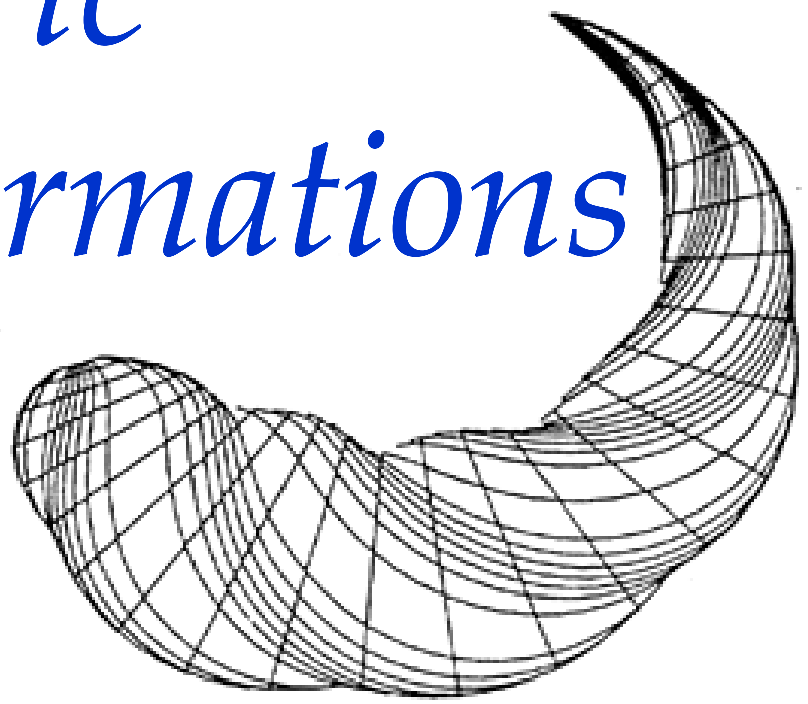
Geometric Modeling

Geometric Transformations



Alexander Pasko, Evgenii Maltsev, Dmitry Popov

Contents

- Matrix operations
- 2D transformations
- 3D transformations
- Reference materials

Points representation

2D space

$[x, y]$

or

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

3D space

$[x, y, z]$

or

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Points in 2D and 3D spaces are represented as row or a column matrix. The object transformations are presented in matrix form.

Matrices and Matrix Operators

Matrix is a rectangular table of elements with rows and columns:

$$A = (a_{i,j})_{m \times n}$$

(m, n – dimensions)

$$\begin{bmatrix} x_1 \\ . \\ . \\ . \\ x_n \end{bmatrix} \quad A = \begin{bmatrix} u_{11}\ u_{12}\ \ldots\ \ldots\ \ldots\ u_{1k} \\ u_{21}\ u_{22}\ \ldots\ \ldots\ \ldots\ u_{2k} \\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots . \\ u_{n1}\ u_{n2} \ldots\ \ldots\ \ldots .. u_{nk} \end{bmatrix}$$

$$A + B = B + A$$
$$A + (B + C) = (A + B) + C$$
$$(cd)A = c(dA)$$
$$1A = A$$
$$c(A + B) = cA + cB$$
$$(c + d)A = cA + dA$$

Matrix Operations:

- ✓ Addition/ Subtraction
- ✓ Identity
- ✓ Multiplication

Scalar multiplication

If a Matrix ***A*** and a number ***c*** are given, we may define the scalar multiplication ***cA*** by

$$(c\,A)\,[\,i, j\,] = c\,A\,[\,i, j\,]$$

$$2\begin{bmatrix}1 & 8 & -3\\ 4 & -2 & 5\end{bmatrix} = \begin{bmatrix}2\times 1 & 2\times 8 & 2\times -3\\ 2\times 4 & 2\times -2 & 2\times 5\end{bmatrix} = \begin{bmatrix}2 & 16 & -6\\ 8 & -4 & 10\end{bmatrix}$$

Matrix Multiplication

- Multiplication of two matrices is well-defined only if the number of columns of the first matrix is the same as the number of rows of the second matrix.
- If A is an m-by-n matrix (m rows, n columns) and B is an n-by-p matrix (n rows, p columns), then their product AB is the m-by-p matrix (m rows, p columns) given by

$$(AB)[i, j] = A[i, 1] * B[1, j] + A[i, 2] * B[2, j] + \ldots + A[i, n] * B[n, j]$$

for each pair i and j.

Matrix Multiplication

- It is easy to remember how to do this by imagining the first matrix as being built out of row vectors and the second matrix as being built out of (column) vectors:

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

and

$$B = \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \\ b_5 & b_6 \end{bmatrix} = \begin{bmatrix} V_3 & V_4 \end{bmatrix}$$

Then

$$A \times B = \begin{bmatrix} V_1 V_3 & V_1 V_4 \\ V_2 V_3 & V_2 V_4 \end{bmatrix}$$

where in each product above one multiplies a row vector by a column vector by multiplying the corresponding entries and adding up the results

Matrix Multiplication Properties

This multiplication has the following properties:

$$(AB)C = A(BC)$$

for all k-by-m matrices A, m-by-n matrices B and n-by-p matrices C ("associativity").

$$(A + B)C = AC + BC$$

for all m-by-n matrices A and B and n-by-k matrices C ("right distributivity").

$$C(A + B) = CA + CB$$

for all m-by-n matrices A and B and k-by-m matrices C ("left distributivity").

It is important to note that commutativity does ***not*** generally hold; that is, given matrices A and B and their product defined, then generally $AB \neq BA$.

Matrix Determinants

A single real number

Computed recursively

$$det\ A = \sum_{i=1}^{n} (-1)^{1+i} A_{1i}$$

Example:

$$\det\begin{bmatrix} a & c \\ b & d \end{bmatrix} = ad - bc$$

Matrix Transpose and Inverse

Matrix Transpose:
Swap rows and cols:

$$A = \begin{bmatrix} 7 \\ 5 \end{bmatrix} \quad A^T = \begin{bmatrix} 7 & 5 \end{bmatrix}$$

$$(A^T)^T = A$$

$$(A+B)^T = A^T + B^T$$

$$(cA)^T = c(A^T)$$

$$(AB)^T = B^T A^T$$

Matrix Inverse: Given A, find B such that

$$AB = BA = I$$

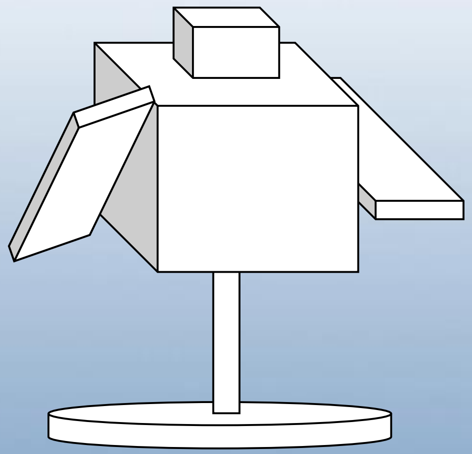
Transformations

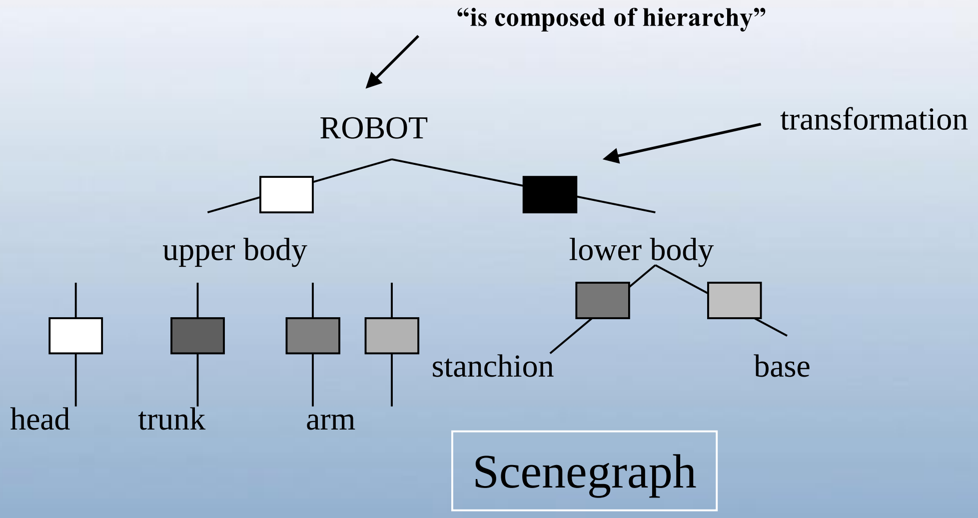
❖ 2D transformations

- Translation
- Rotation
- Scaling
- Shear
- Matrix representation
- Homogeneous coordinates

How Are Geometric Transformations Used?

- Object construction using assemblies/ hierarchy of parts; leaves contain primitives, nodes contain transformations.

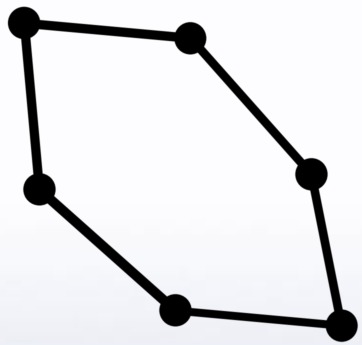




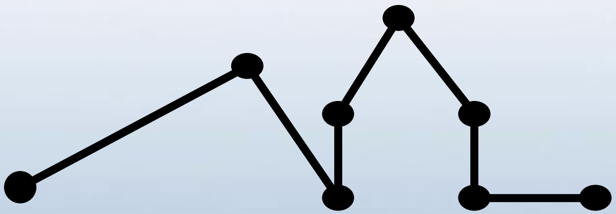
2D Object Definition using Points

Lines and Polylines

- Lines drawn between ordered points to create more complex forms called *polylines*

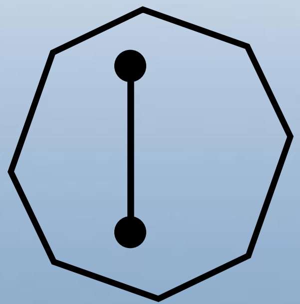


- Same first and last point make *closed polyline* or *polygon*
- If it does not intersect itself, called *simple polygon*

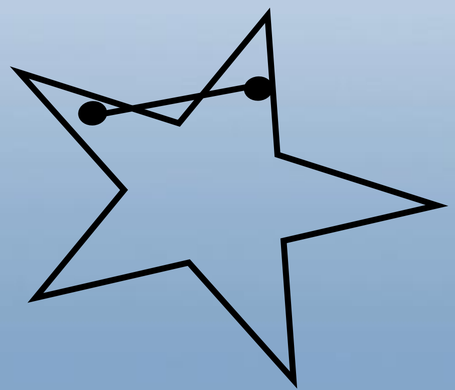


Convex vs. Concave Polygons

Convex : For every pair of points in the polygon, the line between them is fully contained in the polygon.

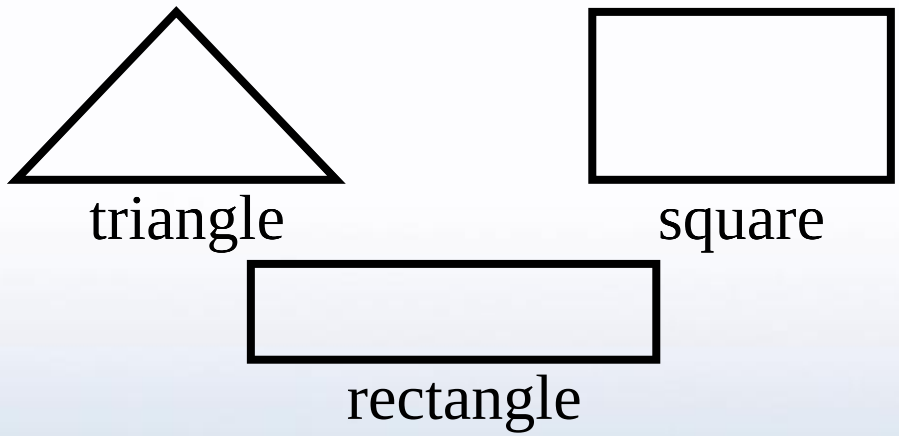


Concave (Not convex): some two points in the polygon are joined by a line not fully contained in the polygon.



2D Object Definition

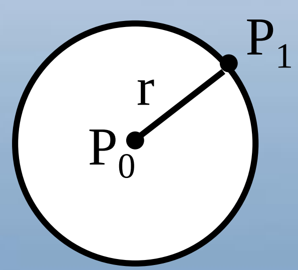
Special polygons

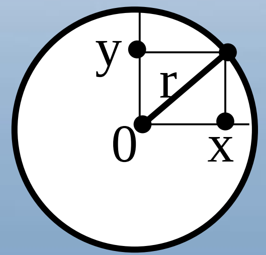


Circle

- Consists of all points equidistant from one predetermined point (the center)
- (radius) *r* = *c*, where *c* is a constant
- In the Cartesian coordinates with center of circle at origin equation is

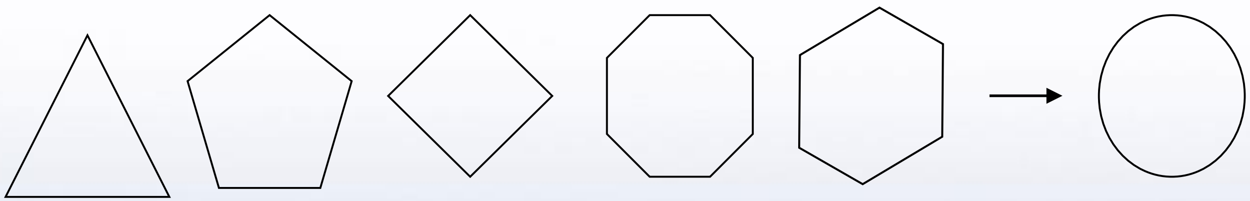
$$r^2 = x^2 + y^2$$





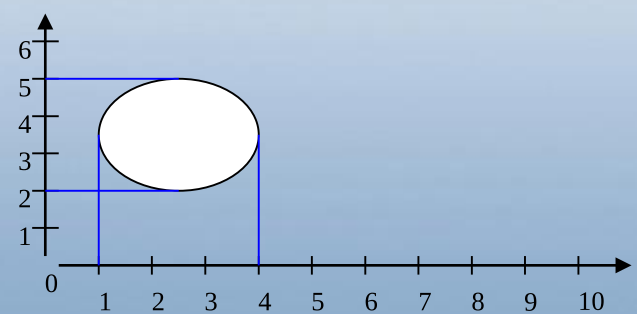
Circle as polygon

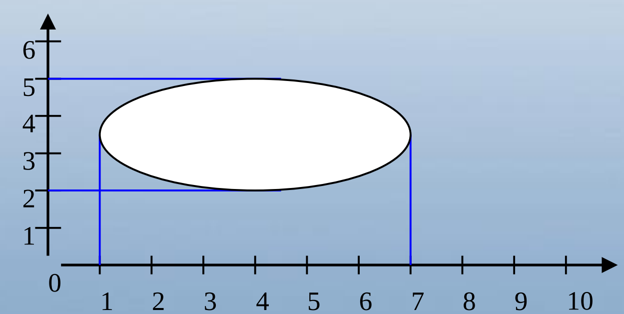
- Informally: a regular polygon with > 15 sides



(Aligned) Ellipses

A circle scaled along the x or y axis





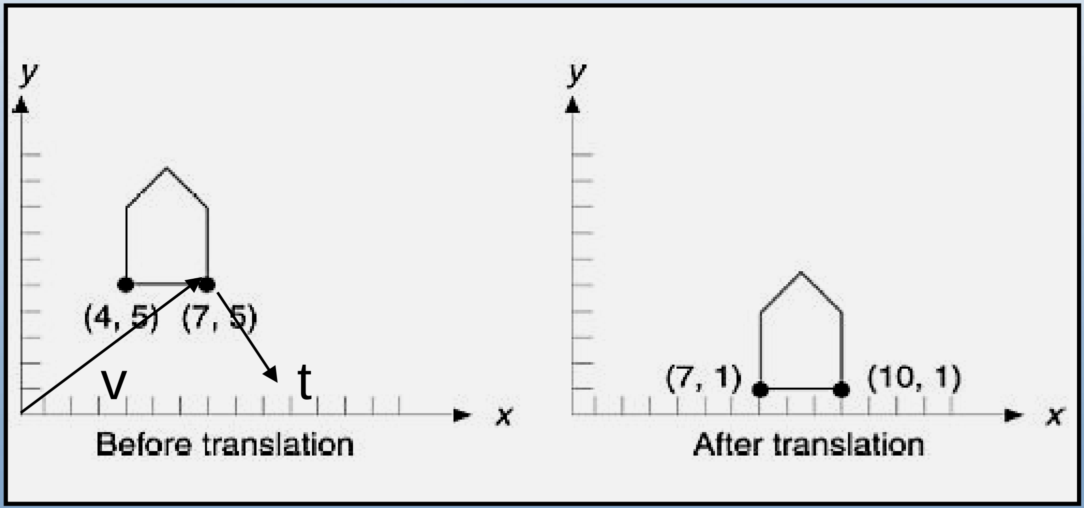
Example: height, on *y-axis*, remains 3, while length, on *x-axis*, changes from 3 to 6

2D Translation

- Component-wise addition of vectors **v' = v + t**
- Translation of points in the (x,y) plane to a new position by adding translation amount to the coordinates of the point

$$x' = x + dx$$

$$y' = y + dy$$



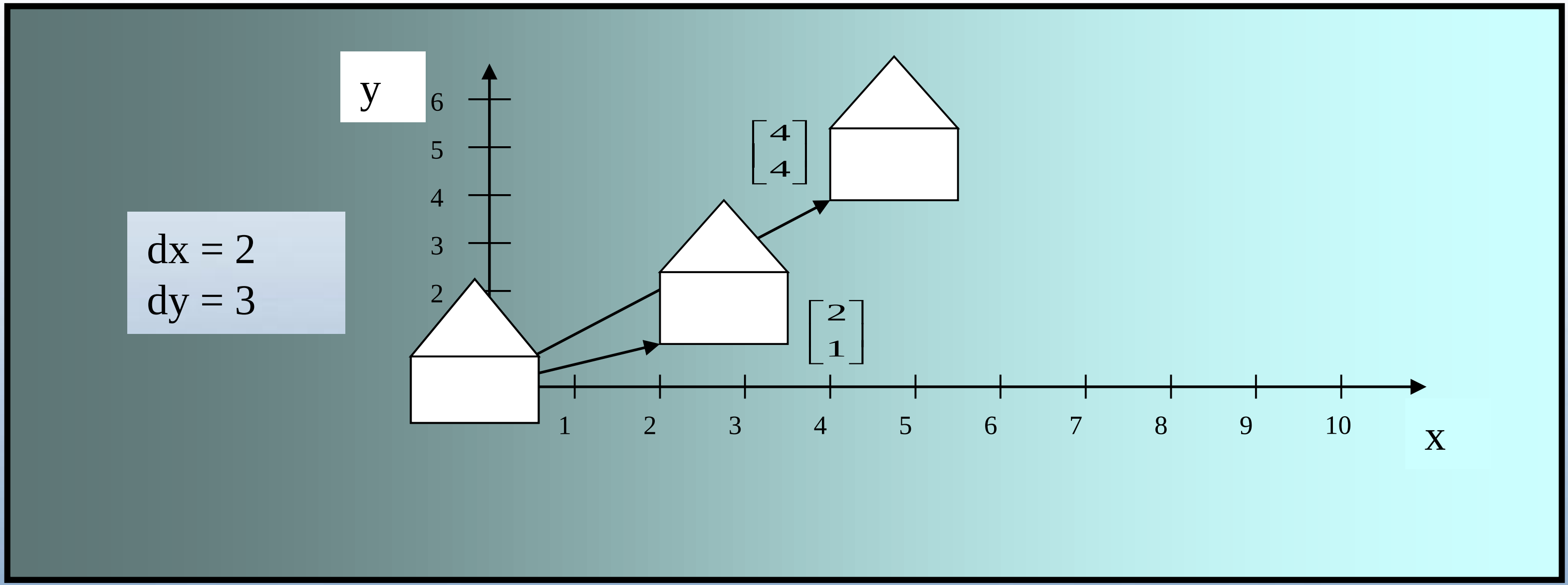
In Matrix form:

$$v = \begin{bmatrix} x \\ y \end{bmatrix}, \quad v' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad t = \begin{bmatrix} dx \\ dy \end{bmatrix}$$

$$v' = v + t \quad \rightarrow \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

2D Translation

- To move polygons: just translate vertices (vectors) and then redraw lines between them
- Preserves lengths (isometric)
- Preserves angles (conformal)



House shifts position relative to origin

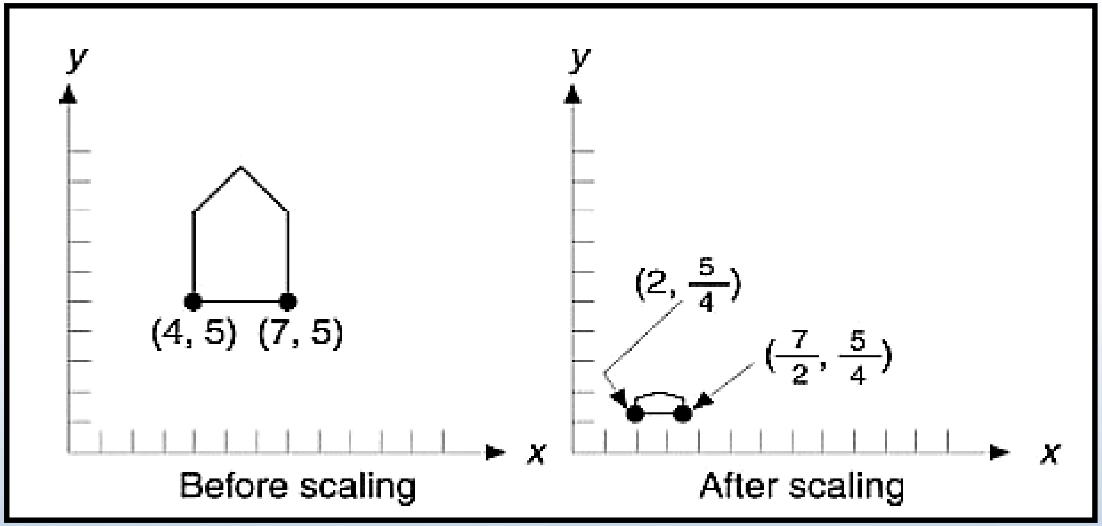
A translation by (0,0), i.e. no translation at all, gives us the identity matrix, as it should.

2D Scaling

- Component-wise scalar multiplication of vectors

$$v' = S \cdot v$$

- Point can be scaled (stretched) by s_x along the x axis and by s_y along the y axis into new points by the multiplication:



$$x' = s_x x$$
$$y' = s_y y$$

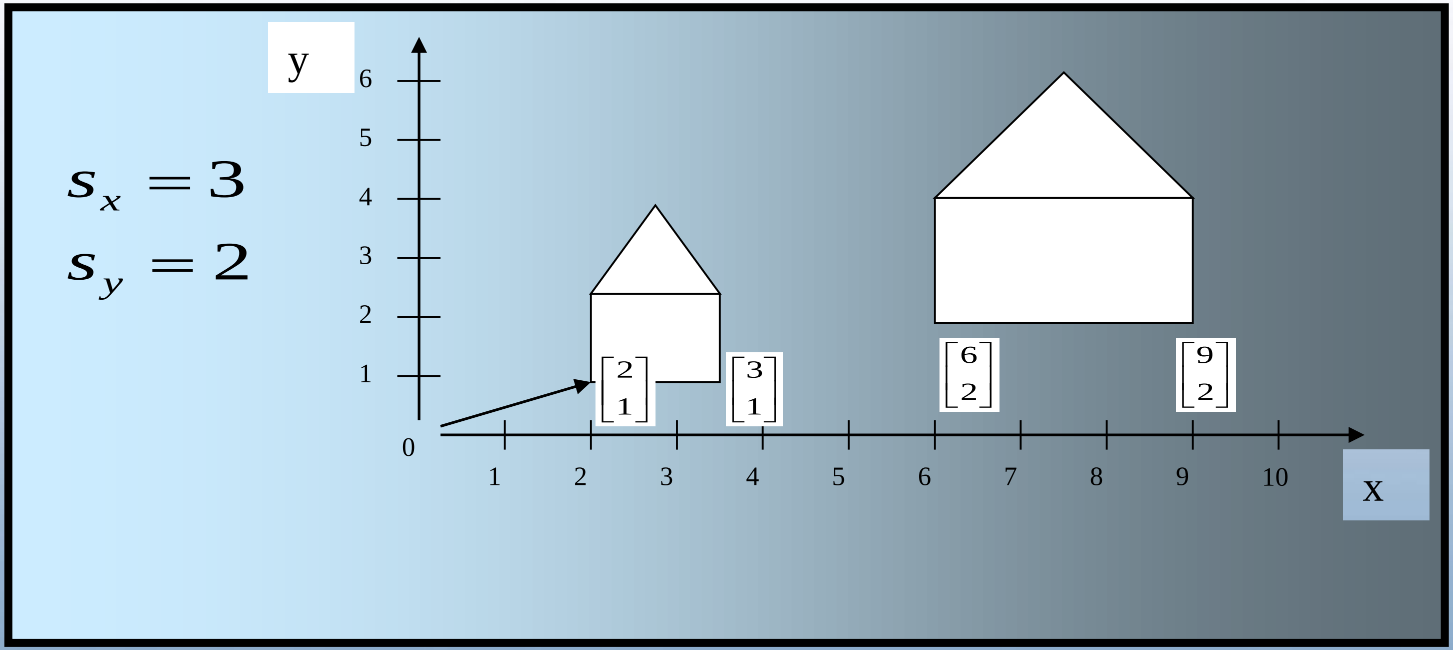
In Matrix form:

$$v = \begin{bmatrix} x \\ y \end{bmatrix}, \quad v' = \begin{bmatrix} x' \\ y' \end{bmatrix} \qquad S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

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

2D Scaling

- Does not preserve lengths
- Does not preserve angles (except when scaling is uniform)

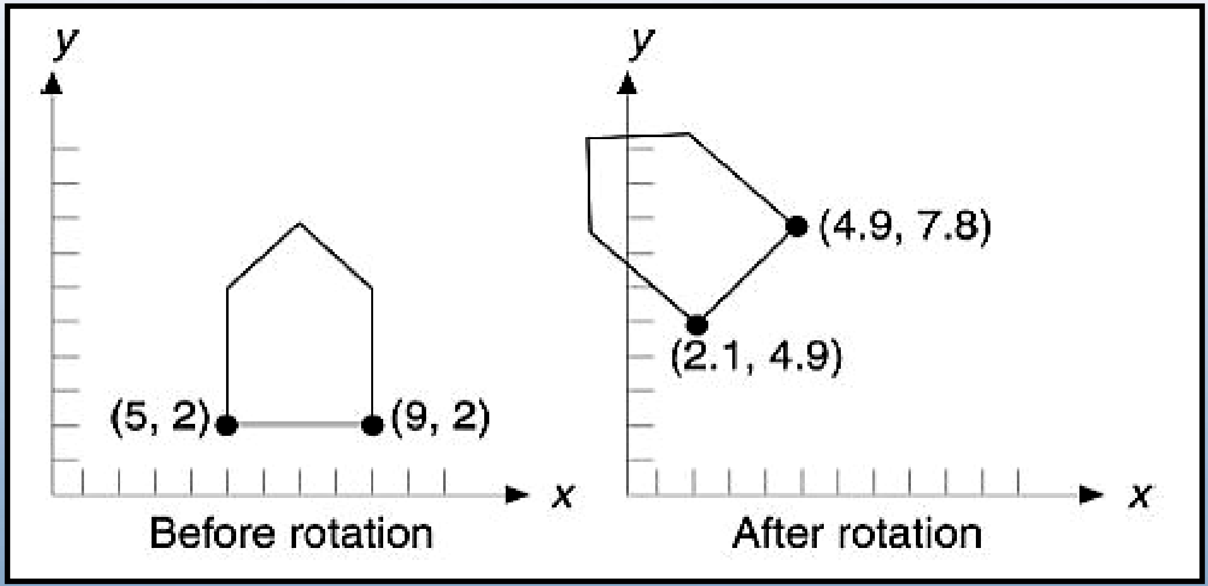


Note: House shifts position relative to origin

2D Rotation

Rotation of vectors through an angle θ about the origin $v' = R_\theta \cdot v$

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



In Matrix form:

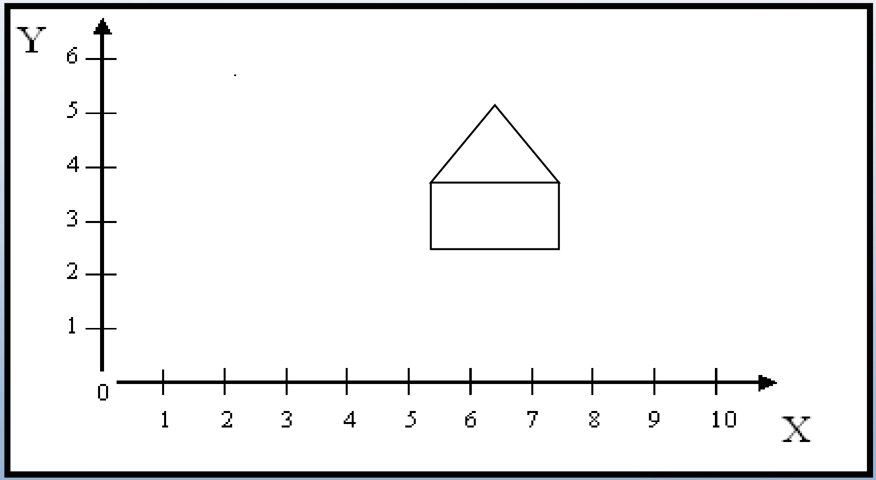
$$v = \begin{bmatrix} x \\ y \end{bmatrix}, \quad v' = \begin{bmatrix} x' \\ y' \end{bmatrix} \qquad R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

R_θ – rotation Matrix

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

2D Rotation and Scale are Relative to Origin

- Suppose object is not centered at origin?
- Solution: move it to the origin, then scale and/or rotate, then move it back.



- Composition of the successive transformations

Homogenous Coordinates

- Translation, scaling and rotation are expressed as:

translation: $v' = v + t$

scale: $v' = S \cdot v$

rotation: $v' = R \cdot v$

- Composition is difficult to express, since translation not expressed as a Matrix multiplication
- Homogeneous coordinates allow all transformations (translation, scaling and rotation) to be expressed homogeneously, allowing composition via multiplication by 3x3 matrices

Homogenous Coordinates

Point is presented by a triple *(x,y,w) or* $\begin{bmatrix} x \\ y \\ w \end{bmatrix}$

Two sets of homogenious coordinates *(x,y,w) and (x’,y’,w’)* are presents the same point if and only if one a multiple of the other.

The same points by different coordinate triples: (2,3,7), (6,9,21);

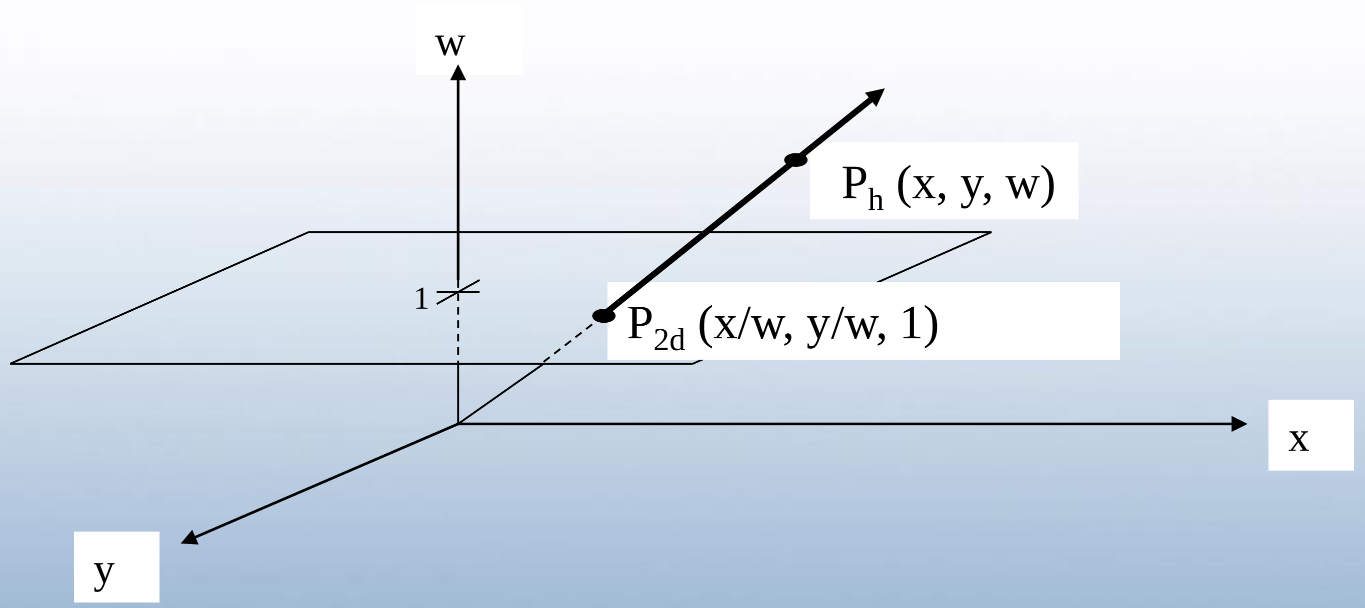
$$P_{2d}(x,y) \to P_h(wx, wy, w), \quad w \neq 0$$

$$P_h(x', y', w), \quad w \neq 0$$

$$P_{2d}(x,y) = P_{2d}\left(\frac{x'}{w}, \frac{y'}{w}\right)$$

Homogenous Coordinates

- w is 1 for affine transformations in graphics
- P_{2d} is intersection of line determined by P_h with the *w = 1* plane



- So an infinite number of points correspond to (*x*, *y*, 1): they constitute the whole line (*tx*, *ty*, *tw*)

2D Homogeneous Coordinate Transformations

- For points written in homogeneous coordinates $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$,

  translation, scaling and rotation relative to the origin are expressed homogeneously as:

$$T(dx, dy) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix}; \quad v' = T(dx, dy)v$$

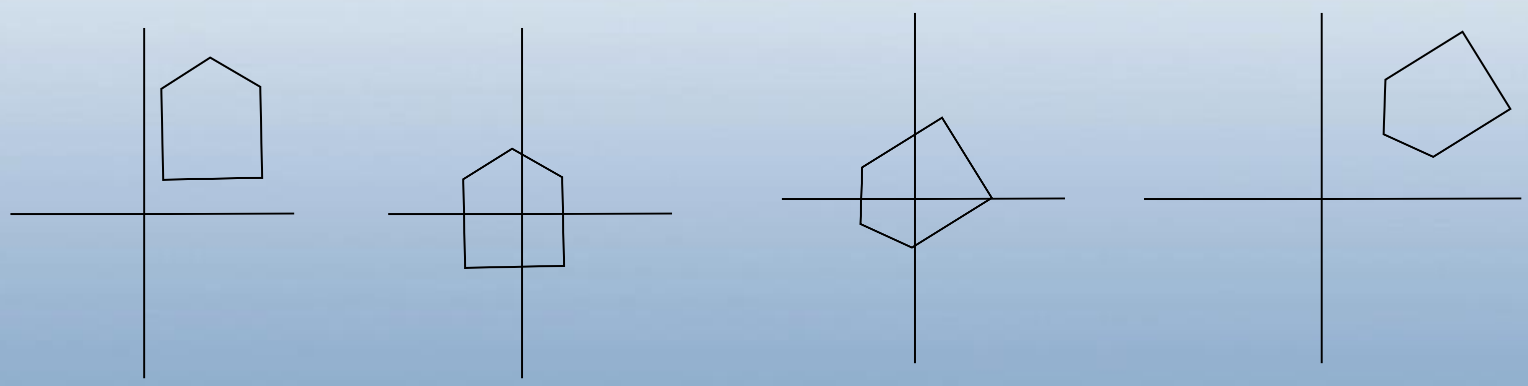
$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad v' = S(s_x, s_y)v$$

$$R(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad v' = R(\phi)v$$

Matrix Compositions

With the T Matrix, can avoid unwanted translation introduced when we scale or rotate an object not centered at origin:

- translate the object to the origin
- perform the scale or rotate
- translate back.



$House\ (H)$ $\quad T(dx, dy)H$ $\quad R(\theta)T(dx, dy)H$ $\quad T(-dx, -dy)R(\theta)T(dx, dy)H$

Matrix Compositions

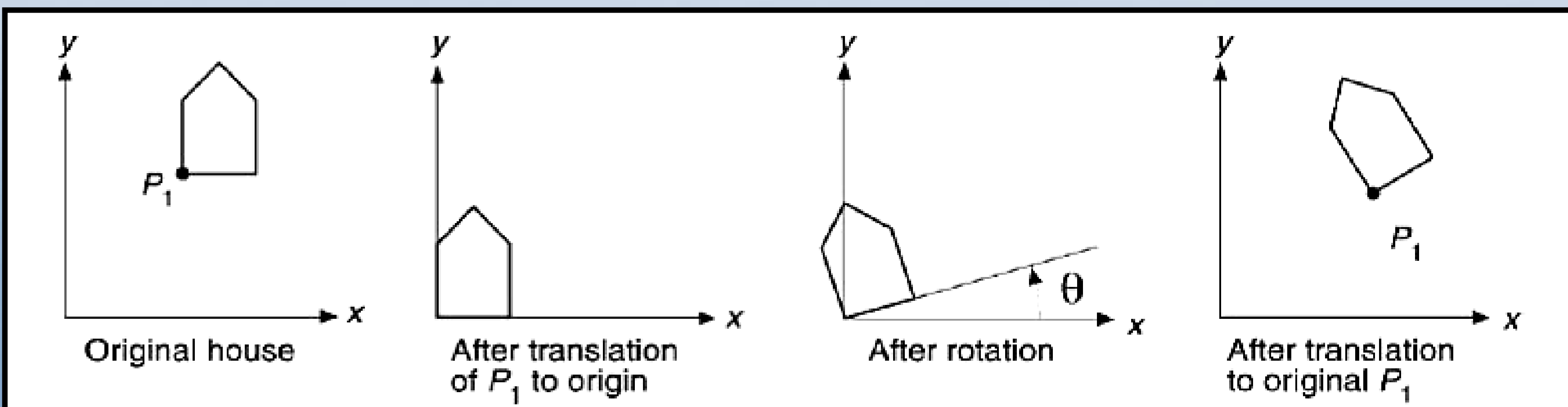
Rotate about a point *P1*

- Translate *P1* to origin
- Rotate
- Translate back to *P1*

$$T(x_1, y_1) \cdot R(\theta) \cdot T(-x_1, -y_1) =$$

$$= \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix}$$

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



Matrix Compositions

Scale object around point *P1*

- *P1* to origin
- Scale
- Translate back to *P1*
- Compose into T

$P' = T \cdot P :$

$$T(x_1, y_1) \cdot S(S_x, S_y) \cdot T(-x_1, -y_1)$$

$$= \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix}$$

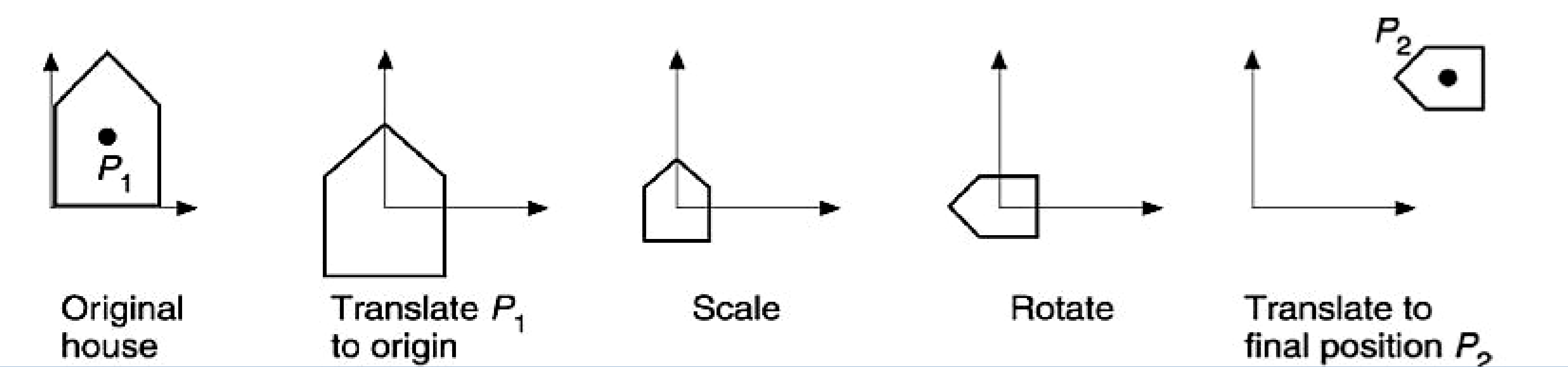
$$= \begin{bmatrix} S_x & 0 & x_1(1 - S_x) \\ 0 & S_y & y_1(1 - S_y) \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix Compositions

- Scale + rotate object around point *P1* and move to *P2*

  - *P1* to origin
  - Scale
  - Rotate
  - Translate to *P2*

$$T(x_2, y_2) \cdot R(\theta) \cdot S(s_x, s_y) \cdot T(-x_1, -y_1)$$



Matrix Compositions

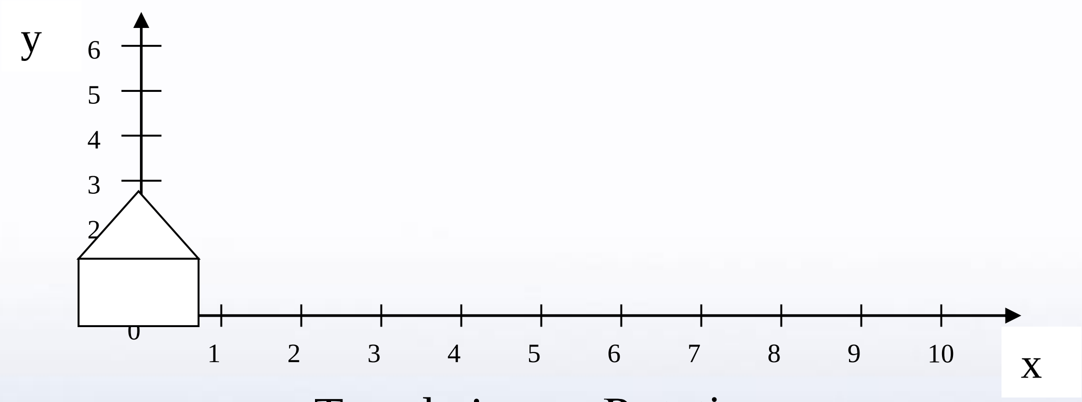
Multiple transformations in proper order:

$$P' = \mathrm{T} \cdot P$$

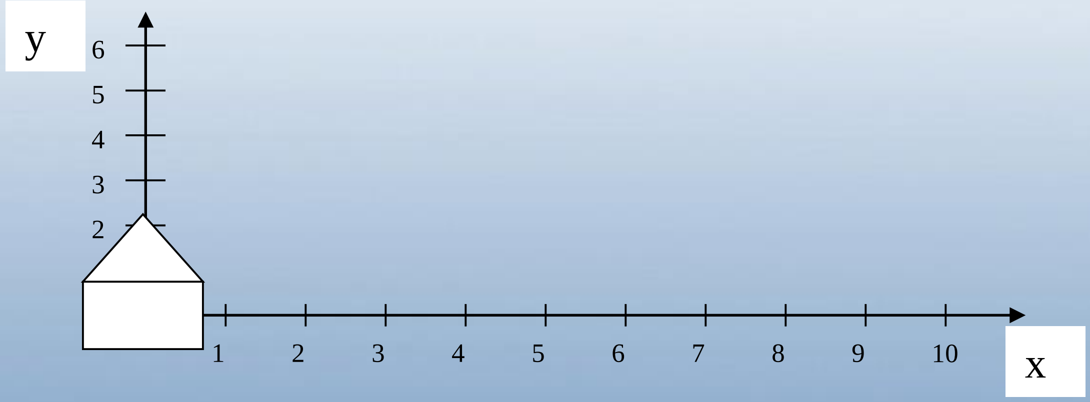
$$P' = ((T \cdot (R \cdot (S \cdot T))) \cdot P)$$

$$P' = (T \cdot (R \cdot (S \cdot (T \cdot P))))$$

Transformations are NOT Commutative



Translation → Rotation

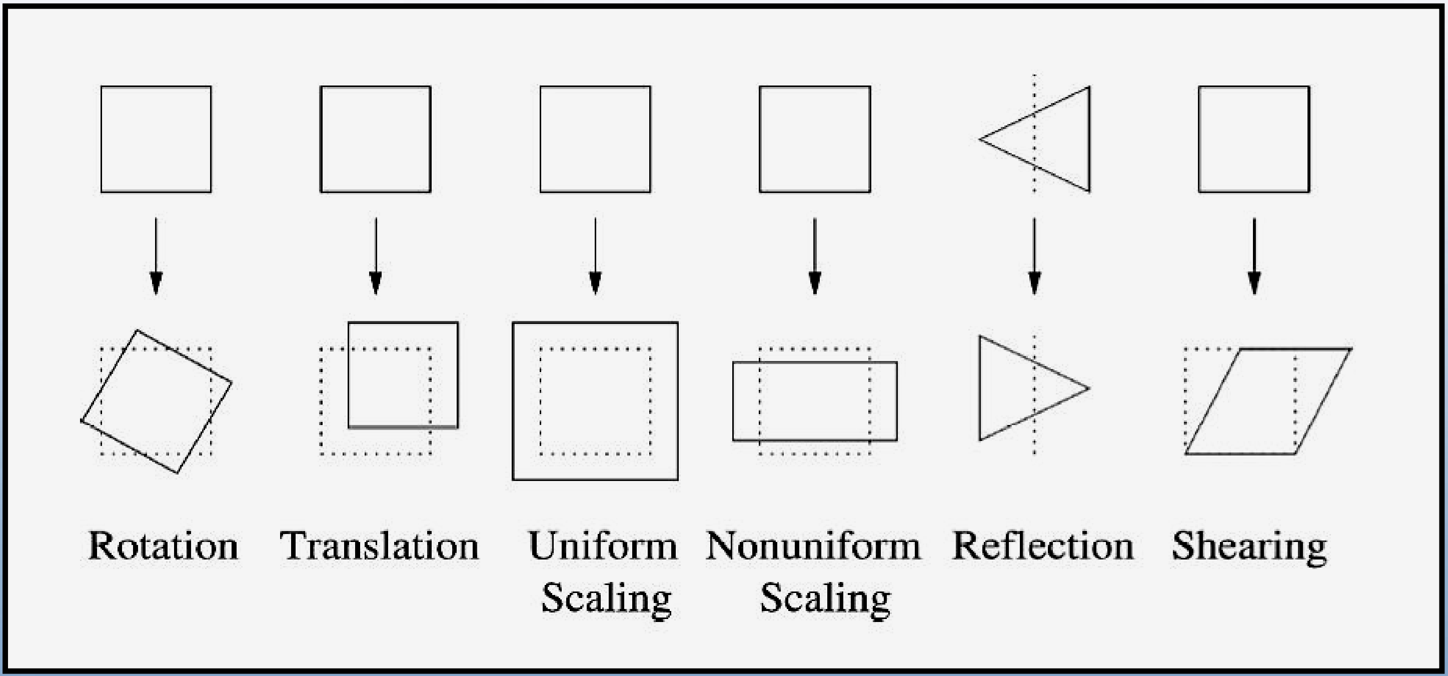


Rotation → Translation

2D Affine Transformations

All represented as Matrix operations on vectors
Parallel lines preserved, angles/ lengths not

- Scale
- Rotate
- Translate
- Reflect
- Shear



Pics/Math courtesy of Dave Mount @ UMD-CP

Matrix Representation of 2D Affine Transformations

Translation: $$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scale: $$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotation: $$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos\ \theta & -sin\ \theta & 0 \\ sin\ \theta & cos\ \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shear: $$SH_x = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Reflection: $$F_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2D Shear

$$\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{p}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x + ay \\ y \end{bmatrix}$$

Shear operation

$$Sh_x(a) = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{p}' = Sh_x(a)\mathbf{p}$$

– Preserves parallels

– Does not preserve lengths and angles

Shear operation along y axis

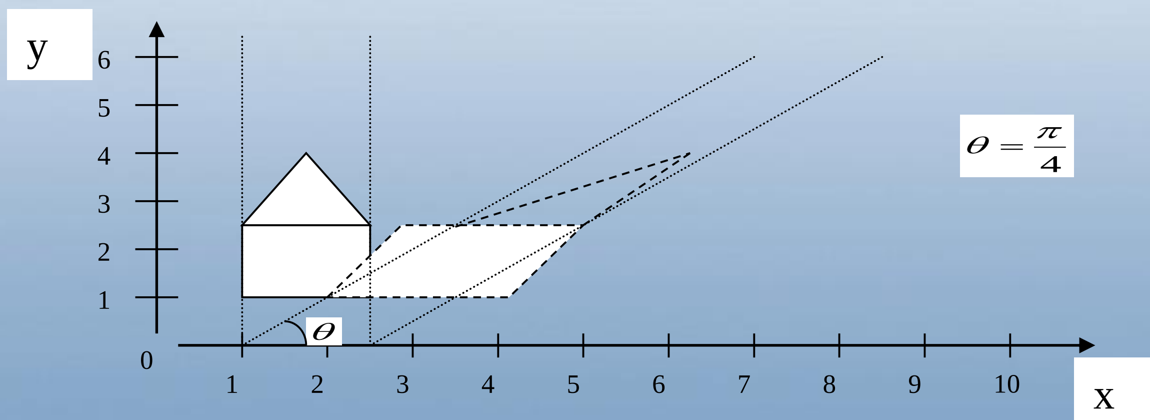
$$Sh_y(b) = \begin{bmatrix} 1 & 0 \\ b & 1 \end{bmatrix}$$

Skew/Shear/Translate

- Take a scene and “skew” it to the side

$$Skew_{\theta} = \begin{bmatrix} 1 & \dfrac{1}{\tan\theta} \\ 0 & 1 \end{bmatrix}$$

- Squares become parallelograms - x coordinates skew to the right, while y coordinates stay the same
- 90^0 between axes becomes θ
- Like taking a deck of cards and pushing top to the side – each card shifts relative to the one below it
- Notice that the base of the house (at y=1) remains horizontal, but shifts to the right…



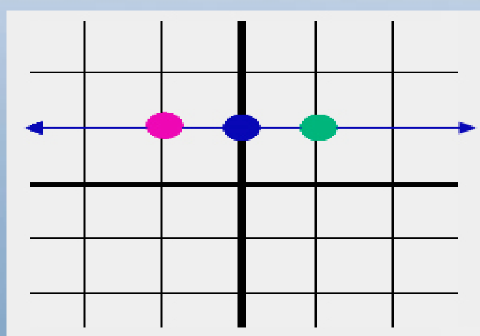
NB: A skew of 0 angle, i.e. no skew at all, gives us the identity Matrix, as it should

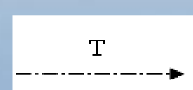
Skew/Shear/Translate

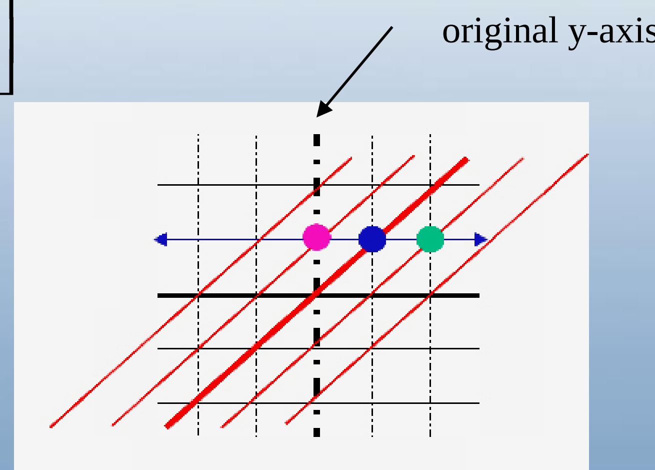
- Everything along the line y=1 stays on the line y=1, but is translated to the right
- Distance between points on this line is preserved
- A 1D homogeneous coordinate translation looks like a 2D skew transformation

$$\begin{bmatrix} 1 & \dfrac{1}{\tan\theta} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & dx \\ 0 & 1 \end{bmatrix}$$

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





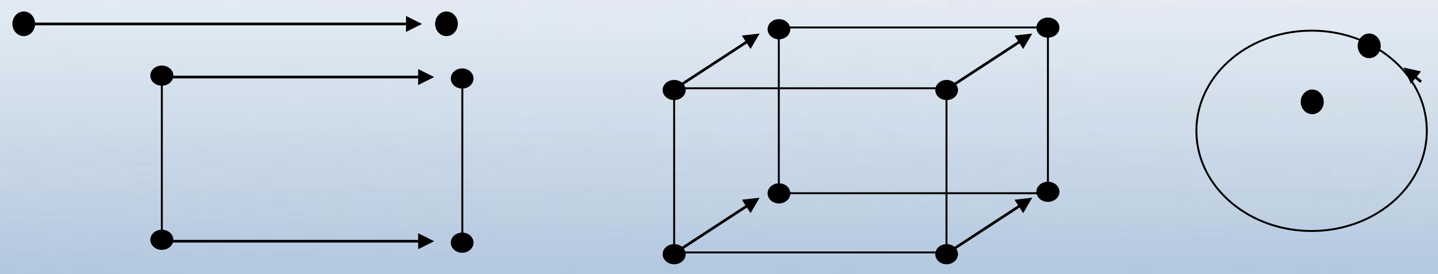


2D to 3D Object Definition

Vertices in motion ("Generative object description")

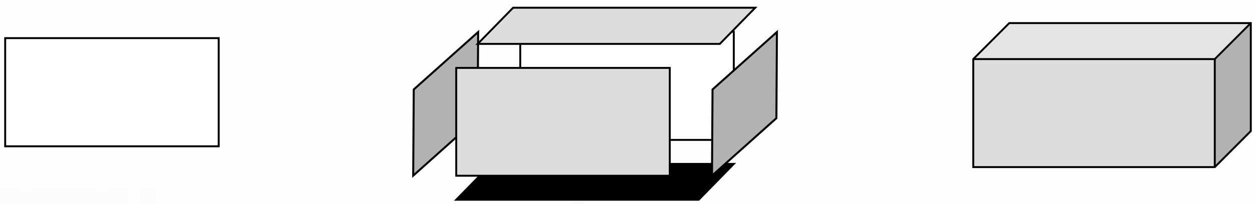
- Line is drawn by tracing path of a point as it moves (one dimensional entity)

- Square drawn by tracing vertices of a line as it moves perpendicularly to itself (two dimensional entity)

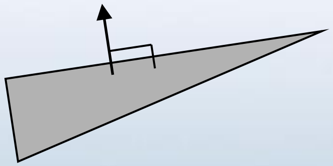


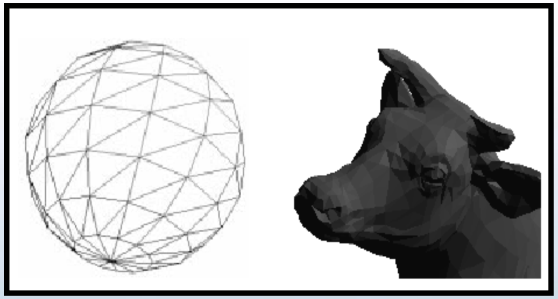
- Cube drawn by tracing paths of vertices of a square as it moves perpendicularly to itself (three-dimensional entity)
- Circle drawn by swinging a point at a fixed length around a center point

Building 3D Primitives

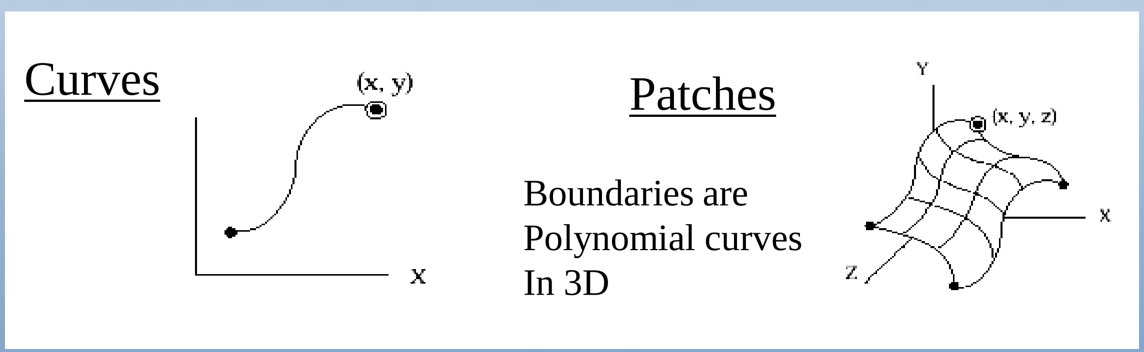


- Triangles and tri-meshes





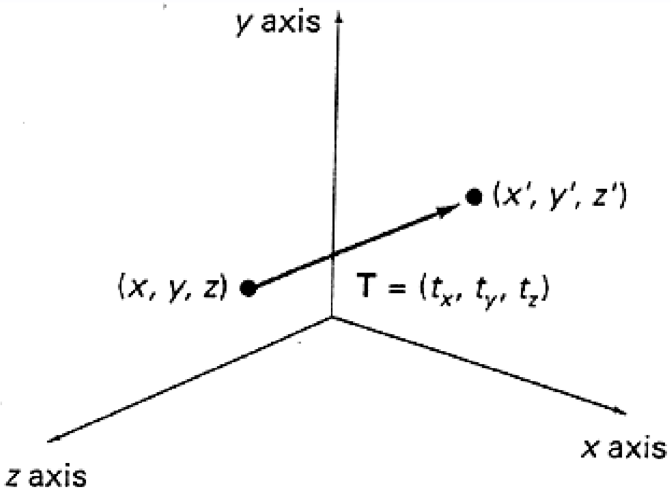
- Often parametric polynomials, called splines



3D Transformations

- Affine transformations
  - Translation
  - Scaling
  - Rotation
- Deformations
  - Twisting
  - Tapering
  - Bending
- Set-theoretic operations
- Metamorphosis

Translation



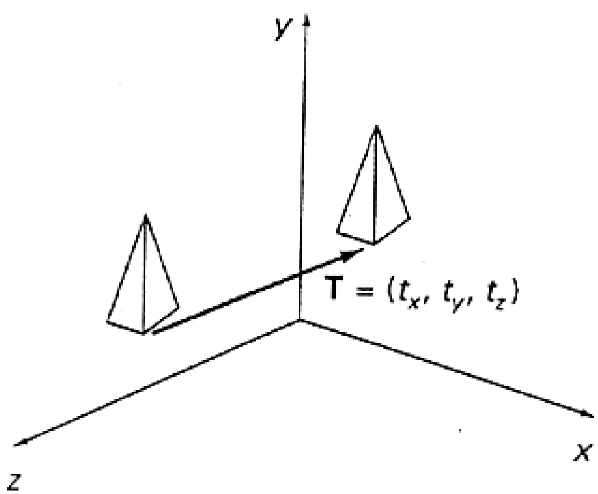
$$x' = x + t_x,$$

$$y' = y + t_y,$$

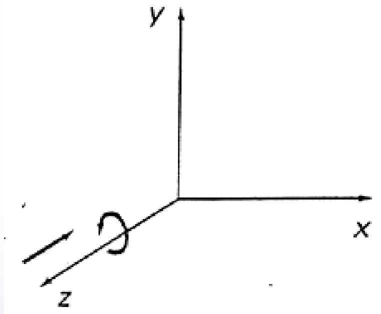
$$z' = z + t_z$$

In a three-dimensional homogeneous coordinate representation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



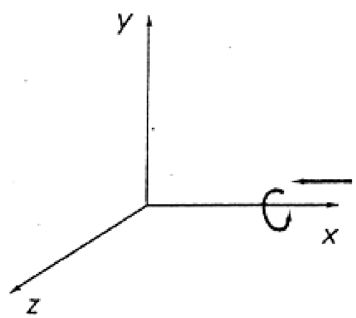
Coordinate-axes rotations



z-axis rotation

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

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



x-axis rotation

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

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scaling

$$x' = x \cdot s_x, \quad y' = y \cdot s_y, \quad z' = z \cdot s_z$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scaling with respect to a selected fixed position (x_f, y_f, z_f) can be represented with the following transformation sequence:

1. Translate the fixed point to the origin
2. Scale the object relative to the coordinate origin
3. Translate the fixed point back to its original position

$$T(x_f, y_f, z_f) \cdot S(s_x, s_y, s_z) \cdot T(-x_f, -y_f, -z_f) = \begin{bmatrix} s_x & 0 & 0 & (1 - s_x)x_f \\ 0 & s_y & 0 & (1 - s_y)y_f \\ 0 & 0 & s_z & (1 - s_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Deformations

(x,y,z) - original point

(X,Y,Z) - point of a deformed object

Forward mapping

For polygonal and parametric forms

$$\Phi: (x,y,z) \to (X,Y,Z) \text{ or}$$

$$(X,Y,Z) = (\phi_1(x,y,z), \phi_2(x,y,z), \phi_3(x,y,z))$$

Inverse mapping

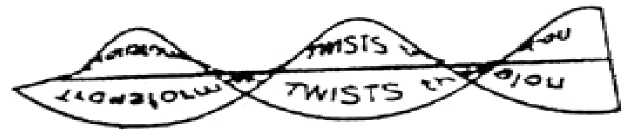
For implicit form

$$\Phi^{-1}: (X,Y,Z) \to (x,y,z) \text{ or}$$

$$(x,y,z) = (\phi^{-1}{}_1(X,Y,Z), \phi^{-1}{}_2(X,Y,Z), \phi^{-1}{}_3(X,Y,Z))$$

Twisting

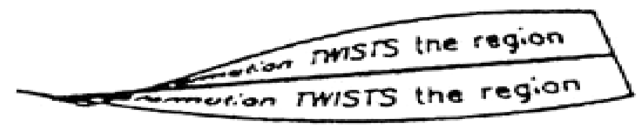
Forward mapping



$$\theta = f(z) \qquad X = xC_\theta - yS_\theta,$$

$$C_\theta = cos(\theta) \qquad Y = xS_\theta + yC_\theta,$$

$$S_\theta = sin(\theta) \qquad Z = z.$$



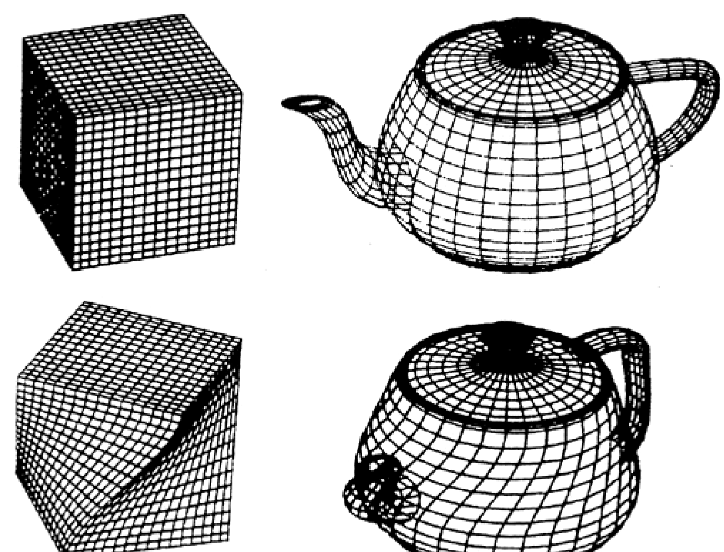
Inverse mapping

$$\theta = f(Z),$$

$$x = XC_\theta + YS_\theta,$$

$$y = -XS_\theta + YC_\theta,$$

$$z = Z$$



Images by A. Barr

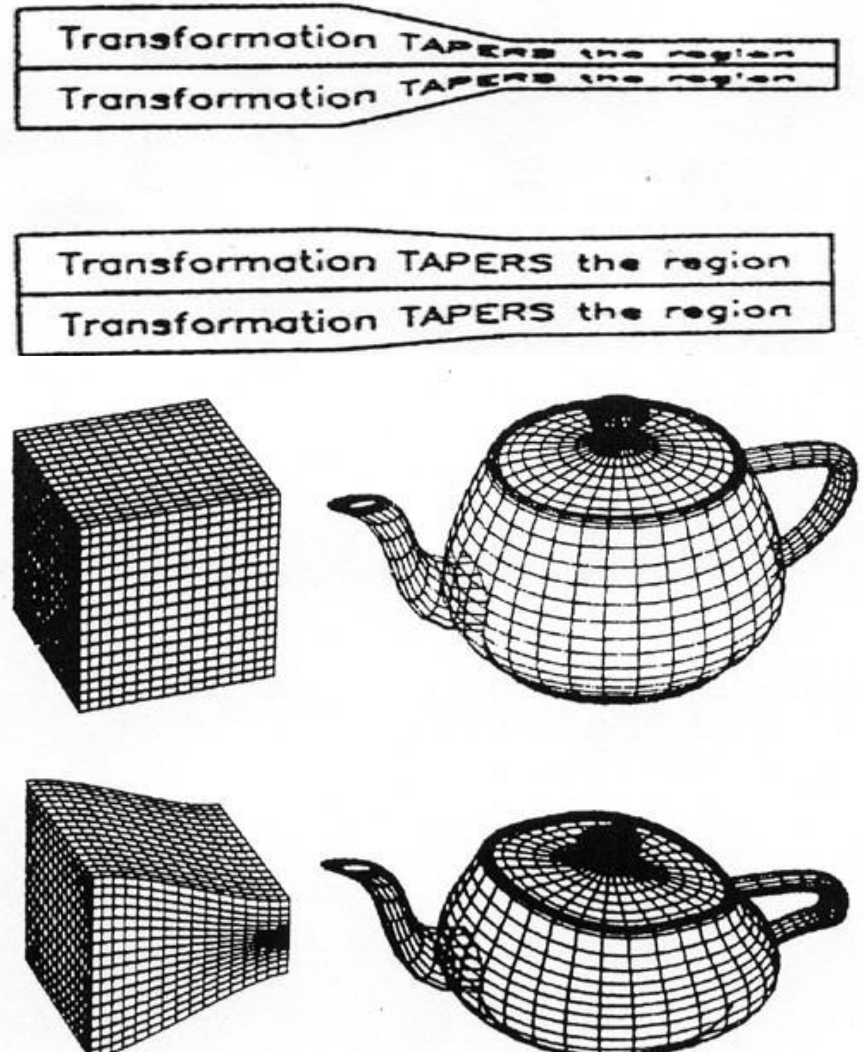
Tapering

Forward mapping

$$r = f(z),$$
$$X = rx,$$
$$Y = ry,$$
$$Z = z$$

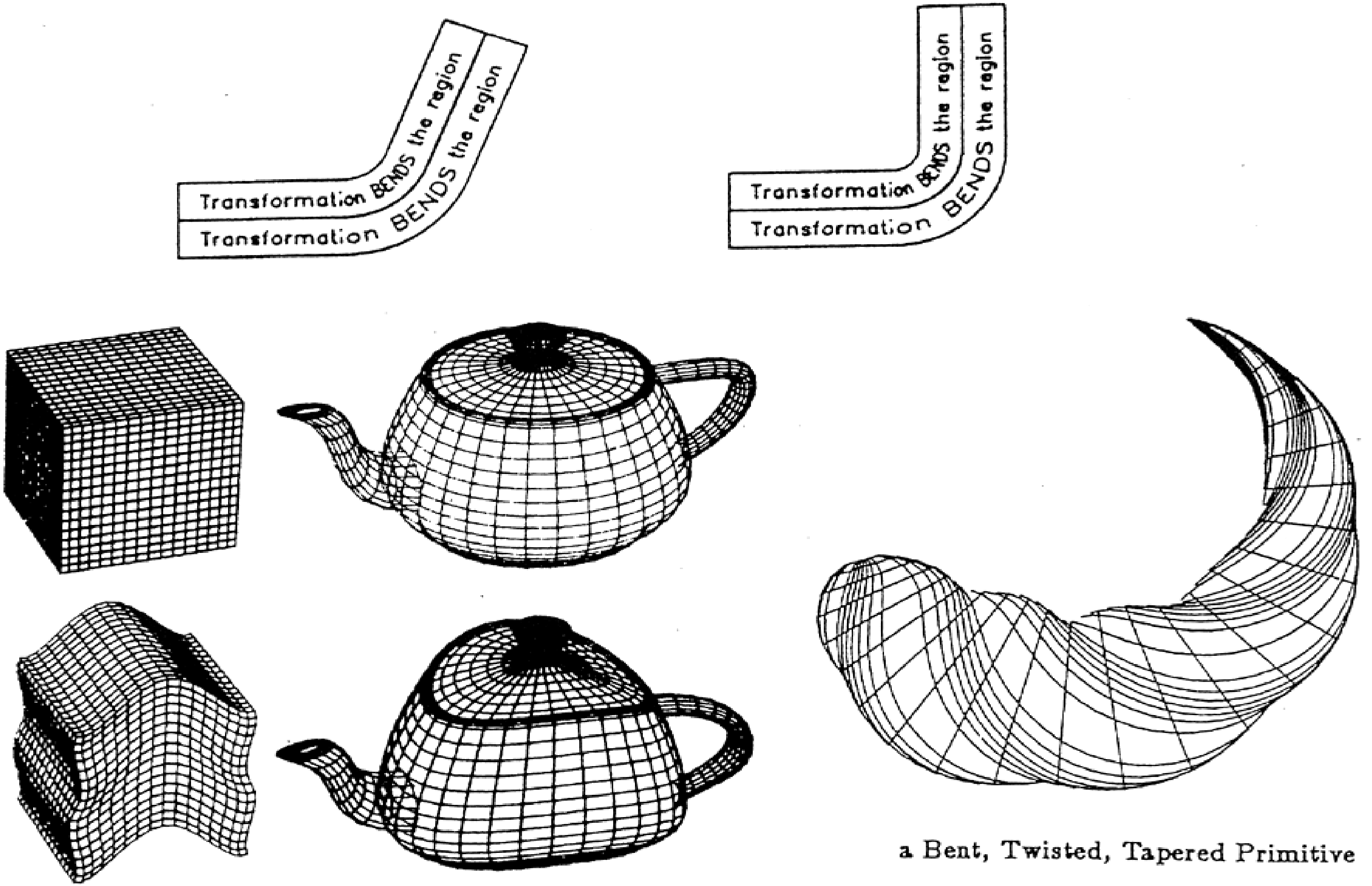
Inverse mapping

$$r(Z) = f(Z),$$
$$x = X/r,$$
$$y = Y/r,$$
$$z = Z$$



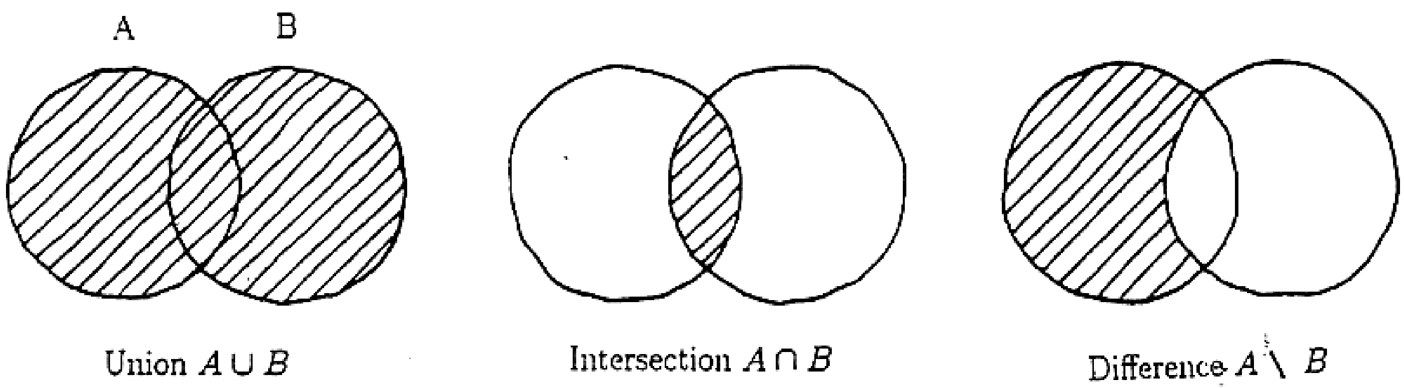
Images by A. Barr

Bending

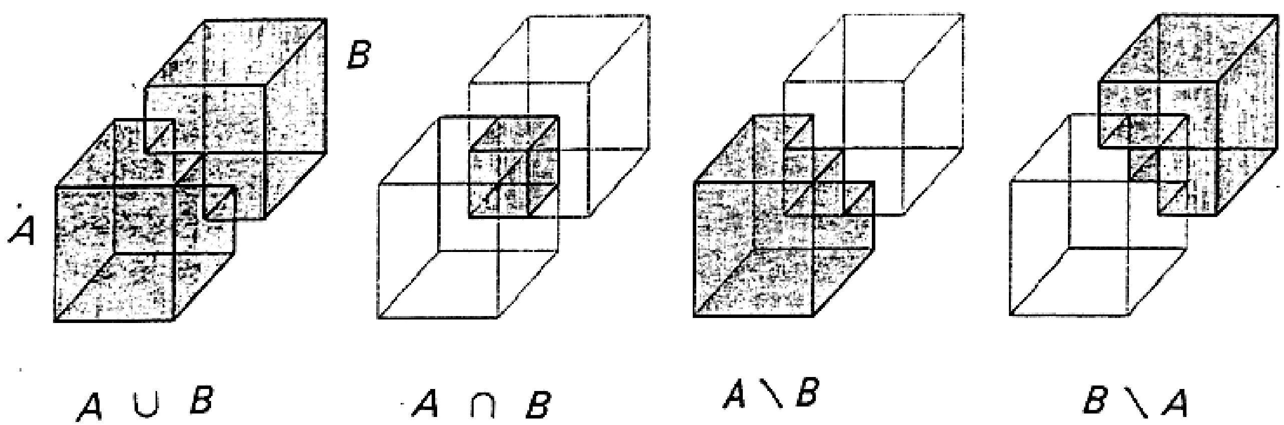


Images by A. Barr

Set-theoretic operations



A Venn diagram showing the operators of set-theory



Metamorphosis

Metamorphosis (morphing, warping, shape transformation) changes a geometric object from one given shape to another.

Polygonal objects

Two steps: 1) search for correspondence between points;
2) interpolation between two surfaces.

Problems:
- different number of points in two objects;
- constant topology (for example, how to transform a sphere in three intersecting tori?);
- possible self-intersections.

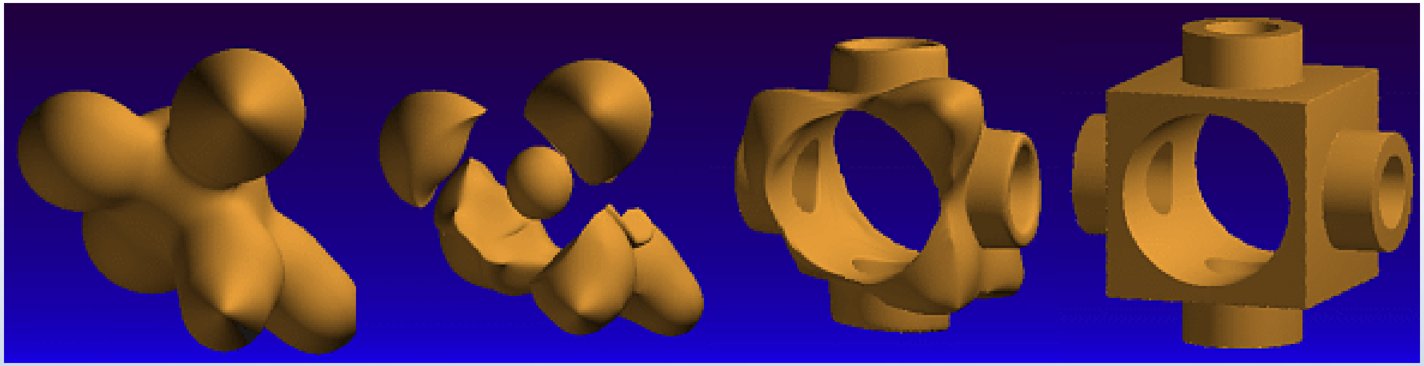
Implicit form

Metamorphosis is defined as a transformation between two functions. The simplest form is

$$\mathbf{f_3(X) = f_1(X)\ (1\text{-}t) + f_2(X)\ t\ ,}$$

where $0 \le t \le 1$.

Metamorphosis of implicit surfaces

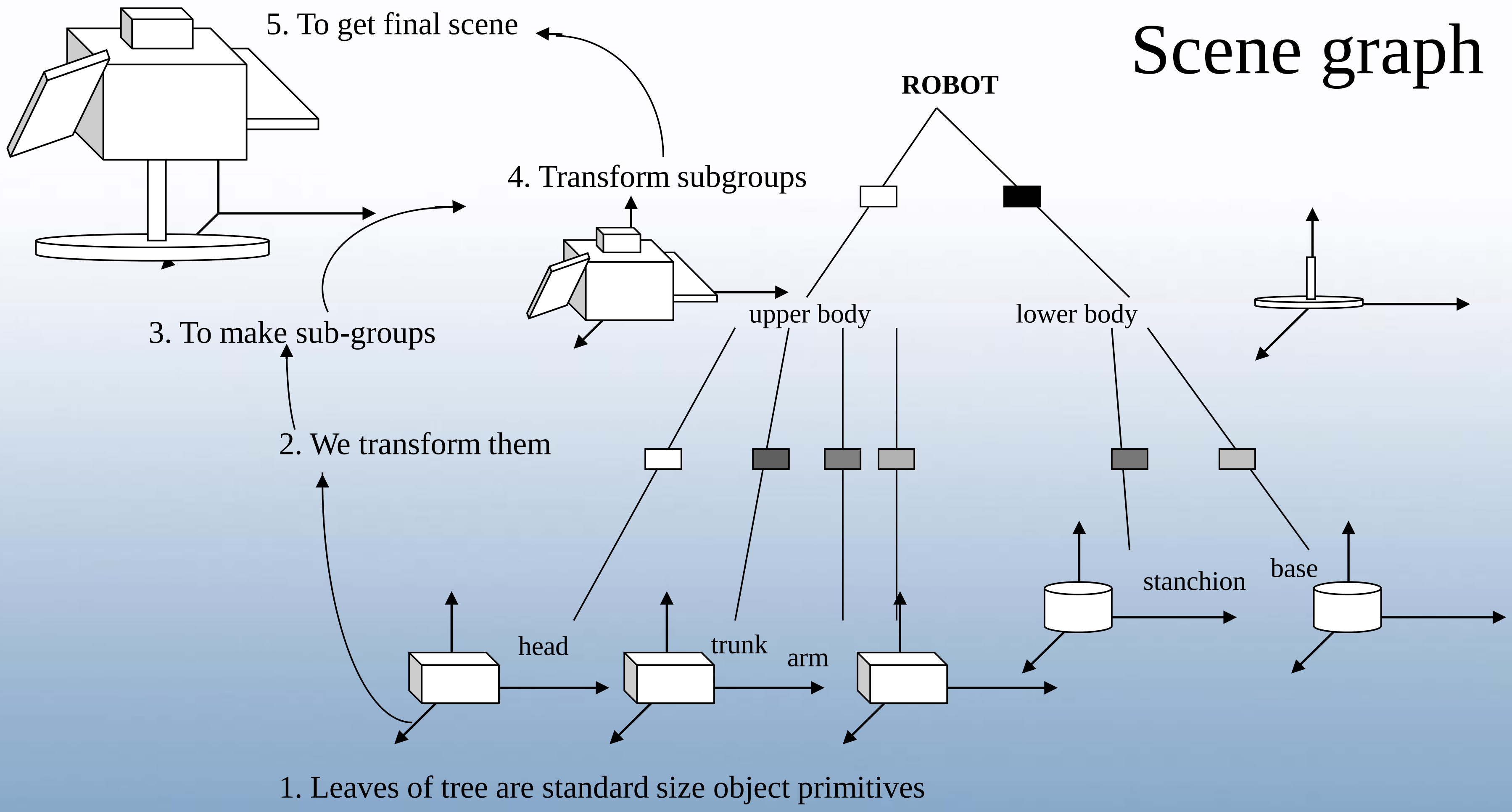


Can a constructive solid have an implicit surface?

Transforms in Scene Graphs

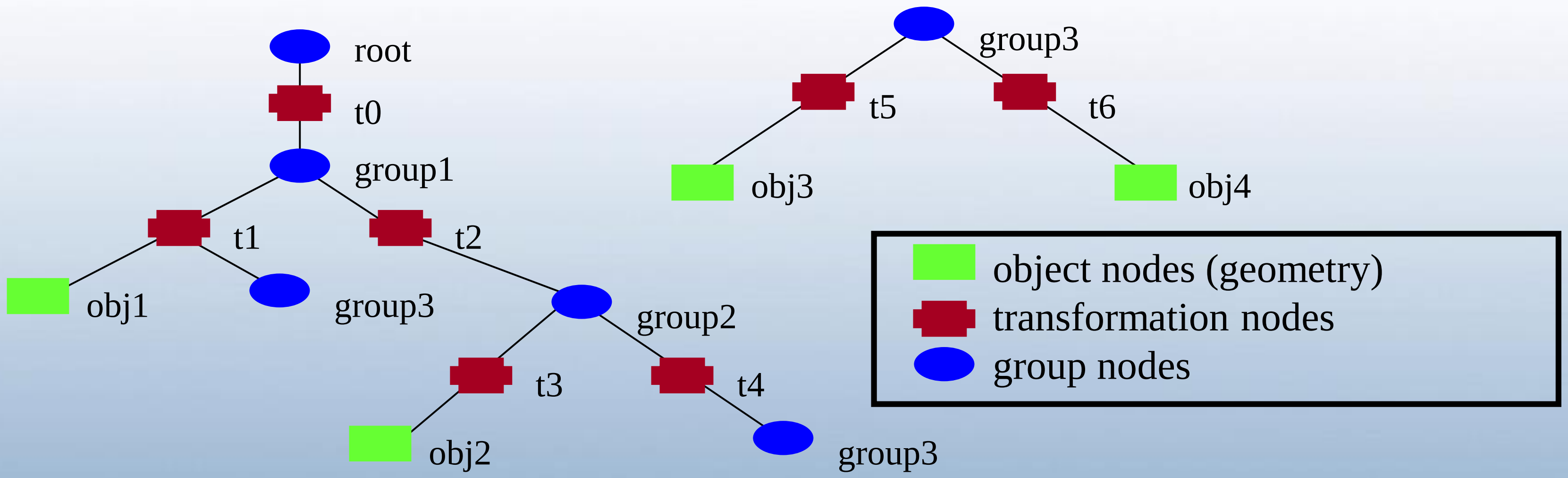
- 3D scenes are typically stored in a directed acyclic graph (DAG) called a *scene graph*
  - Open Scene Graph (used in the Cave)
  - Sun's Java3D™
  - x3D ™ (ex VRML ™)
- Typical scene graph format (there are hundreds of packages!)
  - objects (cubes, sphere, cone, polyhedra etc.) with basic defaults (located at the origin within unit box) stored as nodes
  - attributes (color, texture map, etc.) and transformations are also nodes in scene graph (labeled edges on slide 2 are an abstraction)

Transforms in Scene Graphs



Transforms in Scene Graphs

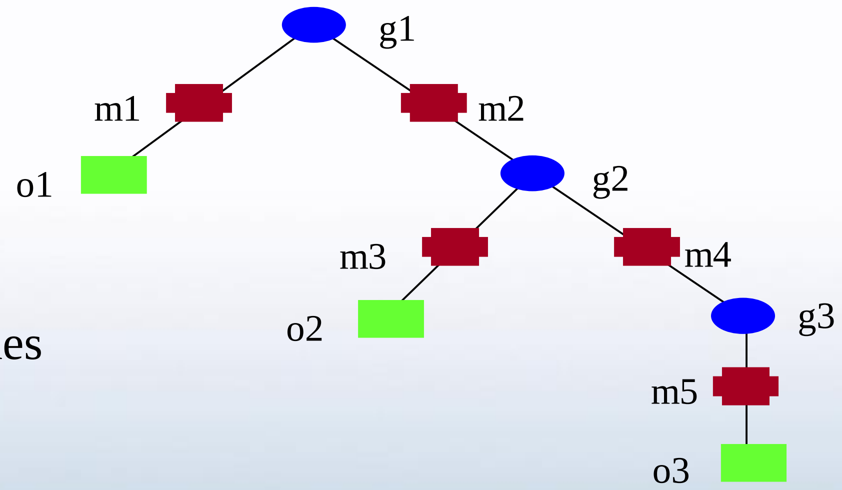
- In the scene graph below, transformation t0 will affect all objects, but t2 will only affect obj2 and one instance of group3 (which includes an instance of obj3 and obj4)
  - t2 doesn’t affect obj1, other instance of group3



- Note that if you want to use multiple instances of a sub-tree, such as group3 above, you must define it before it’s used
  - this is so that it’s easier to implement

Transforms in Scene Graphs

- An example:



g: group nodes

m: matrices of the transform nodes

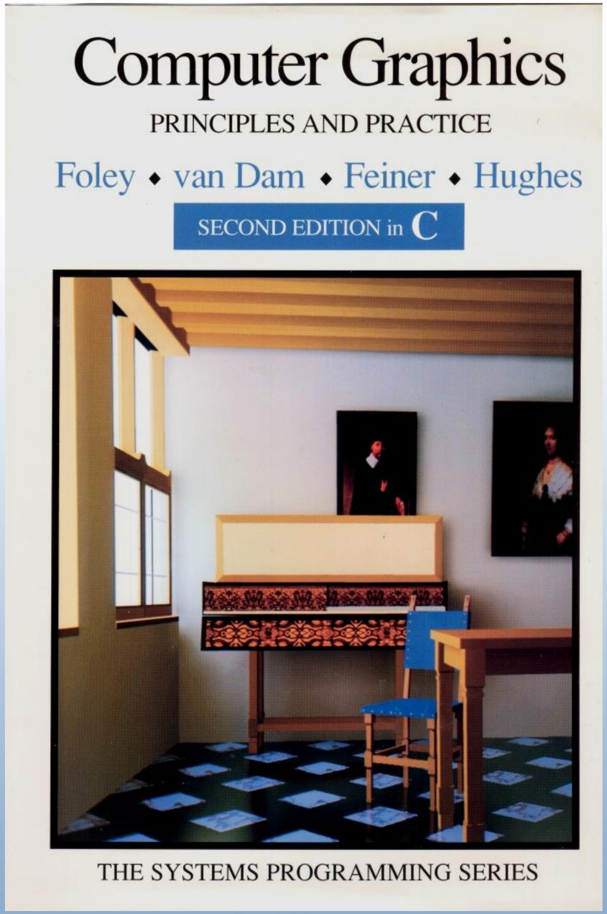
o: object nodes

- for o1, CTM = m1
- for o2, CTM = m2* m3
- for o3, CTM = m2* m4* m5
- for a vertex v in o3, its position in the world (root) coordinate system is:

CTM v = (m2*m4*m5)v

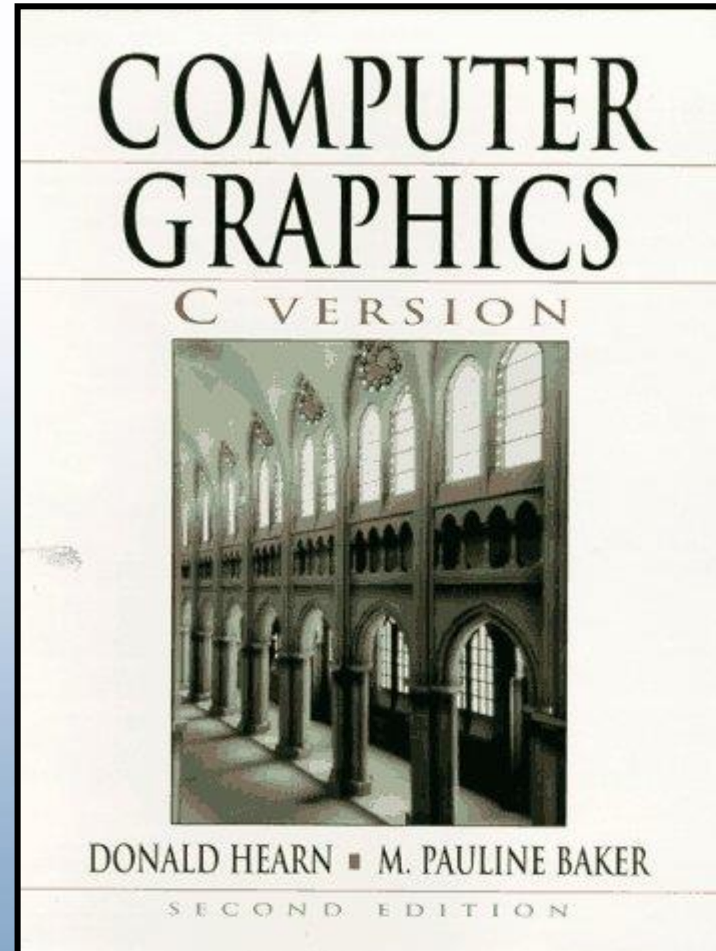
References

- James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer Graphics: Principles and Practice (2nd Edition in C), Addison-Wesley, Reading, MA, 1997.



References

Hearn Donald, Baker M.P., Computer Graphics, New York, Prentice Hall, 2nd Edition, 1994.



References

- I.N. Bronshtein, K.A. Semendyayev, G. Musiol, H. Muehlig, H. Mühlig, Handbook of Mathematics, Springer, 2003

- MathWorld, 2006 http://mathworld.wolfram.com

- Wikipedia, 2006 http://en.wikipedia.org/wiki