Lecture 03: 2D Transformations

September 1, 2020
Previous Lecture & Today

• Last Thursday
  – Scalars, Vectors and Points
  – Vector Spaces
  – Affine Spaces
  – Euclidean Spaces
  – Know and love the dot product

• Today
  – Projection (dot product) and rotation,
  – Homogeneous Coordinates for 2D
Dot Product as Projection

The linear algebraic definition of a dot product is clearly taught everywhere: products between first, second, third values etc. However, one of the more meaningful definitions of the dot product is sometimes neglected. Not here. Much of what depends critically upon a reflexive understanding of the dot product as a projection, understand what we mean by ‘projection’, read on.

Ross Beveridge

August 27, 2020
Poll 1: Go the Other Way

What happens if we flip the sign on the x and y terms of vector u?

- d becomes 0
- d becomes -2
- d stays 2
Above you see how almost all texts and courses introduction 2D rotation. This is entirely correct, but there is a more intuitive way to understand rotation.
Consider an alternate basis

\[ |x| = u \begin{vmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{vmatrix} + v \begin{vmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{vmatrix} \]

\[ |u| = \begin{vmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{vmatrix} |y| \]

\[ |v| = \begin{vmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{vmatrix} |x| \]
Poll 2: Lengthy Question

What is the length of vector \( w \)?

one over square root of 2
one
square root of 2

\[
w = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}
\]
Welcome to 2D Rotation

These are the same!

\[ |u| = \begin{vmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{vmatrix} |x| \]

\[ |u| = \begin{vmatrix} \cos(45^\circ) & \sin(45^\circ) \\ -\sin(45^\circ) & \cos(45^\circ) \end{vmatrix} |x| \]
Rotate by $\theta$

$$M = RP$$

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

$$P = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} \cos(\theta) \cdot x - \sin(\theta) \cdot y \\ \sin(\theta) \cdot x + \cos(\theta) \cdot y \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$
But Notice the two Dot Products!

\[ M = RP \]

\[ R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad P = \begin{bmatrix} x \\ y \end{bmatrix} \]

\[ \begin{bmatrix} \cos(\theta) x - \sin(\theta) y \\ \sin(\theta) x + \cos(\theta) y \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \]

\[ \cos(\theta) x - \sin(\theta) y = \begin{bmatrix} \cos(\theta) \\ -\sin(\theta) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \]

Row by row, matrix multiplication is dot product!
Recap: Rotation is Projection

In [4]:
    el = matrix([[1,1],[3,1],[3,2],[2,2],[2,4],[1,4]])
    el.transpose()
    gel = polygon(list(el),color='green')

In [5]:
    bnd = 5.0
    gu = arrow((0,0),u)
    gv = arrow((0,0),v)
    gud = line([(0,0),bnd*u], linestyle='-.')
    gvd = line([(0,0),bnd*v], linestyle='-.')
    gos = gu + gv + gud + gvd
    gos.show(xmin=-bnd, ymin=-bnd, xmax=bnd, ymax=bnd, aspect_ratio=1)
Uniform Scaling

The first of several 2D canonical matrices...

\[ S = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}, \quad P = \begin{bmatrix} x \\ y \end{bmatrix}, \quad M = S \cdot P \]

\[ \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s \cdot x \\ s \cdot y \end{bmatrix} \]

\[ I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad M = s \cdot I \cdot P, \quad M = s \begin{bmatrix} x \\ y \end{bmatrix} \]
Non-uniform Scaling

\[ S = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \quad P = \begin{bmatrix} x \\ y \end{bmatrix} \]

\[ M = S \cdot P \]

\[ \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_1 x \\ s_2 y \end{bmatrix} \]

*Note orientation shift in line*
Poll 3: Angling for a Change

Can the application of non-uniform scaling change to orientation between 2 different vectors?

Yes
No
Flip an Axis...

\[
\begin{bmatrix}
  x \\
  -y
\end{bmatrix}
= \begin{bmatrix}
  1 & 0 \\
  0 & -1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
\]

What does this do to appearance of objects?
Flip the Horizontal

March, 23, 2012

Enantiomorph

Look in the mirror. What do you see? A reflection? Nonsense! A reader of Uncommon Parlance observes an enantiomorph: the fancy-pants term for a mirror image. Enantiomorphism also crops up in the field of chemistry where it refers to crystals that are structurally mirror images of each other. Etymology: from Ancient Greek ἐναντίος or enantios (opposite) + μορφή or morphē (form).

“The cardinal looked himself in the eye and curled his lip into a sneer. In the mirror his enantiomorph exhibited the same self-disgust and followed suit.”

Credits: Uncommon Parlance

\[
\begin{bmatrix}
    x_2 \\
    y_2
\end{bmatrix} =
\begin{bmatrix}
    -1 & 0 \\
    0 & 1
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    y_1
\end{bmatrix}
\]
Swap Axes

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

Why would you do this?
Translation

\[ P(x, y) = \begin{bmatrix} x \\ y \end{bmatrix} \]

\[ P(u, v) = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \]

I am intentionally drawing the alternative geometry, i.e. move the origin not the point.

Plug in some values and draw yourself some pictures.

Addition, not multiplication!
Canonical Transformations

\[ \text{Rotate} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \]

\[ \text{Scale} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \]

\[ \text{Flip} = \begin{bmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{bmatrix} \]

\[ \text{Translate} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \]

Add, don’t multiply
Composition ...

To apply transformation A to point $p$, and then transform the result by transformation B:

$$p' = (B \ A) \ p = B \ (A \ p)$$

Question: why is this important?
Problem: Translation

Unfortunately, we often need to translate points, and translation is matrix addition, not multiplication.

\[
\begin{bmatrix}
  x_2 \\
  y_2
\end{bmatrix} = \begin{bmatrix}
  x_1 \\
  y_1
\end{bmatrix} + \begin{bmatrix}
  t_x \\
  t_y
\end{bmatrix}
\]

We need some way to make translation into a matrix multiplication operation, so that all transformations can be composed...
In homogeneous coordinates, a two-dimensional point is represented as a vector of length 3.

In homogeneous coordinates, a three-dimensional point is represented as a vector of length 4.

In general, homogeneous coordinates represent an $N$-dimensional point with a vector of length $N+1$. 
Homogeneous Coordinates (cont.)

In particular, the 2D point \((x,y)\) is:

\[
\begin{vmatrix}
  x & 2x & nx \\
  y & 2y & ny \\
  1 & 2 & n \\
\end{vmatrix}
\]

For any \(n \neq 0\)

*Question: what is the last coordinate (conceptually)?*
Poll 4: Homogeneous Where?

Where would you draw the point P on a 2D grid?
- At position (8, 4)
- At position (4, 2)
- At Position (2, 1)
Homogeneous Coordinates (cont...)  

• Note that homogeneous coordinates are non-unique, but  

• Translation in homogeneous coordinates is multiplication:  

\[
\begin{pmatrix}
  x + t_x \\
  y + t_y \\
  1
\end{pmatrix}
=  
\begin{pmatrix}
  1 & 0 & t_x \\
  0 & 1 & t_y \\
  0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
  x \\
  y \\
  1
\end{pmatrix}
\]
• **2D Rotation looks pretty much the same:**

\[
\begin{bmatrix}
  x_2 \\
  y_2 \\
  w_2
\end{bmatrix} = \begin{bmatrix}
  \cos(\theta) & -\sin(\theta) & 0 \\
  \sin(\theta) & \cos(\theta) & 0 \\
  0 & 0 & 1
\end{bmatrix}\begin{bmatrix}
  x_1 \\
  y_1 \\
  w_1
\end{bmatrix}
\]

• **As does 2D non-uniform scaling:**

\[
\begin{bmatrix}
  x_2 \\
  y_2 \\
  w_2
\end{bmatrix} = \begin{bmatrix}
  s_x & 0 & 0 \\
  0 & s_y & 0 \\
  0 & 0 & 1
\end{bmatrix}\begin{bmatrix}
  x_1 \\
  y_1 \\
  w_1
\end{bmatrix}
\]
Rotate about a Point

- Rotate the Cat’s Head about its Nose
  1. Translate the Nose to the Origin
  2. Rotate by the desired amount
  3. Invert the translation

\[(t_x, t_y)\]
Rotation about a Point (II)

- Translate to origin
  Note the negations: we want to bring \((t_x, t_y)\) to the origin, so subtract \(t_x, t_y\).
  \[
  M_T = \begin{bmatrix}
  1 & 0 & -t_x \\
  0 & 1 & -t_y \\
  0 & 0 & 1
  \end{bmatrix}
  \]

- Rotate about origin
  What was the point \((t_x, t_y)\) is now at the origin.
  \[
  M_R = \begin{bmatrix}
  \cos(\theta) & -\sin(\theta) & 0 \\
  \sin(\theta) & \cos(\theta) & 0 \\
  0 & 0 & 1
  \end{bmatrix}
  \]

- Translate back
  Finally, what started as \((t_x, t_y)\) is again \((t_x, t_y)\).
  \[
  M_{T^{-1}} = \begin{bmatrix}
  1 & 0 & t_x \\
  0 & 1 & t_y \\
  0 & 0 & 1
  \end{bmatrix}
  \]
Rotation about a Point (III)

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

- Think about order!
- Operations right before those on left.
- Therefore, read from right to left.
Rotation about a Point (IV)

• Reminder, in matrix multiplication:

\[ AB \neq BA \]

• The equation to rotate a matrix of points \( P \) around \((t_x, t_y)\) is:

\[ P' = T^{-1}RTP \]
Rotation about a Point (V)

- Compose the three transformations.

\[ P' = \left( T^{-1}RT \right) P \]

\[
\begin{bmatrix}
\cos(\theta) & -\sin(\theta) & \sin(\theta)t_y + (1-\cos(\theta))t_x \\
\sin(\theta) & \cos(\theta) & -\sin(\theta)t_x + (1-\cos(\theta))t_y \\
0 & 0 & 1
\end{bmatrix}
\]
Scaling About Point P

- Scaling also operates relative to the origin.
- To make an object bigger without moving it
  - Translate origin to object centroid.
  - Apply scaling.
  - Invert the translation.

\[
M_4 = \begin{bmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
s_x & 0 & 0 \\
0 & s_y & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
1 & 0 & -t_x \\
0 & 1 & -t_y \\
0 & 0 & 1 \\
\end{bmatrix} =
\begin{bmatrix}
s_x & 0 & -s_x t_x + t_x \\
0 & s_y & -s_y t_y + t_y \\
0 & 0 & 1 \\
\end{bmatrix}
\]
One More Transform - Shear

Shearing in the $X$ dimension

- Metaphor - Wind Blows Figure.

- Basic Matrix Form.

- Can $X$-Shear do This?

$$\begin{pmatrix}
1 & sh_x & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}$$
Shear (cont.)

Shearing in the $Y$ dimension

• Metaphor - Same thing, other direction.

• Basic Matrix Form.

\[
\begin{pmatrix}
1 & 0 & 0 \\
sh_y & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

• Note: Parallel Lines Stay Parallel
Use Notebook 3 to develop an intuition as well as mechanical understanding!
Poll 5: What Happened

The transformation below is?
Rotation only
Translation only
Scale only
Rotation and Translation
Rotation and Translation and Scaling
The End