Geometrical Transformation

October 2, 2008

1 2D Transformations

- Translation: A vector $P(x, y)$ - a point in (x,y)-plane with coordinates (x, y) and $P'(x', y')$ with

  $$x' = x + d_x,\ \ y' = y + d_y$$

  We can represent

  $$P' = \begin{bmatrix} x' \\ y' \end{bmatrix},\ P = \begin{bmatrix} x \\ y \end{bmatrix},\ T = \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

  Then, the translation (from P to P') can be represented by

  $$P' = P + T$$

- Scaling: A vector $P(x, y)$ - a point in (x,y)-plane with coordinates (x, y) and $P'(x', y')$ with

  $$x' = s_x * x,\ \ y' = s_y * y$$

  We can represent

  $$P' = \begin{bmatrix} x' \\ y' \end{bmatrix},\ P = \begin{bmatrix} x \\ y \end{bmatrix},\ S = \begin{bmatrix} s_x \\ s_y \end{bmatrix}$$

  Then, the scaling (of P to P') can be represented by

  $$P' = S.P$$

- Rotating: A vector $P(x, y)$ - a point in (x,y)-plane with coordinates (x, y) and $P'(x', y')$ with

  $$x' = x * cos(\theta) - y * sin(\theta),\ \ y' = x * sin(\theta) + y * cos(\theta)$$

  We can represent

  $$P' = \begin{bmatrix} x' \\ y' \end{bmatrix},\ P = \begin{bmatrix} x \\ y \end{bmatrix},\ R = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

  Then, the rotating (of P to P') can be represented by

  $$P' = R.P$$

2 2D Homogeneous Coordinated System

In homogeneous coodinated system, a point is represented by three componens (a matrix of 3×1)

$$P(x, y, W) = \begin{bmatrix} x \\ y \\ W \end{bmatrix}$$

where $(0, 0, 0)$ is not allowed. Two cooridnates (x, y, W) and (x', y', W') represent the *same point* if $x' = kx$, $y' = ky$, and $W' = kW$ for some k. (x, y, W) with $W \neq 0$ can be *homogenized* to $(x/W, y/W, 1)$.

3 Transformation in 2D Homogeneous Coordinated System

- Translation:

$$T(d_x, d_y) = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix}$$

- Scaling:

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Rotating:

$$R(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformation matrix is of the form

$$\begin{bmatrix} r_{11} & r_{12} & d_x \\ r_{21} & r_{22} & d_y \\ 0 & 0 & 1 \end{bmatrix}$$

Special case: Shear transformation (on x-axis or y-axis)

$$SH_x = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad or \qquad SH_y = \begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

4 Composition of 2D Transformations

The question is: what is the final transformation resulting from a sequence of transformations? As an example, to turn a point $Q(x_q, y_q)$ by θ degree around a given point $P(x_p, y_p)$, one needs to do three steps:

- Translate P to the origin, applying this translation to Q (so, $T(-x_p, -y_p)$)
- Rotate Q' by θ degree (the result of Q after $T(-x_p, -y_p)$) (so, $R(\theta)$)
- Translate Q'' (the result of Q' after $R(\theta)$) back to its coordinates, relative to P (so, $T(x_p, y_p)$)

The above series of transformations results in the transformation defined by

$$T(x_p, y_p) \,.\, R(\theta) \,.\, T(-x_p, -y_p)$$

5 Window to Viewport Transformation

This is the problem of converting real-world coordinates (graphics specified in real-world coordinates) to viewport (a window on the display monitor) coordinates. The real-world coordinates are assumed to be specified by the rectangle given by (x_{min}, y_{min}) and (x_{max}, y_{max}) and the viewport coordinates are given by (u_{min}, v_{min}) and (u_{max}, v_{max}). The transformation is simply the composition of the three transformations:

- Translate the window (of the world) to the origin (so, $T(-x_{min}, -y_{xmin})$)

- Scaling so the window and the viewport has the same size (so, $S(\frac{u_{max}-u_{min}}{x_{max}-x_{min}}, \frac{v_{max}-v_{min}}{y_{max}-y_{min}}))$

- Translate the window to the viewport (so, $T(u_{min}, v_{min}))$

This results in the following matrix M_{wv}:

$$M_{wv} = \begin{bmatrix} 1 & 0 & u_{min} \\ 0 & 1 & v_{min} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{u_{max}-u_{min}}{x_{max}-x_{min}} & 0 & 0 \\ b & \frac{v_{max}-v_{min}}{y_{max}-y_{min}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_{min} \\ 0 & 1 & -y_{min} \\ 0 & 0 & 1 \end{bmatrix}$$

which is

$$\begin{bmatrix} \frac{u_{max}-u_{min}}{x_{max}-x_{min}} & 0 & -x_{min} * \frac{u_{max}-u_{min}}{x_{max}-x_{min}} + u_{min} \\ b & \frac{v_{max}-v_{min}}{y_{max}-y_{min}} & -y_{min} * \frac{v_{max}-v_{min}}{y_{max}-y_{min}} + v_{min} \\ 0 & 0 & 1 \end{bmatrix}.$$

6 3D Homogeneous Coordinated System

In homogeneous coodinated system, a 3D point is represented by four componens (a matrix of 4×1)

$$P(x, y, z, W) = \begin{bmatrix} x \\ y \\ z \\ W \end{bmatrix}$$

where $(0, 0, 0, 0)$ is not allowed. Two cooridnates (x, y, z, W) and (x', y', z', W') represent the *same point* if $x' = kx$, $y' = ky$, $z' = kz$, and $W' = kW$ for some k. (x, y, z, W) with $W \neq 0$ can be *homogenized* to $(x/W, y/W, z/W, 1)$.

7 Transformation in 3D Homogeneous Coordinated System

Use the right-handed coordinated system. Rotating will be done about the x-, y-, or z-axis. When rotating about the x-axis, goes from y to z; about the y-axis, goes from z to x; and about the z-axis, goes from x to y.

- Translation:

$$T(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Scaling:

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Rotating about z-axis:

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

- Rotating about x-axis:

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

- Rotating about y-axis:

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

8 Composition of 3D Transformations

As with the composition of 2D transformations, the same question applied for a sequence of transformations? It is not the computation that makes this problem difficult (though it is a tedious one!). The main difficulty with 3D transformations lies in the identification of various steps that would help achieve the final result. The example from the textbook is a typical one.

9 Transformations as a Change in Coodinate Systems

The starting point: Objects might have been specified in different coordinates. $P^{(i)}$ denotes the coordinates of P in the coordinate system i. So, the question is what is the transformation that allows us to convert the coordinates of P in i to coordinates of P in j?

We denote by $M_{i \leftarrow j}$ the transformation that allows us to convert the coordinates of P in i to coordinates of P in j, i.e., $P^{(i)} = M_{i \leftarrow j} P^{(j)}$. To determine $M_{i \leftarrow j}$, we should do:

- Translate the origin of j to i
- Rotate so that the axies (of the different systems) align
- Scaling so unit of the two systems are the same

A few properties of $M_{i \leftarrow j}$:

- $M_{i \leftarrow j} = M^{-1}_{j \leftarrow i}$
- $M_{i \leftarrow j} = M_{i \leftarrow k} . M_{k \leftarrow j}$

The next important question is about the changes of a transformation matrix in one system over the other. More precisely, let $Q^{(j)}$ be a transformation in the coordinate system j. What is the transformation $Q^{(i})$ in the coordinate system i which has **the same effects** as $Q^{(j)}$ on a point $P^{(i)}$ as $Q^{(j)}$ has on the point $P^{(j)}$? To derive $Q^{(i})$, we observe that it can be obtained by

- Translate $P^{(i)}$ to $P^{(j)}$ (so, $M_{j \leftarrow i}$)
- Apply $Q^{(j)}$
- Translate back to $P^{(i)}$ (so, $M_{i \leftarrow j}$)

This gives us:

$$Q^{(i}) = M_{i \leftarrow j} . Q^{(j)} . M_{j \leftarrow i} = M_{i \leftarrow j} . Q^{(j)} . M^{-1}_{i \leftarrow j}$$