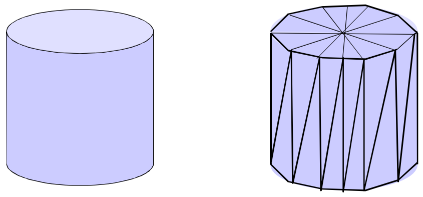
DEFINING OBJECTS - 3D REPRESENTATIONS

3D surface representation - continued
- Sweep functions

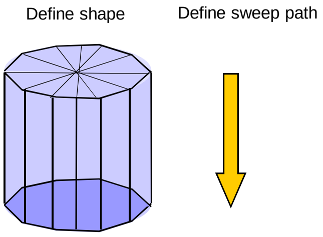
Elementary 3D transformations
- Translation
- Scaling
- Rotation

Surface representations

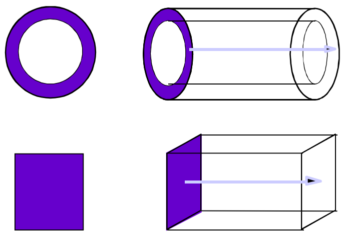
Object triangulation



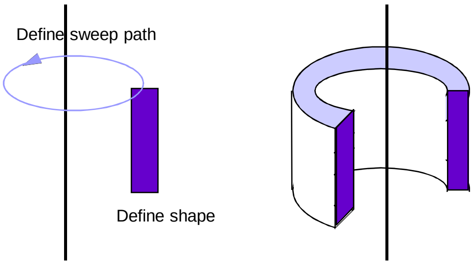
Sweep representations



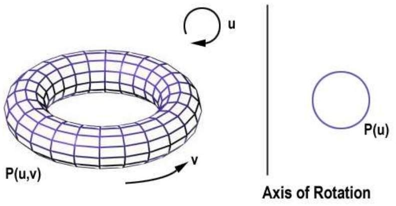
Sweep representations



Sweep representations

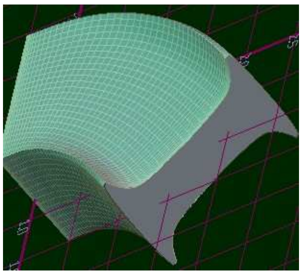


Sweep representations



Source: http://groups.csail.mit.edu/graphics/classes/6.837/F98/talecture/

Sweep representations



Source: http://www.h-dm.com/resources/CADZilla/manual/multi_section_sweep.jpg

7

Sweep functions: implementation

3D Transformations – quick revision

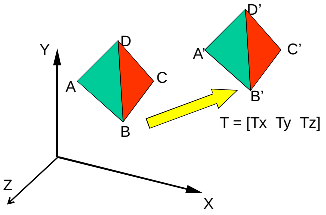
8

Basic transformations

- Translation (shift)
- Scaling
- Rotation

9

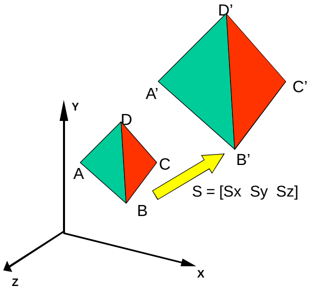
Translation



x' = x + Tx
y' = y + Ty
z' = z + Tz

10

Scaling
about the origin



x' = x * Sx
y' = y * Sy
z' = z * Sz

S > 1	- enlarge
0 < S < 1	- reduce
S < 0	- mirror

11

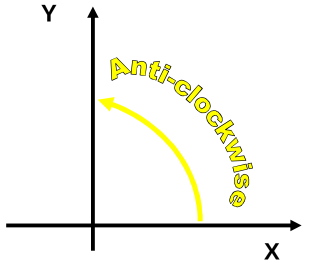
Rotation
in the right-handed coordinate system

Positive angle of rotation is counter-clockwise when the axis about which it occurs points toward the observer

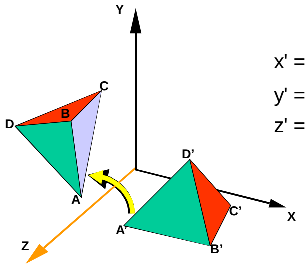
12

Positive angle of rotation for Z axis

Z axis points at the observer



Rotation about Z axis



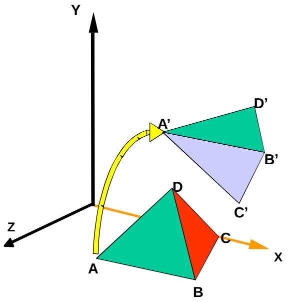
$x' = x \cdot \cos\varphi - y \cdot \sin\varphi$

$y' = x \cdot \sin\varphi + y \cdot \cos\varphi$

$z' = z$

Positive angle

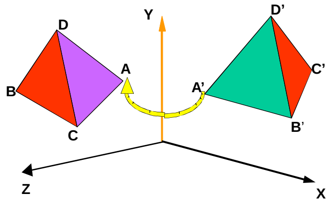
Rotation about X axis



- $y' = y \cdot \cos\varphi - z \cdot \sin\varphi$
- $z' = y \cdot \sin\varphi + z \cdot \cos\varphi$
- $x' = x$

Positive angle

Rotation about Y axis



- $z' = z \cdot \cos\varphi - x \cdot \sin\varphi$
- $x' = z \cdot \sin\varphi + x \cdot \cos\varphi$
- $y' = y$

Positive angle

Rotation

Axis of rotation is	Direction of positive rotation is
X	from Y to Z
Y	from Z to X
Z	from X to Y

Matrix representation
Homogeneous coordinates

- Common notation for ALL transformations
- Common computational mechanism for ALL transformations
- Simple mechanism for combining a number of transformations => computational efficiency

Homogeneous coordinates

- Point P = (x, y, z) represented by a vector

$$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = [x\ \ y\ \ z\ \ 1]^T$$

- Transformations

  All represented by a 4 x 4 matrix M

$$M = \begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Point transformation in homogeneous coordinates

- Implemented by matrix multiplication

  P’ = **M** · P

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Transformation matrices for elementary transformations

- 4 x 4 matrix
- Homogeneous coordinates
- Translation, scaling, rotation and perspective projection, all defined through matrices

See this website for a nice explanation of homogeneous coordinates:

http://www.devmaster.net/forums/showthread.php?t=2092

Translation

$x' = x + Tx$

$y' = y + Ty$

$z' = z + Tz$

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

Scaling

$x' = x * Sx$

$y' = y * Sy$

$z' = z * Sz$

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

Rotation about Z axis

$x' = x \cdot \cos\varphi - y \cdot \sin\varphi$

$y' = x \cdot \sin\varphi + y \cdot \cos\varphi$

$z' = z$

$$Rz = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about X axis

y' = y·cos φ - z·sin φ
z' = y·sin φ + z·cos φ
x' = x

$$Rx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi & 0 \\ 0 & \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about Y axis

z' = z·cosφ - x·sinφ
x' = z·sinφ + x·cosφ
y' = y

$$Ry = \begin{bmatrix} \cos\varphi & 0 & \sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\varphi & 0 & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Key concepts for 3D transformations in Java

Class:

- Transform3D

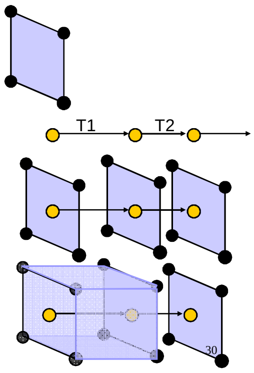
Example methods

- setTranslation, setRotation, set Scale
- SetTransform
- rotX, rotY, rotZ
- transform
- mul

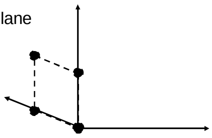
End of revision

Sweep functions: implementation

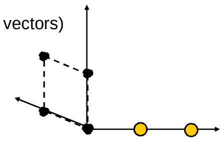
- Translational sweep
  - Define a shape as a polygon vertex table
  - Define a sweep path as a sequence of translation vectors
  - Translate the shape, continue building a vertex table
  - Define a surface table



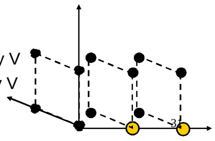
```
% Define a crossection of a unit cube in the YZ plane
crossection=[0  0  0 1
             0 1  0 1
             0 1  1 1
             0 0  1 1];
K=4;
```

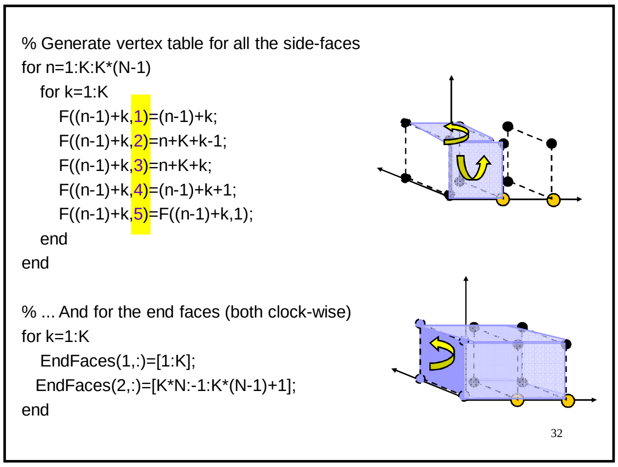


```
% Define a sweep path (relative displacement vectors)
path=[0 0 0
      1 0 0
      2 0 0];
```

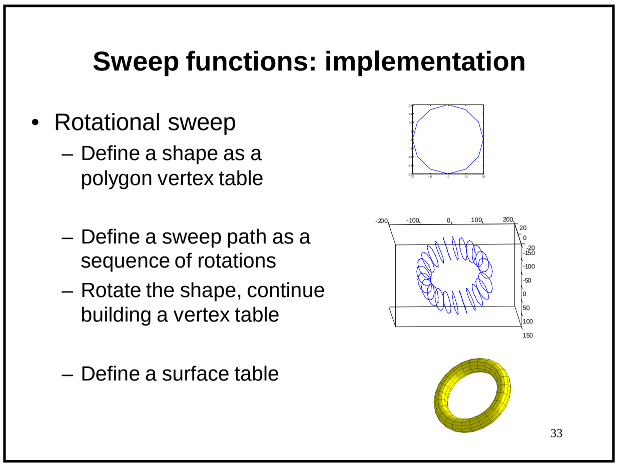


```
% Create a vertex table
for n=1:N
   Tm=makehgtform('translate',path(:,n));
   i1=K*(n-1)+1;   % current start index to array V
   i2=K*(n-1)+K;   % current end index to array V
   V(:,i1:i2)=Tm*C;
end
```



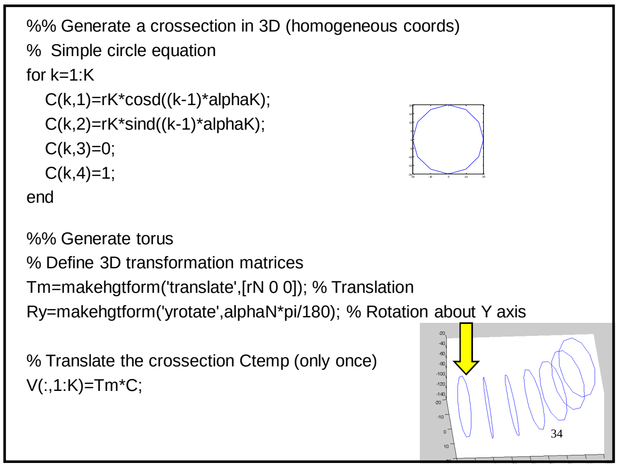


```
% Generate vertex table for all the side-faces
for n=1:K:K*(N-1)
   for k=1:K
      F((n-1)+k,1)=(n-1)+k;
      F((n-1)+k,2)=n+K+k-1;
      F((n-1)+k,3)=n+K+k;
      F((n-1)+k,4)=(n-1)+k+1;
      F((n-1)+k,5)=F((n-1)+k,1);
   end
end

% ... And for the end faces (both clock-wise)
for k=1:K
   EndFaces(1,:)=[1:K];
  EndFaces(2,:)=[K*N:-1:K*(N-1)+1];
end
```

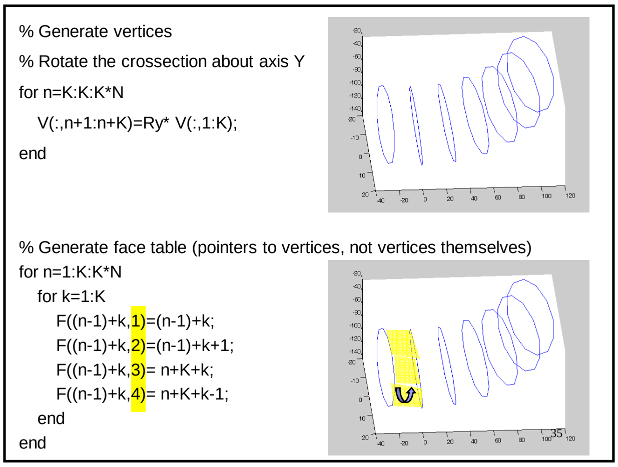


Sweep functions: implementation

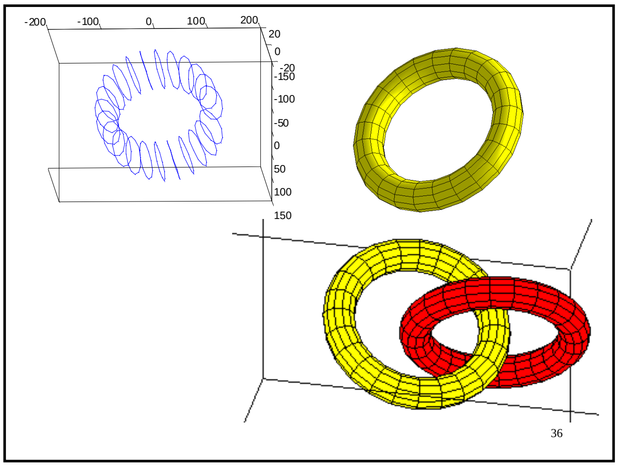
- Rotational sweep
  - Define a shape as a polygon vertex table
  - Define a sweep path as a sequence of rotations
  - Rotate the shape, continue building a vertex table
  - Define a surface table

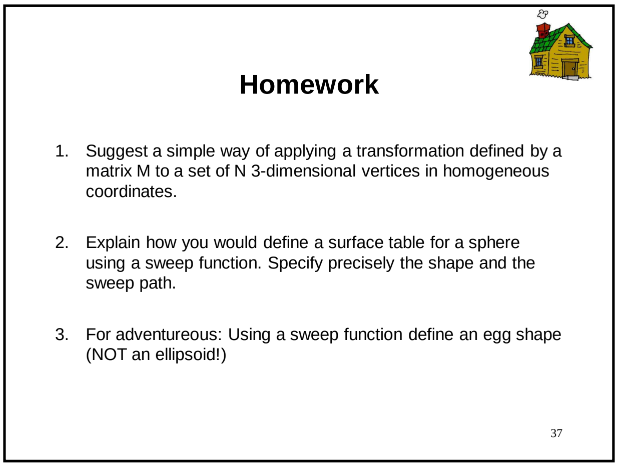


```
%% Generate a crossection in 3D (homogeneous coords)
%  Simple circle equation
for k=1:K
   C(k,1)=rK*cosd((k-1)*alphaK);
   C(k,2)=rK*sind((k-1)*alphaK);
   C(k,3)=0;
   C(k,4)=1;
end

%% Generate torus
% Define 3D transformation matrices
Tm=makehgtform('translate',[rN 0 0]); % Translation
Ry=makehgtform('yrotate',alphaN*pi/180); % Rotation about Y axis

% Translate the crossection Ctemp (only once)
V(:,1:K)=Tm*C;
```



```
% Generate vertices

% Rotate the crossection about axis Y

for n=K:K:K*N

   V(:,n+1:n+K)=Ry* V(:,1:K);

end


% Generate face table (pointers to vertices, not vertices themselves)
for n=1:K:K*N
   for k=1:K
      F((n-1)+k,1)=(n-1)+k;
      F((n-1)+k,2)=(n-1)+k+1;
      F((n-1)+k,3)= n+K+k;
      F((n-1)+k,4)= n+K+k-1;
   end
end
```





Homework

1. Suggest a simple way of applying a transformation defined by a matrix M to a set of N 3-dimensional vertices in homogeneous coordinates.
2. Explain how you would define a surface table for a sphere using a sweep function. Specify precisely the shape and the sweep path.
3. For adventureous: Using a sweep function define an egg shape (NOT an ellipsoid!)

Matlab exercise

- Extend the code in ‘ex2_torus.m’ to define and display five linked toruses in different colours. Matlab code for ‘ex2_torus.m’ is in file www.cs.bham.ac.uk/~exc/Teaching/Graphics/ex2_torus.m

  You will also need file ‘torus.m’ which is a function generating the face and the vertex table for a torus.
- Write Matlab code to generate a surface table for a sphere using a sweep function.

Reminder about Matlab tutorials

- http://www.cyclismo.org/tutorial/matlab/
  - Work through the tutorial should take you 2-3 hours.
- http://web.mit.edu/6.094/www/lecnotes/lec1.ppt
  - Ignore first five pages which have information relevant to the MIT course
- Matlab Help
  - Have a look at the “Programming” and “Graphics” sections

Next lecture

Height maps

Parametric surfaces