent divided by 0, you should check first whether a degree of shearing is 90°.
- To shear an object back to its original, simply use $\mathbf{H}_y(-\theta_x, -\phi_z)$ since $\cot(-\theta) = -\cot\theta$.
- Let's check $(y_0 = 0)$:

$$\mathbf{H}_y(-\theta_x,-\phi_z)\mathbf{H}_y(\theta_x,\phi_z) = \begin{bmatrix} 1 & \cot(-\theta_x) & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \cot(-\phi_z) & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \cot\theta_x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \cot\phi_z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \cot\theta_x + \cot(-\theta_x) & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \cot(-\phi_z) + \cot\phi_z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shear

- If a shearing degree is unknown but a target is known



- From the above image, $y_0 = 0$, we have

$$\tan\theta = \frac{y_1 - 0}{x'_1 - x_1} = \frac{y_1}{x'_1 - x_1} \rightsquigarrow \cot\theta = \frac{x'_1 - x_1}{y_1} \rightsquigarrow \theta = \cot^{-1}(\frac{x'_1 - x_1}{y_1})$$

$$\tan\theta = \frac{y_2 - 0}{x'_2 - x_2} = \frac{y_2}{x'_2 - x_2} \rightsquigarrow \cot\theta = \frac{x'_2 - x_2}{y_2} \rightsquigarrow \theta = \cot^{-1}(\frac{x'_2 - x_2}{y_2})$$

Shear

- A shearing matrix where an object is sheared about z-axis and the center of shearing is at the origin:

$$\mathbf{H}_z(\theta_x, \phi_y) = \begin{bmatrix} 1 & 0 & \cot\theta_x & 0 \\ 0 & 1 & \cot\phi_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- A shearing matrix where an object is sheared about x-axis and the center of shearing is at the origin:

$$\mathbf{H}_x(\theta_y, \phi_z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \cot\theta_y & 1 & 0 & 0 \\ \cot\phi_z & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Concatenation of Transformations

- We can apply a series of transformation to a point $\mathbf{p}$ to get a point $\mathbf{q}$

$$\mathbf{p} \longrightarrow \boxed{\mathbf{A}} \longrightarrow \boxed{\mathbf{B}} \longrightarrow \boxed{\mathbf{C}} \longrightarrow \mathbf{q}$$

- In homogeneous coordinate, this process can be viewed as

$$\mathbf{q} = \mathbf{C}(\mathbf{B}(\mathbf{A}\mathbf{p}))$$

- Let $\mathbf{M} = \mathbf{CBA}$, we have

$$\mathbf{q} = \mathbf{M}\mathbf{p}$$

- $\mathbf{M}$ can be computed only once and load it into the graphic pipeline which will be applied to multiple vertices.

Uniform Variables

- **Uniform** variables can be used to send a transformation matrix between an application and the graphic pipeline
- Example in a vertex shader:

```
#version 130

in vec4 vPosition;
in vec4 vColor;
out vec4 color;

uniform mat4 ctm;

void main()
{
    color = vColor;
    gl_Position = ctm * vPosition;
}
```

- The `transformation_matrix` will be applied to all vertices.
- The `mat4` (4×4 matrix) and matrix multiplication are part of `GLSL`

Uniform Variables

- To communicate with a uniform variable in a vertex shader, we need to create two global variables:

```
GLuint ctm_location;
mat4 tr_matrix;
```

- These two variables may required by multiple functions
- In the `init()` function:

```
ctm_location = glGetUniformLocation(program, "ctm");
tr_matrix = m4_identity();
```

- The name `transformation_matrix` must match with a uniform variable in the vertex shader
- The `m4_identity()` function simply returns a 4×4 identity matrix (column major).

Uniform Variables

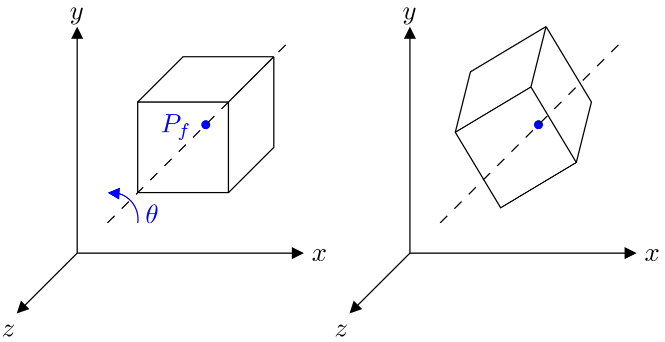
- Simply send a transformation matrix before rendering in the `display()` function:

```
glClear(...);

glUniformMatrix4fv(ctm_location, 1, GL_FALSE,
                                    (GLfloat *) &tr_matrix);

glDrawArrays(...);
```

- Arguments (from first to last)
    - The location (in `GLuint`) from the `glGetUniformLocation()` function
    - The number of elements (1 matrix in this case)
    - Transpose (no transpose in this case)
    - Pointer to the matrix

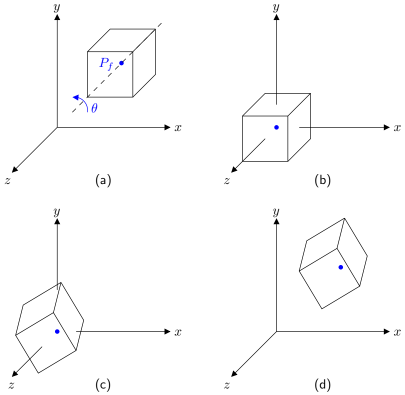
Rotation about a Fixed Point

- So far our fixed point is at the origin
- Generally, it does not have to be at the origin as shown below:



- This can be done by the following steps:
  1. Translate the fixed point P_f to the origin,
  2. Rotate about the z-axis for θ, and
  3. Translate the fixed point back.

Rotation about a Fixed Point



(b) Move the fixed point P_f to the origin

(c) Rotate about the z-axis

(d) Move the fixed point back

Rotation about a Fixed Point

- Suppose a fixed point is located at $P_f = (x_f, y_f, z_f)$ and we want to rotate about z-axis for θ degree
- To move a fixed point P_f to an origin, we can the **translation matrix** $\mathbf{T}(-\mathbf{p}_f)$:

$$\mathbf{T}(-\mathbf{p}_f) = \begin{bmatrix} 1 & 0 & 0 & -x_f \\ 0 & 1 & 0 & -y_f \\ 0 & 0 & 1 & -z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- To rotate about z-axis for θ degree, we use **rotation matrix** $\mathbf{R}_z(\theta)$:

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

- To move the origin to fixed point P_f, we use the **translation matrix** $\mathbf{T}(\mathbf{p}_f)$:

$$\mathbf{T}(\mathbf{p}_f) = \begin{bmatrix} 1 & 0 & 0 & x_f \\ 0 & 1 & 0 & y_f \\ 0 & 0 & 1 & z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about a Fixed Point

- Our transformation matrix $\mathbf{M} = \mathbf{T}(\mathbf{p}_f)\mathbf{R}_z(\theta)\mathbf{T}(-\mathbf{p}_f)$ can be calculated as

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & x_f \\ 0 & 1 & 0 & y_f \\ 0 & 0 & 1 & z_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_f \\ 0 & 1 & 0 & -y_f \\ 0 & 0 & 1 & -z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & x_f \\ 0 & 1 & 0 & y_f \\ 0 & 0 & 1 & z_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 & -x_f\cos\theta + y_f\sin\theta \\ \sin\theta & \cos\theta & 0 & -x_f\sin\theta - y_f\cos\theta \\ 0 & 0 & 1 & -z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & 0 & x_f - x_f\cos\theta + y_f\sin\theta \\ \sin\theta & \cos\theta & 0 & y_f - x_f\sin\theta - y_f\cos\theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

General Rotation

- An arbitrary rotation about the origin can be achieve by rotating about each axis one at a time
  - rotate about z-axis by an angle α,
  - rotate about y-axis by an angle β, and
  - rotate about x-axis by an angle γ.
- This series of rotation corresponds to a rotation matrix

$$\mathbf{R} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z$$

- The difficult part is finding appropriate values for α, β, and γ for an arbitrary rotation.

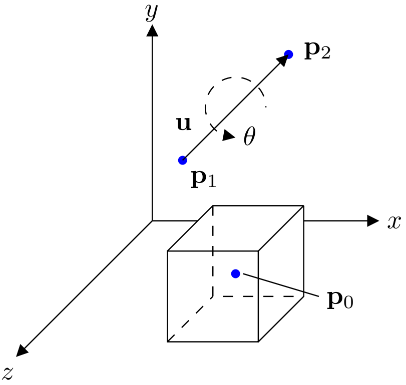
- We generally prefer to draw an object in a familiar way
- For example, a cube:
  - center of mass at the origin,
  - each side is 1×1 unit, and
  - each side aligned with an axis
- Then we can transform an instance of this object with desired size, orientation, and location (**instance transformation**)
- This transformation is of the form

$$\mathbf{M} = \mathbf{TRS}$$

where

  - $\mathbf{S}$ is a scaling matrix,
  - $\mathbf{R}$ is a rotation matrix, and
  - $\mathbf{T}$ is a translation matrix.

- We can also rotate an object about an arbitrary point and line



- Example: Rotate the above cube about the vector $\mathbf{u}$ for θ degree where the fixed point is $\mathbf{p}_0$

Rotation About an Arbitrary Axis

- If $\mathbf{u}$ is defined by two points $\mathbf{p}_1$ and $\mathbf{p}_2$, we have

$$\mathbf{u} = \mathbf{p}_2 - \mathbf{p}_1$$

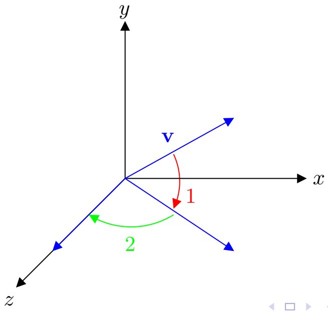
- Since we only need direction from $\mathbf{u}$, we can **normalized** it into a unit vector $\mathbf{v}$ where

$$\mathbf{v} = \frac{\mathbf{u}}{|\mathbf{u}|} = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix}$$

- Using the unit vector $\mathbf{v}$ will simply our calculation in the next part
- What we need are
  1. move the fixed point $\mathbf{p}_0$ to the origin using the translation matrix $\mathbf{T}(-\mathbf{p}_0)$
  2. rotate the object using the rotation matrix $\mathbf{R}$, and
  3. move the fixed point back using the translation matrix $\mathbf{T}(\mathbf{p}_0)$.

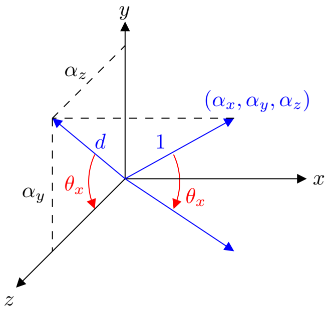
Rotation About an Arbitrary Axis

- To rotate about $\mathbf{v}$ for θ degree where the fixed point is at the origin, we can perform the following step:
  1. Align the axis of $\mathbf{v}$ with the z-axis by a rotation
  2. Rotate by θ about z-axis
  3. Rotate $\mathbf{v}$ back to its original direction
- Rotating the axis of $\mathbf{v}$ to align with the z-axis can be done in two steps:
  1. Rotate $\mathbf{v}$ into the pane $y = 0$
  2. Rotate $\mathbf{v}$ into the pane $x = 0$



Rotate $\mathbf{v}$ to the Pane $y = 0$

- To rotate $\mathbf{v}$ to the pane $y = 0$, it is the same as rotating the projection of $\mathbf{v}$ on the pane $x = 0$ about the x-axis to the pane $y = 0$



- From the above picture, we have
  - $d = \sqrt{\alpha_y^2 + \alpha_z^2}$,
  - $\sin\theta_x = \alpha_y / d$, and
  - $\cos\theta_x = \alpha_z / d$.

Rotate $\mathbf{v}$ to the Pane $y = 0$

- Recall the rotation matrix about x-axis

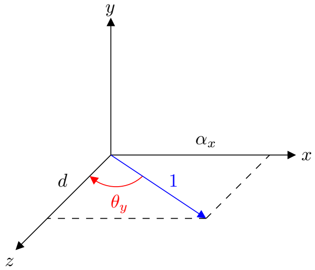
$$\mathbf{R}_x = \mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Thus we have

$$\mathbf{R}_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha_z/d & -\alpha_y/d & 0 \\ 0 & \alpha_y/d & \alpha_z/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotate $\mathbf{v}$ to the Pane $x = 0$

- Once we rotate $\mathbf{v}$ about x-axis to the pane $y = 0$, we need to rotate about y-axis to align $\mathbf{v}$ with the z-axis



- From the above picture, we have
  - $\sin\theta_y = \alpha_x/1 = \alpha_x$, and
  - $\cos\theta_y = d/1 = d$.
- **Note** that this is a clockwise rotation ($\sin -\theta = -\sin\theta$ and $\cos -\theta = \cos\theta$)

Rotate $\mathbf{v}$ to the Pane $x = 0$

- Recall the rotation matrix about x-axis

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

- Thus we have

$$\mathbf{R}_y(-\theta_y) = \begin{bmatrix} d & 0 & -\alpha_x & 0 \\ 0 & 1 & 0 & 0 \\ \alpha_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Put everything together, we get our final transformation matrix

$$\mathbf{M} = \mathbf{T}(\mathbf{p}_0)\mathbf{R}_x(-\theta_x)\mathbf{R}_y(-\theta_y)\mathbf{R}_z(\theta)\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x)\mathbf{T}(-\mathbf{p}_0)$$

  where

  - $\mathbf{T}(-\mathbf{p}_0)$ is the translation matrix that moves the fixed point to the origin
  - $\mathbf{R}_x(\theta_x)$ is the rotation matrix that rotate the vector to the pane $y = 0$
  - $\mathbf{R}_y(\theta_y)$ is the rotation matrix that rotate the vector to align with z-axis
  - $\mathbf{R}_z(\theta)$ is the rotation matrix that rotate the object by θ degree
  - $\mathbf{R}_y(-\theta_y)$ and $\mathbf{R}_x(-\theta_x)$ rotates the vector back to its original direction
  - $\mathbf{T}(\mathbf{p}_0)$ moves the fixed point back to its original location

Example

- Suppose we want to rotate an object by θ degree about the line (vector $\mathbf{u}$) from the origin to the point $(1, 2, 3)$.
- First normalize the vector $\mathbf{u}$ to the unit vector $\mathbf{v}$

$$\mathbf{v} = \frac{\mathbf{u}}{|\mathbf{u}|} = \frac{1}{|\mathbf{u}|}\left(\begin{bmatrix}1\\2\\3\\1\end{bmatrix} - \begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right) = \frac{1}{|\mathbf{u}|}\begin{bmatrix}1\\2\\3\\0\end{bmatrix}$$

- Since $|\mathbf{u}| = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{14}$, we have

$$\mathbf{v} = \begin{bmatrix}1/\sqrt{14}\\2/\sqrt{14}\\3/\sqrt{14}\\0\end{bmatrix} = \begin{bmatrix}\alpha_x\\\alpha_y\\\alpha_z\\0\end{bmatrix}$$

and

$$d = \sqrt{\alpha_y^2 + \alpha_z^2} = \sqrt{(\tfrac{2}{\sqrt{14}})^2 + (\tfrac{3}{\sqrt{14}})^2} = \sqrt{\frac{4}{14} + \frac{9}{14}} = \sqrt{\frac{13}{14}}$$

Example

- First rotate $\mathbf{v}$ about x-axis to the pane $y = 0$ using the rotation matrix $\mathbf{R}_x(\theta_x)$
- Recall that $\cos\theta_x = \alpha_z/d = \frac{3/\sqrt{14}}{\sqrt{13/14}} = \frac{3}{\sqrt{13}}$. Thus $\theta_x = \cos^{-1}\frac{3}{\sqrt{13}}$.
- Or we can simply use $\mathbf{R}_x(\theta_x)$ as discussed earlier:

$$\mathbf{R}_x(\cos^{-1}\frac{3}{\sqrt{13}}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha_z/d & -\alpha_y/d & 0 \\ 0 & \alpha_y/d & \alpha_z/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{3}{\sqrt{13}} & -\frac{2}{\sqrt{13}} & 0 \\ 0 & \frac{2}{\sqrt{13}} & \frac{3}{\sqrt{13}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{R}_x(-\cos^{-1}\frac{3}{\sqrt{13}}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{3}{\sqrt{13}} & \frac{2}{\sqrt{13}} & 0 \\ 0 & -\frac{2}{\sqrt{13}} & \frac{3}{\sqrt{13}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example

- Next, we rotate the new $\mathbf{v}$ to align with z-axis using the rotation matrix $\mathbf{R}_y(\theta_y)$
- Recall that $\cos\theta_y = d/1 = \sqrt{\frac{13}{14}}$ and this is a clockwise rotation. Thus $\theta_y = -\cos^{-1}\sqrt{\frac{13}{14}}$
- Or we can simply use $\mathbf{R}_y(\theta_y)$ as discussed earlier:

$$\mathbf{R}_y(-\cos^{-1}\sqrt{\frac{13}{14}}) = \begin{bmatrix} d & 0 & -\alpha_x & 0 \\ 0 & 1 & 0 & 0 \\ \alpha_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{13}{14}} & 0 & -\frac{1}{\sqrt{14}} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{\sqrt{14}} & 0 & \sqrt{\frac{13}{14}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{R}_y(\cos^{-1}\sqrt{\frac{13}{14}}) = \begin{bmatrix} \sqrt{\frac{13}{14}} & 0 & \frac{1}{\sqrt{14}} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{\sqrt{14}} & 0 & \sqrt{\frac{13}{14}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example

- Next, we rotate the new $\mathbf{v}$ about the z-axis by θ degree using the rotation matrix $\mathbf{R}_z(\theta)$ ($\mathbf{R}_z(45)$ in this case)
- From the rotation about z-axis, we have

$$\mathbf{R}_z = \mathbf{R}_z(45) = \begin{bmatrix} \cos 45 & -\sin 45 & 0 & 0 \\ \sin 45 & \cos 45 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that $\sin 45 = \cos 45 = \frac{\sqrt{2}}{2}$

Example

- Put them all together, we have our rotation matrix

$$\mathbf{R} = \mathbf{R}_x(-\cos^{-1}\frac{3}{\sqrt{13}})\mathbf{R}_y(\cos^{-1}\sqrt{\frac{13}{14}})\mathbf{R}_z(45)\mathbf{R}_y(-\cos^{-1}\sqrt{\frac{13}{14}})\mathbf{R}_x(\cos^{-1}\frac{3}{\sqrt{13}})$$

- Let's check:

$$\begin{aligned}
\mathbf{M}_1 &= \mathbf{R}_y(-\cos^{-1}\sqrt{\frac{13}{14}})\mathbf{R}_x(\cos^{-1}\frac{3}{\sqrt{13}}),\\
\mathbf{M}_2 &= \mathbf{R}_z(45)\mathbf{M}_1,\\
\mathbf{M}_3 &= \mathbf{R}_y(\cos^{-1}\sqrt{\frac{13}{14}})\mathbf{M}_2, \text{ and}\\
\mathbf{M}_4 &= \mathbf{R}_x(-\cos^{-1}\frac{3}{\sqrt{14}})\mathbf{M}_3 = \mathbf{R}.
\end{aligned}$$

Example

- $\mathbf{M}_1 = \mathbf{R}_y(-\cos^{-1}\sqrt{\frac{13}{14}})\mathbf{R}_x(\cos^{-1}\frac{3}{\sqrt{13}})$, we have

$$\mathbf{M}_1 = \begin{bmatrix} \sqrt{\frac{13}{14}} & 0 & -\frac{1}{\sqrt{14}} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{\sqrt{14}} & 0 & \sqrt{\frac{13}{14}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{3}{\sqrt{13}} & -\frac{2}{\sqrt{13}} & 0 \\ 0 & \frac{2}{\sqrt{13}} & \frac{3}{\sqrt{13}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{\frac{13}{14}} & -\frac{2}{\sqrt{13}\sqrt{14}} & -\frac{3}{\sqrt{13}\sqrt{14}} & 0 \\ 0 & \frac{3}{\sqrt{13}} & -\frac{2}{\sqrt{13}} & 0 \\ \frac{1}{\sqrt{14}} & \frac{2}{\sqrt{14}} & \frac{3}{\sqrt{14}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- $\mathbf{M}_2 = \mathbf{R}_z(45)\mathbf{M}_1$, we have

$$\mathbf{M}_2 = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{\frac{13}{14}} & -\frac{2}{\sqrt{13}\sqrt{14}} & -\frac{3}{\sqrt{13}\sqrt{14}} & 0 \\ 0 & \frac{3}{\sqrt{13}} & -\frac{2}{\sqrt{13}} & 0 \\ \frac{1}{\sqrt{14}} & \frac{2}{\sqrt{14}} & \frac{3}{\sqrt{14}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\sqrt{2}\sqrt{13}}{2\sqrt{14}} & \frac{-3\sqrt{7}-\sqrt{2}}{\sqrt{14}\sqrt{13}} & \frac{4\sqrt{7}-3\sqrt{2}}{2\sqrt{13}\sqrt{14}} & 0 \\ \frac{\sqrt{2}\sqrt{13}}{2\sqrt{14}} & \frac{3\sqrt{7}-\sqrt{2}}{\sqrt{13}\sqrt{14}} & \frac{-4\sqrt{7}-3\sqrt{2}}{2\sqrt{13}\sqrt{14}} & 0 \\ \frac{1}{\sqrt{14}} & \frac{2}{\sqrt{14}} & \frac{3}{\sqrt{14}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example

- $\mathbf{M}_3 = \mathbf{R}_y(\cos^{-1}\sqrt{\frac{13}{14}})\mathbf{M}_2$, we have

$$\mathbf{M}_3 = \begin{bmatrix} \sqrt{\frac{13}{14}} & 0 & \frac{1}{\sqrt{14}} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{\sqrt{14}} & 0 & \sqrt{\frac{13}{14}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}\sqrt{13}}{2\sqrt{14}} & \frac{-3\sqrt{7}-\sqrt{2}}{\sqrt{14}\sqrt{13}} & \frac{4\sqrt{7}-3\sqrt{2}}{2\sqrt{13}\sqrt{14}} & 0 \\ \frac{\sqrt{2}\sqrt{13}}{2\sqrt{14}} & \frac{3\sqrt{7}-\sqrt{2}}{\sqrt{13}\sqrt{14}} & \frac{-4\sqrt{7}-3\sqrt{2}}{2\sqrt{13}\sqrt{14}} & 0 \\ \frac{1}{\sqrt{14}} & \frac{2}{\sqrt{14}} & \frac{3}{\sqrt{14}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{2+13\sqrt{2}}{28} & \frac{2-\sqrt{2}-3\sqrt{7}}{14} & \frac{6-3\sqrt{2}+4\sqrt{7}}{28} & 0 \\ \frac{\sqrt{2}\sqrt{13}}{2\sqrt{14}} & \frac{3\sqrt{7}-\sqrt{2}}{\sqrt{13}\sqrt{14}} & \frac{-4\sqrt{7}-3\sqrt{2}}{2\sqrt{13}\sqrt{14}} & 0 \\ \frac{2\sqrt{13}-\sqrt{2}\sqrt{13}}{28} & \frac{26+\sqrt{2}+3\sqrt{7}}{14\sqrt{13}} & \frac{78+3\sqrt{2}-4\sqrt{7}}{28\sqrt{13}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Let $\mathbf{M}_4 = \mathbf{R}_x(-\cos^{-1}\frac{3}{\sqrt{14}})\mathbf{M}_3$, we have

$$\mathbf{M}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{3}{\sqrt{13}} & \frac{2}{\sqrt{13}} & 0 \\ 0 & -\frac{2}{\sqrt{13}} & \frac{3}{\sqrt{13}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{2+13\sqrt{2}}{28} & \frac{2-\sqrt{2}-3\sqrt{7}}{14} & \frac{6-3\sqrt{2}+4\sqrt{7}}{28} & 0 \\ \frac{\sqrt{2}\sqrt{13}}{2\sqrt{14}} & \frac{3\sqrt{7}-\sqrt{2}}{\sqrt{13}\sqrt{14}} & \frac{-4\sqrt{7}-3\sqrt{2}}{2\sqrt{13}\sqrt{14}} & 0 \\ \frac{2\sqrt{13}-\sqrt{2}\sqrt{13}}{28} & \frac{26+\sqrt{2}+3\sqrt{7}}{14\sqrt{13}} & \frac{78+3\sqrt{2}-4\sqrt{7}}{28\sqrt{13}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{2+13\sqrt{2}}{28} & \frac{2-\sqrt{2}-3\sqrt{7}}{14} & \frac{6-3\sqrt{2}+4\sqrt{y}}{28} & 0 \\ \frac{2-\sqrt{2}+3\sqrt{7}}{14} & \frac{4+5\sqrt{2}}{14} & \frac{6-3\sqrt{2}-\sqrt{7}}{14} & 0 \\ \frac{6-3\sqrt{2}-4\sqrt{7}}{28} & \frac{6-3\sqrt{2}+\sqrt{7}}{14} & \frac{18+5\sqrt{2}}{28} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Vertices of an object are transformed by **current transformation matrix** ($\mathbf{C}$),

$$\mathbf{p}' = \mathbf{C}\mathbf{p}$$

  for a vertex $\mathbf{p}$
- Can be done in either application or shader.
- The value of $\mathbf{C}$ can simply be set

$$\mathbf{C} = \mathbf{I}, \quad \mathbf{C} = \mathbf{T}, \quad \mathbf{C} = \mathbf{S}, \quad \mathbf{C} = \mathbf{R}, \text{ etc}$$

  or by modifying the CTM by multiplication

$$\mathbf{C} = \mathbf{TC}, \mathbf{C} = \mathbf{CT}, \mathbf{C} = \mathbf{SC}, \mathbf{C} = \mathbf{CS}, \mathbf{C} = \mathbf{RC}, \mathbf{C} = \mathbf{CR}, \text{ etc}$$

- It is a good idea to have functions that we can use to generate transformation matrices and matrix multiplication:

```
mat4 rotate_about_x(float theta);
mat4 rotate_about_y(float theta);
mat4 rotate_about_z(float theta);
mat4 translate(float dx, float dy, float dz);
mat4 scale(float sx, float sy, float sz);
mat4 m4m4_multiplication(mat4 lm, mat4 rm);
```

- Where mat4 is defined as a structure as follows:

```
typedef struct {
    GLfloat x;
    GLfloat y;
    GLfloat z;
    GLfloat w;
} vec4;

typedef struct {
    vec4 x;
    vec4 y;
    vec4 z;
    vec4 w;
} mat4;
```

Example

- Matrices in OpenGL are column-major
- If we define a variable `m` as follows:

```
GLfloat m[12];
```

  `m` can be considered as a 4×4 matrix in OpenGL as follows:

$$\begin{bmatrix} \texttt{m[0]} & \texttt{m[4]} & \texttt{m[8]} & \texttt{m[12]} \\ \texttt{m[1]} & \texttt{m[5]} & \texttt{m[9]} & \texttt{m[13]} \\ \texttt{m[2]} & \texttt{m[6]} & \texttt{m[10]} & \texttt{m[14]} \\ \texttt{m[3]} & \texttt{m[7]} & \texttt{m[11]} & \texttt{m[15]} \end{bmatrix}$$

- Note that 4×4 matrix type has been defined in shader already
  - Name `mat4`
  - Multiplication has been defined as well and allow us to perform the following:

```
mat4 ctm = rx * ry;
```

    where `rx` and `ry` are 4×4 matrix
  - Can be initialized using a constructor (again **column-major**):

```
mat4 m = mat4(vec4(1.0, 0.0, 0.0, 0.0), vec4(0.0, 1.0, 0.0, 0.0),
              vec4(0.0, 0.0, 1.0, 0.0), vec4(0.0, 0.0, 0.0, 1.0));
```

Example

- The following function changes the matrix c into a rotation matrix about z-axis for theta degree:

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

```
mat4 rotate_z(float theta)
{
    mat4 result;
    float s = (M_PI/180.0)*theta;

    result.x.x = cos(s); result.y.x = -sin(s); result.z.x = 0; result.w.x = 0;
    result.x.y = sin(s); result.y.y = cos(s);  result.z.y = 0; result.w.y = 0;
    result.x.z = 0;      result.y.z = 0;       result.z.z = 1; result.w.z = 0;
    result.x.w = 0;      result.y.w = 0;       result.z.w = 0; result.w.w = 1;

    return result;
}
```

Example

- Create a transformation matrix for a 45-degree rotation about the line through the origin and the point $(1, 2, 3)$ with a fixed point of $(4, 5, 6)$:
- Recall that

$$\mathbf{v} = \mathbf{u}/|\mathbf{u}| = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 0 \end{bmatrix} / \sqrt{1^2 + 2^2 + 3^2} = \begin{bmatrix} 1/\sqrt{14} \\ 2/\sqrt{14} \\ 3/\sqrt{14} \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \\ 0 \end{bmatrix}$$

and

- $d = \sqrt{\alpha_y^2 + \alpha_z^2} = \sqrt{13/14}$,
- $\alpha_x = \cos^{-1}(\alpha_z/d) = \cos^{-1}(\frac{3}{\sqrt{14}}/\sqrt{\frac{13}{14}}) = \cos^{-1}(3/\sqrt{13})$, and
- $\alpha_y = \cos^{-1} d = \cos^{-1} \sqrt{13/14}$.

Example

```
mat4 m1, m2, m3, ctm;
float theta_x, theta_y;
float radian_to_degree = 180.0/M_PI;

theta_x = radian_to_degree * acos(3.0 / sqrt(13.0));
theta_y = radian_to_degree * acos(sqrt(13.0/14.0));

rotate_x(m1, theta_x);                 // m1 = Rx(+)
rotate_y(m2, -theta_y);                // m2 = Ry(-)
mat4_multiplication(m3, rmy, rx);      // m3 = Ry(-)Rx(+)
rotate_z(m1, 45.0);                    // m1 = Rz(45)
mat4_multiplication(m2, m1, m3);       // m2 = Rz(45)Ry(-)Rx(+)
rotate_y(m1, theta_y);                 // m1 = Ry(+)
mat4_multiplication(m3, m1, m2);       // m3 = Ry(+)Rz(45)Ry(-)Rx(+)
rotate_x(m1, -theta_x);                // m1 = Rx(-)
mat4_multiplication(ctm, m1, m3);      // ctm = Rx(-)Ry(+)Rz(45)Ry(-)Rx(+)
```

Quaternions

- Quaternions are a number system (an extension to the complex number system)
- Suitable for three-dimensional rotations

Taylor Series

- Taylor Series is defined as

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots$$

- Let's take a look at e^{ix} where i is the imaginary unit ($i^2 = -1$):

$$\begin{aligned} e^{ix} = \sum_{n=0}^{\infty} \frac{(ix)^n}{n!} &= 1 + ix + \frac{(ix)^2}{2!} + \frac{(ix)^3}{3!} + \frac{(ix)^4}{4!} + \frac{(ix)^5}{5!} + \dots \\ &= 1 + ix - \frac{x^2}{2!} - \frac{ix^3}{3!} + \frac{x^4}{4!} + \frac{ix^5}{5!} - \dots \\ &= (1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots) + i(x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots) \end{aligned}$$

Taylor Series

- A function $f(x)$ can be expressed as

$$f(x) = c_0 + c_1(x-a) + c_2(x-a)^2 + c_3(x-a)^3 + \dots$$

- Find c_0:

$$\begin{aligned} f(a) &= c_0 + c_1(a-a) + c_2(a-a)^2 + c_3(a-a)^3 + \dots \\ &= c_0 \end{aligned}$$

- Find c_1:

$$\begin{aligned} f'(x) &= c_1 + 2c_2(x-a) + 3c_3(x-a)^2 + 4c_4(x-a)^3 \dots \\ f'(a) &= c_1 + 2c_2(a-a) + 3c_3(a-a)^2 + 4c_4(a-a)^3 \dots \\ &= c_1 \end{aligned}$$

- Find c_2:

$$\begin{aligned} f''(x) &= 2c_2 + 2 \cdot 3c_3(x-a) + 3 \cdot 4c_4(x-a)^2 + \dots \\ &= (2!)c_2 + (3!)c_3(x-a) + 3 \cdot 4c_4(x-a)^2 + \dots \\ f''(a) &= (2!)c_2 + (3!)c_3(a-a) + 3 \cdot 4c_4(a-a)^2 + \dots \\ &= (2!)c_2 \\ f''(a)/(2!) &= c_2 \end{aligned}$$

Taylor Series

- Keep doing the derivative, we get

$$c_0 = f(a)$$
$$c_1 = \frac{f'(a)}{1!}$$
$$c_2 = \frac{f''(a)}{2!}$$
$$c_3 = \frac{f'''(a)}{3!}$$
$$\vdots$$

- Therefore:

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \ldots$$

Taylor Series

- Let $f(x) = \cos(x)$
  - $f'(x) = -\sin(x),\ f''(x) = -\cos(x),\ f'''(x) = \sin(x),\ \ldots$
- Thus

$$\cos(x) = \cos(a) - \frac{\sin(a)}{1!}(x-a) - \frac{\cos(a)}{2!}(x-a)^2 + \frac{sin(a)}{3!}(x-a)^3 + \ldots$$

- Let $a = 0$ ($\cos(0) = 1$ and $sin(0) = 0$) we obtain:

$$\begin{aligned}\cos(x) &= 1 - \frac{0}{1!}(x-0) - \frac{1}{2!}(x-0)^2 + \frac{0}{3!}(x-0)^3 + \ldots \\ &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \ldots\end{aligned}$$

- Similarly, if we let $f(x) = \sin(x)$, we get:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots$$

- Recall e^{ix}:

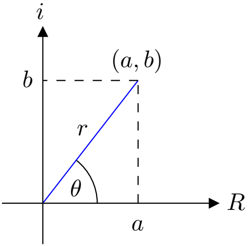
$$\begin{aligned}e^{ix} &= (1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \ldots) + i(x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots) \\ &= \cos(x) + i\sin(x)\end{aligned}$$

Euler's Identity

- Euler's identity
$$e^{i\pi} + 1 = 0$$
where
  - e is the Euler's number ($e = \sum_{n=0}^{\infty} \frac{1}{n!} \approx 2.718282\ldots$)
  - i is the imaginary unit ($i^2 = -1$)
- Recall that:
$$e^{ix} = \cos(x) + i\sin(x)$$
- Since $\cos(\pi) = -1$ and $\sin(\pi) = 0$, we get:
$$\begin{aligned} e^{i\pi} &= \cos(\pi) + i\sin(\pi) \\ &= -1 + i(0) \\ &= -1 \end{aligned}$$
- Therefore, $e^{i\pi} + 1 = 0$

Quaternion

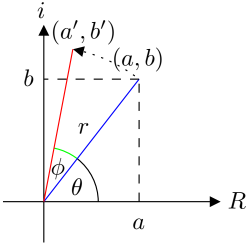
- Consider an imaginary number $a + bi$ in the imaginary pane:



- We know the followings:
  - $r^2 = a^2 + b^2$
  - $a = r\cos\theta$
  - $b = r\sin\theta$
  - $\theta = \tan^{-1}(\frac{b}{a})$
- Thus

$$\begin{aligned} a + bi &= r\cos\theta + ir\sin\theta \\ &= r(\cos\theta + i\sin\theta) \\ &= re^{i\theta} \end{aligned}$$

- Suppose we rotate (a, b) about the origin by ϕ to (a', b') as follows:



- We have

$$\begin{aligned} a' + b'i &= re^{i(\theta+\phi)} \\ &= re^{i\theta}e^{i\phi} \end{aligned}$$

- This shows that $e^{i\phi}$ is an operator for rotation in the complex plane.

Quaternions

- To rotate a point (a, b) to a new point (a', b') about the origin, we only need one scalar value ϕ.
- But in three-dimensional space, we need both direction (vector) and amount of rotation.
- A quaternion consists of four scalars

$$a = (q_0, q_1, q_2, q_3) = (q_0, \mathbf{q})$$

  where $\mathbf{q} = (q_1, q_2, q_3)$.
- **Note** that $\mathbf{q}$ is a vector in three-dimensional space.
- A quaternion generally represented by

$$a = q_0 + q_1 i + q_2 j + q_3 k$$

  where i, j, and k are imaginary numbers such that:

$$i^2 = j^2 = k^2 = ijk = -1$$

Quaternions

- Note that quaternions is not commutative ($ij \neq ji$)
- Recall the property $i^2 = j^2 = k^2 = ijk = -1$

$$\begin{aligned} ijk &= -1 \\ ijkk &= -1 \cdot k \\ ij(k^2) &= -k \\ ij(-1) &= -k \\ -ij &= -k \\ ij &= k \end{aligned}$$

$$\begin{aligned} ijk &= -1 \\ iijk &= i(-1) \\ (-1)jk &= -i \\ -jk &= -i \\ jk &= i \\ jjk &= ji \\ (-1)k &= ji \\ -k &= ji \end{aligned}$$

Quaternions

- From the property $i^2 = j^2 = k^2 = ijk = -1$, we obtain the following:

$$\begin{array}{ll} ij = k & ji = -k \\ jk = i & kj = -i \\ ki = j & ik = -j \end{array}$$

- Consider two quaternions
  - $a = q_0 + q_1 i + q_2 j + q_3 k = (q_0, \mathbf{q})$ where $\mathbf{q} = (q_1, q_2, q_3)$
  - $b = p_0 + p_1 i + p_2 j + p_3 k = (p_0, \mathbf{p})$ where $\mathbf{p} = (p_1, p_2, p_3)$
- Quaternion addition:

$$\begin{aligned} a + b &= (q_0 + q_1 i + q_2 j + q_3 k) + (p_0 + p_1 i + p_2 j + p_3 k) \\ &= (q_0 + p_0) + (q_1 + p_1)i + (q_2 + p_2)j + (q_3 + p_3)k \\ &= (q_0, \mathbf{q}) + (p_0, \mathbf{p}) \\ &= (q_0 + p_0, \mathbf{q} + \mathbf{p}) \end{aligned}$$

where $\mathbf{q} + \mathbf{p}$ is a vector addition.

- Quaternion Multiplication

$$
\begin{aligned}
ab =&(q_0 + q_1 i + q_2 j + q_3 k)(p_0 + p_1 i + p_2 j + p_3 k)\\
=&q_0p_0 + q_0p_1 i + q_0p_2 j + q_0p_3 k+\\
&q_1p_0 i + q_1p_1 ii + q_1p_2 ij + q_1p_3 ik+\\
&q_2p_0 j + q_2p_1 ji + q_2p_2 jj + q_2p_3 jk+\\
&q_3p_0 k + q_3p_1 ki + q_3p_2 kj + q_3p_3 kk\\
=&(q_0p_0 - q_1p_1 - q_2p_2 - q_3p_3)+\\
&(q_0p_1 + q_1p_0 + q_2p_3 - q_3p_2)i+\\
&(q_0p_2 - q_1p_3 + q_2p_0 + q_3p_1)j+\\
&(q_0p_3 + q_1p_2 - q_2p_1 + q_3p_0)k
\end{aligned}
$$

- Recall that if $\mathbf{q} = q_1 i + q_2 j + q_3 k$ and $\mathbf{p} = p_1 i + p_2 j + p_3 k$ and consider i, j, and k as unit vectors, we have
  - $\mathbf{q} \cdot \mathbf{p} = q_1 p_1 + q_2 p_2 + q_3 p_3$
  - $\mathbf{q} \times \mathbf{p} = (q_2 p_3 - q_3 p_2) i + (q_3 p_1 - q_1 p_3) j + (q_1 p_2 - q_2 p_1) k$
  - $q_0 \mathbf{p} = q_0 p_1 i + q_0 p_2 j + q_0 p_3 k$
  - $p_0 \mathbf{q} = q_1 p_0 i + q_2 p_0 j + q_3 p_0 k$
- Recall from previous slide:

$$
\begin{aligned}
ab =& (q_0 p_0 - q_1 p_1 - q_2 p_2 - q_3 p_3) + \\
& (q_0 p_1 + q_1 p_0 + q_2 p_3 - q_3 p_2) i + \\
& (q_0 p_2 - q_1 p_3 + q_2 p_0 + q_3 p_1) j + \\
& (q_0 p_3 + q_1 p_2 - q_2 p_1 + q_3 p_0) k \\
=& (q_0 p_0 - \mathbf{q} \cdot \mathbf{p}) + q_0 \mathbf{p} + p_0 \mathbf{q} + \mathbf{q} \times \mathbf{p} \\
=& (q_0 p_0 - \mathbf{q} \cdot \mathbf{p}, q_0 \mathbf{p} + p_0 \mathbf{q} + \mathbf{q} \times \mathbf{p})
\end{aligned}
$$

Quaternions

- Magnitude of a quaternion $a = q_0 + q_1 i + q_2 j + q_3 k = (q_0, \mathbf{q})$ can be calculated as

$$\begin{aligned} |a|^2 &= q_0^2 + q_1^2 + q_2^2 + q_3^2 \\ &= q_0^2 + \mathbf{q} \cdot \mathbf{q} \end{aligned}$$

- Multiplicative identity in quaternion is $(1, \mathbf{0})$

$$(q_0 + q_1 i + q_2 j + q_3 k)(1 + 0i + 0j + 0k) = (q_0 + q_1 i + q_2 j + q_3 k)$$

Quaternions

- Let $a = (q_0 + q_1 i + q_2 j + q_3 k)$ and $b = (q_0 - q_1 i - q_1 j - q_1 k)$

$$
\begin{aligned}
ab =& (q_0 + q_1 i + q_2 j + q_3 k)(q_0 - q_1 i - q_2 j - q_3 k) \\
=& q_0 q_0 - q_0 q_1 i - q_0 q_2 j - q_0 q_3 k + \\
& q_1 q_0 i - q_1 q_1 ii - q_1 q_2 ij - q_1 q_3 ik + \\
& q_2 q_0 j - q_2 q_1 ji - q_2 q_2 jj - q_2 q_3 jk + \\
& q_3 p_0 k - q_3 q_1 ki - q_3 q_2 kj - q_3 p_3 kk \\
=& q_0^2 + q_1^2 + q_2^2 + q_3^2
\end{aligned}
$$

- From above, if $a^{-1} = \frac{1}{|a|^2} b$, then $aa^{-1} = 1 = (1, \mathbf{0})$
- In other words, if $a = q_0 + q_1 i + q_2 j + q_3 k = (q_0, \mathbf{q})$, then the multiplicative identity of a is

$$a^{-1} = \frac{1}{|a|^2}(q_0, -\mathbf{q})$$

Quaternions and Rotation

- A quaternion $p = (0, \mathbf{p})$ where $\mathbf{p} = (x, y, z)$ can be used to represent a point in three-dimensional space.
- Consider a quaternion $r = (\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\mathbf{v})$ where v has a unit length ($|v| = 1$)
- Consider the magnitude of r:

$$
\begin{aligned}
|r|^2 &= (\cos^2\frac{\theta}{2}) + (\sin^2\frac{\theta}{2})|v|^2 \\
&= (\cos^2\frac{\theta}{2}) + (\sin^2\frac{\theta}{2}) \\
&= 1
\end{aligned}
$$

- Thus

$$
r^{-1} = \frac{1}{|r|^2}(\cos\frac{\theta}{2}, -\sin\frac{\theta}{2}\mathbf{v}) = (\cos\frac{\theta}{2}, -\sin\frac{\theta}{2}\mathbf{v})
$$