Section – III:

TRANSFORMATIONS

in 2-D

2D TRANSFORMATIONS AND MATRICES

Representation of Points:

2 x 1 matrix: $\begin{bmatrix} X \\ Y \end{bmatrix}$

General Problem: [B] = [T] [A]

[T] represents a generic operator to be applied to the points in A. T is the geometric transformation matrix.

If A & T are known, the transformed points are obtained by calculating B.

General Transformation of 2D points:

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

$$x' = ax + cy$$

$$y' = bx + dy$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix}^T = \begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$x' = ax + cy$$

$$y' = bx + dy$$

Solid body transformations – the above equation is valid for all set of points and lines of the object being transformed.

Special cases of 2D Transformations:

1) T = identity matrix:

a=d=1, b=c=0 => x'=x, y'=y

2) *Scaling & Reflections*:

b=0, c=0 => x' = a.x, y' = d.y;

This is scaling by a in x, d in y.

If, a = d > 1, we have enlargement;

If, 0 < a = d < 1, we have compression;

If a = d, we have uniform scaling, else non-uniform scaling.

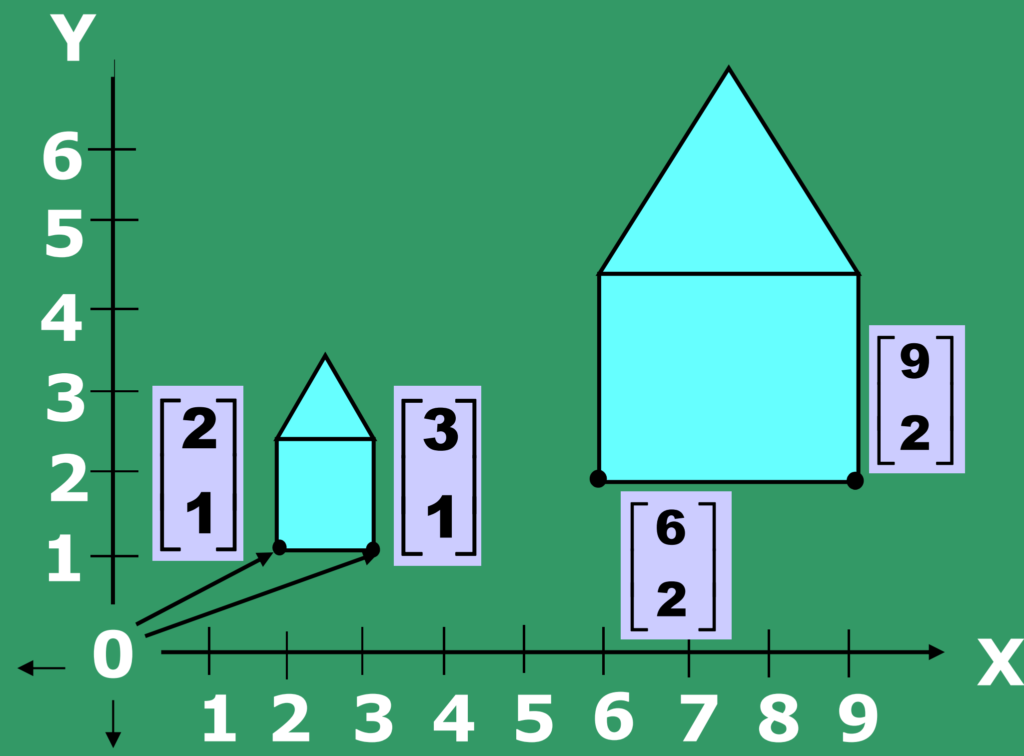
Scale matrix: let S_x = a, S_y = d:

$$\begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

Example of Scaling

$S_x = 3$

$S_y = 2$



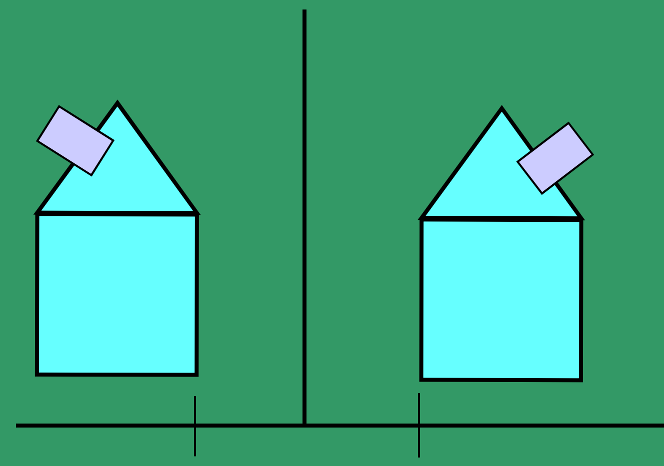
What if S_x and/ or $S_y < 0$ (are negative)?
Get reflections through an axis or plane.

Only diagonal terms are involved in scaling and reflections.

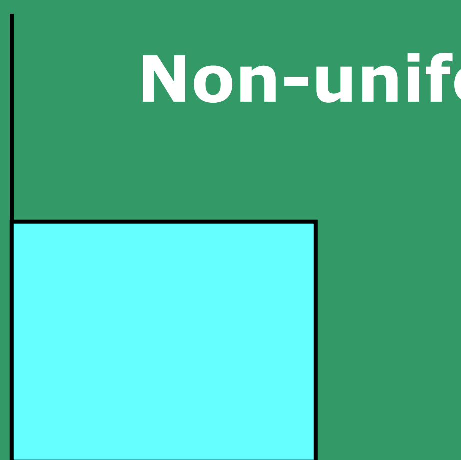
Note : House shifts position relative to origin

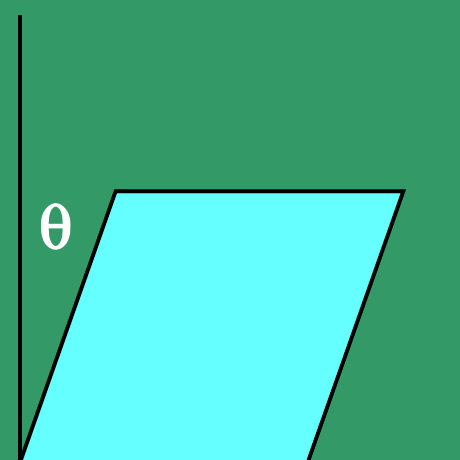
More examples of Scaling and reflection

Reflection (about the Y-axis)



Non-uniform scaling





Special cases of Reflections (|T| = -1)

Matrix T	Reflection about
$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	Y=0 Axis (or X-axis)
$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	X=0 Axis (or Y-axis)
$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	Y = X Axis
$\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$	Y = -X Axis

Off diagonal terms are involved in SHEARING;

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

$$x' = ax + cy$$

$$y' = bx + dy$$

a = d = 1;

let, c = 0, b = 2

x' = x
y' = 2x + y ;

y' depends linearly on x ; This effect is called shear.

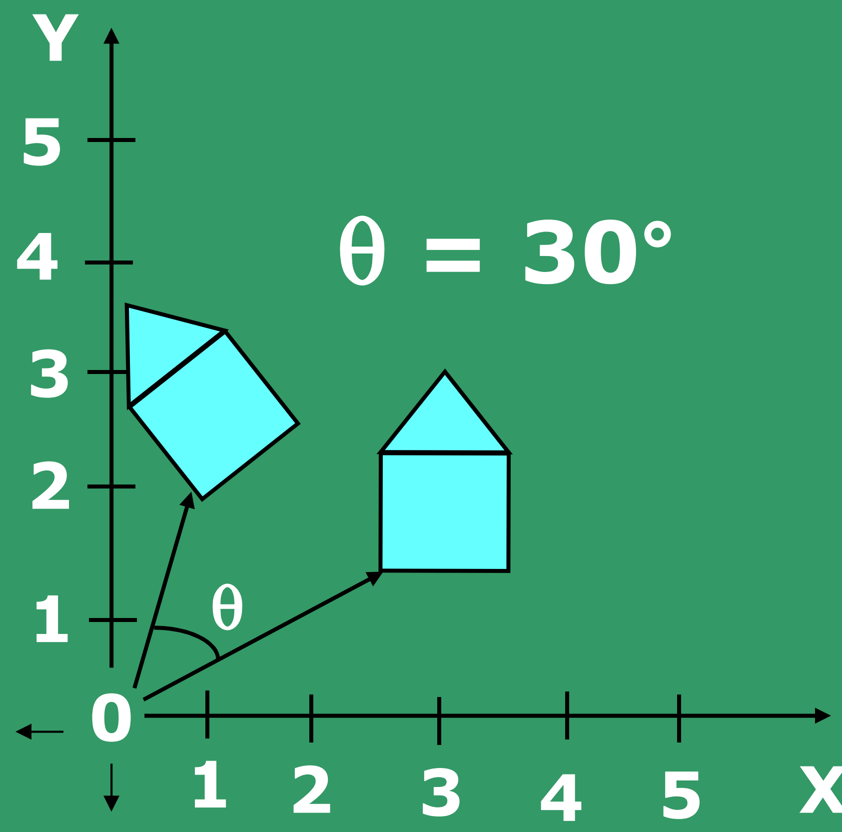
Similarly for b=0, c not equal to zero. The shear in this case is proportional to y-coordinate.

ROTATION

$$X' = x\cos(\theta) - y\sin(\theta)$$
$$Y' = x\sin(\theta) + y\cos(\theta)$$

In matrix form, this is :

$$T = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$



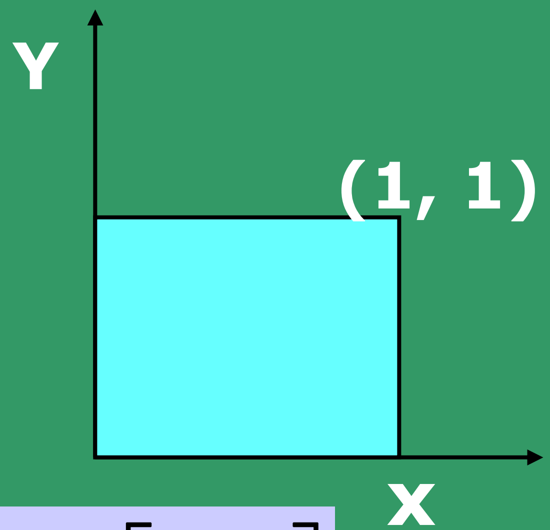
Positive Rotations: counter clockwise about the origin

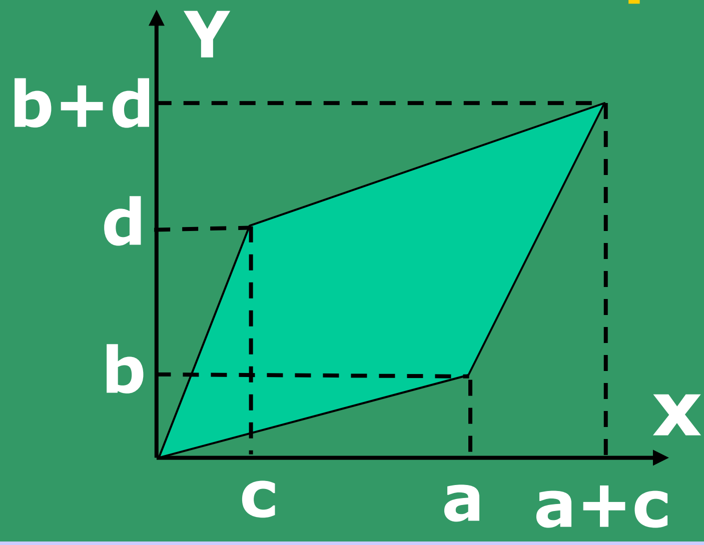
For rotations, $|T| = 1$ and $[T]^T = [T]^{-1}$. Rotation matrices are orthogonal.

Special cases of Rotations

θ (in degrees)	Matrix T
90	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
180	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
270 or -90	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
360 or 0	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Example - Transformation of a Unit Square





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

$$S' = S\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ a & b \\ a+c & b+d \\ c & d \end{bmatrix}$$

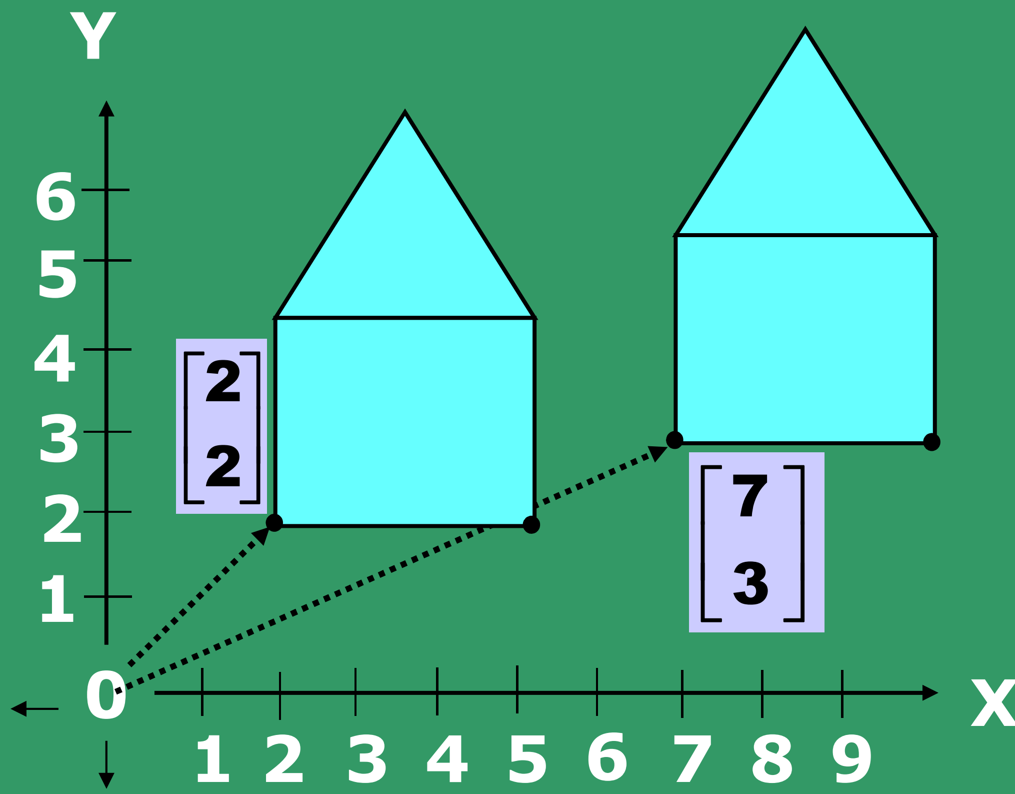
Area of the unit square after transformation

Extend this idea for any arbitrary area.

Translations

$t_x = 5$

$t_y = 1$



Translations

$$B = A + T_d\text{, where } T_d = [t_x \; t_y]^T$$

Where else are translations introduced?

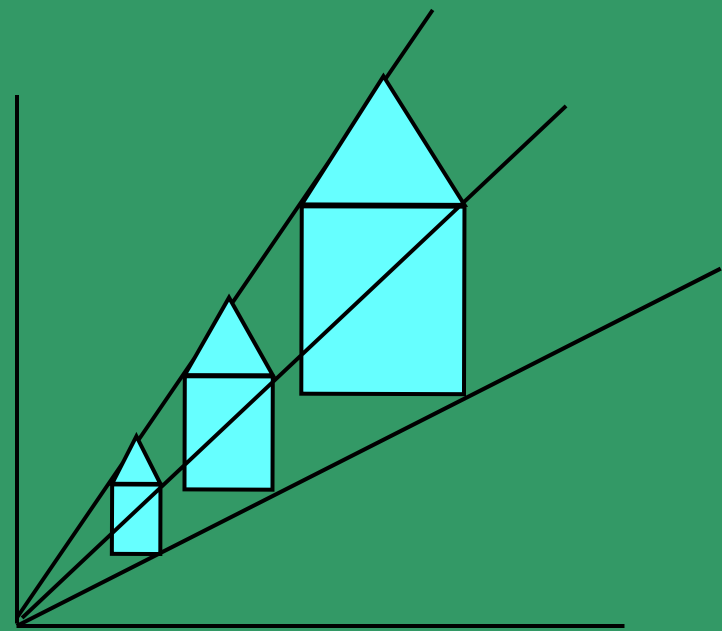
1) Rotations - when objects are not centered at the origin.

2) Scaling - when objects/lines are not centered at the origin - if line intersects the origin, no translation.

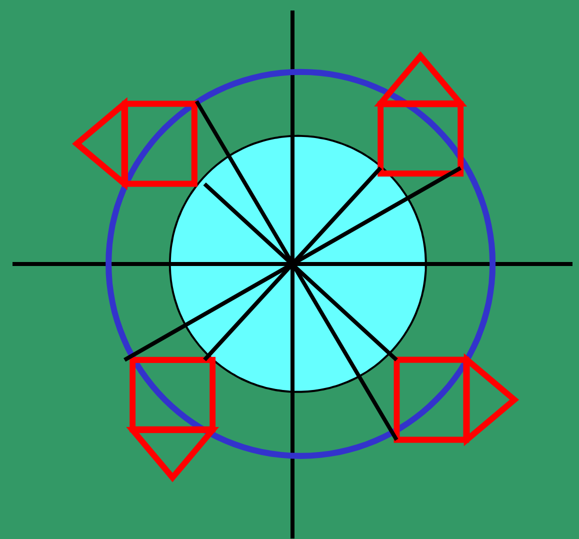
Origin is invariant to Scaling, reflection and Shear – not translation.

Note: we cannot directly represent translations as matrix multiplication, as we can for:

SCALING



ROTATION



Can we represent translations in our general transformation matrix?

Yes, by using homogeneous coordinates

HOMOGENEOUS COORDINATES

Use a 3 x 3 matrix:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & c & t_x \\ b & d & t_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

We have:

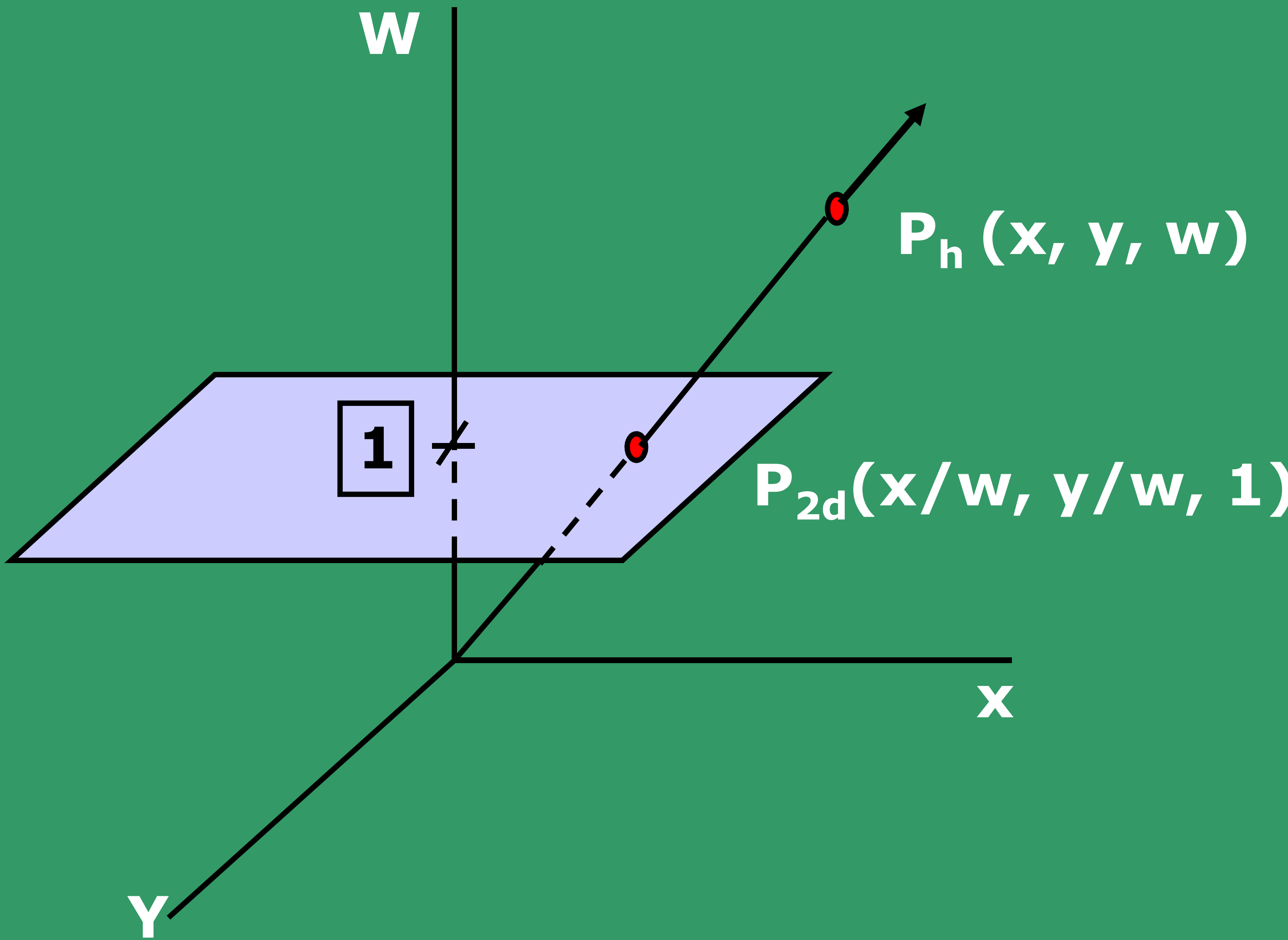
$x' = ax + cy + t_x$

$y' = bx + cy + t_y$

Each point is now represented by a triplet: (x, y, w).

(x/w, y/w) are called the Cartesian coordinates of the homogeneous points.

Interpretation of Homogeneous Coordinates



Two homogeneous coordinates (x_1, y_1, w_1) & (x_2, y_2, w_2) may represent the same point, iff they are multiples of one another: say, (1,2,3) & (3,6,9).

There is no unique homogeneous representation of a point.

All triples of the form (t.x, t.y, t.W) form a line in x,y,W space.

Cartesian coordinates are just the plane w=1 in this space.

W=0, are the points at infinity

General Purpose 2D transformations in homogeneous coordinate representation

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

Parameters involved in scaling, rotation, reflection and shear are: **a, b, c, d**

If B = T.A, then

Translation parameters: (p, q)

If B = A.T, then

Translation parameters: (m, n)

What about S ?

COMPOSITE TRANSFORMATIONS

If we want to apply a series of transformations T_1, T_2, T_3 to a set of points, We can do it in two ways:

1) **We can calculate $p'=T_1*p$, $p''= T_2*p'$, $p'''=T_3*p''$**
2) **Calculate $T= T_1*T_2*T_3$, then $p'''= T*p$.**

Method 2, saves large number of additions and multiplications (computational time) – needs approximately 1/3 of as many operations. Therefore, we concatenate or compose the matrices into one final transformation matrix, and then apply that to the points.

Translations:

Translate the points by tx_1, ty_1, then by tx_2, ty_2:

$$\begin{bmatrix} 1 & 0 & (tx_1 + tx_2) \\ 0 & 1 & (ty_1 + ty_2) \\ 0 & 0 & 1 \end{bmatrix}$$

Scaling:

Similar to translations

Rotations:

Rotate by θ_1, then by θ_2:

(i) stick the $(\theta_1 + \theta_2)$ in for θ, or

(ii) calculate T_1 for θ_1, then T_2 for θ_2 & multiply them.

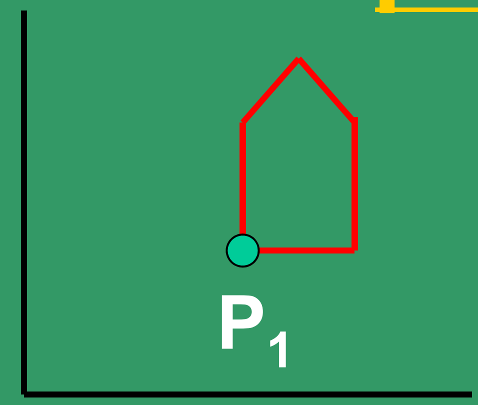
Exercise: Both gives the same result – work it out

Rotation about an arbitrary point P in space

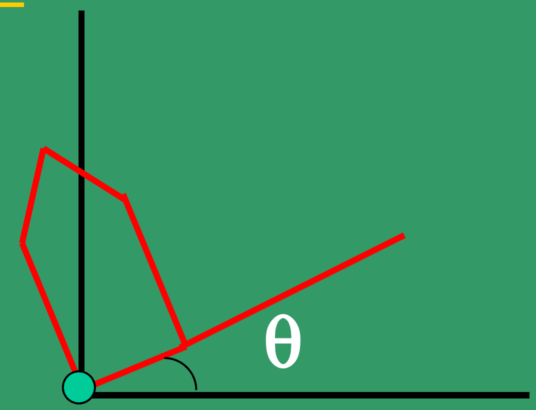
As we mentioned before, rotations are applied about the origin. So to rotate about any arbitrary point P in space, translate so that P coincides with the origin, then rotate, then translate back. Steps are:

- Translate by $(-P_x, -P_y)$
- Rotate
- Translate by (P_x, P_y)

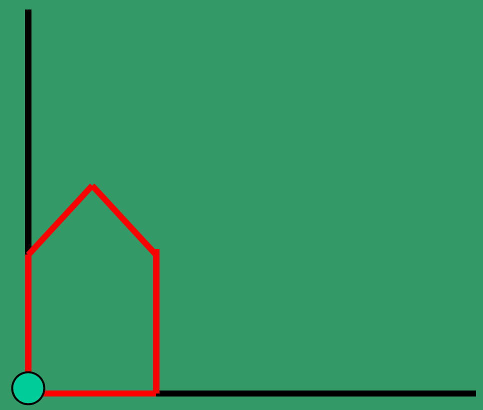
Rotation about an arbitrary point P in space



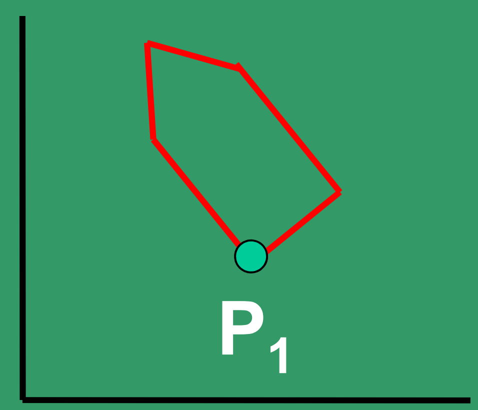
House at P_1



Rotation by θ



Translation of P_1 to Origin



Translation back to P_1

Rotation about an arbitrary point P in space

$$T = T_3(P_x, P_y) * T_2(\theta) * T_1(-P_x, -P_y)$$

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

Scaling about an arbitrary point in Space

Again,

- **Translate P to the origin**

- **Scale**

- **Translate P back**

$T = T_1(P_x, P_y) * T_2(S_x, S_y) * T_3(-P_x, -P_y)$

$$T = \begin{bmatrix} Sx & 0 & \{Px*(1-Sx)\} \\ 0 & Sy & \{Py*(1-Sy)\} \\ 0 & 0 & 1 \end{bmatrix}$$

Reflection through an arbitrary line

Steps:

- **Translate line to the origin**
- **Rotation about the origin**
- **Reflection matrix**
- **Reverse the rotation**
- **Translate line back**

$$T_{GenRfl} = T_r \; R \; T_{rfl} \; R^T \; T_r^{-1}$$

Commutivity of Transformations

If we scale, then translate to the origin, and then translate back, is that equivalent to translate to origin, scale, translate back?

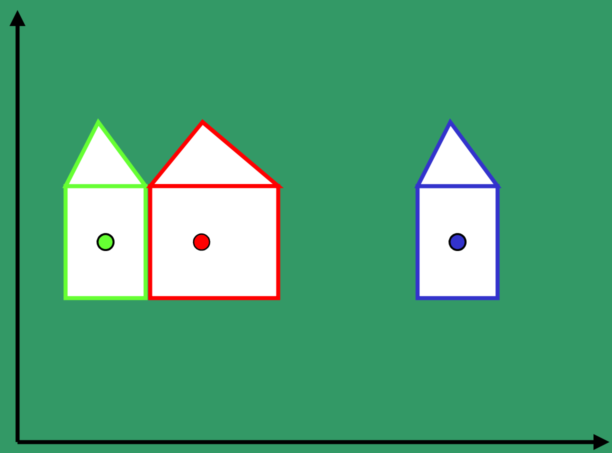
When is the order of matrix multiplication unimportant?

When does $T_1 * T_2 = T_2 * T_1$?

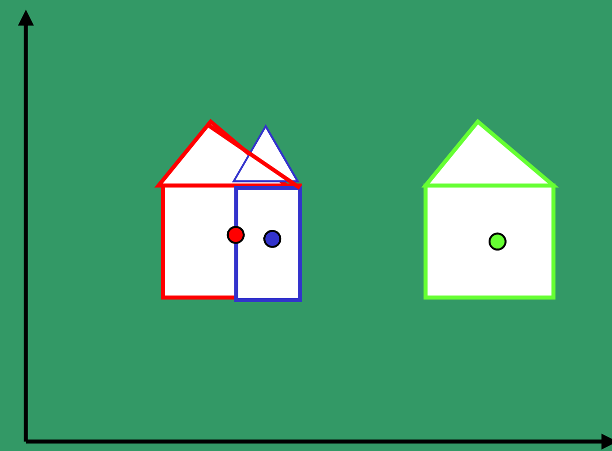
Cases where $T_1 * T_2 = T_2 * T_1$:

T_1	T_2
translation	translation
scale	scale
rotation	rotation
scale(uniform)	rotation

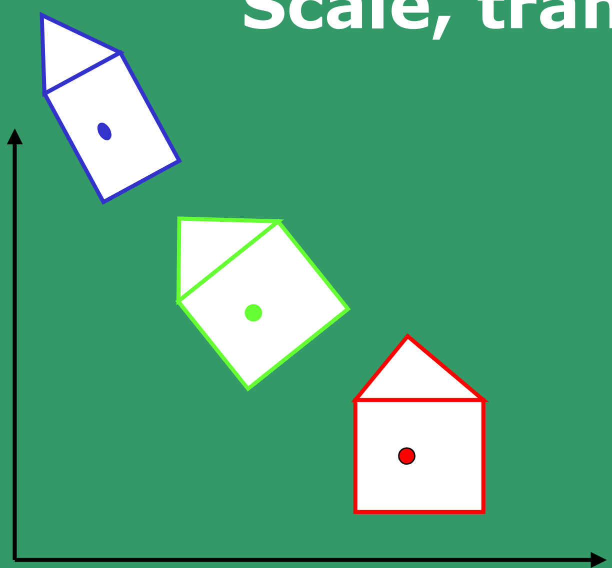
Order:
R-G-B



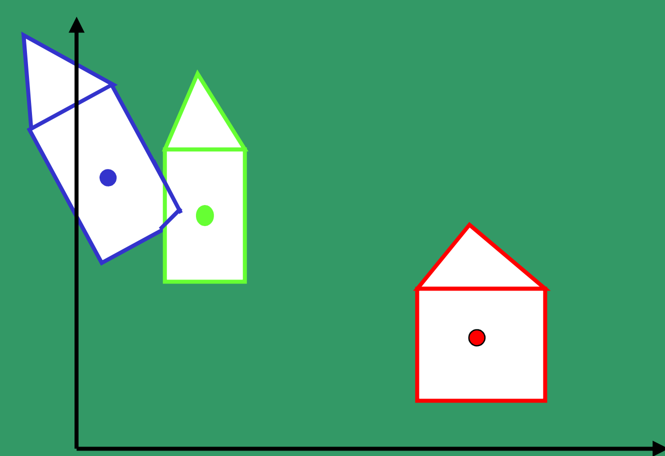
Scale, translate



Translate, scale



Rotate, differential scale



Differential scale, rotate

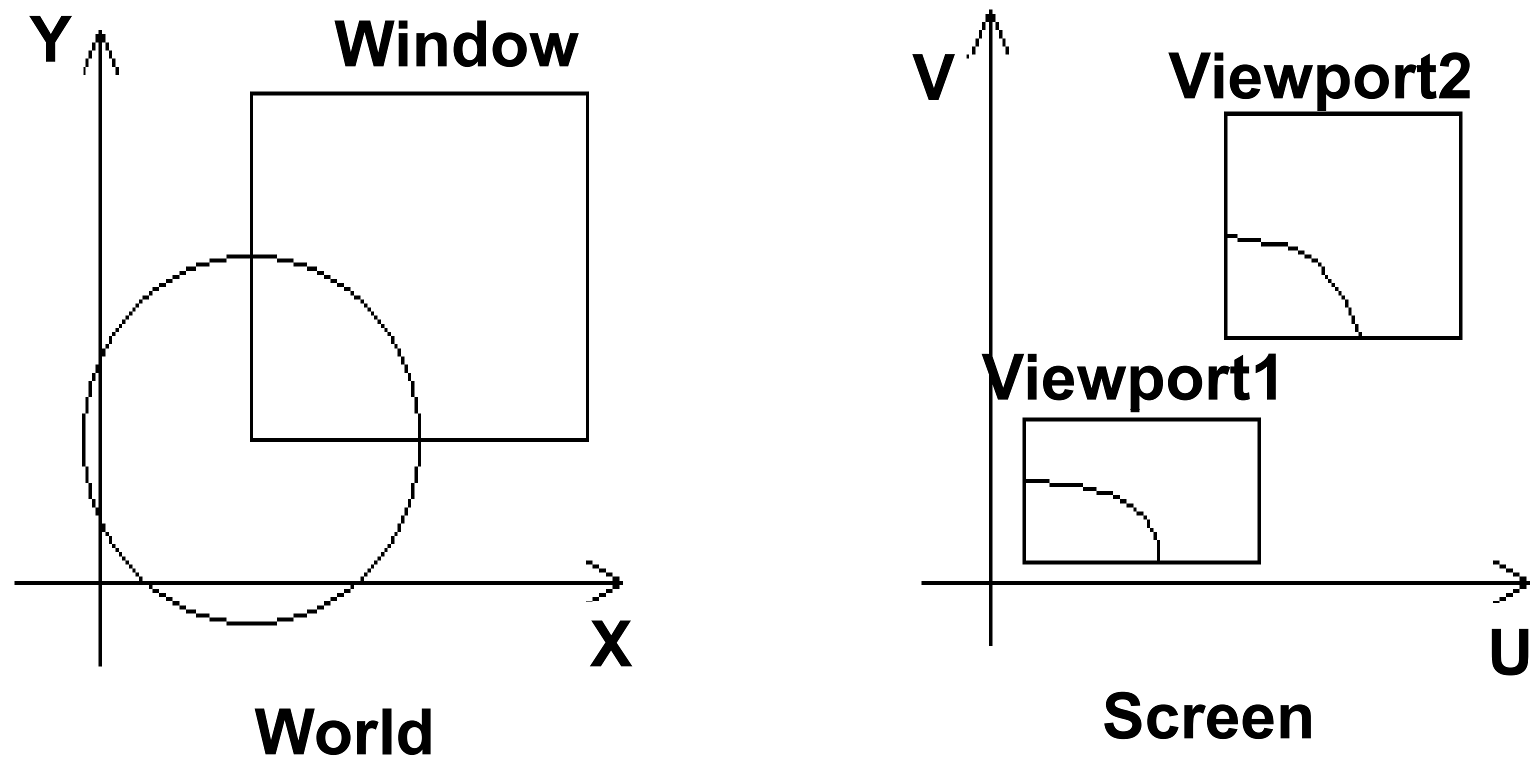
COORDINATE SYSTEMS

Screen Coordinates: The coordinate system used to address the screen (device coordinates)

World Coordinates: A user-defined application specific coordinate system having its own units of measure, axis, origin, etc.

Window: The rectangular region of the world that is visible.

Viewport: The rectangular region of the screen space that is used to display the window.


Y
Window
X
World
V
Viewport2
Viewport1
U
Screen

WINDOW TO VIEWPORT TRANSFORMATION

Purpose is to find the transformation matrix that maps the window in world coordinates to the viewport in screen coordinates.

Window: (x, y space) denoted by:

$x_{min}, y_{min}, x_{max}, y_{max}$

Viewport: (u, v space) denoted by:

$u_{min}, v_{min}, u_{max}, v_{max}$

The overall transformation:

- Translate the window to the origin
- Scale it to the size of the viewport
- Translate it to the viewport location

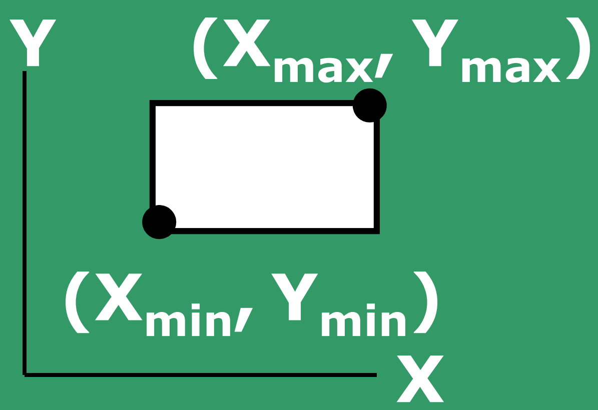
$$M_{WV} = T(U_{\min}, V_{\min}) * S(S_x, S_y) * T(-x_{\min}, -y_{\min});$$

$$S_x = (U_{\max} - U_{\min}) / (x_{\max} - x_{\min});$$

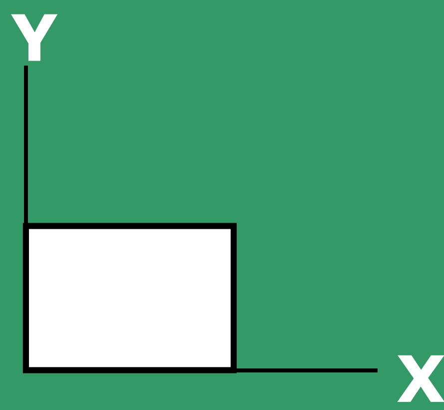
$$S_y = (V_{\max} - V_{\min}) / (y_{\max} - y_{\min});$$

$$M_{WV} = \begin{bmatrix} S_x & 0 & (-x_{\min} * S_x + U_{\min}) \\ 0 & S_y & (-y_{\min} * S_y + V_{\min}) \\ 0 & 0 & 1 \end{bmatrix}$$

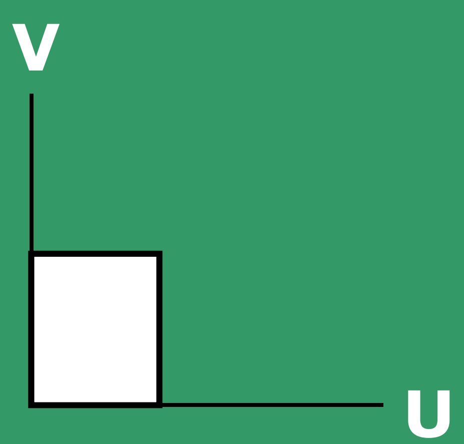
Window – Viewport Transformation



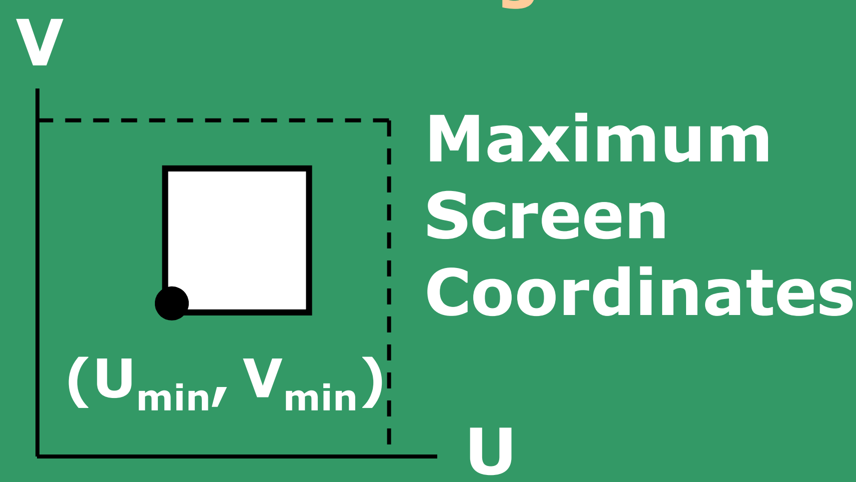
Window in World Coordinates



Window translated to origin



Window Scaled to size to Viewport



Viewport Translated to final position

Exercise -Transformations of Parallel Lines

Consider two parallel lines:

(i) A[X_1, Y_1] to B[X_2, Y_2] and

(ii) C[X_3, Y_3] to B[X_4, Y_4].

Slope of the lines:

$$m = \frac{Y_2 - Y_1}{X_2 - X_1} = \frac{Y_4 - Y_3}{X_4 - X_3}$$

Solve the problem:

If the lines are transformed by a matrix:

$$T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

The slope of the transformed lines is:

$$m' = \frac{b + dm}{a + cm}$$