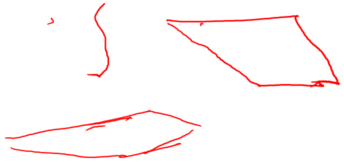
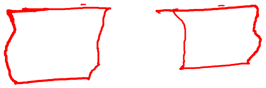
Transformations

Transformations

- We used the window to viewport transformation to scale and translate objects in the world window to their size and position in the viewport.
- We gain more flexible control over the size, orientation, and position of objects of interest.
- To do so, we will use the powerful **affine transformation**.



Affine Transformation



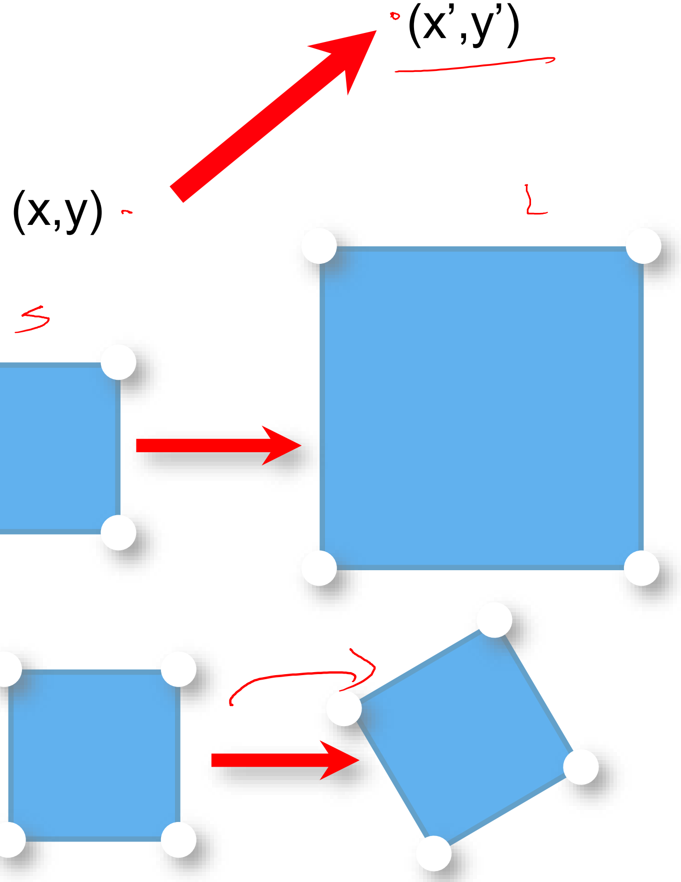
An *affine transformation* is any transformation that preserves collinearity (i.e., all points lying on a line initially still lie on a line after transformation) and ratios of distances (e.g., the midpoint of a line segment remains the midpoint after transformation).

Using affine transformation we gain more flexible control over the size, orientation and position of objects of interest.

Affine transformation has a simple form: The coordinates of Q are linear combination of those of P.

2D Transforms

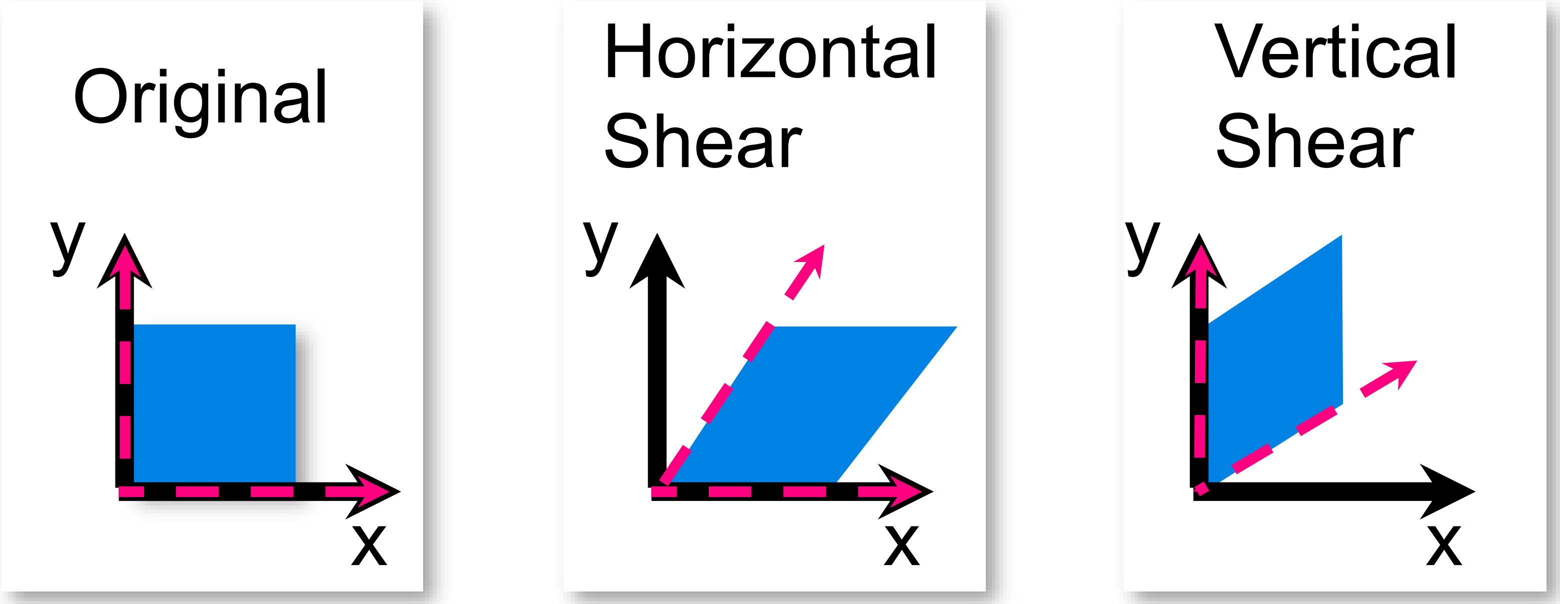
- What am I talking about when I say "transforms"?



  - Translation
  - Scaling
  - Rotation

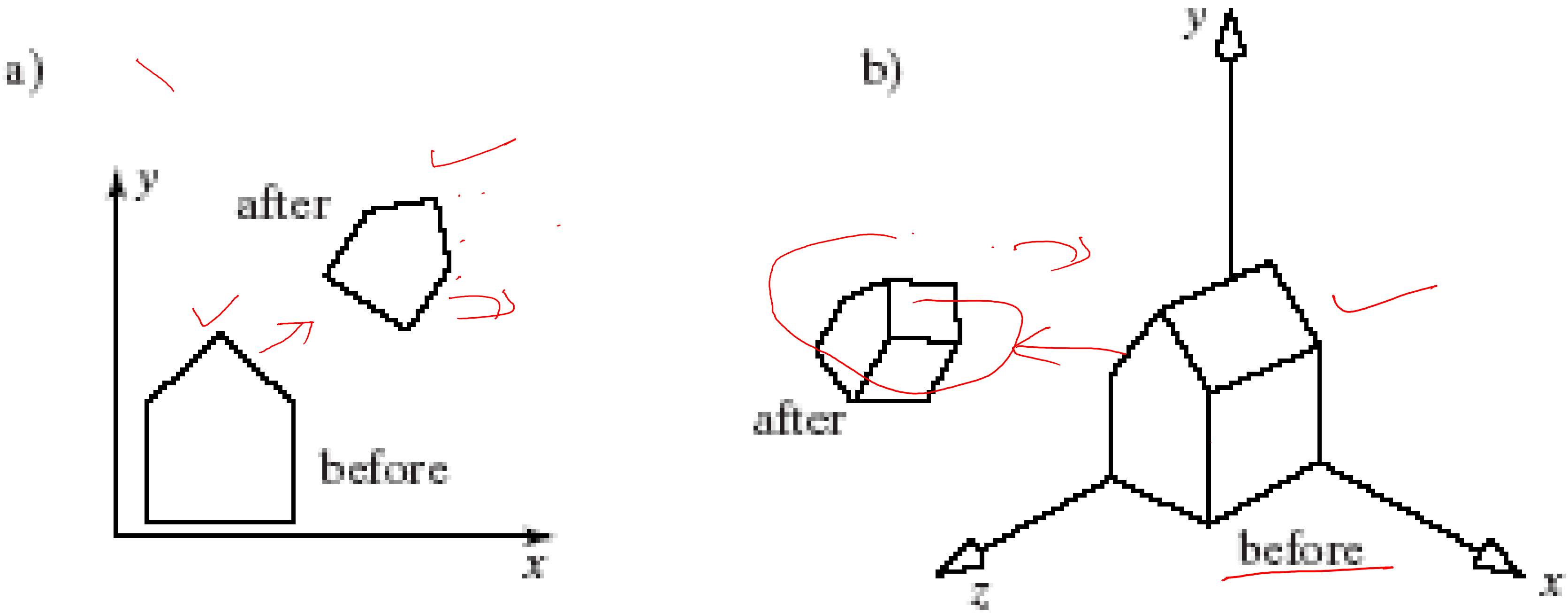
Shearing

It



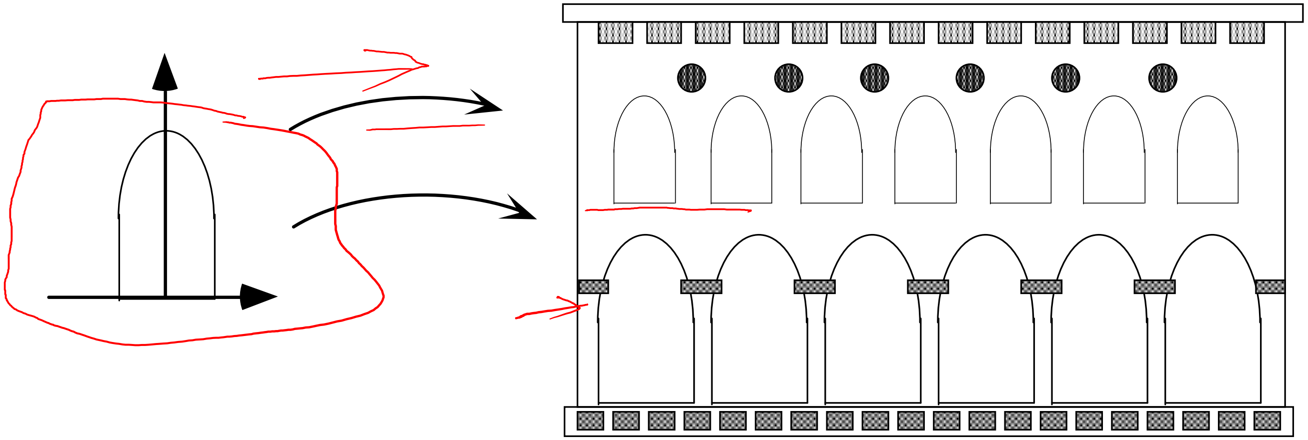
Example of Affine Transformations

- The house has been scaled, rotated and translated, in both 2D and 3D.



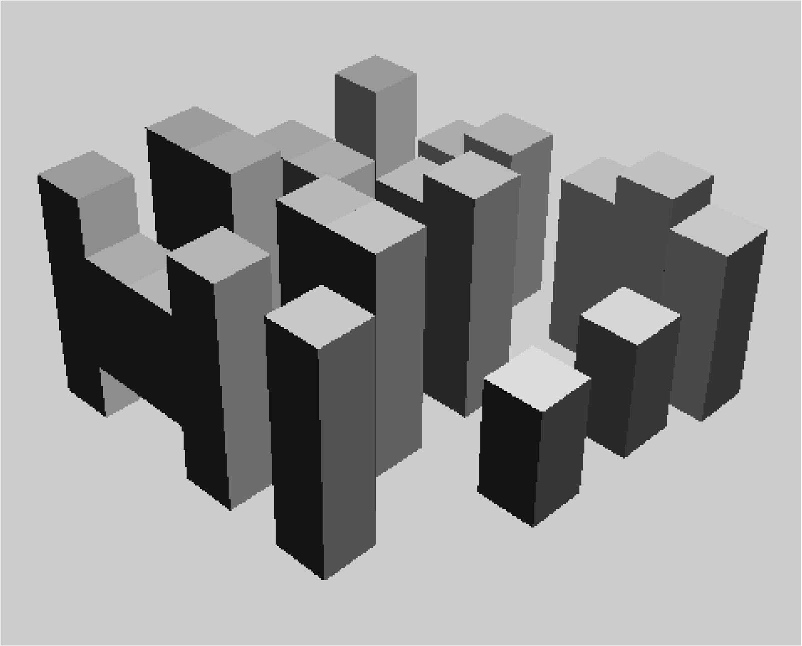
Using Transformations

- The arch is designed in its own coordinate system.
- The scene is drawn by placing a number of instances of the arch at different places and with different sizes.



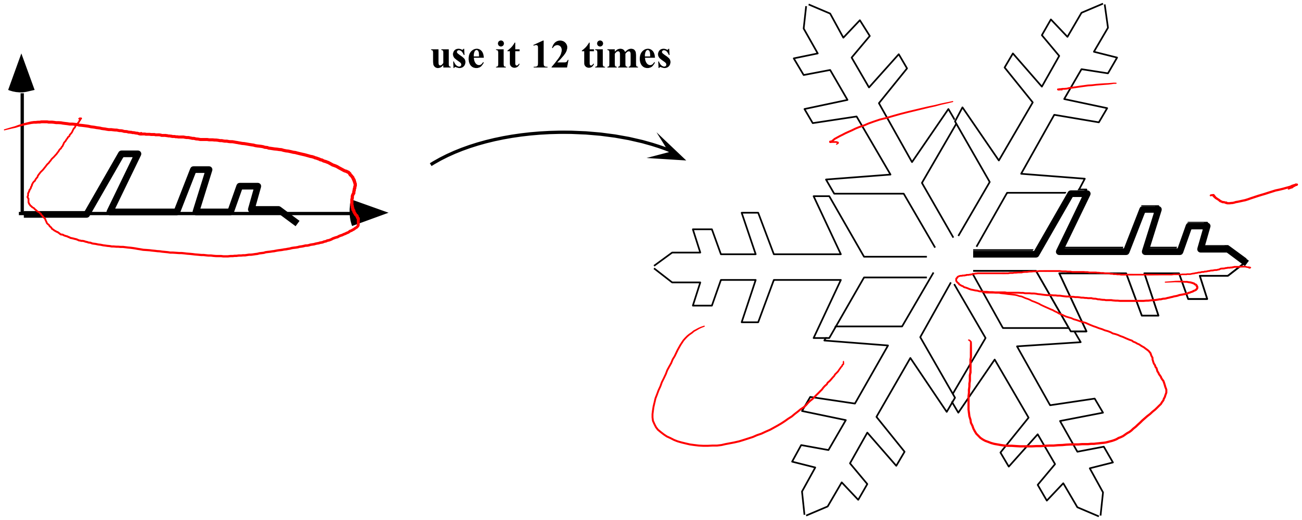
Using Transformations (2)

- In 3D, many cubes make a city.



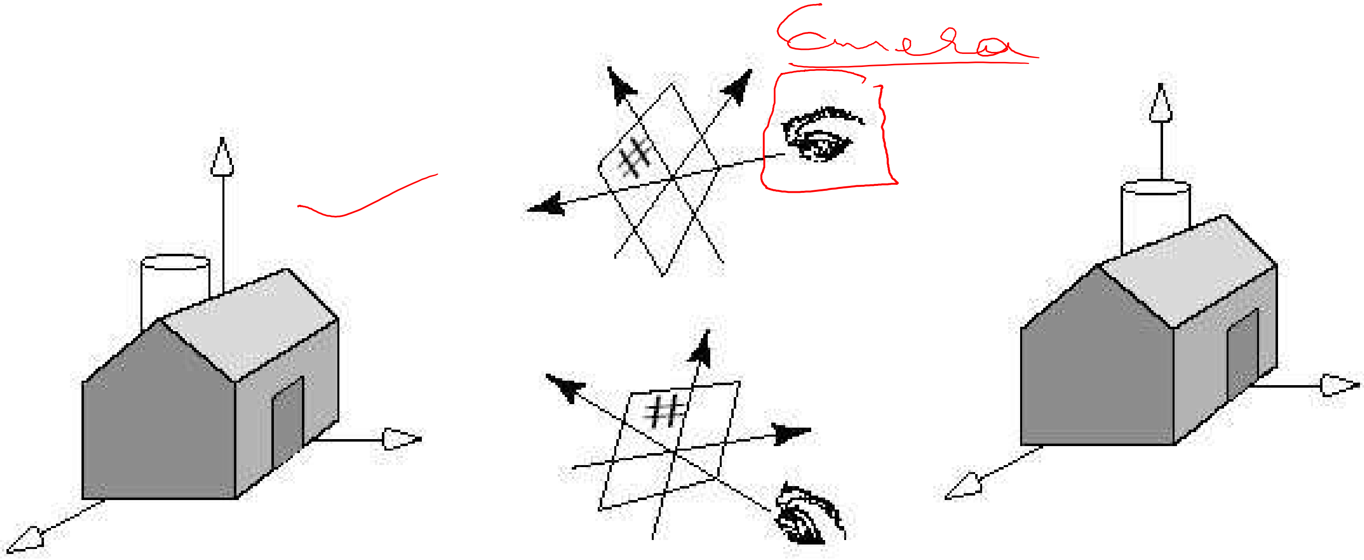
Using Transformations (3)

- The snowflake exhibits symmetries.
- We design a single **motif** and draw the whole shape using appropriate reflections, rotations, and translations of the motif.



Using Transformations (4)

- Positioning and reorienting a camera can be carried out through the use of 3D affine transformations.

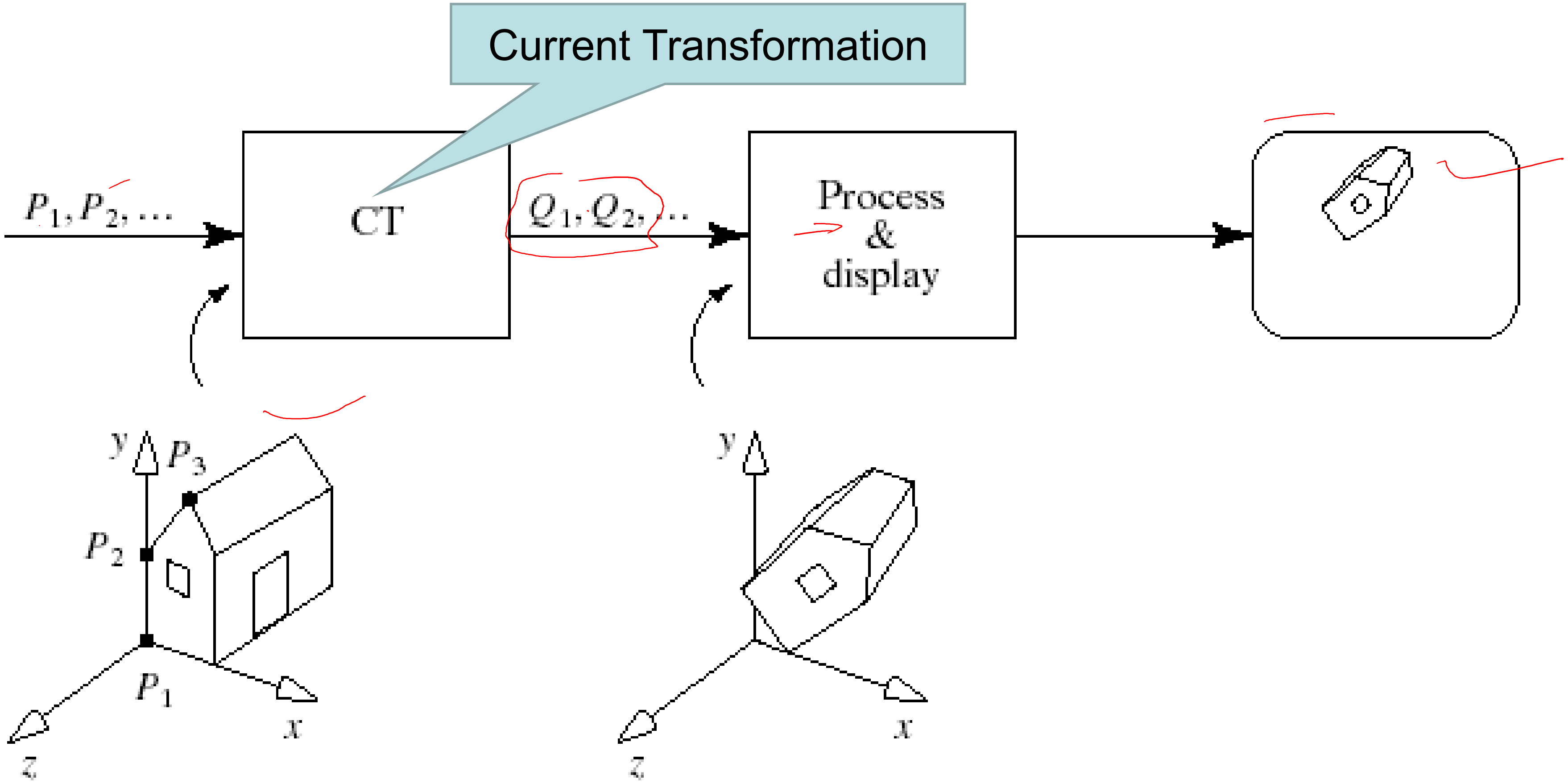


Using Transformations (5)

- In a computer animation, objects move.
- We make them move by translating and rotating their local coordinate systems as the animation proceeds.
- A number of graphics platforms, including OpenGL, provide a graphics pipeline: a sequence of operations which are applied to all points that are sent through it.
- A drawing is produced by processing each point.

The OpenGL Graphics Pipeline

- This version is simplified.



Current Transformation

Graphics Pipeline (2)

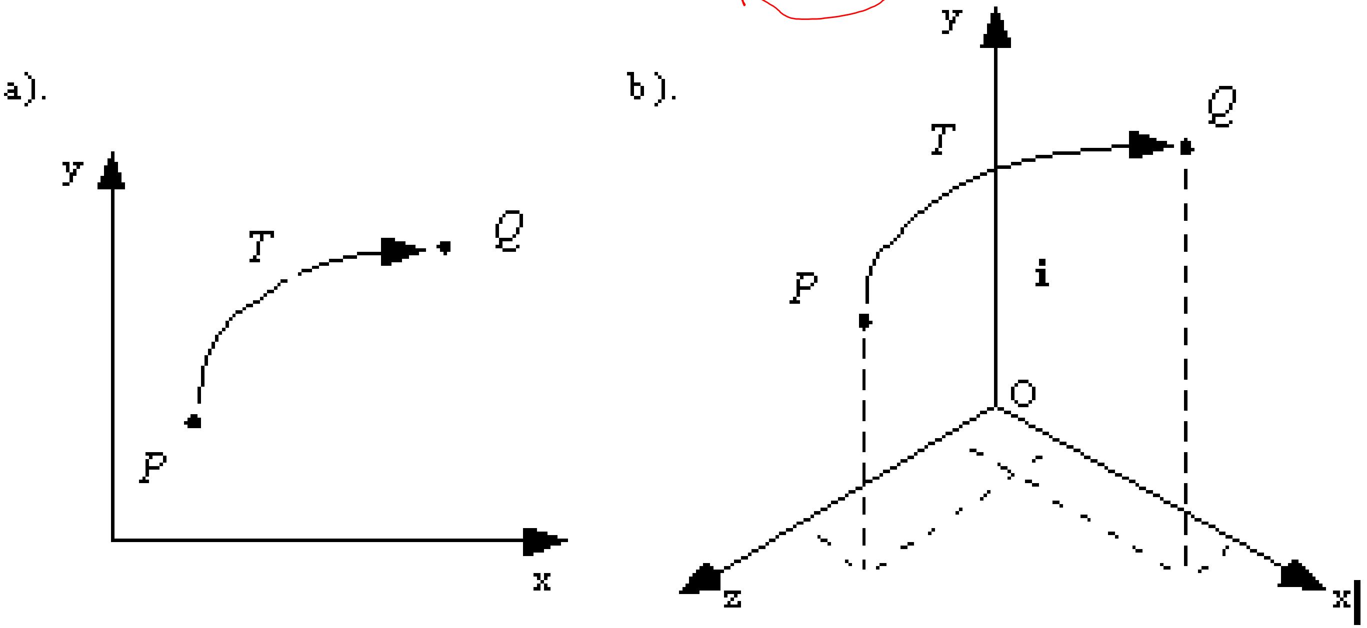
- An application sends the pipeline a sequence of points P_1, P_2, ... using commands such as:

```
glBegin(GL_LINES);
  glVertex3f(...); // send P1 through the pipeline
  glVertex3f(...); // send P2 through the pipeline
  ...
glEnd();
```

- These points first encounter a transformation called **the current transformation** (CT), which alters their values into a different set of points, say Q_1, Q_2, Q_3.

Transformations

- A (2D or 3D) transformation *T*( ) alters each point, *P* into a new point, *Q*, using a specific formula or algorithm: *Q*= *T*(*P*).



Transformations (2)

- An arbitrary point *P* in the plane is **mapped** to *Q*.
- *Q* is the **image** of *P* under the mapping *T*.
- We transform an object by transforming each of its points, using the *same* function *T*() for each point.
- The **image** of line *L* under *T*, for instance, consists of the images of *all* the individual points of L.

Transformations (3)

- Most mappings of interest are continuous, so the image of a straight line is still a connected curve of some shape, although it's not necessarily a straight line.
- Affine transformations, however, *do* preserve lines: the image under *T* of a straight line is also a straight line.

Transformations (4)

- We use an explicit coordinate frame when performing transformations.
- A coordinate frame consists of a point $\mathcal{O}$, called the **origin**, and some mutually perpendicular vectors (called **i** and **j** in the 2D case; **i, j,** and **k** in the 3D case) that serve as the axes of the coordinate frame.
- In 2D,

$$\widetilde{P} = \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}, \widetilde{Q} = \begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix}$$

Transformations (5)

- Recall that this means that point $\mathscr{P}$ is at location $= \mathscr{P}_x \mathbf{i} + \mathscr{P}_y \mathbf{j} + \mathscr{O}$, and similarly for $\mathscr{Q}$.
- $\mathscr{P}_x$ and $\mathscr{P}_y$ are the coordinates of $\mathscr{P}$.
- To get from the origin to point $\mathscr{P}$, move amount $\mathscr{P}_x$ along axis $\mathbf{i}$ and amount $\mathscr{P}_y$ along axis $\mathbf{j}$.

Transformations (6)

- Suppose that transformation *T* operates on any point $\mathcal{P}$ to produce point $\mathcal{Q}$:
- $\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = T(\begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix})$ or $\mathcal{Q} = \mathrm{T}(\mathcal{P})$.
- T may be any transformation: e.g.,

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} \underline{\cos}(P_x)\underline{e}^{-P_x} \\ \boxed{\dfrac{\ln(P_y)}{1+P_x^2}} \\ 1 \end{pmatrix}$$

Transformations (7)

- To make **affine** transformations we restrict ourselves to much simpler families of functions, those that are *linear* in P_x and P_y.
- Affine transformations make it easy to scale, rotate, and reposition figures.
- Successive affine transformations can be combined into a single overall affine transformation.

Affine Transformations

- Affine transformations have a compact matrix representation.
- The matrix associated with an affine transformation operating on 2D vectors or points must be a three-by-three matrix.
  - This is a direct consequence of representing the vectors and points in homogeneous coordinates.
  - Any point in the projective plane is represented by a triple (X, Y, Z), called the **homogeneous coordinates** or projective **coordinates** of the point, where X, Y and Z are not all 0

Affine Transformations (2)

- Affine transformations have a simple form.
- Because the coordinates of $\mathcal{Q}$ are *linear* combinations of those of $\mathcal{P}$, the transformed point may be written in the form:

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11}P_x + m_{12}P_y + m_{13} \\ m_{21}P_x + m_{22}P_y + m_{23} \\ 1 \end{pmatrix}$$

Affine Transformations (3)

- There are six given constants: m_{11}, m_{12}, etc.
- The coordinate Q_x consists of portions of both P_x and P_y, and so does Q_y.
- This *combination* between the *x*- and *y*-components also gives rise to rotations and shears.

Affine Transformations (4)

- Matrix form of the affine transformation in 2D:

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{pmatrix}}_{M} \underbrace{\begin{pmatrix} P_x \\ P_y \\ \boxed{1} \end{pmatrix}}_{} \quad D$$

- For a 2D affine transformation the third row of the matrix is always (0, 0, 1).

Affine Transformations (5)

- Some people prefer to use row matrices to represent points and vectors rather than column matrices: e.g., $P = (P_x, P_y, 1)$
- In this case, the *P* vector must *pre-multiply* the matrix, and the transpose of the matrix must be used: $Q = P\,\boxed{M^T}$. ✓

$$M^T = \begin{pmatrix} m_{11} & m_{21} & 0 \\ m_{12} & m_{22} & 0 \\ m_{13} & m_{23} & 1 \end{pmatrix}$$

Affine Transformations (6)

- Vectors can be transformed as well as points.
- If a 2D vector **v** has coordinates V_x and V_y then its coordinate frame representation is a column vector with third component 0.

Affine Transformations (7)

- When vector **V** is transformed by the same affine transformation as point P, the result is

$$\begin{pmatrix} W_x \\ W_y \\ 0 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ \boxed{0} \end{pmatrix}$$

- Important: to transform a point *P* into a point *Q, post-multiply M* by *P*: Q = M P.

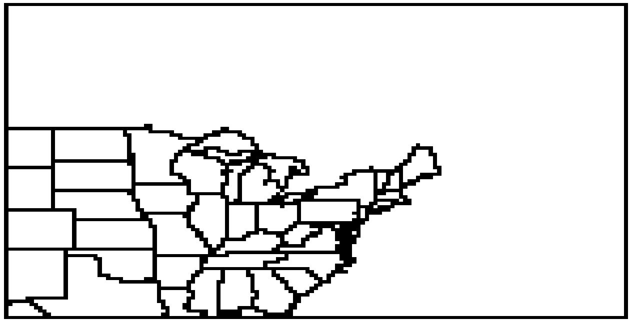
Affine Transformations (8)

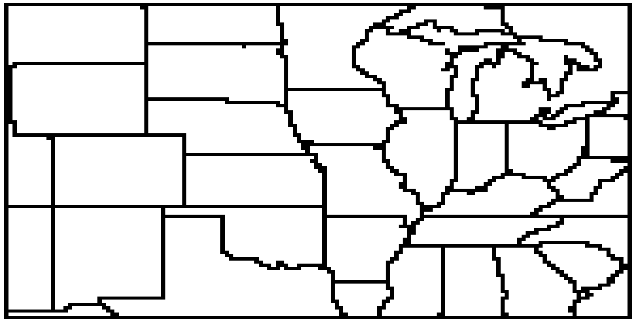
- Example: find the image Q of point P = (1, 2, 1) using the affine transformation

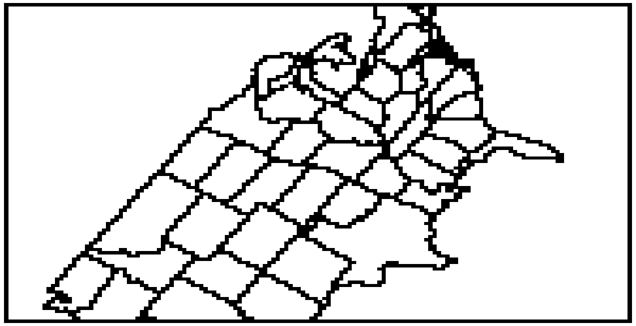
$$M = \begin{pmatrix} 3 & 0 & 5 \\ -2 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}; Q = \begin{pmatrix} 8 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 5 \\ -2 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

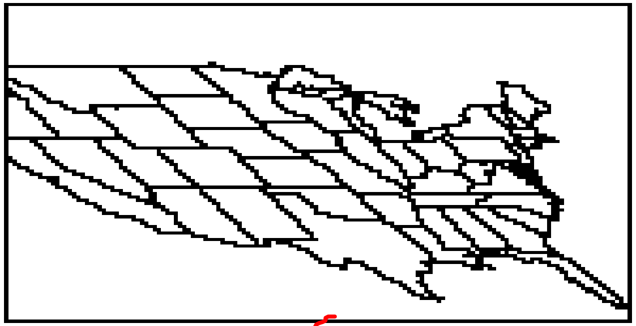
Geometric Effects of Affine Transformations

- Combinations of four elementary transformations: (a) a translation, (b) a scaling, (c) a rotation, and (d) a shear (all shown below).


a)


b)


c)


d)

Translations

- The amount *P* is translated does not depend on P's position.
- It is meaningless to translate vectors.
- To translate a point P by a in the x direction and b in the y direction use the matrix:

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix} = \begin{pmatrix} Q_x + a \\ Q_y + b \\ 1 \end{pmatrix}$$

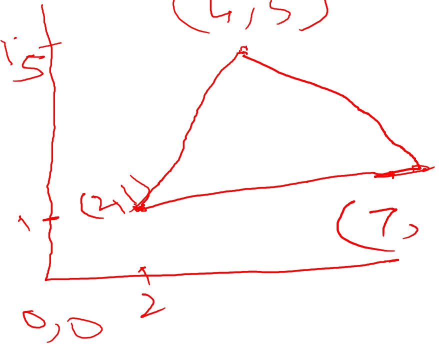
- Only using homogeneous coordinates allow us to include translation as an affine transformation.

translate 2D

$P(2,3)$ $\boxed{5, 7}$

$Q\ (7,10)$

$(4,5)$



$$\begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 7 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 + 0 + 5 \\ 0 + 3 + 7 \\ 0 + 0 + 1 \end{bmatrix} = \begin{bmatrix} 7 \\ 10 \\ 1 \end{bmatrix}$$

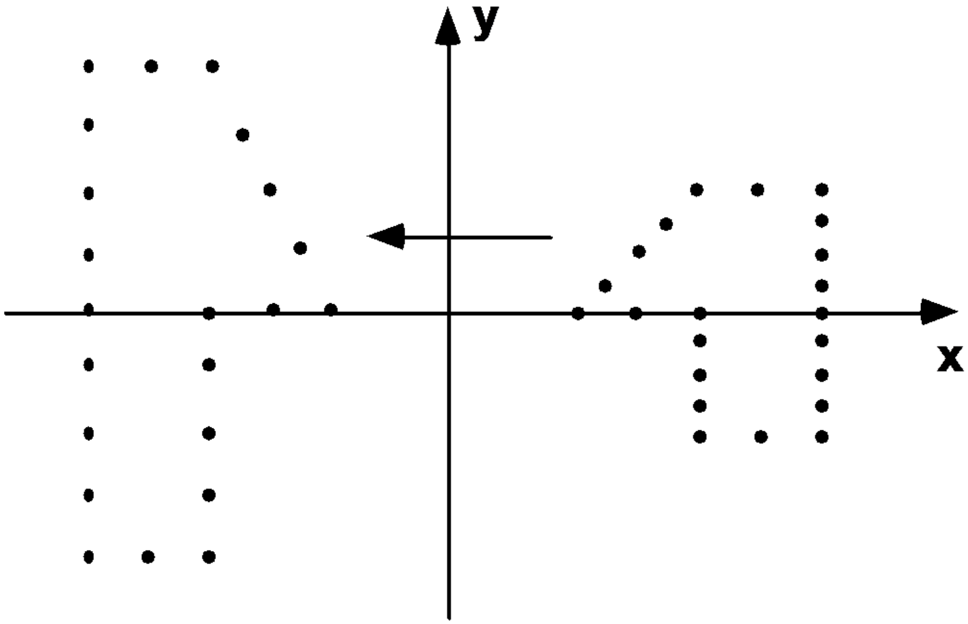
Scaling

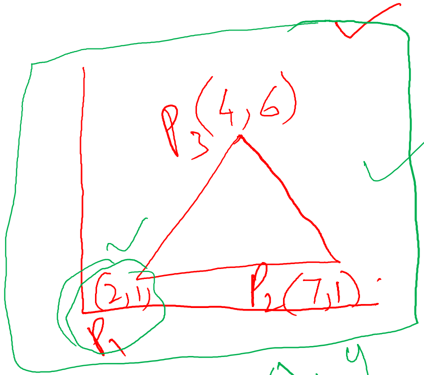
- Scaling is about the origin. If $S_x = S_y$ the scaling is uniform; otherwise it distorts the image.
- If S_x or $S_y < 0$, the image is reflected across the x or y axis.
- The matrix form is

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$

Example of Scaling

- The scaling (*S*x, *S*y) = (-1, 2) is applied to a collection of points. Each point is both reflected about the *y*-axis and scaled by 2 in the *y*-direction.

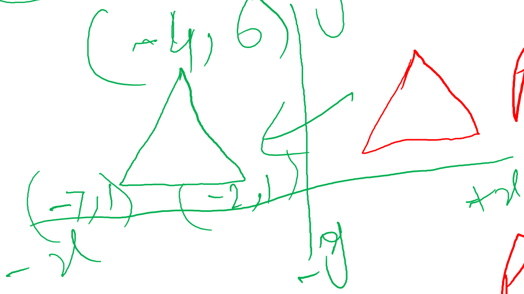




$S_x = 2 \quad S_y = 2$

$S_x = -1 , S_y = 1$

$1/2 = 0.5$



$$P_1' = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix}$$

$$P_2' = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 14 \\ 2 \\ 1 \end{bmatrix}$$

$$P_3' = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 6 \\ 1 \end{bmatrix} = \begin{bmatrix} 8 \\ 12 \\ 1 \end{bmatrix}$$

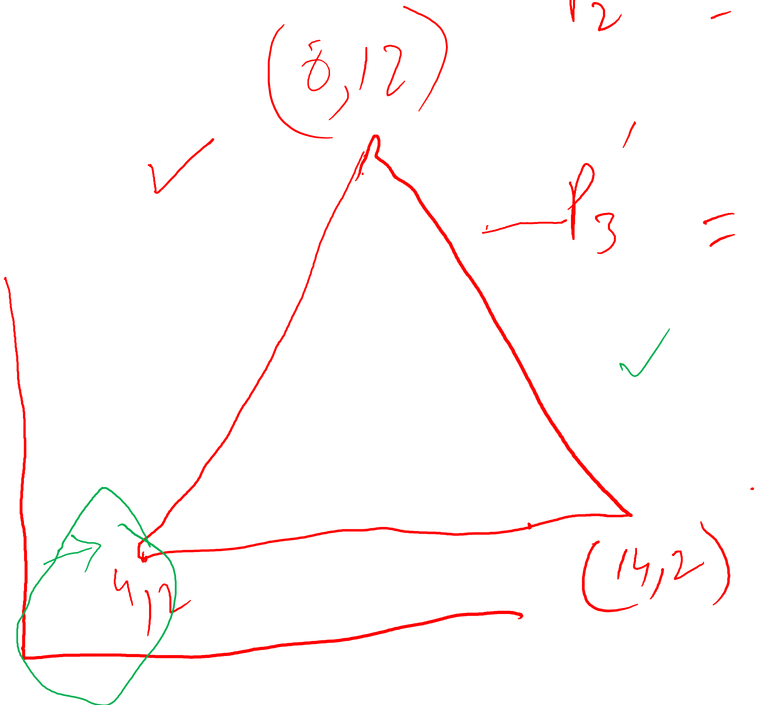
$$P_1' = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

$= (-2, 1)$

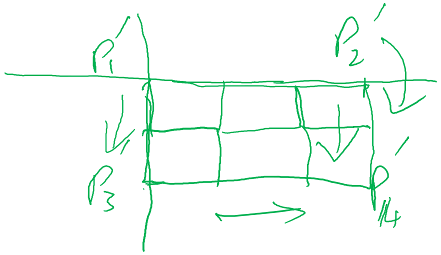
$$P_2' = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 1 \\ 1 \end{bmatrix}$$

$(-7, 1)$

$P_3' = (-4, 6)$

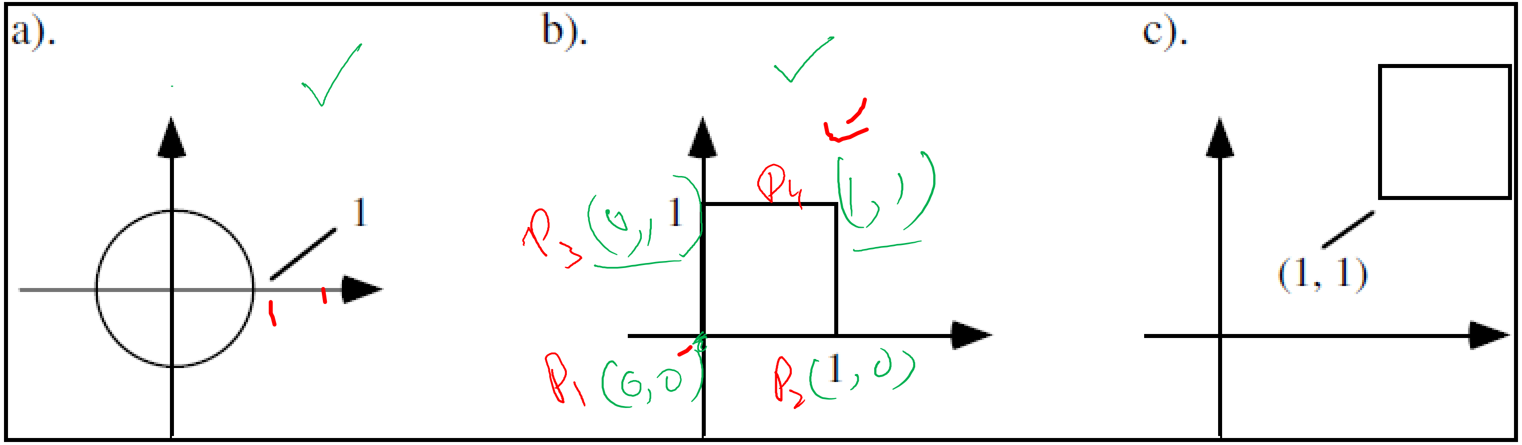


Example



- A pure scaling affine transformation uses scale factors $Sx = 3$ and $Sy = -2$. Find the image of each of the three objects.

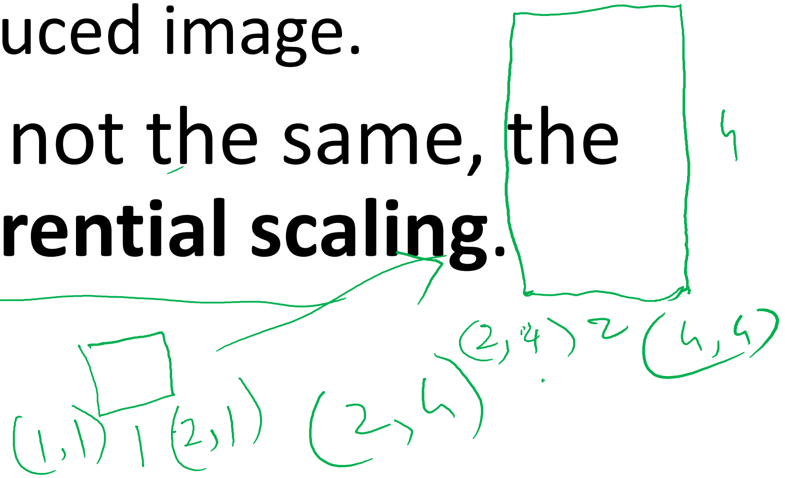
$P_1'(0,0)$, $P_2'(3,0)$, $P_3'(0,-2)$, $P_4'(3,-2)$



Types of Scaling

- Pure reflections, for which each of the scale factors is +1 or -1.
- A **uniform scaling**, or a magnification about the origin: $S_x = S_y$, magnification |S|.
  - Reflection also occurs if S_x or S_y is negative.
  - If $|S| < 1$, the points will be moved closer to the origin, producing a reduced image.
- If the scale factors are not the same, the scaling is called a **differential scaling**.

$S_x = 2$, $S_y = 4$



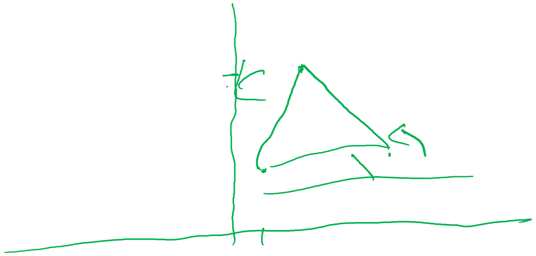
4,6 → (2.96, 4.36)

(3,4)

(5.56, 1.86)

P(3,1) (7,1) →

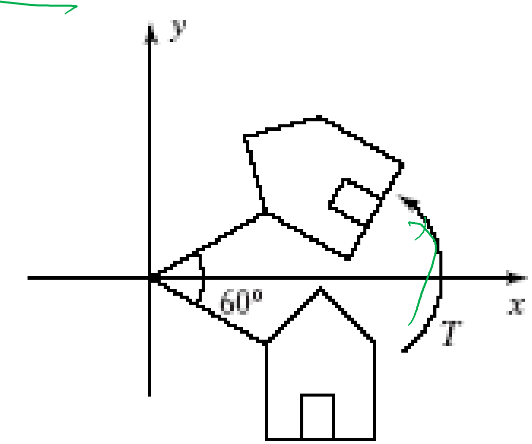
(6,2)



Rotation

- Counterclockwise around origin by angle θ:

$$\begin{pmatrix}Q_x\\ Q_y\\ 1\end{pmatrix} = \begin{pmatrix}\cos(\theta) & -\sin(\theta) & 0\\ \sin(\theta) & \cos(\theta) & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}P_x\\ P_y\\ 1\end{pmatrix}$$



$$P_1 = \begin{bmatrix}\cos 30 & -\sin 30 & 0\\ \sin 30 & \cos 30 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}2\\ 1\\ 1\end{bmatrix} = \begin{bmatrix}0.866 & -0.5 & 0\\ 0.5 & 0.866 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}2\\ 1\\ 1\end{bmatrix}$$

(1.232, 1.866)

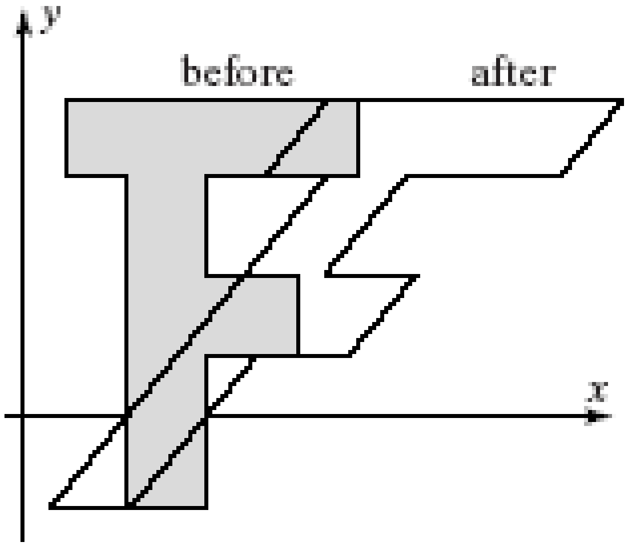
(1, 2)

Degrees	0	30°	45°	60°	90°
Radians	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$
sin θ	0	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$	1
cos θ	1	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	0
tan θ	0	$\frac{\sqrt{3}}{3}$	1	$\sqrt{3}$	Undefined

Shear

- Shear H about origin: x depends linearly on y in the figure.
- Shear along x: h ≠ 0, and P_x depends on P_y (for example, *italic* letters).
- Shear along y: g ≠ 0, and P_y depends on P_x.
- Into which point does (3, 4) shear when $h = .3$

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & \boxed{h} & 0 \\ \boxed{g} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$



P$_1$(-3, 4) P$_2$(2, 4)   g=0.4

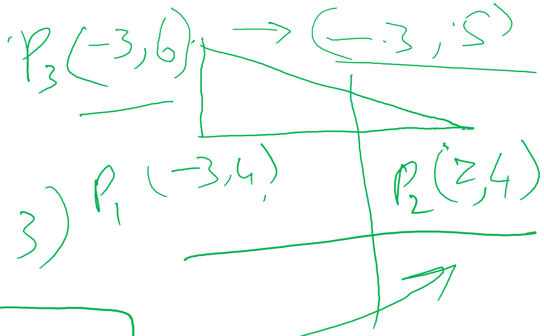
$$\boxed{\begin{array}{r} 4.0 \\ -1.2 \\ \hline \end{array}}$$

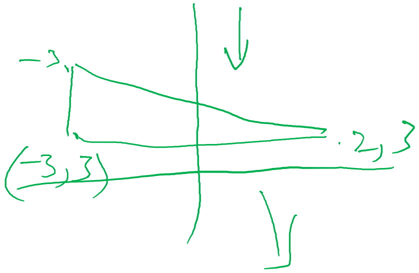
$$\begin{bmatrix} 1 & 0 & 0 \\ 0.4 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -3 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} -3 \\ 2.8 \\ 1 \end{bmatrix} = (-3, 3)$$

$$\boxed{(2, \overset{3}{2.8})}$$

$$\begin{array}{r} -1.2 \\ +4 \\ \hline 2.8 \end{array}$$

$$\begin{array}{r} 6 \\ -1.2 \\ \hline 4.8 \end{array}$$





Inverse Translation and Scaling

- Inverse of translation
  T^{-1}:

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$

$\boxed{2}$ $\frac{1}{2} = 0.5$

- Inverse of scaling
  S^{-1}:

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1/S_x & 0 & 0 \\ 0 & 1/S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$

Inverse Rotation and Shear

- Inverse of rotation $R^{-1} = R(-\theta)$:

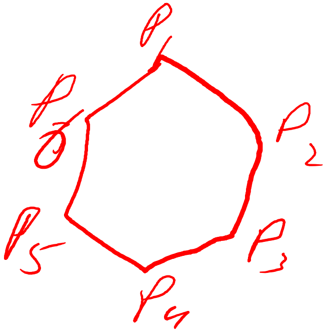
$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$

- Inverse of shear H^{-1}: generally $h \neq 0$ or $g \neq 0$.

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & -h & 0 \\ -g & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix} \boxed{\frac{1}{1-gh}}$$

Composing Affine Transformations

- Usually, we want to apply several affine transformations in a particular order to the figures in a scene: for example,
  - translate by (3, - 4)
  - then rotate by 30º
  - then scale by (2, - 1) and so on.



18

9

- Applying successive affine transformations is called **composing** affine transformations.

Composing Affine Transformations(2)

$$\begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1.5 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} .707 & -.707 & 0 \\ .707 & .707 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \boxed{\begin{pmatrix} 1.06 & -1.06 & 3 \\ -1.414 & -1.414 & 5 \\ 0 & 0 & 1 \end{pmatrix}}$$

P_1 P

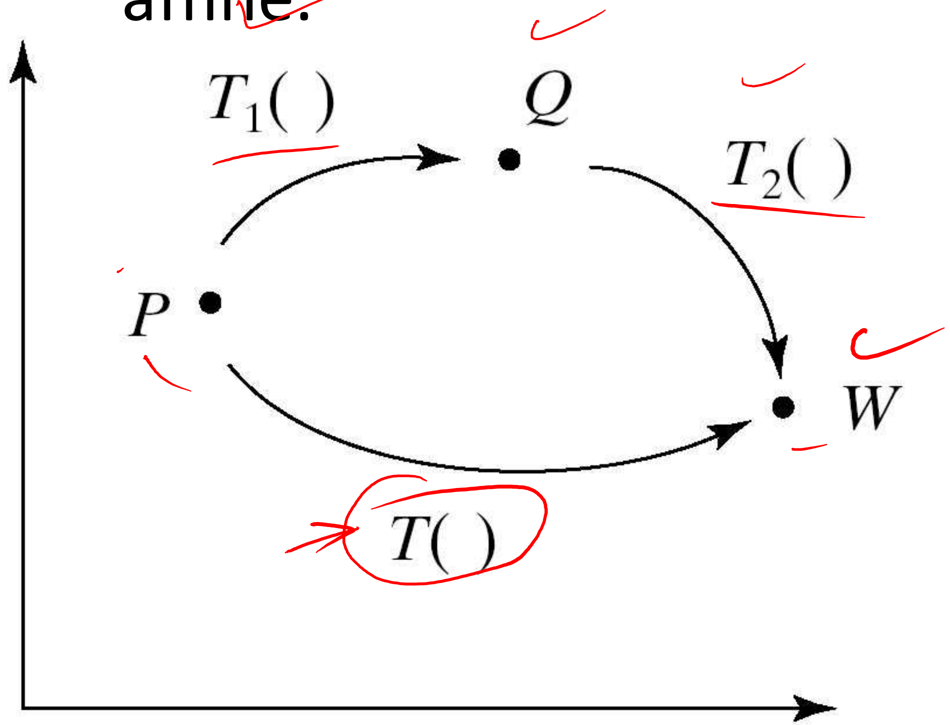
1) Rotate at 45°

2) Scale $S_x = 1.5$, $S_y = -2$

3) translate 3, 5

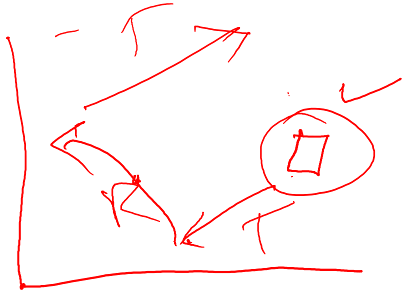
Composing Affine Transformations (3)

- $T_1(\ )$ maps P into Q, and $T_2(\ )$ maps Q into point W. Is $W = T_2(Q) = T_2(T_1(P))$affine?
- Let T_1=M_1 and T_2=M_2, where M_1 and M_2 are the appropriate matrices.
- W = $M_2(M_1P))$ = $(M_2M_1)P$ =MP by associativity.
- So M = M_2M_1, the product of 2 matrices (in reverse order of application), which is affine.



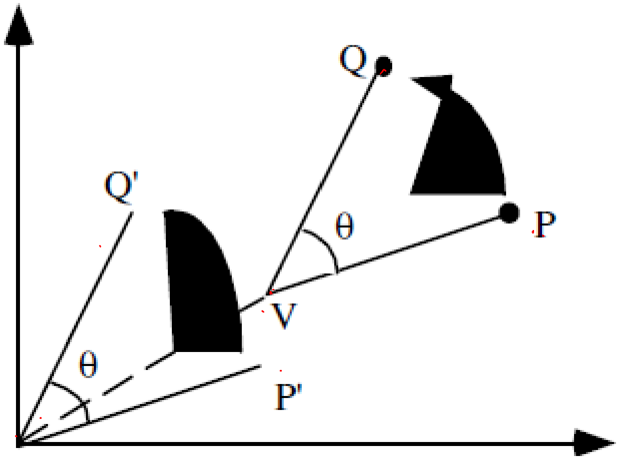
Composing Affine Transformations: Examples

- To rotate around an arbitrary point P: translate P to the origin, rotate, translate P back to original position.
  - $Q = T_P\, R\, T_{-P}\, P$
- Shear around an arbitrary point:
  - $Q = T_P\, H\, T_{-P}\, P$
- Scale about an arbitrary point:
  - $Q = T_P S T_{-P}\, P$

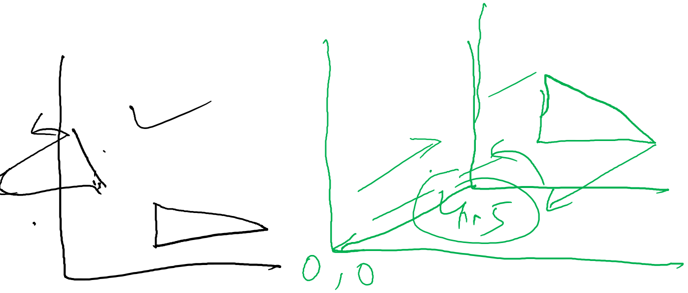


Rotating About an Arbitrary Point

- Translate point *P* through vector
  - **v** = (-*V*x, -*V*y)
- Rotate about the origin through angle θ
- Translate *P* back through **v**



$$\begin{pmatrix} 1 & 0 & V_x \\ 0 & 1 & V_y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & -V_x \\ 0 & 1 & -V_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & d_x \\ \sin(\theta) & \cos(\theta) & d_y \\ 0 & 0 & 1 \end{pmatrix}$$



Example

P3'(3.20 , 8.20)
P2'(−0.23, 1.07)
P1'(−0.23 , 4.22)

O

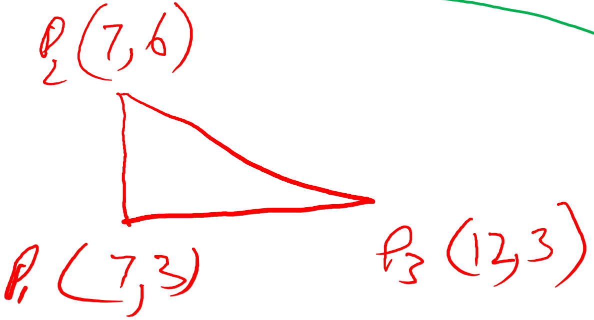
- Find the transformation that rotates points through 30° about (-2, 3),
  - determine to which point the point (1, 2) maps

$$\begin{pmatrix}1&0&4\,\cancel{V_x}\\0&1&5\,\cancel{V_y}\\0&0&1\end{pmatrix}\begin{pmatrix}\cos(\overset{60}{\theta})&-\sin(\overset{60}{\theta})&0\\ \sin(\overset{60}{\theta})&\cos(\overset{60}{\theta})&0\\0&0&1\end{pmatrix}\begin{pmatrix}1&0&-\overset{4}{\cancel{V_x}}\\0&1&-\overset{5}{\cancel{V_y}}\\0&0&1\end{pmatrix}=\begin{pmatrix}\cos(\theta)&-\sin(\theta)&d_x\\ \sin(\theta)&\cos(\theta)&d_y\\0&0&1\end{pmatrix}$$

rotate along

O'(4,5)

60°



$$\begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(60) & -\sin 60 & 0 \\ \sin(60) & \cos 60 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{bmatrix}$$

$P_1'(7.23, 6.6)$

$P_2'(4.6, 8.1)$

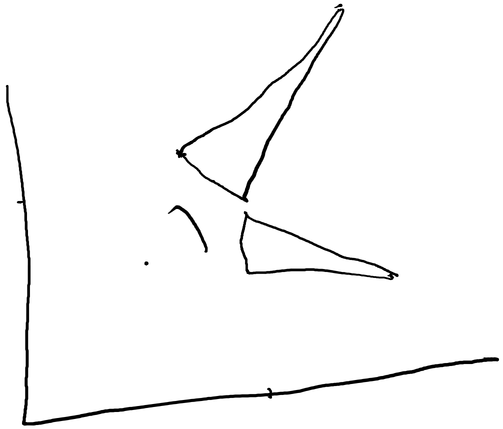
$P_3'(9.7, 10.9)$

$$\begin{bmatrix} 0.5000 & -0.8660 & 6.3300 \\ 0.8660 & 0.5000 & -0.9640 \\ 0 & 0 & 1 \end{bmatrix}$$

$P_1'(7, 7)$

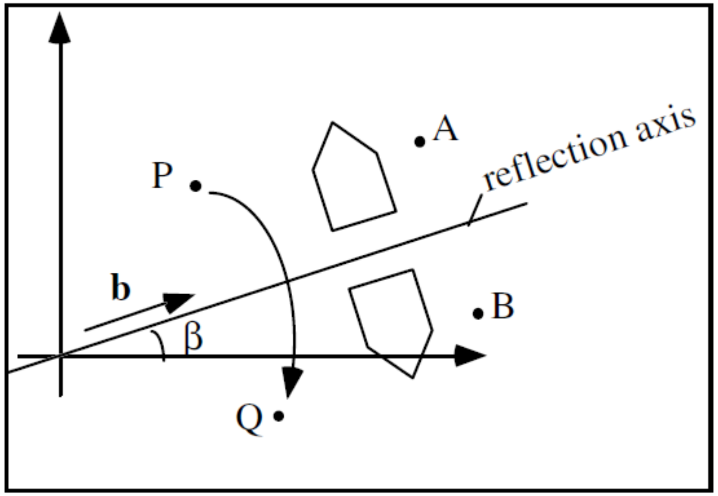
$P_2'(5, 8)$

$P_3'(10, 11)$



Reflections about a tilted line

- A rotation through angle -β (so the axis coincides with the *x*-axis);
- A reflection about the *x*-axis;
- A rotation back through β that "restores" the axis.



$$\begin{pmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c^2 - s^2 & -2cs & 0 \\ -2cs & s^2 - c^2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Composing Affine Transformations (Examples)

- Reflect across an arbitrary line through the origin $\mathcal{O}$: Q = R(θ) S R(-θ) P
- The rotation transforms the axis to the x-axis, the reflection is a scaling, and the last rotation transforms back to the original axis.
- Window-viewport: Translate by -w.l, -w.b, scale by A, B, translate by v.l, v.b.

Properties of 2D and 3D Affine Transformations

- Affine transformations *preserve* affine combinations of points.
  - $W = a_1P_1 + a_2P_2$ is an affine combination.
  - $MW = a_1MP_1 + a_2MP_2$
- Affine transformations preserve lines and planes.
  - A line through A and B is $L(t) = (1-t)A + tB$, an affine combination of points.
  - A plane can also be written as an affine combination of points: $P(s, a) = sA + tB +(1 – s – t)C$.

Properties of Transformations (2)

- Every affine transformation is composed of elementary operations.
- A matrix may be factored into a product of elementary matrices in various ways. One particular way of factoring the matrix associated with a 2D affine transformation yields

  M = (shear)(scaling)(rotation)(translation)
- That is, any 3 x 3 matrix that represents a 2D affine transformation can be written as the product of (reading right to left) a translation matrix, a rotation matrix, a scaling matrix, and a shear matrix.