Applications of Linear Algebra to Computer Graphics

2D & 3D Transformations · Composition · Projections

SIT292 – Enrichment Task 2 (Topic 4.5)
Eunjin Kim
Student ID: 223715707
Trimester: T2 2025

Overview

Linear transformations move points between frames. Matrices handle translation, rotation, scaling, and projection. Homogeneous coordinates let us express every step as a matrix product and compose them cleanly.

> **Key idea.** Use homogeneous matrices so chaining transforms is just multiplication. Order matters.

Learning Goals

- Explain coordinate frames and homogeneous coordinates.
- Compare 2D and 3D transforms and the effect of order.
- Compute numeric compositions on sample shapes (worked examples).
- Contrast orthographic and perspective projection with matrix forms.

Coordinates & Homogeneous Form

Represent 2D points as (x, y) and 3D points as (x, y, z). For translation as a matrix multiply, use $(x, y, 1)$ and $(x, y, z, 1)$.

2D transforms

Rotation by θ, scaling (s_x, s_y), translation (t_x, t_y):

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}.$$

3D transforms

Rotation about z (similar for x, y), translation (t_x, t_y, t_z):

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

2D vs 3D Transformations

2D pipeline

- Use $3times3$ matrices on $(x, y, 1)$.
- Common order: T, S, R (translate after rotate/scale).
- Good for UI, sprites, maps.

3D pipeline

- Use $4times4$ matrices on $(x, y, z, 1)$.
- Include camera transforms and projection.
- Z-depth affects perspective scaling.

> **Order matters.** $TSRneRTS$. Rotating after translating spins around a different center.

Worked Example · 2D (Rotation → Scale → Translation)

Triangle $P(0,0)$, $Q(2,0)$, $R(0,1)$. Apply $\theta = 30°$, $S = \text{diag}(2, 0.5)$, then $t = (3,1)$. Using the column-vector convention, we have $M = T\,S\,R$.

With $\cos 30° = \frac{\sqrt{3}}{2} \approx 0.866$ and $\sin 30° = 0.5$:

$$R = \begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad S = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

$$S\,R = \begin{bmatrix} 1.732 & -1.000 & 0 \\ 0.250 & 0.433 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad M = T(SR) = \begin{bmatrix} 1.732 & -1.000 & 3 \\ 0.250 & 0.433 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Transforming points

$P' = (3,1)$
$Q' \approx (6.464,\ 1.5)$
$R' \approx (2,\ 1.433)$

What changed?

- Rotation tilts; scale stretches x and shrinks y.
- Translation shifts by $(3,1)$.

Worked Example · 3D (Rotate about z then Translate)

Point $A(1,0,2)$. Apply $R_z(45°)$ then $T=(1,2,3)$. With $\cos 45° = \sin 45° = \frac{\sqrt{2}}{2} \approx 0.7071$:

$$R_z = \begin{bmatrix} 0.7071 & -0.7071 & 0 & 0 \\ 0.7071 & 0.7071 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad M = T\,R_z = \begin{bmatrix} 0.7071 & -0.7071 & 0 & 1 \\ 0.7071 & 0.7071 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$A' = M[1,0,2,1]^T = (1.7071,\ 2.7071,\ 5,\ 1)$.

Orthographic vs Perspective Projection

Orthographic (parallel)

Drops z; objects keep size regardless of depth.

$$P_{\text{ortho}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (x,y,z,1)^T \mapsto (x,y,0,1)^T.$$

Perspective (pinhole)

Distant objects appear smaller. With focal length f and homogeneous divide:

$$P_{\text{persp}} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$v' = P_{\text{persp}}[x, y, z, 1]^T = (fx,\ fy,\ z,\ z),$$

$$(x', y') = \left(\frac{fx}{z},\ \frac{fy}{z}\right).$$

Numeric contrast

Points $B(1, 0, 2)$ and $C(1, 0, 4)$ with $f = 2$. Ortho: both $(1, 0)$.
Perspective: $B \to (1, 0)$, $C \to (0.5, 0)$ — depth halves apparent x.

Takeaway

- Orthographic preserves sizes (CAD, UI, grids).
- Perspective mimics vision (realistic 3D scenes).

Common Pitfalls

- Order reversed (e.g., RTS instead of TSR).
- Mixing row- vs column-vector conventions.
- Forgetting the homogeneous divide in perspective.
- Degrees vs radians (in code).

Conclusion

Matrix algebra unifies motion and camera models. Homogeneous coordinates let us translate with matrices and compose steps into one product. Orthographic and perspective projections complete the pipeline by mapping 3D to 2D for different goals.

References (IEEE style)

1. W. K. Nicholson, *Linear Algebra with Applications*, 7th ed. McGraw-Hill, 2015, ch. 4.5.
2. J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*, 2nd ed. Addison-Wesley, 1996.

3. E. Lengyel, *Mathematics for 3D Game Programming and Computer Graphics*, 3rd ed. Course Technology, 2011.
4. T. Akenine-Möller, E. Haines, and N. Hoffman, *Real-Time Rendering*, 4th ed. CRC Press, 2018.