Affine Transformations

Reading

- Foley et al., Chapter 5.6 and Chapter 6
- Supplemental
**Affine Geometry**

- Points: location in 3D space
- Vectors: quantity with a direction and magnitude, but no fixed position
- Scalar: a real number

\[ s = 5.3 \]

\[ P \]

**Affine Spaces**

Affine space consists of points and vectors related by a set of axioms:
- Difference of two points is a vector:
- Head-to-tail rule for vector addition:

**Affine Operations**

Legal affine operations:
- vector + vector → vector
- scalar · vector → vector
- point − point → vector
- point + vector → point

... example of an “illegal” operation:
- point + point → nonsense

Useful combination of affine operations:

\[ P(\alpha) = P_0 + \alpha v \]

What is it?
**Affine Combination**

Affine combination of two points:

\[ Q = \alpha_1 Q_1 + \alpha_2 Q_2 \]

where \( \alpha_1 + \alpha_2 = 1 \) is defined to be the point \( Q = Q_1 + \alpha_1(Q_2 - Q_1) \)

We can generalize affine combination to multiple points:

\[ Q = \alpha_1 Q_1 + \alpha_2 Q_2 + \cdots + \alpha_n Q_n \]

where \( \sum \alpha_i = 1 \)

---

**Affine Frame**

A frame can be defined as a set of vectors and a point:

\( (v_1, \ldots, v_n, O) \)

Where \( v_1, \ldots, v_n \) form a basis and \( O \) is a point in space.

Any point \( P \) can be written as

\[ P = p_1 v_1 + \cdots + p_n v_n + O \]

And any vector as:

\[ u = u_1 v_1 + \cdots + u_n v_n \]

---

**Matrix representation of points and vectors**

Coordinate axiom:

\[ 0 \cdot P = 0 \]

\[ 1 \cdot P = P \]

So every point in the frame \( F = (v_1, \ldots, v_n, O) \) can be written as

\[ P = p_1 v_1 + p_2 v_2 + \cdots + p_n v_n + 1 \cdot O \]

\[ = [v_1 \ v_2 \ \cdots \ v_n \ O] \ldots \]

And every vector as

\[ u = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n + 0 \cdot O \]

\[ = [v_1 \ v_2 \ \cdots \ v_n \ O] \ldots \]

---

**Changing frames**

Given a point \( P \) in frame \( F \), what are the coordinates of \( P \) in frame \( F' = (v'_1, \ldots, v'_n, O') \)

\[ P = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} p'_1 \\ p'_2 \\ \vdots \\ p'_n \end{bmatrix} \]

\[ \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} \begin{bmatrix} v_1' \\ v_2' \\ \cdots \\ v_n' \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix} \]

Since each element of \( F \) can be written in coordinates relative to \( F' \)

\[ v_i = f_{i1} v'_1 + \cdots + f_{in} v'_n \]

\[ O = f_{o1} v'_1 + \cdots + f_{on} v'_n + O' \]
Changing frames cont’d

Written in a matrix form

\[
\begin{bmatrix}
    v'_1 & v'_2 & \cdots & v'_n \end{bmatrix}' = A \begin{bmatrix}
    v_1 & v_2 & \cdots & v_n \end{bmatrix}
\]

Euclidean and Cartesian spaces

A Euclidean space is an affine space with an inner product:

\[ \langle u, v \rangle = u \cdot v = u^T v \]

A Cartesian space is a Euclidean space with a standard orthonormal frame. In 3D: \((i, j, k, O)\)

\[ e_i \cdot e_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \]

Useful properties and operations in Cartesian spaces

Length: \( |v| = \sqrt{v \cdot v} \)

Distance between points: \( |P - Q| \)

Angle between vectors: \( \cos^{-1} \left( \frac{u \cdot v}{|u| \cdot |v|} \right) \)

Perpendicular (orthogonal): \( u \cdot v = 0 \)

Parallel: \( \frac{u \cdot v}{|u| \cdot |v|} = \pm 1 \)

Cross product (in 3D): \( u \times v = w \)

Affine Transformations

\( F : A \rightarrow B \) is an affine transformation if it preserves affine combinations:

\[ F \left( \sum \alpha_i Q_i \right) = \sum \alpha_i F(Q_i) \]

Where \( \sum \alpha_i = 1 \). The same applies to vectors.

Affine coordinates are preserved:

\[ F(O + \sum p_i v_i) = F(O) + \sum p_i F(v_i) \]

Lines map to lines:

\[ F(P_0 + \alpha v) = F(P_0) + \alpha F(v) \]

Parallelism is preserved:

\[ F(Q_0 + \beta v) = F(Q_0) + \beta F(v) \]

Ratios are preserved:

\[ \text{Ratio}(Q_1, Q_2, Q_3) = \text{Ratio}(F(Q_1), F(Q_2), F(Q_3)) \]
2D Affine Transformations

\[ P = [x, y, 1] \]

\( P \) is a column vector

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

\( P \) is a row vector

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

Identity

Doesn't move points at all

\[
\begin{bmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
  0 & 0 & 1
\end{bmatrix}
\]

Translation

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & c \\
  0 & 1 & f \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\]

\[ x' = x + c \]

\[ y' = y + f \]

Scaling

Changing the diagonal elements performs scaling

If \( a = f \) scaling is uniform

\[
\begin{bmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
  0 & 0 & 1
\end{bmatrix}
\]

What if \( a, f < 0 \)

\[
\begin{bmatrix}
  -1 & 0 & 0 \\
  0 & -1 & 0 \\
  0 & 0 & 1
\end{bmatrix}
\]
Shearing

What about the off-diagonal elements?
The matrix
\[
\begin{bmatrix}
1 & 0 & 0 \\
d & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
Gives
\[
x' = x \\
y' = dx + y
\]

Effect on unit square

\[
\begin{bmatrix}
a & b & 0 \\
d & e & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1
\end{bmatrix}
= \begin{bmatrix}
0 & a & a+b & b \\
0 & d & d+e & e \\
1 & 1 & 1 & 1
\end{bmatrix}
\]

- \( M \) can be determined just by knowing how corners \([1,0,1]\) and \([0,1,1]\) are mapped
- \( A \) and \( f \) give \( x \)- and \( y \)-scaling
- \( B \) and \( d \) give \( x \)- and \( y \)-shearing

Rotation

- Rotation of points \([1,0,1]\) and \([0,1,1]\) by angle \( \alpha \) around the origin:
\[
\begin{bmatrix}
1 \\
0 \\
1
\end{bmatrix}
\rightarrow
\begin{bmatrix}
\cos(\alpha) \\
\sin(\alpha) \\
1
\end{bmatrix}
\]
\[
\begin{bmatrix}
0 \\
-\sin(\alpha) \\
\cos(\alpha)
\end{bmatrix}
\rightarrow
\begin{bmatrix}
1 \\
1
\end{bmatrix}
\]

The Matrices

Identity (do nothing):
\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
Scale by \( s_x \) in the \( x \) and \( s_y \) in the \( y \) direction
\((s_x < 0 \text{ or } s_y < 0 \text{ is reflection})\):
\[
\begin{bmatrix}
s_x & 0 & 0 \\
0 & s_y & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Rotate by angle \( \theta \) (in radians):
\[
\begin{bmatrix}
\cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Shear by amount \( a \) in the \( x \) direction:
\[
\begin{bmatrix}
1 & a & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Shear by amount \( b \) in the \( y \) direction:
\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & b \\
0 & 0 & 1
\end{bmatrix}
\]

Translate by the vector \((t_x, t_y)\):
\[
\begin{bmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
0 & 0 & 1
\end{bmatrix}
\]
Transformation Composition

Applying transformations $F$ to point $P$ and transformation $G$ to the result

\[ P' = FP \]
\[ P'' = GP' \]

Combining two transformations

\[ P'' = G(FP) \]
\[ = (GF)P \]

Let’s play a game

- Problems 2, 3, 4, 14, 17, 18

Rotation around arbitrary point

\[ \theta \]
\[ p \]

Reflection around arbitrary axis

\[ \theta \]
Reflection around arbitrary axis

Properties of Transforms

- Compact representation
- Fast implementation
- Easy to invert
- Easy to compose

3D Scaling

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

3D Translation

\[
\begin{bmatrix}
x' \\
y' \\
z'
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & t_x \\
0 & 0 & 0 & t_y \\
0 & 0 & 0 & t_z \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]
Rotation in 3D

• Rotation now has more possibilities in 3D:

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

\[
R_y(\theta) = \begin{bmatrix}
\cos \theta & 0 & \sin \theta \\
0 & 1 & 0 \\
-\sin \theta & 0 & \cos \theta
\end{bmatrix}
\]

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

Use right hand rule

Rotation in 3D

• What about the inverses of 3D rotations?

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

\[
R_y(\theta) = \begin{bmatrix}
\cos \theta & 0 & \sin \theta \\
0 & 1 & 0 \\
-\sin \theta & 0 & \cos \theta
\end{bmatrix}
\]

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

Shearing in 3D

• Shearing is also more complicated. Here is one example:

\[
x' = \begin{bmatrix}
1 & b & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]

Properties of affine transformations

• All of the transformations we've looked at so far are examples of “affine transformations.”

• Here are some useful properties of affine transformations:
  - Lines map to lines
  - Parallel lines remain parallel
  - Midpoints map to midpoints (in fact, ratios are always preserved)

\[
\text{ratio} = \frac{\|pq\|}{\|qr\|} = \frac{s}{t} = \frac{\|p'q'\|}{\|q'r'\|}
\]
Rotation that aligns 3 orthonormal vectors with the principal axes