L03

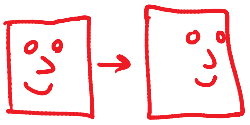
Spatial Transformations

Goal: To learn how to represent affine transformations in an efficient way.

Images (and volumes) can be thought of as fields (or functions) of space. We will talk a lot about transformations of these spatial domains. Some examples are: translation, rotation, scaling, etc.

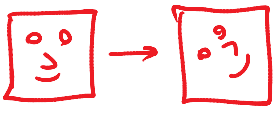
Translation

$$\begin{cases} u = x + a \\ v = y + b \end{cases}$$



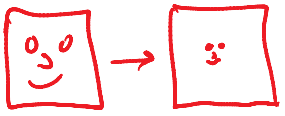
Rotation

$$\begin{cases} u = x\cos\theta + y\sin\theta \\ v = -x\sin\theta + y\cos\theta \end{cases}$$



Scaling

$$\begin{cases} u = s_1 x \\ v = s_2 y \end{cases}$$



Homogeneous Coordinates

Notice above that rotation & scaling can each be written as matrix-vector products.

$$\vec{u} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \vec{x}$$

Rotation

$$\vec{u} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \vec{x}$$

Scaling

So called "affine" transforms.

However, translation cannot. But we can embed translations into a matrix-vector product if we bump up the dimension by 1.

i.e. $\begin{bmatrix} \quad \\ \quad \end{bmatrix} \begin{bmatrix} x \\ \quad \end{bmatrix} = \begin{bmatrix} x+a \\ \quad \end{bmatrix}$

we bump up the dimension by 1.

ie. $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \Longleftrightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Longleftrightarrow \begin{bmatrix} rx \\ ry \\ rz \\ r \end{bmatrix}$ for $r \neq 0$

We call this augmented coordinate system "Homogeneous coordinates".

Consider the transform

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$ in non-homogeneous coords

$$\Rightarrow \begin{bmatrix} u \\ v \\ r \end{bmatrix} = \begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by + e \\ cx + dy + f \\ 1 \end{bmatrix}$$

Example:

Rotate by 30°, then translate by $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$

Rotation matrix is $R_{30^\circ} = \begin{bmatrix} \sqrt{3}/2 & -\frac{1}{2} \\ \frac{1}{2} & \sqrt{3}/2 \end{bmatrix}$

Thus, $P = \begin{bmatrix} \sqrt{3}/2 & -\frac{1}{2} & 2 \\ \frac{1}{2} & \sqrt{3}/2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$

Composition of Transforms

Why do we care about representing translation as a matrix-vector product?

Now we can do this

a matrix vector product.

But we can do this...

→ Two (or more) affine transforms can be combined into one affine transform by simple matrix multiplication.

Consider the composite transform:

1) $P_1 \equiv$ rotate by 30°

2) $P_2 \equiv$ translate by $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$

$$P_1 = \begin{bmatrix} \cos 30 & \sin 30 & 0 \\ -\sin 30 & \cos 30 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad P_2 = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Composite transform is

$$P_2 P_1 = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 30 & \sin 30 & 0 \\ -\sin 30 & \cos 30 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos 30 & \sin 30 & 2 \\ -\sin 30 & \cos 30 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Now suppose we add a 3rd transform to the sequence:

3) $P_3 \equiv$ rotate by 45°, then translate by $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$

$$P_3 = \begin{bmatrix} \cos 45 & \sin 45 & -1 \\ -\sin 45 & \cos 45 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

The transform resulting from the sequential composition of all 3 can be written

The transform resulting from the sequential composition of all 3 can be written

$$P = P_3 P_2 P_1$$

[Pause... Matlab exercise L03_transforms.m]

→ your end result should be

P_all_three =

0.2588	0.9659	1.1213
-0.9659	0.2588	0.2929
0	0	1.0000

Decomposing Affine Transforms

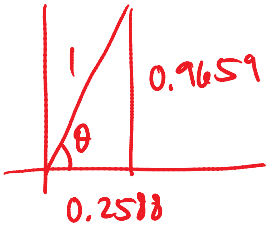
We can decompose a transformation matrix into a single rotation & translation (& scale, skew, etc.)

In the above case

$$\begin{bmatrix} \cos\theta & \sin\theta & t_1 \\ -\sin\theta & \cos\theta & t_2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.2588 & 0.9659 & 1.1213 \\ -0.9659 & 0.2588 & 0.2929 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$\Rightarrow \cos\theta = 0.2588$

$\sin\theta = 0.9659$



$\Rightarrow \theta = 75°$

$t_1 = 1.1213$

$t_2 = -1.7071$

Thus, $P_3 P_2 P_1$ is equivalent to

- Rotation by 75°, followed by
- Translation of $\begin{bmatrix} 1.1213 \\ -1.7071 \end{bmatrix}$

It is often helpful to think of these transforms

It is often helpful to think of these transforms in terms of their affine part & translation part.

$$P = \left[\begin{array}{c|c} A & T \\ \hline 0 & 1 \end{array}\right] \qquad A \in \mathbb{R}^{d \times d} \qquad T \in \mathbb{R}^{d \times 1}$$

$d = 2$ for 2D
$d = 3$ for 3D
etc.

Combining Simple Transforms

Translations:

$$\left[\begin{array}{c|c} I & T_1 \\ \hline 0 & 1 \end{array}\right]\left[\begin{array}{c|c} I & T_2 \\ \hline 0 & 1 \end{array}\right] = \left[\begin{array}{c|c} I & T_1+T_2 \\ \hline 0 & 1 \end{array}\right]$$

Thus, combining translations simply adds the translations.
Inverse of transl. T_1 is $-T_1$

Rotations:

$$\left[\begin{array}{c|c} R_{\theta_1} & 0 \\ \hline 0 & 1 \end{array}\right]\left[\begin{array}{c|c} R_{\theta_2} & 0 \\ \hline 0 & 1 \end{array}\right] = \left[\begin{array}{c|c} R_{\theta_1}R_{\theta_2} & 0 \\ \hline 0 & 1 \end{array}\right] = \left[\begin{array}{c|c} R_{\theta_1+\theta_2} & 0 \\ \hline 0 & 1 \end{array}\right]$$

Thus, combining rotations simply adds their angles.
Inverse of rotation by θ is rotation by $-\theta$.

Scaling:

$$\left[\begin{array}{c|c} S_1 & 0 \\ \hline 0 & 1 \end{array}\right]\left[\begin{array}{c|c} S_2 & 0 \\ \hline 0 & 1 \end{array}\right] = \left[\begin{array}{c|c} S_1 S_2 & 0 \\ \hline 0 & 1 \end{array}\right]$$

Thus, combining scales simply multiplies them.
Inverse of scaling by S is scaling by $\frac{1}{S}$

... multiplies them.

Inverse of scaling by S is scaling by $\frac{1}{S}$.

CAUTION: Be aware of the complexities that arise when you are combining different kinds of simple transforms.

Example: transl. by $\begin{bmatrix}2\\1\end{bmatrix}$, rotate by 30°, transl. by $\begin{bmatrix}-2\\-1\end{bmatrix}$.

$$P = \left[\begin{array}{c|c} I & -T \\ \hline 0 & 1 \end{array}\right]\left[\begin{array}{c|c} R & 0 \\ \hline 0 & 1 \end{array}\right]\left[\begin{array}{c|c} I & T \\ \hline 0 & 1 \end{array}\right] = \left[\begin{array}{c|c} R & RT-T \\ \hline 0 & 1 \end{array}\right]$$

Notice that the translations do not undo each other because there is a rotation between them.

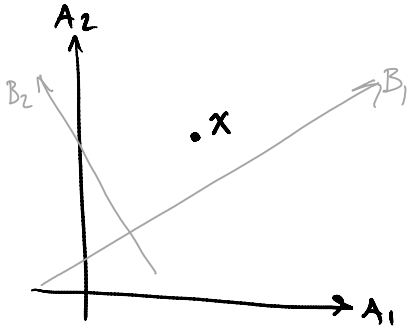
Converting Between Coordinate Systems

Suppose we have a point x and two different coordinate systems (CS) labeled A and B.

$x_A \equiv$ coords of x w.r.t. CS A

$x_B \equiv$ coords of x w.r.t. CS B

eg.



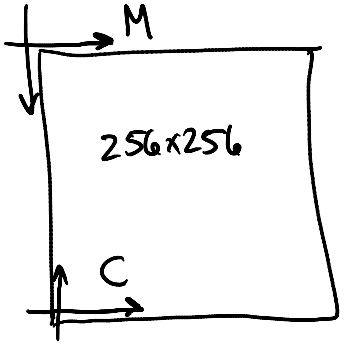
We want to be able to convert x_A to x_B (and vice versa).

$$P_{A\to B}\, x_A = x_B$$

Magic Rule: To get the transform $P_{A \to B}$, we can simply use the same transform that moves the B axes onto the A axes, using A's coord. system.

That is, suppose the B axes is an object in A's coordinate space. Compose the transform that will overlay the B axes on the A axes.

Example: Cartesian to Matlab index coords.



To convert from x_c (Cartesian) to x_m (Matlab index), we move M axes onto C.

1) shift M by $\begin{bmatrix} 1 \\ -256 \end{bmatrix}$ $\longrightarrow$ $P_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -256 \\ 0 & 0 & 1 \end{bmatrix}$

2) reverse M's vertical axis $\longrightarrow$ $P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

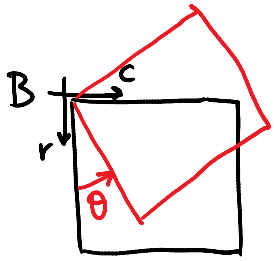
3) swap M's axes $\longrightarrow$ $P_3 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$P_{C \to M} = P_3 P_2 P_1 = \begin{bmatrix} 0 & -1 & 256 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

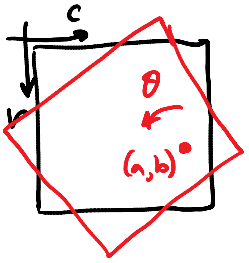
Check: $\begin{bmatrix} 0 \\ 0 \end{bmatrix}_C \longrightarrow \begin{bmatrix} 256 \\ 1 \end{bmatrix}_M$ $\quad$ $\begin{bmatrix} 255 \\ 255 \end{bmatrix}_C \longrightarrow \begin{bmatrix} 1 \\ 256 \end{bmatrix}$

Converting Transforms Between Coordinate Systems

Suppose you want to rotate an image around a specified point with coords (a,b). You already know how to build a rotation matrix, but it rotates about the origin.



Action of R_θ



Rotation of θ about pt. (a,b)

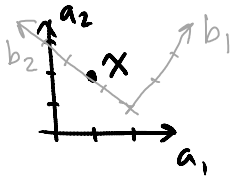
The Theory

Given two coord systems A and B, and a transform matrix M_A, expressed w.r.t. CS A, find the equivalent transform matrix w.r.t. CS B, M_B.

Let P_{AB} be the change-of-basis matrix that converts from A coords to B coords.

i.e. $x_B = P_{AB}\, x_A$

eg.



$x_A = [1 \;\; 2]^T$

$x_B = [0 \;\; \sqrt{2}]^T$

Note that $P_{AB}^{-1} = P_{BA}$

Thus, $x_B = P_{AB}\, x_A$ and $x_A = P_{BA}\, x_B$.

The transform M_A is applied to vectors in CS A.

i.e. $y_A = M_A x_A$

$$P_{BA}\, y_B = M_A\, P_{BA}\, x_B \Rightarrow y_B = P_{AB}\, M_A\, P_{BA}\, x_B$$

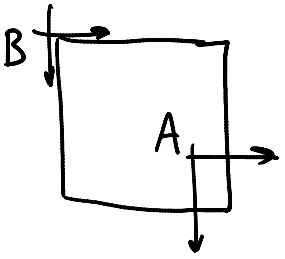
$$y_B = M_B\, x_B$$

Hence, $\boxed{M_B = P_{AB}\, M_A\, P_{BA} \left(= P_{BA}^{-1} M_A P_{BA}\right)}$

Back to our rotation example:

$M_A = R_\theta$ (rotation about A's origin)

$P_{AB} = T(a,b)$ (translation by (a,b))



Hence, $M_B = T(a,b)\, R_\theta\, T(-a,-b)$