

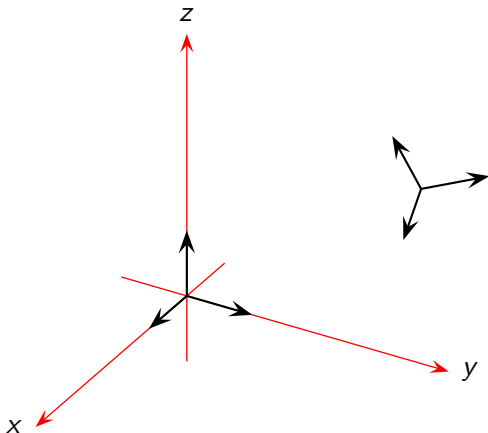
Transforms

Geoffrey Matthews

Department of Computer Science
Western Washington University

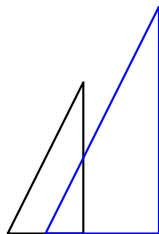
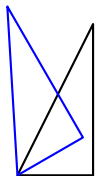
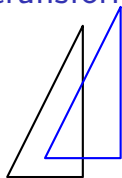
Fall 2011

Transforms



- ▶ Moving from one frame to another.
- ▶ Describe an object in its own frame, then describe all points in the object in the world frame.
- ▶ Describe the world in its natural frame, then describe everything in the world from the camera's frame.
- ▶ Describe everything in the 3D world, then move it to the 2D world of the screen.

Simple transformations

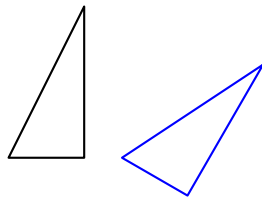
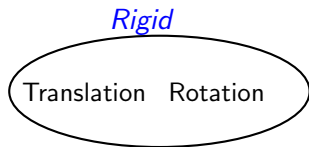


- ▶ Translation
- ▶ Rotation
- ▶ Uniform scaling

Transformations are used

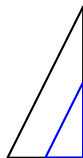
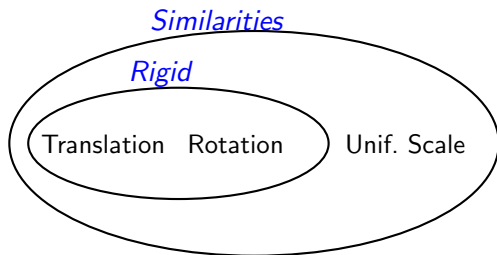
- ▶ Position objects in a scene
- ▶ Change shape of objects
- ▶ Create multiple copies of objects
- ▶ Position camera
- ▶ Projection for virtual cameras
- ▶ Animations

Rigid-body (Euclidean) Transforms



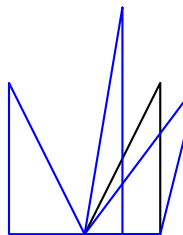
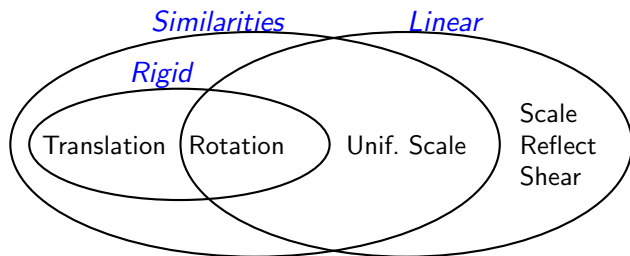
- Preserves distances and angles

Similitudes / Similarity Transforms



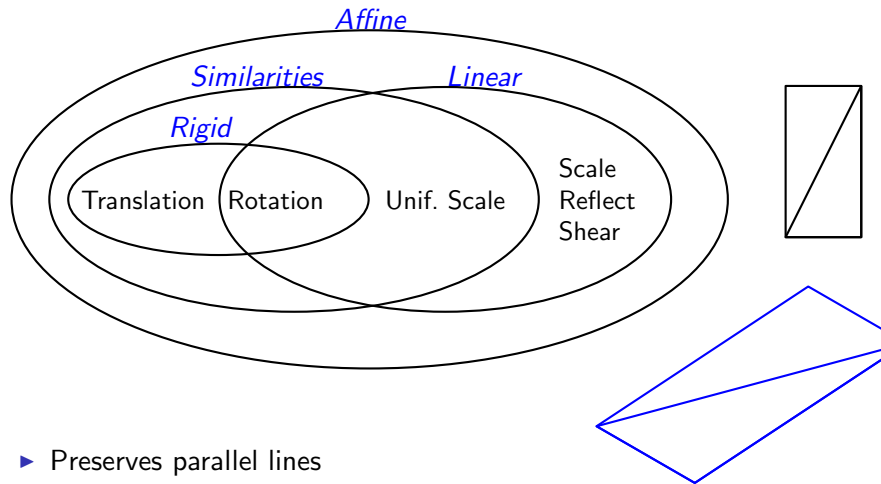
- Preserves angles

Linear transforms



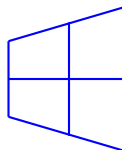
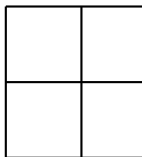
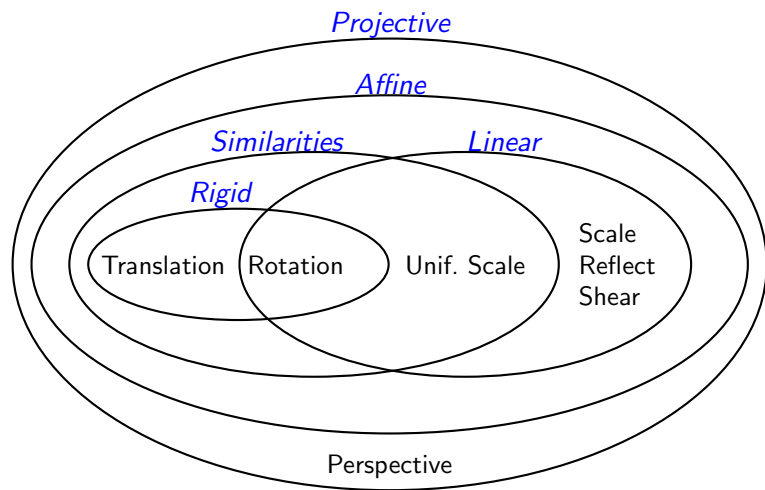
- ▶ $L(p + q) = L(p) + L(q)$
- ▶ $L(ap) = aL(p)$

Affine Transforms



- Preserves parallel lines

Projective Transforms



- Preserves lines

Homogeneous coordinates

- ▶ Recall a 3D frame is three vectors and a point: x, y, z, p .
- ▶ We can represent a point q as *coordinates* in this frame as a 4-vector $(a, b, c, 1)$ because $q = [x, y, z, p] \cdot [a, b, c, 1]^T$
- ▶ Likewise we can represent a vector v as *coordinates* in this frame with a 4-vector $(a, b, c, 0)$ because $v = [x, y, z, p] \cdot [a, b, c, 0]^T$
- ▶ These are called *homogeneous coordinates*
- ▶ They help distinguish between points and vectors
- ▶ They simplify other calculations with points and vectors, in particular, transformations.

Representing transforms with matrices

- ▶ A general linear transformation:

$$x' = ax + by + c$$

$$y' = dx + ey + f$$

- ▶ Multiplication and addition:

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

$$p' = Mp + t$$

Homogeneous coordinates

- If we add another dimension, we can get by with just multiplication.

$$\begin{aligned}x' &= ax + by + c \\y' &= dx + ey + f \\1 &= 1\end{aligned}$$
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
$$p' = Mp$$

- In 2D we use 3×3 matrices.

Homogeneous coordinates

- ▶ In 3D we use 4×4 matrices.
- ▶ Each point has an extra value, w , usually 1.

$$x' = ax + by + cz + d$$

$$y' = ex + fy + gz + h$$

$$z' = ix + jy + kz + l$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

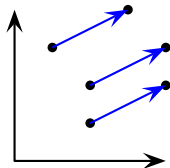
$$p' = Mp$$

Homogeneous coordinates

- ▶ Each point has an extra value, w , usually 1.
- ▶ If M is an *affine* transformation, w will remain 1.
- ▶ We use $w \neq 1$ only in projections.
- ▶ If $w \neq 1$ for a point, we normalize by dividing by w before using.

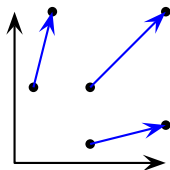
$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix}$$
$$\Leftrightarrow \begin{bmatrix} x'/w' \\ y'/w' \\ z'/w' \\ 1 \end{bmatrix}$$
$$Mp = p'$$

Translate



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

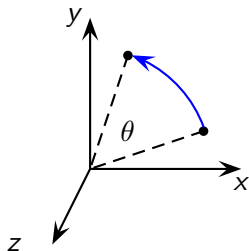
Scale



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- ▶ Isotropic (uniform) scaling: $s_x = s_y = s_z$.
- ▶ Generally avoid scaling; creates difficulties with normals.

Rotation



- ▶ Righthand rotation about the z axis in a righthand frame.
- ▶ Lefthand rotation about the z axis in a lefthand frame.

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

Rotation

- Righthand rotation about the x axis.

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

- Righthand rotation about the y axis.

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

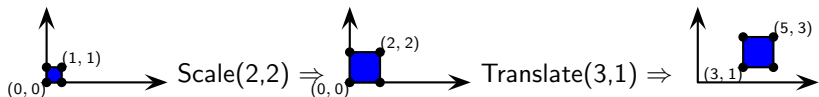
- Righthand rotation about the z axis.

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

Rotation about an arbitrary axis

- ▶ Rodrigues rotation matrix
http://en.wikipedia.org/wiki/Rotation_matrix
- ▶ Fairly easy derivation using vectors.
- ▶ Can also use quaternions.
- ▶ We will find other ways to deal with arbitrary rotations.

How are transforms combined?

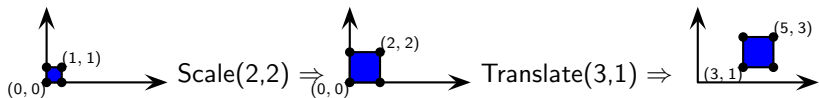


- ▶ Matrix multiplication is associative: $p' = T(Sp) = TS p$

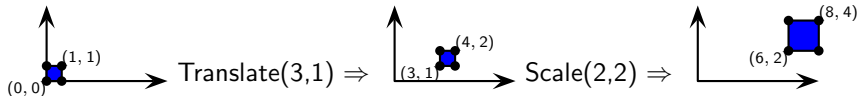
$$TS = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

- ▶ Remember we multiply on the left, so in matrix TS scale is done first, translate second.

Matrix multiplication is not commutative: $TS \neq ST$



$$TS = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$



$$ST = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 6 \\ 0 & 2 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

Easy inverses

- ▶ The inverse of a rotation matrix is its transpose.

Easy inverses

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

Easy inverses

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

Easy inverses

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

Easy inverses

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

Easy inverses

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

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

$$\begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} =$$

Easy inverses

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

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

$$\begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Decomposable Transforms

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

Decomposable Transforms

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

Decomposable Transforms

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

Decomposable Transforms

$$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c & -s & 0 & x \\ s & c & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c & -s & 0 & cx - sy \\ s & c & 0 & sx + cy \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Decomposable Transforms

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

Decomposable Transforms

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

Decomposable Transforms

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

Decomposable Transforms

$$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & x \\ 0 & b & 0 & y \\ 0 & 0 & c & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & ax \\ 0 & b & 0 & by \\ 0 & 0 & c & cz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

More Easy Inverses

$$\begin{bmatrix} c & -s & 0 & x \\ s & c & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} =$$

More Easy Inverses

$$\begin{bmatrix} c & -s & 0 & x \\ s & c & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \left(\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)^{-1}$$
$$=$$

More Easy Inverses

$$\begin{aligned} \begin{bmatrix} c & -s & 0 & x \\ s & c & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} &= \left(\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \\ &= \end{aligned}$$

More Easy Inverses

$$\begin{aligned} \begin{bmatrix} c & -s & 0 & x \\ s & c & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} &= \left(\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & -z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

More Easy Inverses

$$\begin{bmatrix} a & 0 & 0 & x \\ 0 & b & 0 & y \\ 0 & 0 & c & z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} =$$

More Easy Inverses

$$\begin{bmatrix} a & 0 & 0 & x \\ 0 & b & 0 & y \\ 0 & 0 & c & z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \left(\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)^{-1}$$
$$=$$

More Easy Inverses

$$\begin{aligned} \begin{bmatrix} a & 0 & 0 & x \\ 0 & b & 0 & y \\ 0 & 0 & c & z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} &= \left(\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \\ &= \end{aligned}$$

More Easy Inverses

$$\begin{aligned} \begin{bmatrix} a & 0 & 0 & x \\ 0 & b & 0 & y \\ 0 & 0 & c & z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} &= \left(\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 1/a & 0 & 0 & 0 \\ 0 & 1/b & 0 & 0 \\ 0 & 0 & 1/c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & -z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

The Modelview Matrix

- ▶ Most worlds are modelled by
 1. positioning the model in world coordinates
 2. position the camera in world coordinates
- ▶ To put all objects in camera coordinates
 1. multiply each object by model transform
 2. multiply by inverse camera transform
- ▶ The product of these two matrices is called the **modelview** matrix:

$$C^{-1}M$$

- ▶ Each point in an object will be multiplied by this matrix to put it into camera coordinates.

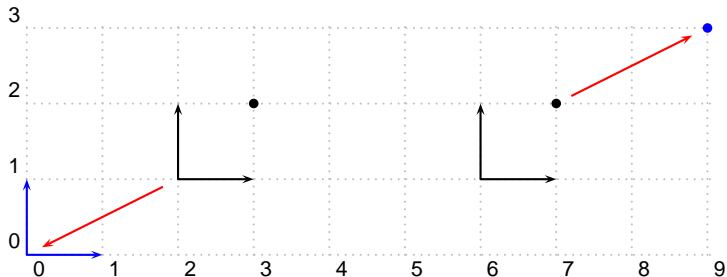
Finding a frame

- ▶ When positioning a model or camera in the world it is generally easy to find a *forward* vector \mathbf{v}_1 and an *up* \mathbf{v}_2 vector in world coordinates.
- ▶ These two vectors need not be orthonormal, just not parallel.
- ▶ A third vector, pointing *right*, can be defined as $\mathbf{v}_3 = \mathbf{v}_1 \times \mathbf{v}_2$.
- ▶ Using the **Gram Schmidt** process you can create an orthonormal frame $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ from $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$

Finding a frame

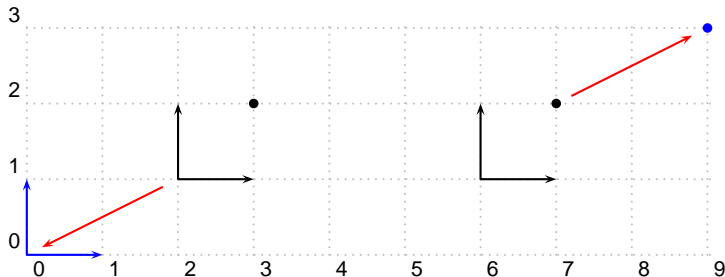
- ▶ When positioning a model or camera in the world it is generally easy to find a *forward* vector \mathbf{v}_1 and an *up* \mathbf{v}_2 vector in world coordinates.
- ▶ These two vectors need not be orthonormal, just not parallel.
- ▶ A third vector, pointing *right*, can be defined as $\mathbf{v}_3 = \mathbf{v}_1 \times \mathbf{v}_2$.
- ▶ Using the **Gram Schmidt** process you can create an orthonormal frame $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ from $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$
- ▶ Slightly faster to Gram-Schmidt-ize the *forward* and *up* vectors, then the cross product is automatically orthonormal.
- ▶ It is usually easy to find a *forward* vector for an object—what is it “looking at”?
- ▶ An *up* vector is also usually easy, can almost always start with $(0, 1, 0)$ and then Gram-Schmidt it.
- ▶ Use *forward* \times *up* to get *right*.

Change of frame



- Does a transform move the object or the frame?

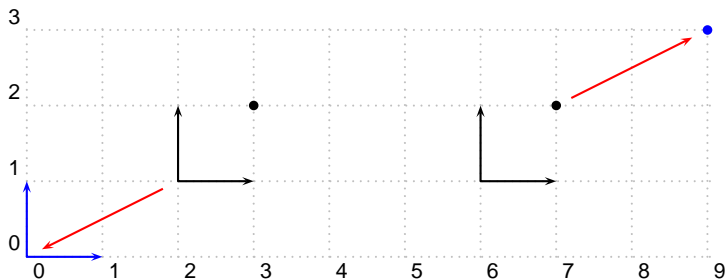
Change of frame



- ▶ Does a transform move the object or the frame?
- ▶ Transform for the above:

$$\begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Change of frame

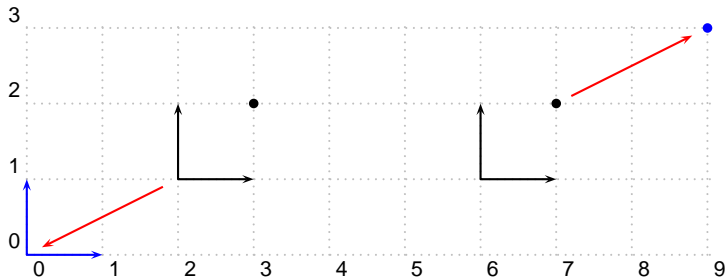


- ▶ Does a transform move the object or the frame?
- ▶ Transform for the above:

$$\begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- ▶ Note that these are *not* inverses! Just a different vocabulary.

Change of frame



- ▶ Does a transform move the object or the frame?
- ▶ Transform for the above:

$$\begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- ▶ Note that these are *not* inverses! Just a different vocabulary.
- ▶ However, if you want to *move a frame*, you need an inverse.

Change of frame

$$\begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} d \\ e \\ f \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} g \\ h \\ i \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} j \\ k \\ l \\ 1 \end{bmatrix}$$

Change of frame

$$\begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ 0 \end{bmatrix}$$

$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{x} = v_1$$

$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{y} = v_2$$

$$\begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} d \\ e \\ f \\ 0 \end{bmatrix}$$

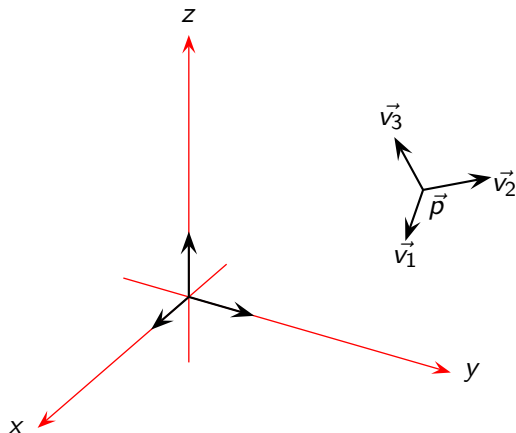
$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{z} = v_3$$

$$\begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} g \\ h \\ i \\ 0 \end{bmatrix}$$

$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{o} = p$$

$$\begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} j \\ k \\ l \\ 1 \end{bmatrix}$$

Change of Frame



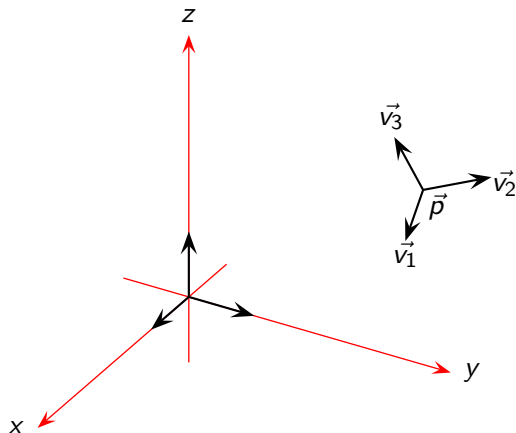
$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{x} = \vec{v}_1$$

$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{y} = \vec{v}_2$$

$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{z} = \vec{v}_3$$

$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{o} = \vec{p}$$

Change of Frame



$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{x} = \vec{v}_1$$

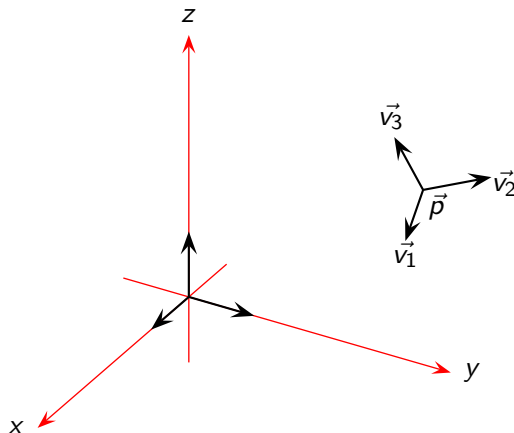
$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{y} = \vec{v}_2$$

$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{z} = \vec{v}_3$$

$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{o} = \vec{p}$$

- Use to put model points into the world.

Change of Frame



$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{x} = \vec{v}_1$$

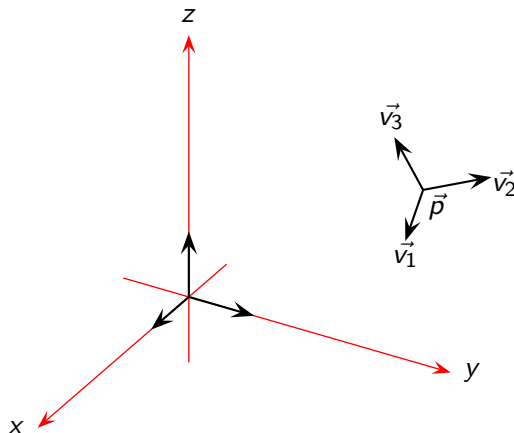
$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{y} = \vec{v}_2$$

$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{z} = \vec{v}_3$$

$$(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{p}) \cdot \vec{o} = \vec{p}$$

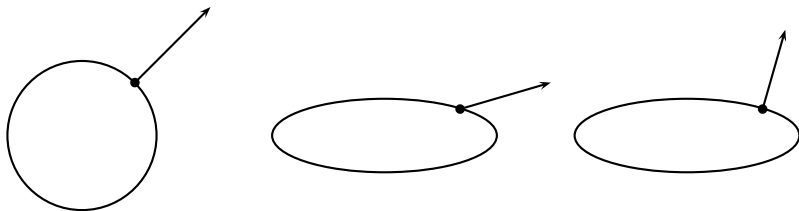
- ▶ Use to put model points into the world.
- ▶ Use inverse to put the world in camera coords.

Camera transforms



- ▶ Cameras and objects generally use only rotation and translation.
- ▶ Can use “easy inverse” for cameras.

Transforming normals



- ▶ Normals do not stay normalized after scale transforms.
- ▶ Must use the inverse transpose

$$(M^{-1})^T$$

- ▶ Might be good to maintain inverses.
- ▶ Rigid transforms OK.

Online Resources

Readings

- ▶ http://en.wikipedia.org/wiki/Transformation_matrix
- ▶ <http://xkcd.com/184/>
- ▶ <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/>,
Computer graphics
- ▶ <http://www.songho.ca/opengl/index.html>

$$\begin{bmatrix} \cos 90^\circ & \sin 90^\circ \\ -\sin 90^\circ & \cos 90^\circ \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -a_1 \end{bmatrix}$$