Reprsentations and 2d Transformations

Overview

- Coordinate Representations
- 2 Dimensional Geometric Transformations
  - Translation
  - Rotation
  - Scaling
- Homogeneous Coordinates
- Reflection
- Shearing

Coordinate Reprsentations

- A **Cartesian coordinate system** specifies each point uniquely in a plane by a pair of numerical **coordinates**, which are the signed distances from the point .
- Graphics package are designed to use with Cartesian coordinate specifications.
- Several different Cartesian reference frames are used to construct and display a scene.
- The geometric part of the rendering process is that it consists of the application of a series of coordinate transformations that takes an object database through a series of coordinate systems.

Coordinate Representations

- *Local or Modelling Coordinate system* :For ease of modeling store the vertices of an object with respect to some point conveniently located in or near the object.
- Ex:Construct the individual objects such as trees or furniture in a scene within separate coordinate reference frames .
- Once an object has been modeled, the next stage is to place it in the scene that we wish to render
- The global coordinate system of the scene is known as the *world coordinate system*

Coordinate Representations

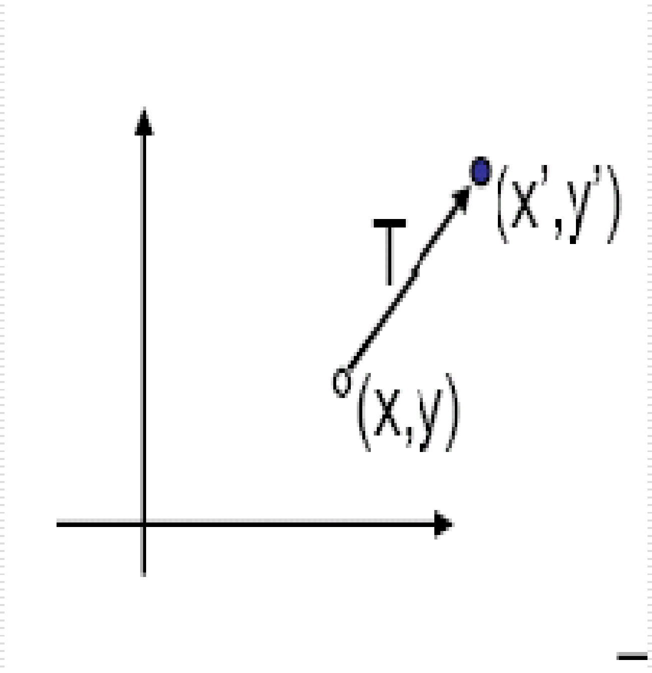
- The world coordinate description of the scene is transferred to one or more output device reference frames for display called device coordinates.
- Modeling and world coordinate definitions allow us to any convenient dimensions.
- Graphics system first converts the world coordinate positions to normalized device coordinates in the range of 0 to 1 before final conversion to specific device coordinates.
- $(x_{mc},y_{mc}) \rightarrow (x_{wc},y_{wc}) \rightarrow (x_{nc},y_{nc}) \rightarrow (x_{dc},y_{dc})$

Geometric Transformation Definition

- In many applications there is need for altering and manipulating displays.
- Geometric Transformations: Operations that are applied to change the geometric description of an object by changing its position, orientation, or size.
- The basic transformations are translation, rotation and scaling.

TRANSLATION

- Translation is applied to an object by repositioning it along a line path from one coordinate location to another.
- Translate a two dimensional point by adding translation distances t_x , t_y
  - $x' = x + t_x$, $y' = y + t_y$
- The translation distance pair (tx, ty) is the translation vector or shift vector.



TRANSLATION

- Translation equations can be expressed as a single matrix equation

- $$P = \begin{bmatrix} x \\ y \end{bmatrix} \quad P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad T = \begin{bmatrix} tx \\ ty \end{bmatrix}$$

- 2D translation equation

$$P' = P + T$$

TRANSLATION

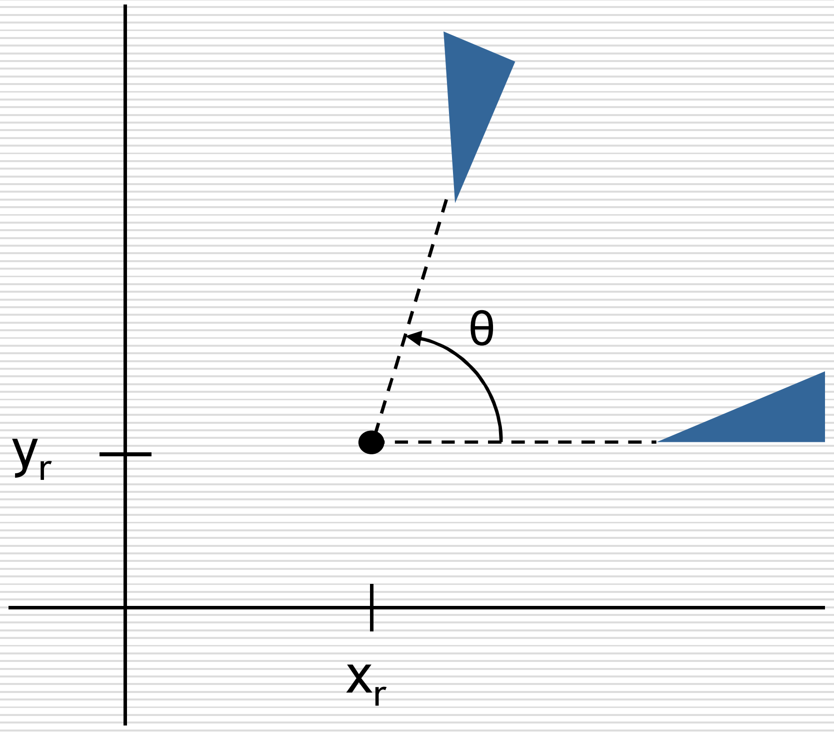
- Rigid body transformation → moves object without deformation
- Every point is translated by the same amount
- Straight line segment is translated by applying the transformations to each of the line endpoints and redrawing the line between the new endpoint positions.
- A triangle with position (10,2), (20,2) and (15,5) is translated with the translation vector (-5.5,3.75). Determine the new positions of the triangle.

2D ROTATION

- A rotation transformation of an object is generated by specifying a rotation axis and a rotation angle.
- All points of the object are then transformed to new positions by rotating the points through the specified angle about the rotation axis.
- 2D rotation is obtained by repositioning the object along a circular path in the *xy* plane.

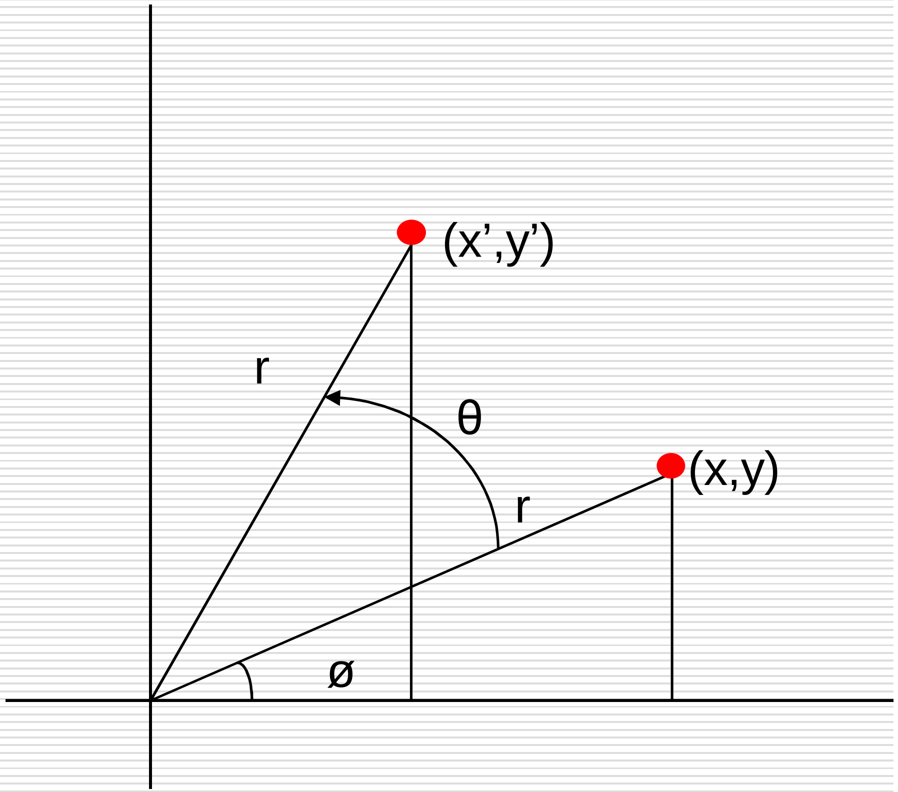
2D ROTATION

- Parameter for 2D rotation:
  - Rotation angle, θ
  - Rotation point (pivot point), (xr, yr)
- Positive θ >> counterclockwise rotation about the pivot point
- Negative θ >> clockwise rotation about the pivot point



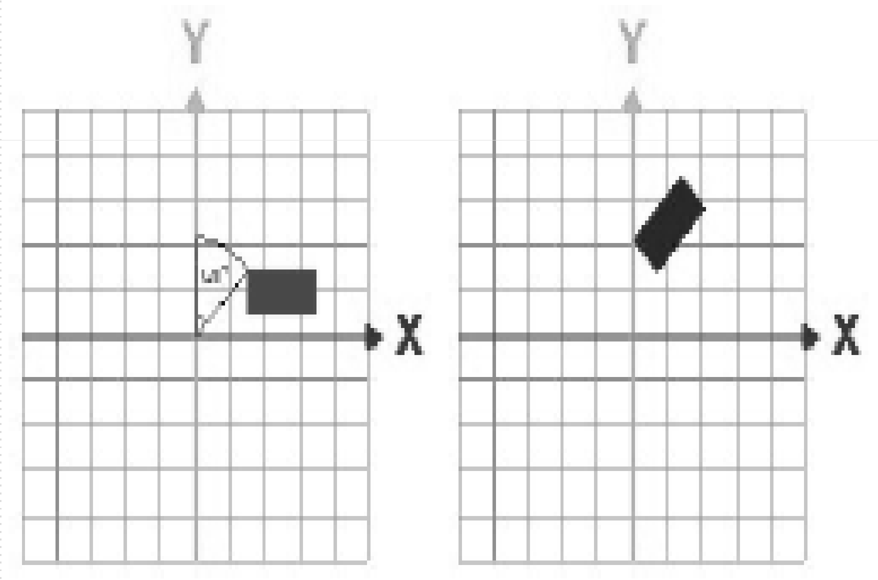
2D Rotation

- Rotation of a point from position (x, y) to position (x', y') through an angle θ relative to the coordinate origin. The original angular displacement of the point from the x axis is Φ



2D ROTATION

- Rotation by 45° counter-clockwise about origin.



2D ROTATION

- Using standard trigonometric identities, transformed coordinates can be expressed in terms of angles θ and Φ as

$$x' = r \cos(\Phi + \theta)$$
$$= r \cos\Phi \cos\theta - r \sin\Phi \sin\theta$$
$$y' = r \sin(\Phi + \theta)$$
$$= r \cos\Phi \sin\theta + r \sin\Phi \cos\theta$$

- The original coordinates of the point in polar coordinates are

$$x = r \cos ø, \qquad y = r \sin ø$$

2D ROTATION

- Substituting expression (5) into (4), we obtain the transformation equations for rotating a point at position (x, y) through an angle θ about the origin:

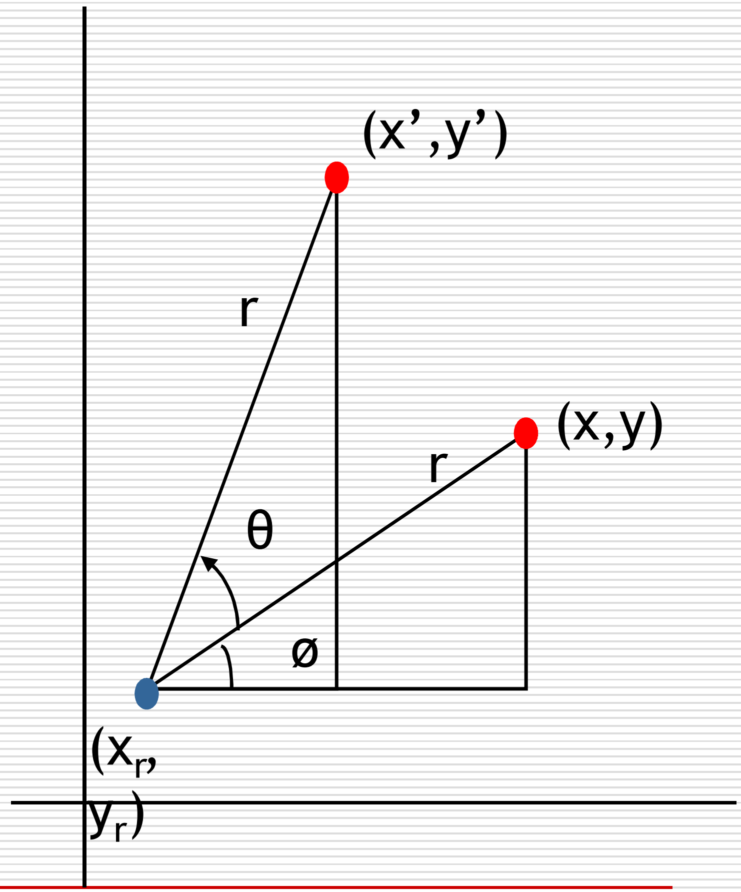
  - $x' = x \cos \theta - y \sin \theta$
  - $y' = x \sin \theta + y \cos \theta$

- Rotation equation in matrix form, P' = R . P where the rotation matrix is

$$R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

2D Rotation

- Rotation of a point about an arbitrary pivot point.
- For rotation of a point about any specified rotation position (x_r, y_r):
- $x' = x_r + (x - x_r) \cos\theta - (y - y_r) \sin\theta$
- $y' = y_r + (x - x_r) \sin\theta + (y - y_r) \cos\theta$
- Rotations are also rigid body transformations that move object without deformation
- Every point in the object is rotated through the same angle



2D SCALING

- To change the size of an object.

- A simple operation is by multiplying object positions (x, y) by scaling factors s_x and s_y to produce the transformed coordinates (x',y') :

$$x' = x \, . \, s_x \;,\quad y' = y \, . \, s_y \qquad (10)$$

can also be written in matrix form (11)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

or $$P' = S \, . \, P$$

2D Scaling

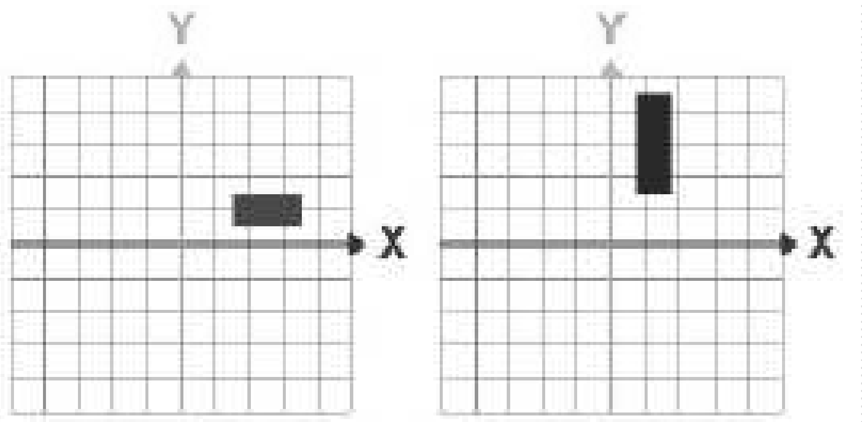
- Any positive value can be assigned to scaling factors s_x and s_y
- Values less than 1 reduce the size and greater than 1 enlarge it.
- Specifying a value of 1 for both s_x and s_y leaves the size unchanged.
- Uniform scaling: maintain relative object proportions (size) when s_x and s_y is assigned same value.

2D Scaling

- Differential scaling: applying unequal values for s_x and s_y.
  - Often use in design applications, where pictures are constructed from few basic shape that can be adjusted by scaling and positioning transformations .
- Objects transformed with Eq. (11) are BOTH scaled and repositioned.
- Scaling factor:
  - |<1| - move objects closer to origin
  - |>1| - move objects farther from the origin

2D Scaling

- Scaling vector : (0.5, 3.0) about origin.

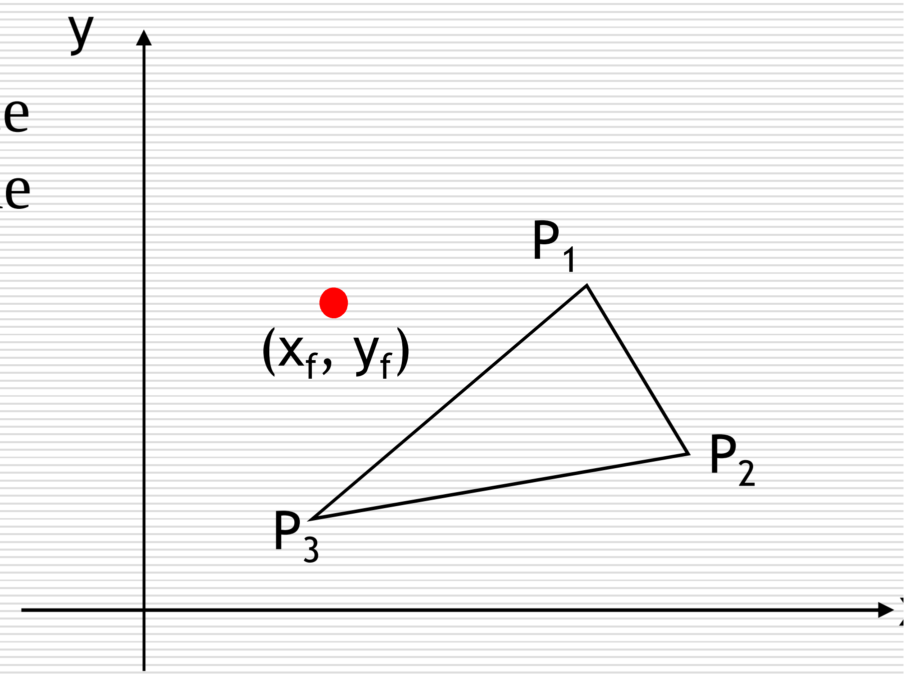


2D SCALING

- The location of the scaled object can be controlled by choosing a position, fixed point, that is to remain unchanged after the transformation.
- The coordinate for fixed point (x_f, y_f) are often chosen at some object position, but any other position can be selected.
- Objects are now resized by scaling the distances between object points and the fixed point.

2D Scaling

- Scaling relative to a chosen fixed point (xf,yf). The distance from each polygon vertex to the fixed point is scaled by transformation equation (13).



$$x' = x \cdot s_x + x_f\,(1 - s_x)$$

$$y' = y \cdot s_y + y_f\,(1 - s_y)$$

HOMOGENEOUS CO-ORDINATES

- Graphics applications involves sequences of geometric transformations.
- Efficient approach is needed to combine the transformations so that the final coordinates are obtained directly.
- Combine the multiplicative and the translational terms for 2d geometric transformations into single matrix multiplication by homogenous coordinates.
- ***Homogeneous coordinates seem unintuitive, but they make graphics operations much easier***
- Represent each 2D coordinate position (x, y) with the homogenous coordinate triple (x_h, y_h, h).

HOMOGENEOUS CO-ORDINATES

- Represent each 2D coordinate position (x, y) with the homogenous coordinate triple (x_h, y_h, h). Where

$$x = \frac{x_h}{h} \qquad y = \frac{y_h}{h} \qquad P = \begin{bmatrix} x_h \\ y_h \\ h \end{bmatrix} = \begin{bmatrix} h \cdot x \\ h \cdot y \\ h \end{bmatrix}$$

- General homogeneous representation can also written as (h.x,h.y,h) set h=1.
- Transformations of translation, scaling and rotation can be represented using Homogeneous coordinates.

Homogeneous Transformation Coordinates

Translation

$$T(t_x, t_y) = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$P' = T(t_x, t_y) \cdot P$$

Rotation

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

$$P' = R(\theta) \cdot P$$

Scaling

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

$$P' = S(s_x, s_y) \cdot P$$

Composite Transformations

- Application of a sequence of transformations to a point:

$$\mathbf{P}' = \mathbf{M}_2 \cdot \mathbf{M}_1 \cdot \mathbf{P}$$
$$= \mathbf{M} \cdot \mathbf{P}$$

- **Composite transformations** is formed by calculating the matrix product of the individual transformations and forming products of transformation matrix.

Composite Transformations-Translation

- First: composition of similar type transformations
- If we apply to successive translations to a point:

$$\mathbf{P}' = \mathbf{T}(t_{2x}, t_{2y}) \cdot \{\mathbf{T}(t_{1x}, t_{1y}) \cdot \mathbf{P}\}$$
$$= \{\mathbf{T}(t_{2x}, t_{2y}) \cdot \mathbf{T}(t_{1x}, t_{1y})\} \cdot \mathbf{P}$$

P AND P' are represented as homogenous coordinate values.

$$T(t_{2x}, t_{2y}) \cdot T(t_{1x}, t_{1y}) = \begin{bmatrix} 1 & 0 & t_{2x} \\ 0 & 1 & t_{2y} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_{1x} \\ 0 & 1 & t_{1y} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_{1x} + t_{2x} \\ 0 & 1 & t_{1y} + t_{2y} \\ 0 & 0 & 1 \end{bmatrix} = T(t_{1x} + t_{2x}, t_{1y} + t_{2y})$$

Successive translations are additive

Composite Transformations-Rotation

Two successive rotations applied to the point p produce the transformed position

P'=R(θ) {R (Φ).P}={R(θ) .R (Φ)}.P

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

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

Two successive rotations are additive.

Composite Transformations-Scaling

Two successive scaling operations produces the following composite scaling matrix

$$\mathbf{S}(s_{2x},s_{2y})\cdot\mathbf{S}(s_{1x},s_{1y})=\begin{bmatrix} s_{2x} & 0 & 0 \\ 0 & s_{2y} & 0 \\ 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} s_{1x} & 0 & 0 \\ 0 & s_{1y} & 0 \\ 0 & 0 & 1 \end{bmatrix}=\begin{bmatrix} s_{1x}\cdot s_{2x} & 0 & 0 \\ 0 & s_{1y}\cdot s_{2y} & 0 \\ 0 & 0 & 1 \end{bmatrix}=\mathbf{S}(s_{1x}\cdot s_{2x},s_{1y}\cdot s_{2y})$$

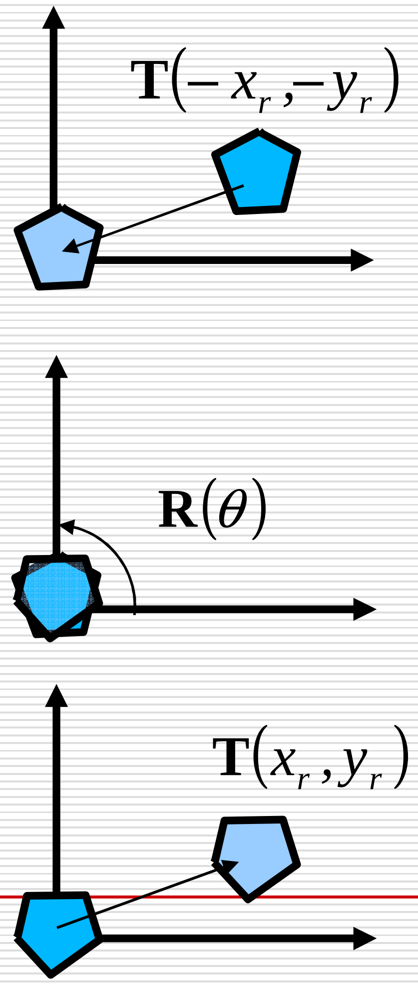
•The resulting matrix indicates the successive operations are multiplicative.

Composite Transformations

- Combining transformations reduces to matrix multiplication, e.g.
  - R(r,) = T(r)* R(θ)* T(-r)
- In general: multiplication of a 3x3 with another 3x3 matrix requires 3*3*3 = 27 multiplications and 2*3*3 additions.
- In 2D transformations, the third row of the matrices is always [0 0 1] and should never be calculated.
- In addition, in homogeneous coordinates the third component of the vectors is always one: (x,y,1).
- Composite converts all to matrix multiplications thus improving computational efficiency

Rotation around a pivot point

- Rotations about any selected pivot point (x_r, y_r) by performing the following sequence:
  - Translate the object so that the pivot point moves to the origin
  - Rotate around origin
  - Translate the object so that the pivot point is back to its original position



$$\mathbf{T}(x_r, y_r)\cdot\mathbf{R}(\theta)\cdot\mathbf{T}(-x_r, -y_r) =$$

$$\begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos\theta & -\sin\theta & x_r(1-\cos\theta)+y_r\sin\theta \\ \sin\theta & \cos\theta & y_r(1-\cos\theta)-x_r\sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

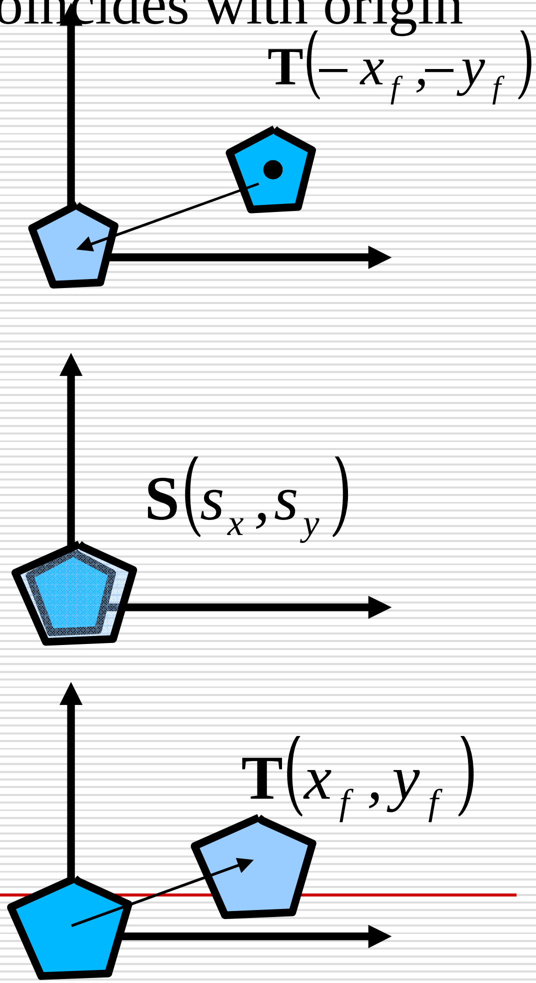
Scaling with respect to a Fixed Point

- Translate object to origin so fixed point coincides with origin
- Scale the object with respect to origin
- Translate back by inverse translation.

$$\mathbf{T}(x_f, y_f) \cdot \mathbf{S}(s_x, s_y) \cdot \mathbf{T}(-x_f, -y_f) =$$

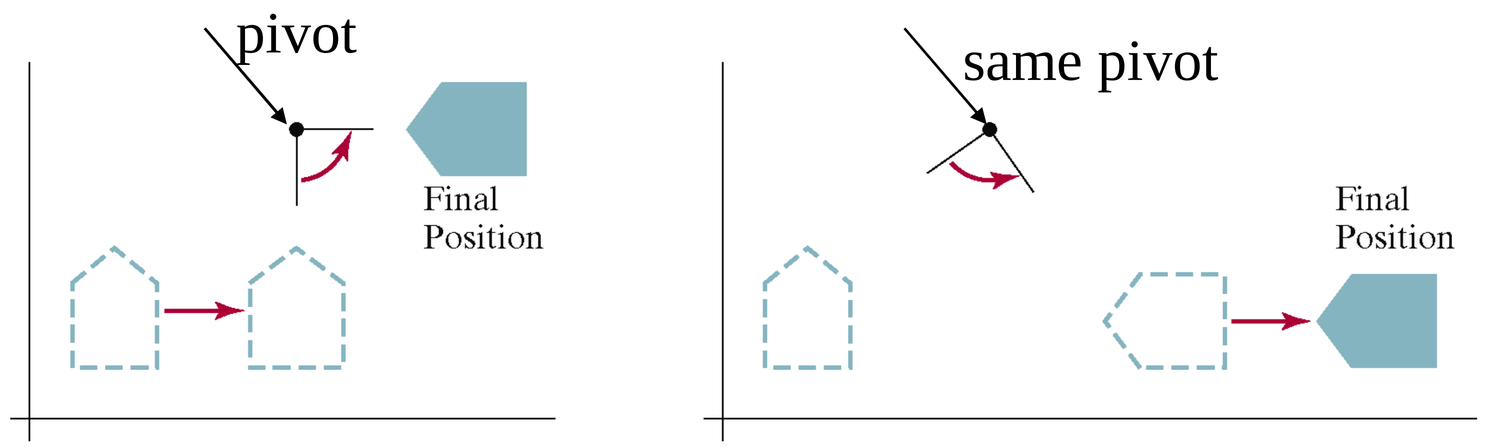
$$\begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} s_x & 0 & x_f(1-s_x) \\ 0 & s_y & y_f(1-s_y) \\ 0 & 0 & 1 \end{bmatrix}$$



Concatenation Properties

- Matrix multiplication is associative, evaluate matrix products using left-to-right or right-to-left associative grouping.
- Matrix composition is not commutative. So careful when applying a sequence of transformations.
- Reversing the order in which the sequence of transformations is performed may affect the transformed position of an object.



REFLECTION

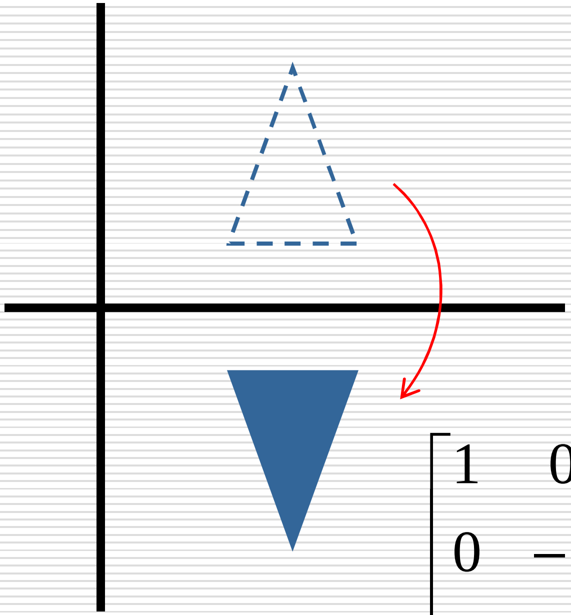
- A transformation that produces a mirror image of an object
- Image is generated relative to an axis of reflection by rotating the object 180° about the reflection axis
  - *Reflection axis is xy* plane – rotation path about the axis is in the plane perpendicular to *xy* plane
  - Reflection axis perpendicular to *xy* plane – rotation path is in the *xy* plane

2D Reflection

x-axis

Reflection about the line

y=0



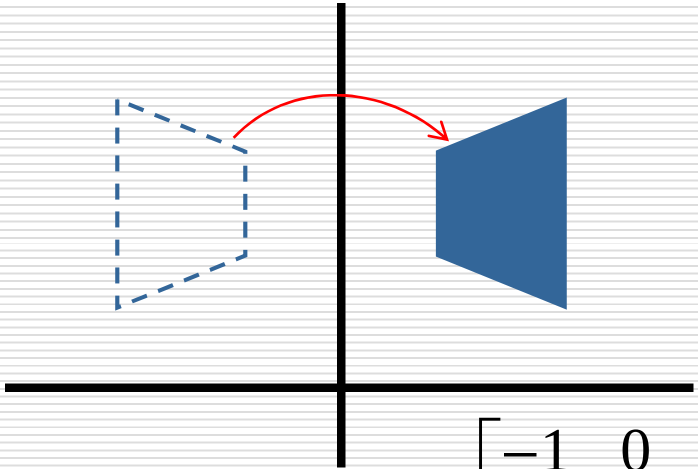
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformation keeps x values but flips the y values

y-axis

Reflection about the line

x=0

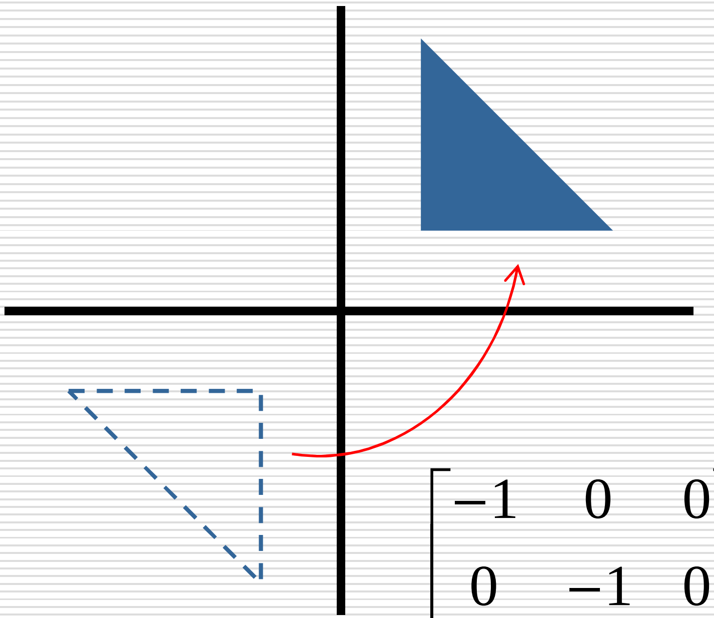


$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformation keeps y values but flips the x values

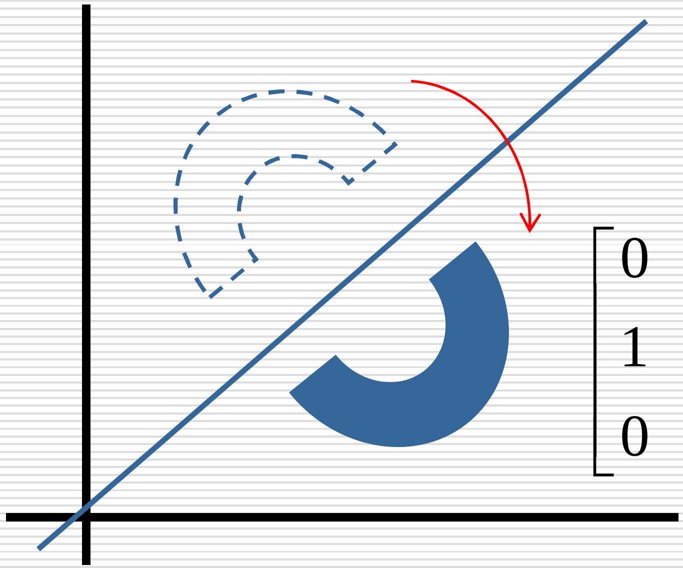
2D Reflection

Reflection relative to the coordinate origin



$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Reflection axis as the diagonal line x=y



$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformation flips both x values and y values by Reflecting relative to the coordinate origin

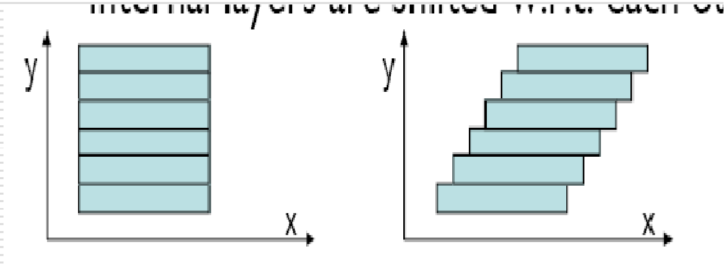
2D Reflection

- Elements of the reflection matrix can be set to values other than ±1.

- Reflection parameter:
  - >1 – shifts the mirror image of a point farther from the reflection axis.
  - <1 – brings the mirror image of a point closer to the reflection axis.

2D Shear

- Transformation that distort the shape of an object.



- Slide to another shape
- Internal layers are shifted w.r.t. each other
- 2 common shearing transformation
  - Shift coordinate *x* values
  - Shift coordinate *y* values

2D Shear

- An x-direction shear relative to the x axis is produced with the transformation matrix

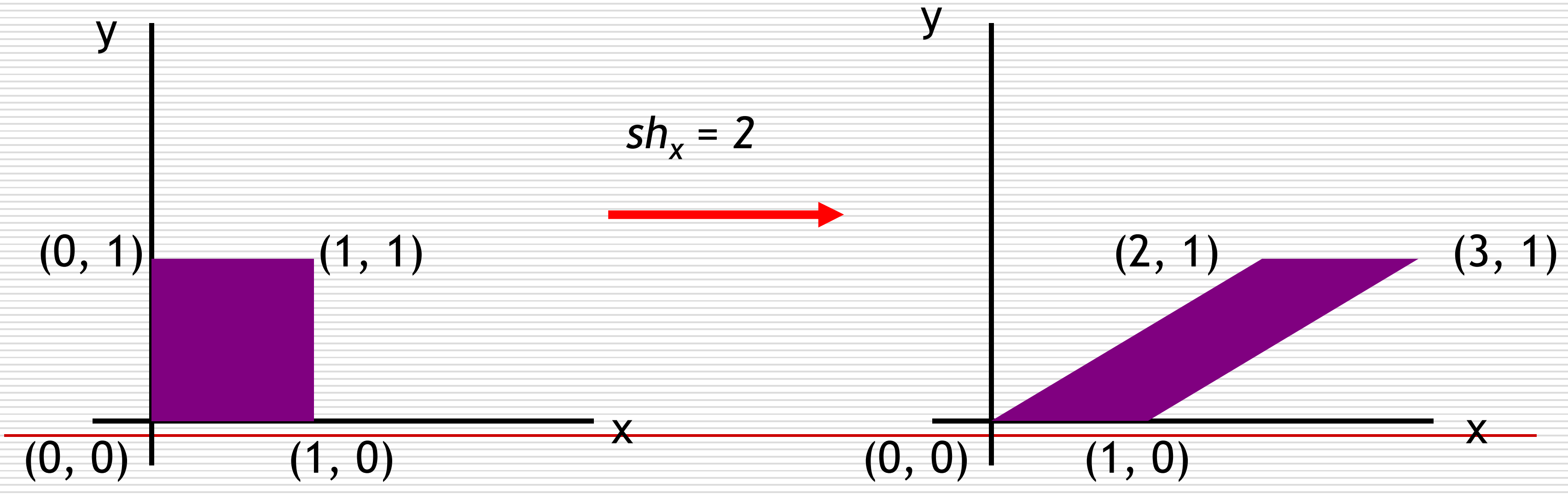
$$\begin{pmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

which transforms coordinate positions as

$$x' = x + sh_x \cdot y, \qquad y' = y$$

2D Shear

- Any real number can be assigned to the shear parameter sh_x.
- A coordinate position (x, y) is then shifted horizontally by an amount proportional to its perpendicular distance (y value) from the x axis.



2D Shear

- We can generate *x*-direction shears relative to other reference lines with

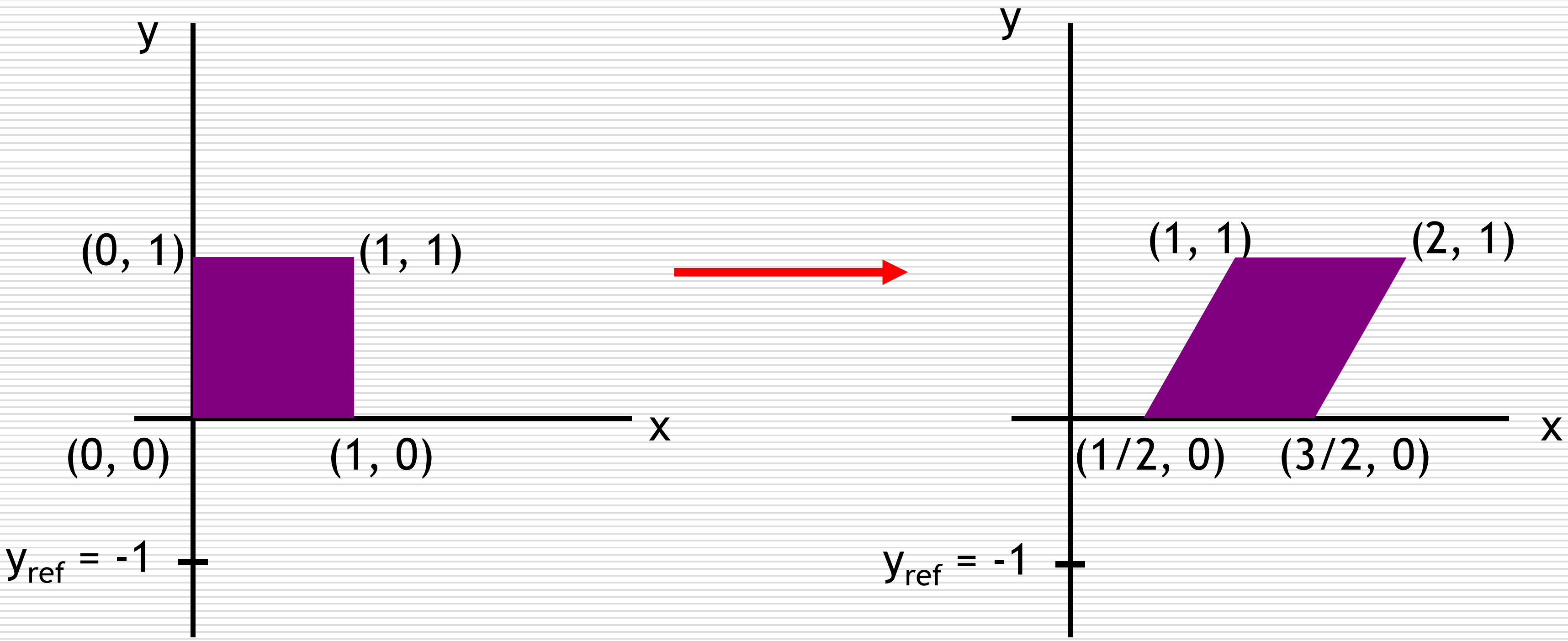
$$\begin{pmatrix} 1 & sh_x & -sh_x \cdot y_{ref} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Now, coordinate positions are transformed as

$$x' = x + sh_x(y - y_{ref}) \,, \qquad y' = y$$

EXAMPLE

- $sh_x=0.5$ and $y_{ref} = -1$



2D Shear

- A *y*-direction shears relative to other reference lines can generate with

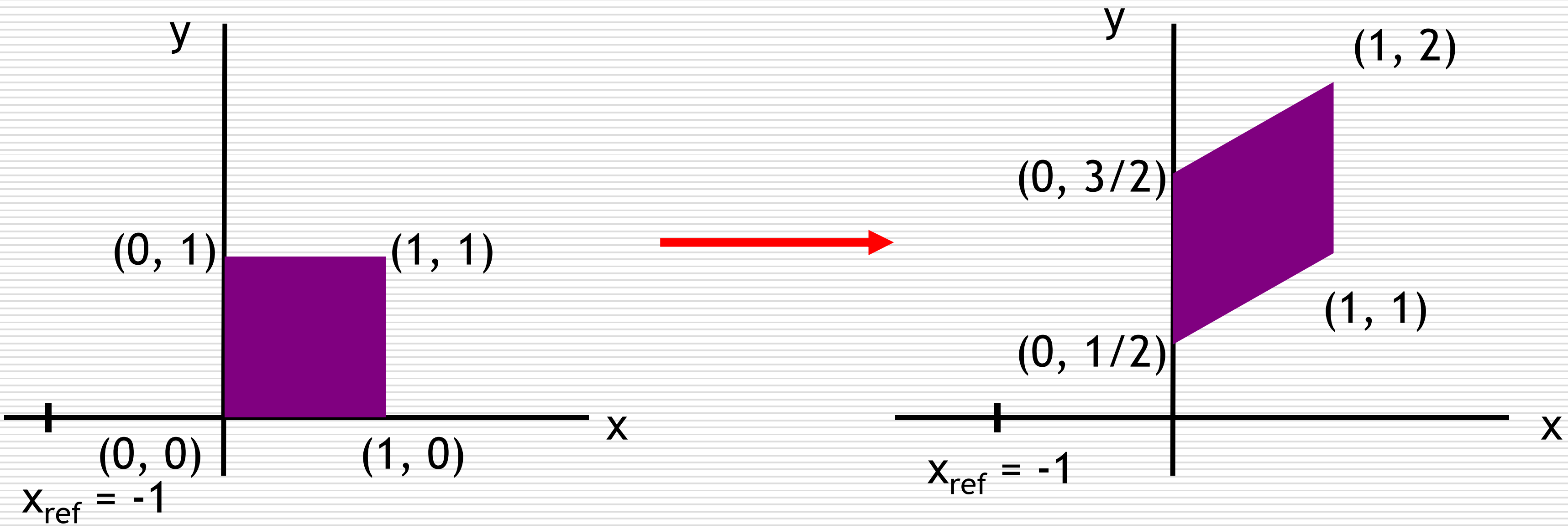
$$\begin{pmatrix} 1 & 0 & 0 \\ sh_y & 1 & -sh_y \cdot x_{ref} \\ 0 & 0 & 1 \end{pmatrix}$$

Now, coordinate positions are transformed as

$$x' = x, \qquad y' = y + sh_y(x - x_{ref})$$

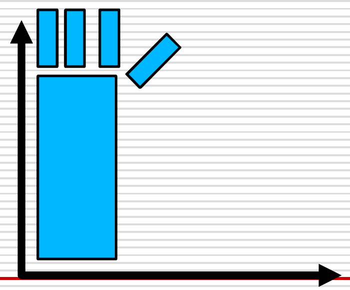
EXAMPLE

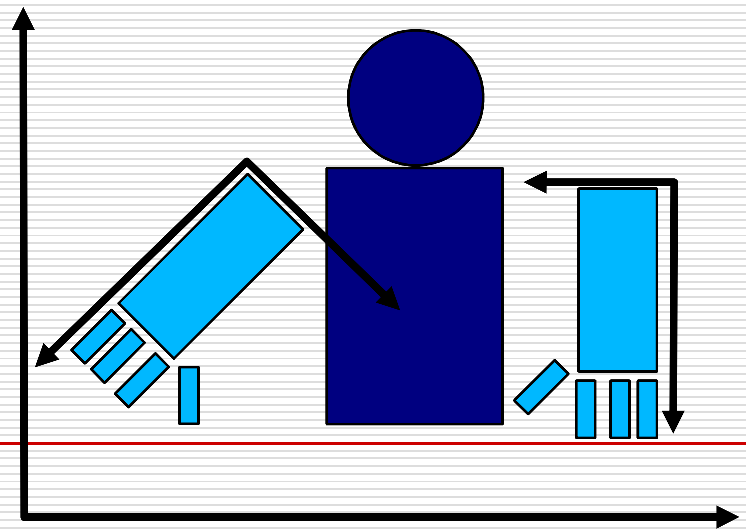
- $sh_y = 0.5$ and $x_{ref} = -1$



Transformations Between the Coordinate Systems

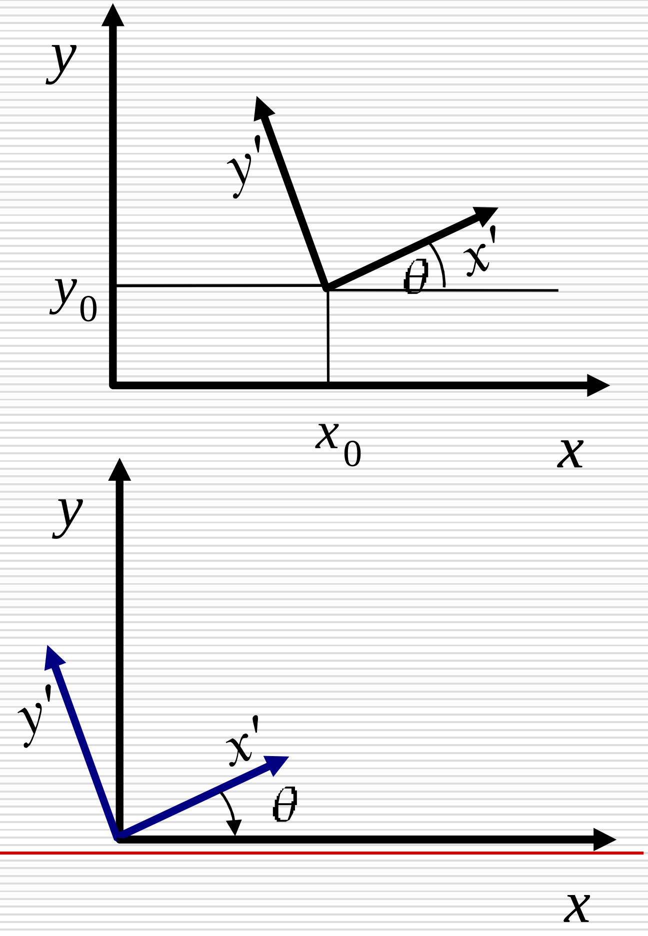
- Between different systems: Polar coordinates to cartesian coordinates
- Between two cartesian coordinate systems. For example, relative coordinates or window to viewport transformation.





Transformations Between the Coordinate Systems

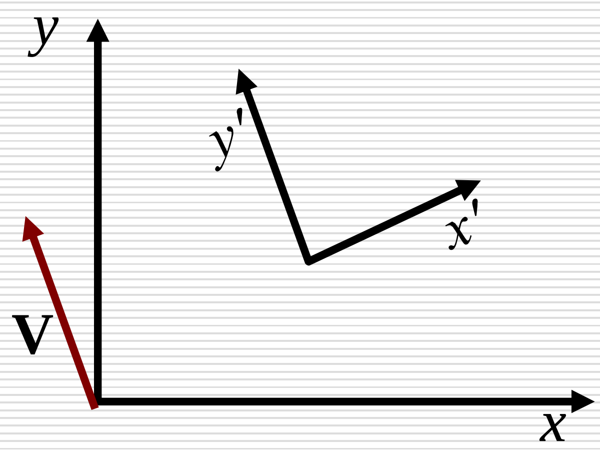
- How to transform from *x,y* to *x',y'* ?
- *Superimpose x',y' to x,y*
- *Transformation:*
  - *Translate so that* (x_0, y_0) *moves to (0,0) of x,y*
  - *Rotate x' axis onto x axis*



$$R(-\theta)\cdot T(-x_{0,}-y_0)$$

Transformations Between the Coordinate Systems

- Alternate method for rotation: Specify a vector **V** for positive y' axis:



unit vector in the y' direction :

$$\mathbf{v} = \frac{\mathbf{V}}{|\mathbf{V}|} = (v_x, v_y)$$

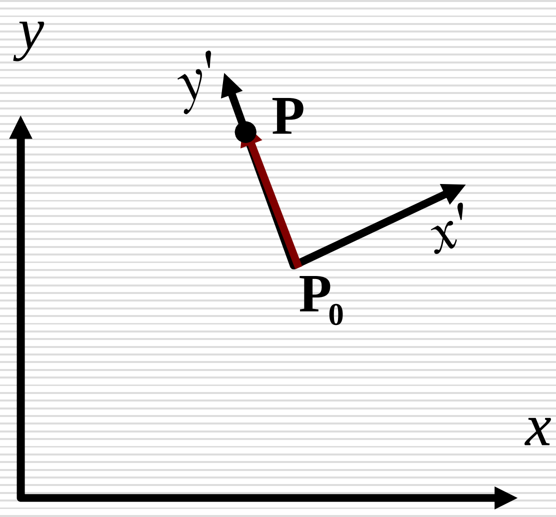
unit vector in the x' direction, rotate $\mathbf{v}$ clockwise 90°

$$\mathbf{u} = (v_y, -v_x) = (u_x, u_y)$$

Transformations Between the Coordinate Systems

- Elements of any rotation matrix can be expressed as elements of a set of orthogonal unit vectors:

$$\mathbf{R} = \begin{bmatrix} u_x & u_y & 0 \\ v_x & v_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} v_y & -v_x & 0 \\ v_x & v_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Choose the directions for v relative to position P0 .

The components of v calculated as

$$\mathbf{V} = \frac{\mathbf{P} - \mathbf{P_0}}{|\mathbf{P} - \mathbf{P_0}|}$$

U is obtained as perpendicular to v

Affine Transformations

- An affine transformation is an important class of linear 2-D geometric transformations which maps variables (*e.g.* pixel **intensity values** located at position (x,y) in an input image) into new variables (*e.g.* in an output image (x',y') by applying a linear combination **of translation, rotation, scaling** and/or shearing (*i.e.* non-uniform scaling in some directions) operations.
- Coordinate transformations of the form:

$$x' = a_{xx}x + a_{xy}y + b_x$$

$$y' = a_{yx}x + a_{yy}y + b_y$$

- Translation, rotation, scaling, reflection, shear. Any affine transformation can be expressed as the combination of these.

- Thank you