Tutorial 4

Supplementary Transformation Notes

Two Main Uses of Sequential Transformations

Transform a point or set of points (an object) from one place to another.

Given a point in one reference frame, find its location in another reference frame.

Transforming an Object

As seen in Tutorial 2, to move an object from one location to another, you directly apply transformations to it to get it to look how you want.

A general transformation, M, can be applied once, or a sequence of transformations, M1, M2, M3, M4, … , can be applied to it.

Transforming an Object - One Transformation

If a general transformation, M, is applied to an object, then every cartesian point, (x,y), on the original object will become (x’,y’) by the following transformation:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \equiv M * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Transforming an Object - Many Transformations

If a sequence of transformations: first M1, then M2, then M3, etc… are applied to an object, the transformations will be done by:

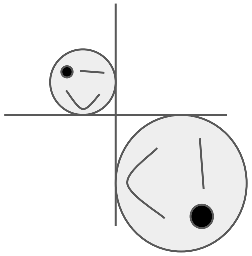
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \equiv M_1 * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix} \equiv M_2 * \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x''' \\ y''' \\ 1 \end{bmatrix} \equiv M_3 * \begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x''' \\ y''' \\ 1 \end{bmatrix} \equiv M_3 * M_2 * M_1 * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Notice how applying transformations to the object means applying them from right to left.

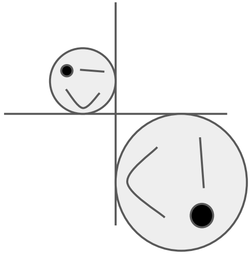
Transforming an Object - Many Transformations

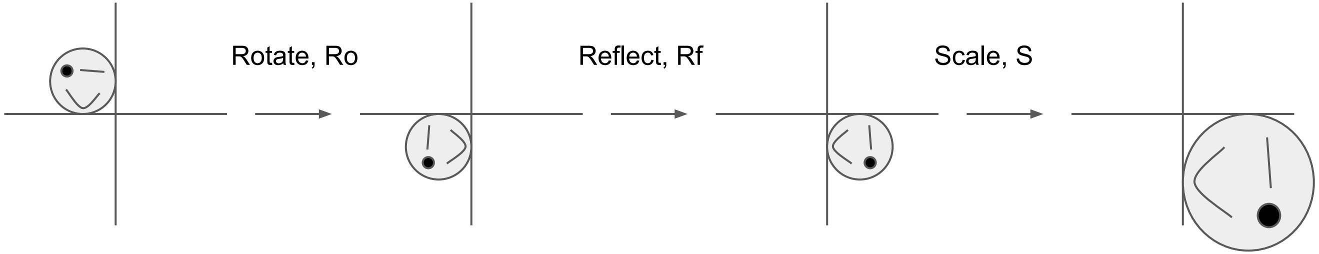
An example of this was done in Tutorial 2. Transform the left face to the right. The left one is centered at (-0.5,0.5) with radius 0.5 and the right is at (1,-1), radius 1:



Transforming an Object - Many Transformations

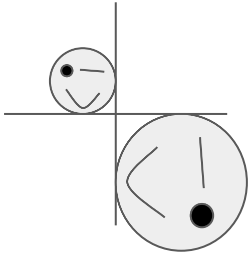
An example of this was done in Tutorial 2. Transform the left face to the right. The left one is centered at (-0.5,0.5) with radius 0.5 and the right is at (1,-1), radius 1:

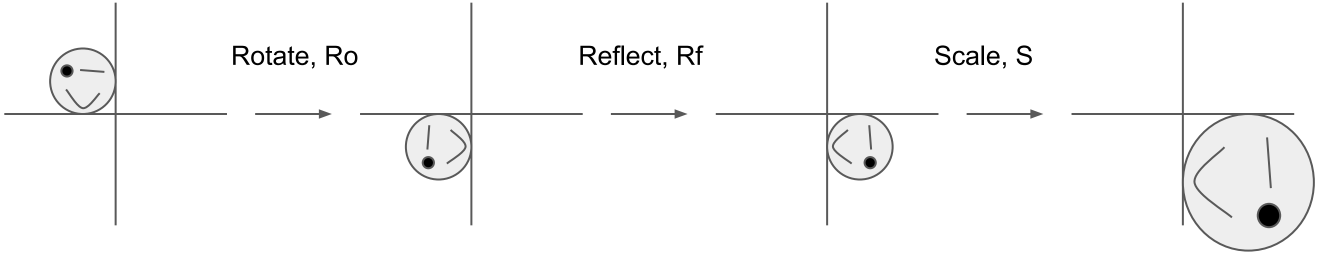




Transforming an Object - Many Transformations

An example of this was done in Tutorial 2. Transform the left face to the right. The left one is centered at (-0.5,0.5) with radius 0.5 and the right is at (1,-1), radius 1:





First Ro is applied, then Rf, then S. This means the overall transformation M is:

M = S * Rf * Ro

Point Conversion

The problem setup here is that you are given a point in reference frame B, pB, and want to find what that point is in reference frame A, pA.

Since you have where the point is in reference frame B, if you knew how to get from reference frame A to reference frame B, then you can get where the point is in reference frame A.

i.e. If the point is 1 “over” in frame B, and frame B is 10 “over” from frame A, then the point is 11 “over” in frame A.

Point Conversion - Known Reference Vectors

If you know the reference vectors of frame B and its origin, all in terms of frame A, then you have the the transformation you need.

In 2D, if you have the vectors u and v (2x1 vectors), which represent the reference vectors of frame B, and the origin of frame B, t (2x1 point), then the transformation that takes a point in frame B to frame A is:

$$M_{BA} = \begin{bmatrix} \vec{u} & \vec{v} & \vec{t} \\ 0 & 0 & 1 \end{bmatrix}$$

Note the matrix is 3x3 in 2D. Getting the point in frame A from frame B is done by:

$$p_A = M_{BA} * p_B$$

Point Conversion - Unknown Reference Vectors

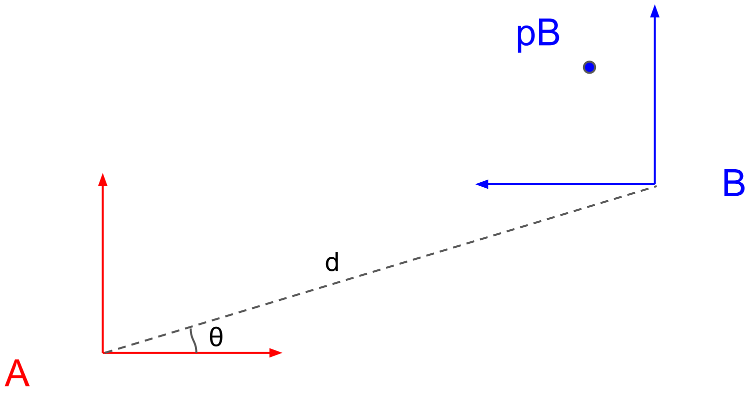
What if you do not know where frame B is in terms of frame A? All you do is find the sequence of transformations that will get frame A to frame B.

It is important to note that to do this, we must transform the reference frame, not a point in the reference frame. This may seem a little backwards and in fact it is. Note that when transforming the reference frame, the origin is always where the reference frame currently is (the definition of the origin), so all rotations and scalings happen at the point the reference frame is located.

This is best shown by example.

Point Conversion - Unknown Reference Vectors

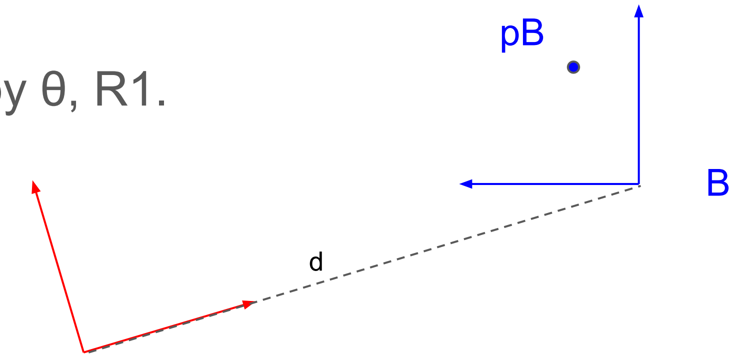
Let us try to build up the transformation that aligns reference frame A with frame B. This transformation will be the one that converts pB (point in frame B) to pA (point in frame A). (Each basic transformation is done on a new slide so it looks clearer.)



Point Conversion - Unknown Reference Vectors

Let us try to build up the transformation that aligns reference frame A with frame B. This transformation will be the one that converts pB (point in frame B) to pA (point in frame A). (Each basic transformation is done on a new slide so it looks clearer.)

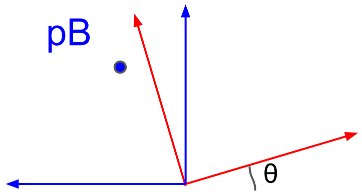
1. Rotate ccw by θ, R1.



Point Conversion - Unknown Reference Vectors

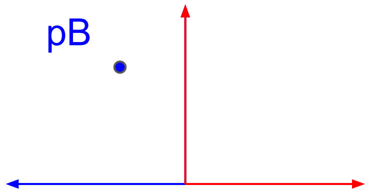
Let us try to build up the transformation that aligns reference frame A with frame B. This transformation will be the one that converts pB (point in frame B) to pA (point in frame A). (Each basic transformation is done on a new slide so it looks clearer.)

1. Rotate ccw by θ, R1.
2. Translate right (in x) by d, T.



Point Conversion - Unknown Reference Vectors

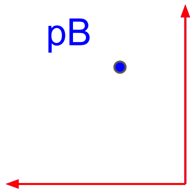
Let us try to build up the transformation that aligns reference frame A with frame B. This transformation will be the one that converts pB (point in frame B) to pA (point in frame A). (Each basic transformation is done on a new slide so it looks clearer.)



1. Rotate ccw by θ, R1.
2. Translate right (in x) by d, T.
3. Rotate cw by θ, R2.

Point Conversion - Unknown Reference Vectors

Let us try to build up the transformation that aligns reference frame A with frame B. This transformation will be the one that converts pB (point in frame B) to pA (point in frame A). (Each basic transformation is done on a new slide so it looks clearer.)



1. Rotate ccw by θ, R1.
2. Translate right (in x) by d, T.
3. Rotate cw by θ, R2.
4. Rotate ccw by 90°, R3.

Point Conversion - Unknown Reference Vectors

1. Rotate ccw by θ, R1.
2. Translate right (in x) by d, T.
3. Rotate cw by θ, R2.
4. Rotate ccw by 90°, R3.

Thus, the matrix that transforms frame A to frame B, and equivalently the matrix that transforms pB (point in frame B) to pA (point in frame A), is:

$$M_{BA} = R_1 * T * R_2 * R_3$$

(Yes, the ordering is from left to right.)

Order of Sequential Transformations

Say you have a sequence of transformations:

$$A * B * C$$

This can be viewed from right to left as:
“First apply C to a point/object, then B to the point/object, then A.”

Or it can be viewed from left to right as:
“First transform the reference frame by A, then transform the reference frame by B, then transform the reference frame by C, then look at or draw the point/object in this new adjusted frame.”

Order of Sequential Transformations

Personally, when drawing or moving objects, I like to think of the transformations being applied to the object from right to left. When getting the location of a point in one frame in another, I like thinking of the transformations being applied to the reference frame from left to right.

It is valid in both cases to think of the transformations in either order. Use whichever way you can understand more intuitively so the assignments and tests can be easier for you.