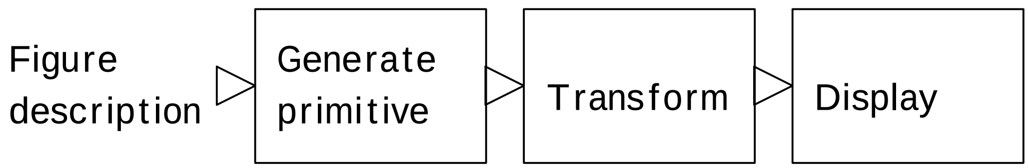
Geometrical Transformations

2D Transformations

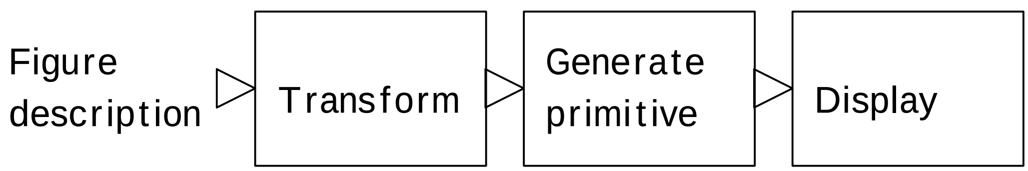
- want to change the position, orientation, and size of objects (for most applications)

- what do we want to transform?

    1. All points - pt. by pt. works always but is very slow.



    2. Vertices (endpts.) only - good for lines, not for curves.



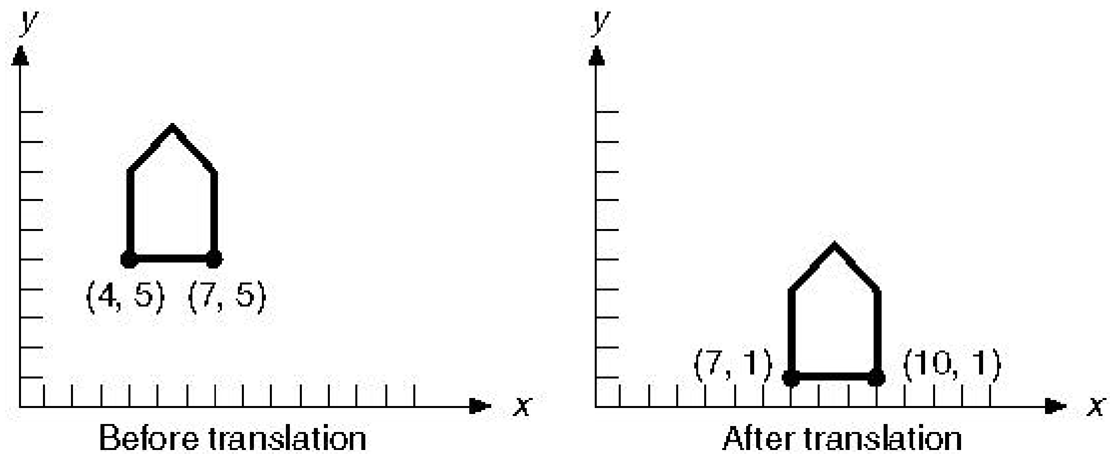
- assume objects consist of straight line segments. Transform endpoints and redraw the segments. (curved lines are approx. by straight segments or splines) So we are just manipulating the endpoints.

Transformations

Points and Vectors will be written as columns

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

1. Translation (moving an object)



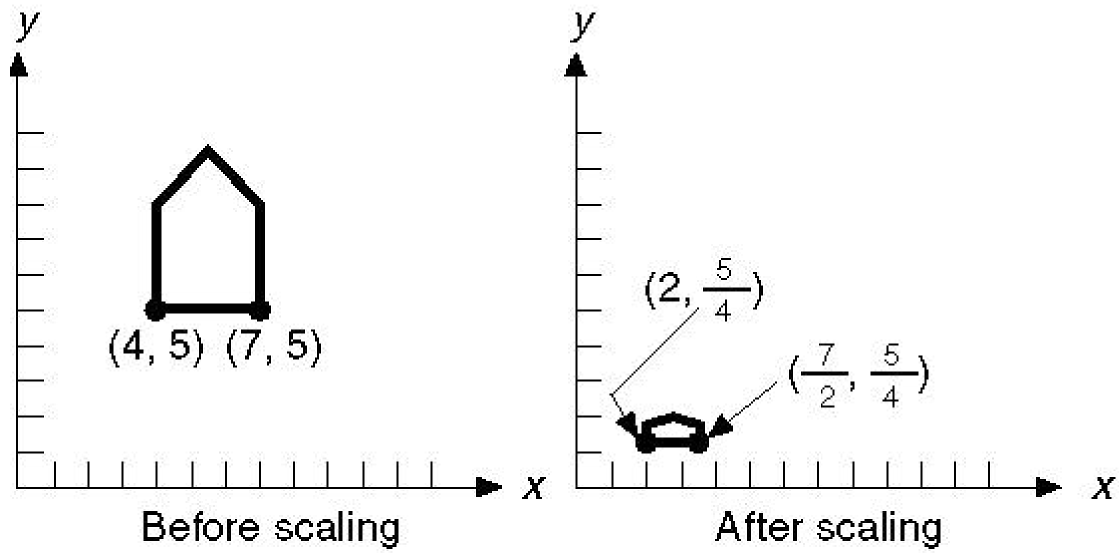
- move points by vector addition

$$P' = P + T$$

$$\begin{aligned} x' &= x + d_x \\ y' &= y + d_y \end{aligned}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

Scaling (changing the size)



- shrink or stretch distances by multiplication

$$P' = S \cdot P$$

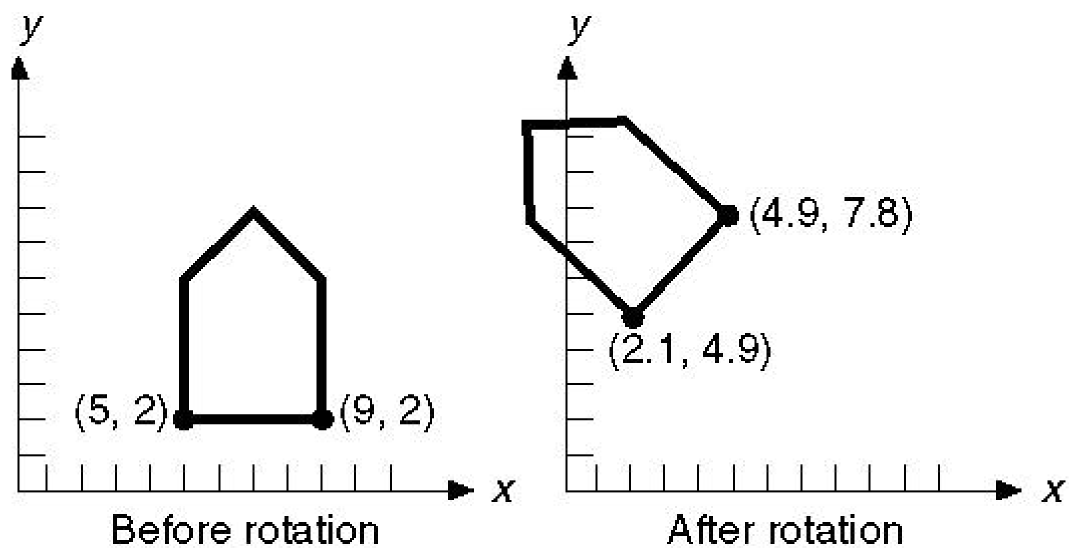
$$x' = s_x \cdot x$$
$$y' = s_y \cdot y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

$k_x = k_y$ for uniform scaling
$k_x \neq k_y$ differential scaling

Note: scaling is about the origin
"house is smaller and closer to the origin"

Rotation (changing position angle)



- change position (angle) by multiplication and addition

$$P' = R \cdot P$$

$$x' = x \cdot \cos\theta - y \cdot \sin\theta$$
$$y' = x \cdot \sin\theta + y \cdot \cos\theta$$

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

- positive angles are measure counterclockwise
- for negative angles:

$\Rightarrow$ cos(-θ) = cos(θ)

$\Rightarrow$ sin(-θ) = -sin(θ)

How is the rotation equation derived?

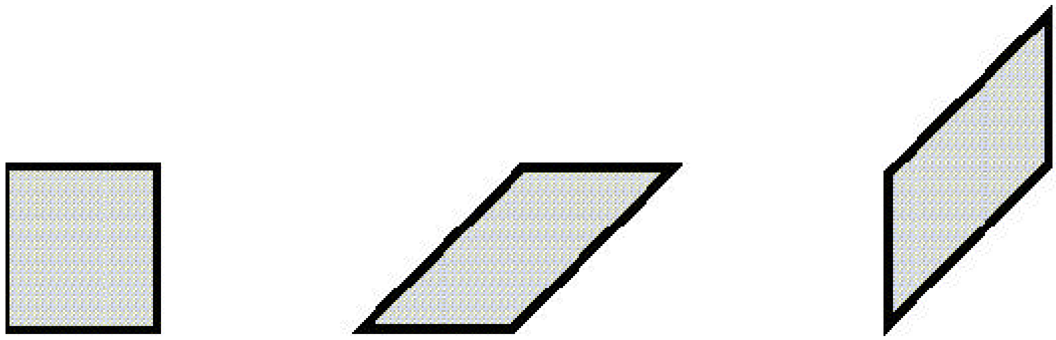
$$
\begin{aligned}
x &= r \cdot \cos\phi \\
y &= r \cdot \sin\phi
\end{aligned}
$$

$$
\begin{aligned}
x\prime &= r \cdot \cos(\theta + \phi) \\
x\prime &= r \cdot \cos\phi \cdot \cos\theta - r \cdot \sin\phi \cdot \sin\theta \\
x\prime &= x \cdot \cos\theta - y \cdot \sin\theta
\end{aligned}
$$

$$
\begin{aligned}
y\prime &= r \cdot \sin(\theta + \phi) \\
y\prime &= r \cdot \cos\phi \cdot \sin\theta + r \cdot \sin\phi \cdot \cos\theta \\
y\prime &= x \cdot \sin\theta + y \cdot \cos\theta
\end{aligned}
$$

2. Shearing (changing slant of the object)

- useful for italics



The unit cube sheared in the x direction

The unit cube sheared in the y direction

- change slant by multiplication and addition

Slant in x

$$P' = SH_x \cdot P$$

$$\begin{aligned} x' &= x + ay \\ y' &= y \end{aligned}$$

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

Slant in y

$$P' = SH_y \cdot P$$

$$\begin{aligned} x' &= x \\ y' &= bx + y \end{aligned}$$

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

Summary:

Translation	P' = T + P
Scaling	P' = S · P
Rotation	P' = R · P
Shearing	P' = SH · P

Unfortunately, translation is different (addition) and we would like to treat all transformations in a consistent way to they can be easily combined.

Problem:

$$P = M_{wv} \begin{bmatrix} ? \\ ? \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x+m \\ y+n \end{bmatrix}$$

$$BUT... \begin{bmatrix} 1 & \frac{m}{y} \\ \frac{n}{x} & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} WRONG!$$

Homogeneous Coordinates

- add a third coordinate to a point
- instead of (x, y) => (x, y, W)

- (x', y', W') and (x, y, W) are the same point if one is a multiple if the other
- there are an infinite number of homogeneous coordinates

$$(x, y) = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \begin{bmatrix} 2x \\ 2y \\ 2 \end{bmatrix} \cdots \begin{bmatrix} Wx \\ Wy \\ W \end{bmatrix}$$

- in general [x y W], $W \neq 0$, represents a point (x/W, y/W) – Cartesian coordinates
- W=1 is normalized
- W=0 are points at infinity

Typically triples of coordinates represent points in 3D space (x, y, z) but here we are using them to represent points in 2D space (x, y, W)

- all triples (tx, ty, tW) where $t \neq 0$ form a line in 3D space
- (x, y, 1) for all x & y form a plane in 3D space

General Transformation Matrix

$$x' = ax + by + l$$
$$y' = cx + dy + m$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & l \\ c & d & m \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

TRANSLATION EQUATION
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

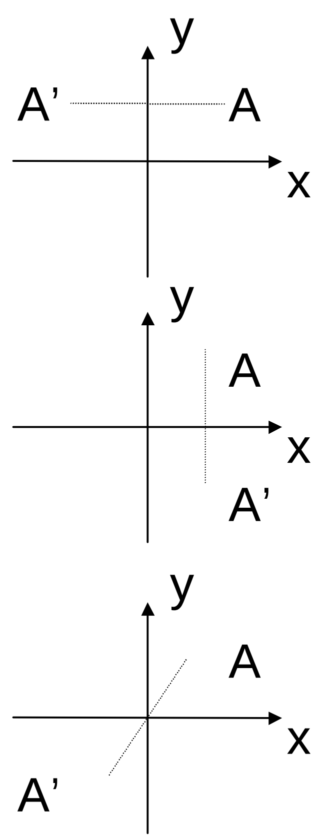
SCALE EQUATION
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

ROTATION EQUATION
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

SHEAR-X EQUATION
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

SHEAR-Y EQUATION
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Symmetries (reflection about axis)



about y
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

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

about origin
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

uniform scaling
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/k \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$T = \begin{bmatrix} a & b & p \\ c & d & q \\ m & n & s \end{bmatrix}$$

a,b,c,d rotation, reflection, shearing, and scaling
p,q translation
s uniform scaling
p,q=0 (will be used in 3D)

Affine transformations

- preserve parallelism but not lengths or angles

Rigid body transformations

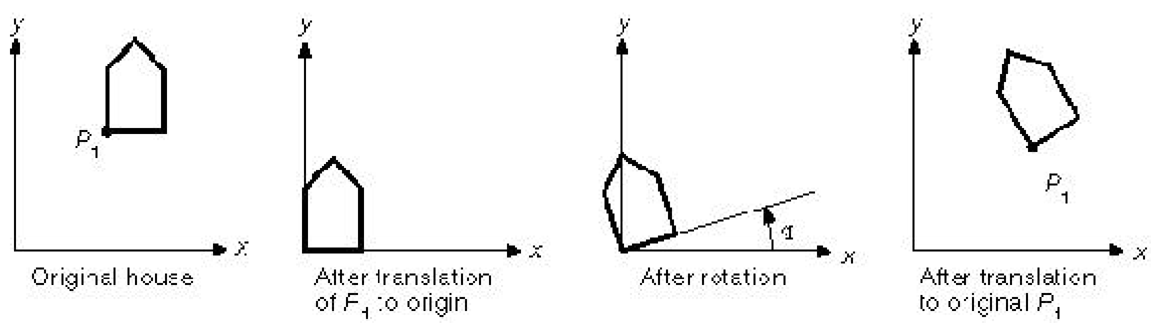
- preserves angles and lengths
- object ("body") is not distorted in any way
- translation? scale? rotate? shear?
- products of rigid body transformations?

Composition

- combine R, S, T, & SH to produce desired general results

Example: Rotation about an arbitrary point P_1

- translate P_1 to origin
- rotate
- translate back



$(T_2 \cdot (R \cdot (T_1 \cdot P)))$

- matrix multiplication is associative
- we can express the three transformations as one matrix:

$(T_2 \cdot (R \cdot T_1)) \cdot P$

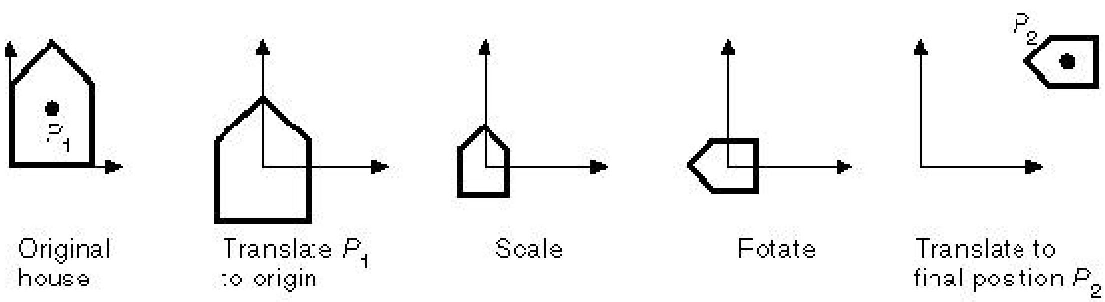
$$T(x_1, y_1) \cdot R(\theta) \cdot T(-x_1, -y_1) = \begin{bmatrix} 1 & 0 & x' \\ 0 & 1 & y' \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} cos(\theta) & -sin(\theta) & x_1(1 - cos(\theta)) + y_1 sin(\theta) \\ sin(\theta) & cos(\theta) & y_1(1 - cos(\theta)) - x_1 sn(\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

Makes a BIG difference when transforming many points

- more efficient
- one composed transformation rather than three matrix operations

Example: Scale, rotate, & position with P_1 as the center for rotation and scaling



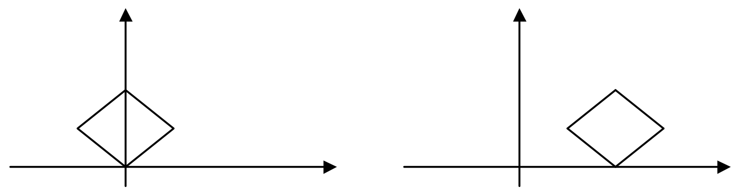
$$T(x_2,y_2) \cdot R(\theta) \cdot S(S_x, S_y) \cdot T(-x_1, -y_1)$$

While matrix multiplication is in general, not commutative, it can be seen that it holds for this example.

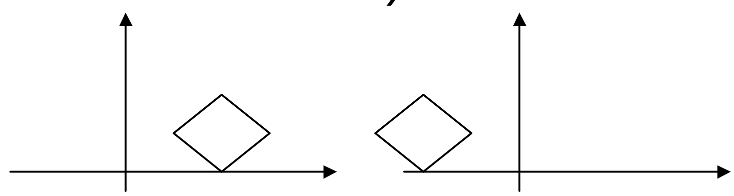
However, other times must be careful of the order in which the transformations are applied, for example.

1. T_1 reflect about y

T_2 translate (m,0)



2. T_2 then T_1, NOT commutative (results differ with order)



NOTE:

- In the text and in our examples we are premultiplying transformation matrices with points:

$$P' = T_2 \cdot T_1 \cdot P \qquad\qquad T = \ldots T_3\, T_2\, T_1$$

- We could also postmultiply:;

$$P' = P \cdot T_1^T \cdot T_2^{\,T} \qquad\qquad T = T_1\, T_2\, T_3 \ldots$$

* we must transpose matrices to go from one convention to the other

Window to Viewport Transformation

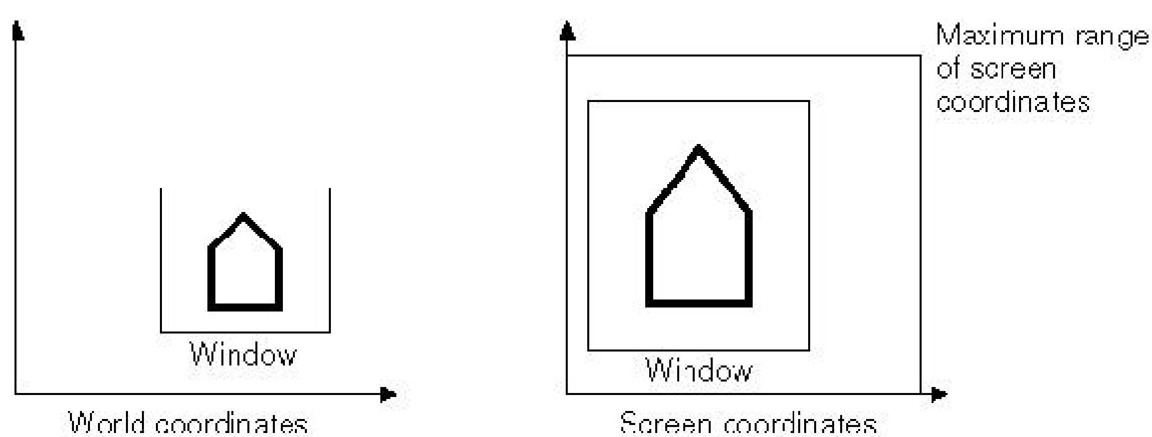
- World-coordinate system: "where the objects reside"
  (also called world space, object space)

- Screen-coordinate system: "display or output objects"
  (screen space, device coordinates, image space)

  OR

- World-coordinate WINDOW: rectangular region in world-space.

- Screen-coordinate VIEWPORT: rectangular region in screen-space.



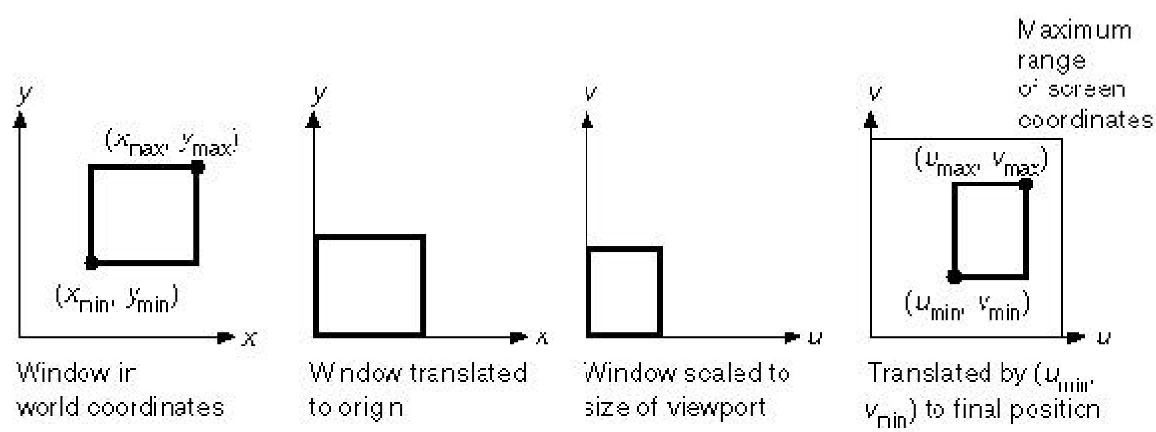
Given a window & viewport, what is the transformation matrix that maps the window from world coordinates into the viewport in screen coordinates

Three steps:

- Translate to origin ($-x_{min}$, $-y_{min}$)
- Scale window to size of viewport

$$\left(\frac{u_{max} - u_{min}}{x_{max} - x_{min}}, \frac{v_{max} - v_{min}}{y_{max} - y_{min}}\right)$$

- Translate to final position (u_{min}, v_{min})



$$M_{wv} = T(u_{min}, v_{min}) \cdot S(\frac{u_{max} - u_{min}}{x_{max} - x_{min}}, \frac{v_{max} - v_{min}}{y_{max} - y_{min}}) \cdot T(-x_{min}, -y_{min})$$

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

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

$$P = M_{wv} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$P = \begin{bmatrix} (x = x_{min}) \cdot \frac{u_{max}-u_{min}}{x_{max}-x_{min}} + u_{min} & (y = y_{min}) \cdot \frac{v_{max}-v_{min}}{y_{max}-y_{min}} + v_{min} & 1 \end{bmatrix}$$

Clipping is generally combined with this mapping