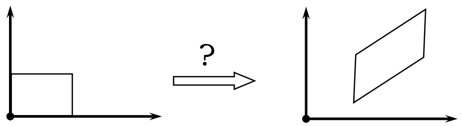
Geometric Transformations

Geometric Transformations

- Why do we need them?
  - Want to define an object in one coordinate system, then place it in another system.
  - Allow us to create multiple instances of objects.
  - Animation (time-dependent transformations).
  - Display using device independent coordinates.
  - 3D viewing (projections).

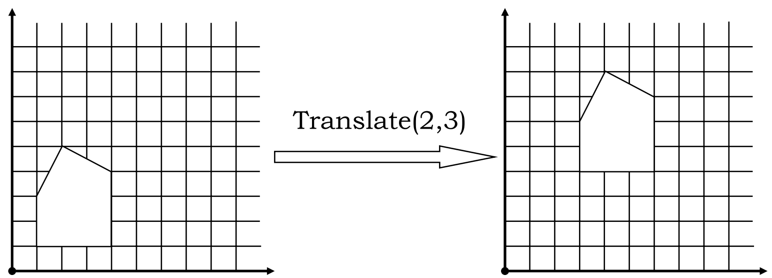
Transformations in 2D

- Reminder: we represent a geometric object as a set of points:
  - Boundary representation: the points form the boundary of the object.
  - Solid representation: the points form the interior of the object.
- Question: how can we transform a geometric object in the plane?



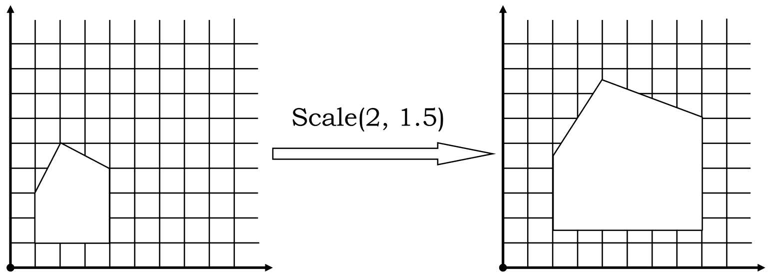
Translation

- Translate(a,b): $(x, y) \rightarrow (x + a, y + b)$



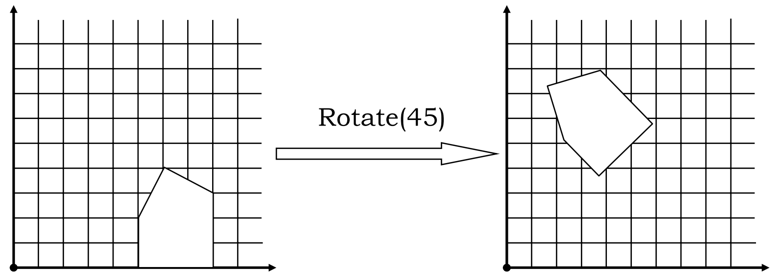
Scaling

- Scale(a,b): $(x, y) \rightarrow (ax, by)$



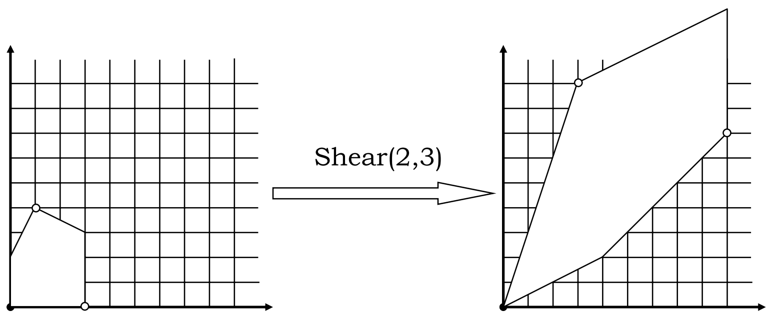
Rotation

- Rotate(θ): $(x, y) \rightarrow (x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta)$



Shearing

- Shear(a,b): $(x, y) \rightarrow (x + ay, bx + y)$



Matrix Notation

- Let's write a point *(x,y)* as a column vector of length 2: $\begin{bmatrix} x \\ y \end{bmatrix}$
- What happens when this vector is multiplied by a 2 by 2 matrix?

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}$$

Scaling

- Scale(a,b):

$$\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax \\ by \end{bmatrix}$$

- What happens when *a* or *b* are negative?

Reflection

- reflection through the *y* axis: $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$
- reflection through the *x* axis: $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
- reflection through *y = x*: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
- reflection through *y = -x*: $\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$

Rotation, Shearing

- Rotate(θ):

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

- Shear(a,b):

$$\begin{bmatrix} 1 & a \\ b & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + ay \\ bx + y \end{bmatrix}$$

Combined Transformations

- A sequence of transformations can be collapsed into a single matrix using matrix multiplication:

$$T_1 T_2 T_3 \begin{bmatrix} x \\ y \end{bmatrix} = T_{1,2,3} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Is the order of transformations important?

Translation

- Translate(*a,b*): $\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x + a \\ y + b \end{bmatrix}$
- Problem: cannot represent translation using 2 by 2 matrices!
- Solution: *homogeneous coordinates* - use a 3 by 3 linear transformation in a special space: the *projective plane*.

Homogeneous Coordinates

- A point in the projective plane P^2 is represented by 3 coordinates, at least one of which is non-zero.
- Two 3-vectors a,b represent the same point in P^2 iff a = hb, where h is a non-zero scalar.
- A 2D point (x,y) in the Euclidean plane corresponds to the 3-vectors (hx,hy,h) in P^2, such as (x,y,1). Note: this is a one-to-many correspondence!
- Geometric interpretation: each point (*x,y*) corresponds to a ray in 3D, from the origin (0,0,0) through the point (*x,y,1*)

Translation

- Translate(*a,b*):

$$\begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + a \\ y + b \\ 1 \end{bmatrix}$$

Homogeneous Matrices

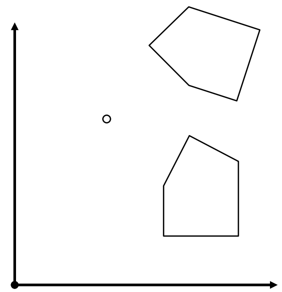
- All of the 2D transformations we have seen so far can now be written as follows:

$$\begin{bmatrix} a & b & m \\ c & d & n \\ 0 & 0 & 1 \end{bmatrix}$$

- What happens when last row is not [0,0,1]?

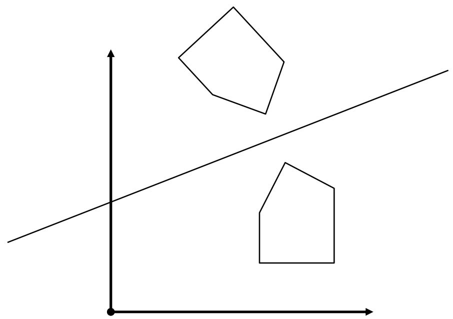
Example 1

- Rotation about an arbitrary point.



Example 2

- Reflection through an arbitrary line



Affine Transformations: Definition

- Let T : $A_1 \to A_2$, where A_1 and A_2 are affine spaces.
- Then T is said to be an affine transformation if:
  - T maps vectors to vectors and points to points
  - T is a linear transformation on vectors
  - T(p + u) = T(p) + T(u)

Affine Transformations: Properties

- Affine transformations preserve affine combinations of points. In other words, given an affine transformation T and a point p:

$$\mathbf{p} = \alpha_1 \mathbf{p}_1 + \cdots + \alpha_k \mathbf{p}_k$$

it holds that: $T(\mathbf{p}) = \alpha_1 T(\mathbf{p}_1) + \cdots + \alpha_k T(\mathbf{p}_k)$

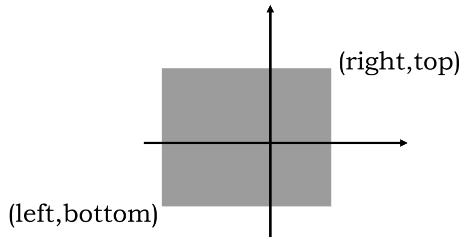
- Parallel lines are preserved.
- Intersections between lines are preserved.

2D Transformations (summary)

- Translation, Rotation, Scaling, Reflection, Shearing
- Rigid-body transformations: preserve angles and lengths
- Affine transformations: preserve parallel lines, but not lengths or angles.

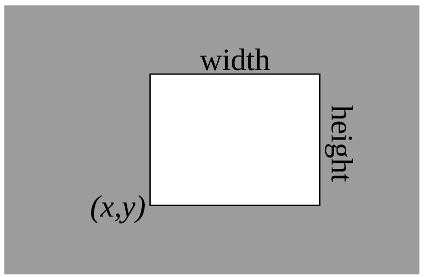
Viewing in 2D

- Objects are given in terms of application dependent *world coordinates (WC)*
- The world is viewed through a WC window:
  gluOrtho2D(left, right, bottom, up)



The Viewport

- The WC window is mapped onto a *device coordinate (DC)* viewport:
  glViewport(x, y, width, height)



2D Viewing Transformation

- Translate WC window to origin:
  Translate(-left, -bottom)
- Scale WC window to match viewport size:
  Scale(width/(right - left), height/(top - bottom))
- Translate to position viewport: Translate(x,y)