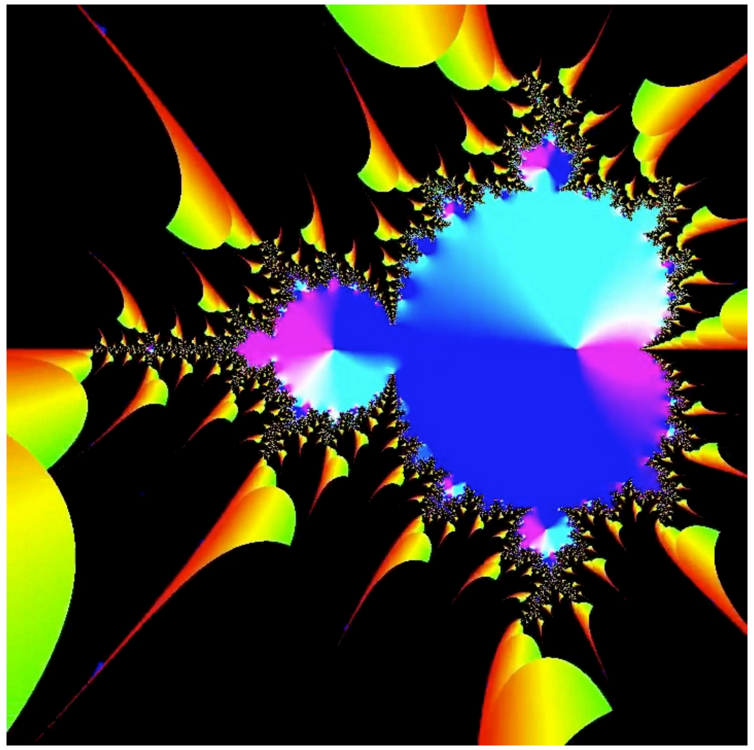
TSBK 07

Computer Graphics

Ingemar Ragnemalm, ISY



Lecture 3

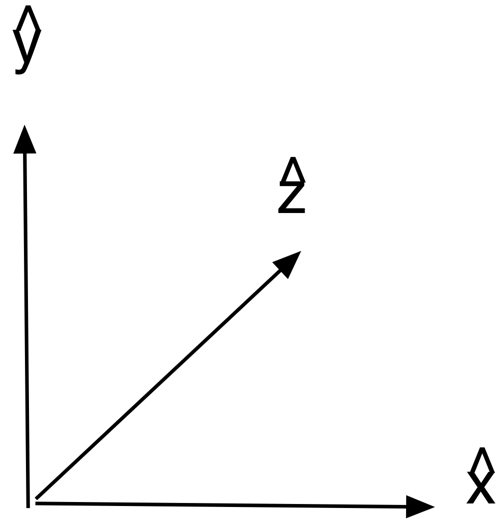
3D concepts

3D transformations

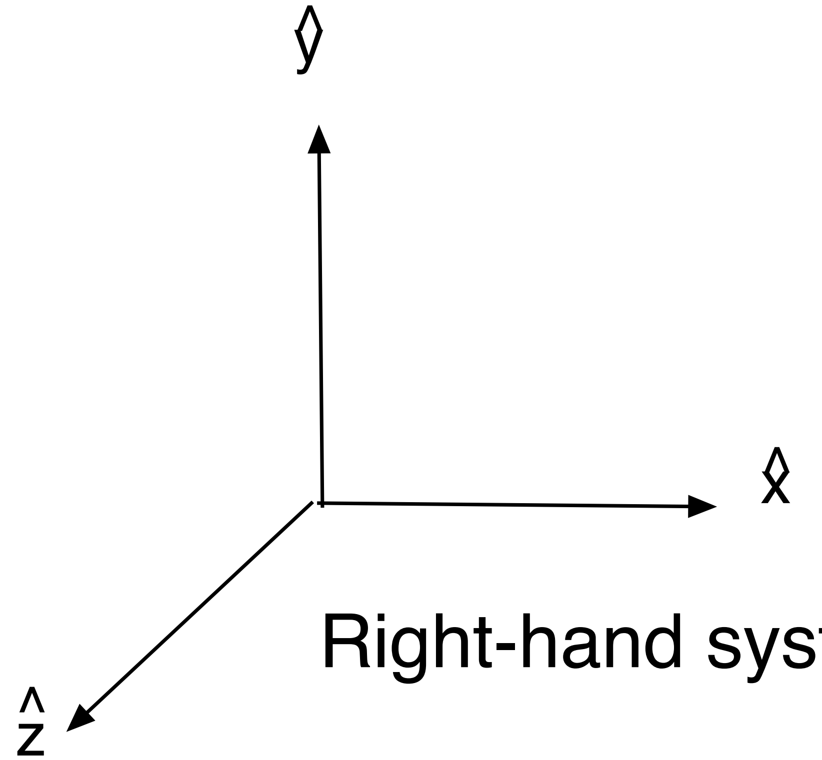
Viewing

Projection

3D coordinate system



Left-hand system



Right-hand system

3D point – (x, y, z) (Positional vector)

Directional vector – (x, y, z)

3D line $\mathbf{p} = \mathbf{p}_1 + \mu * \mathbf{d}$

or $\mathbf{p} = \mathbf{p}_1 + \mu * (\mathbf{p}_2 - \mathbf{p}_1)$

3D line segments e.g. $0 < \mu < 1$

3D plane $A*x + B*y + C*z + D = 0$

Properties of 3D planes:

$\mathbf{n} = (A,B,C)$ Normal vector

Most transformations are trivially expanded to 3D:

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

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

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

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

Rotation

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Around the x axis: $R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Around the y axis: $R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Around the z axis: $R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

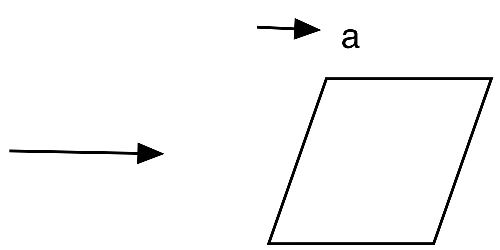
And some more

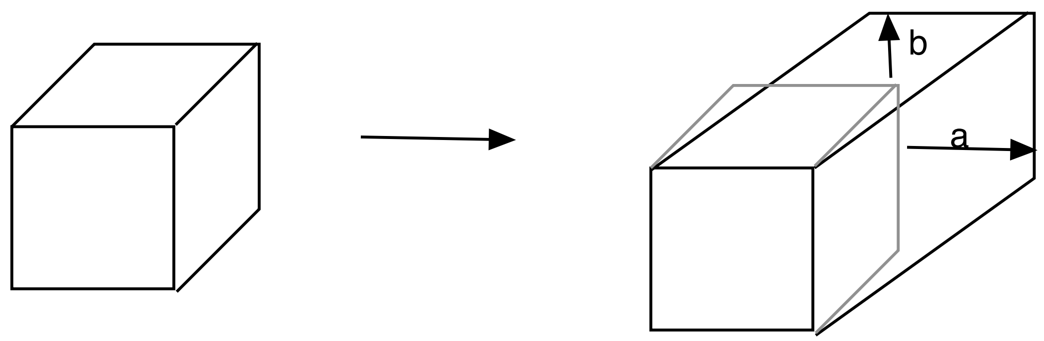
Mirroring: $M_X = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Mirroring: $M_X = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Shearing: $SH_X(a) = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Shearing: $SH_Z(a, b) = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$





In 3D we need a more complex sequence of transformations for:

- **3D viewing**
- **Camera placement**
- **Projection**

3D examples

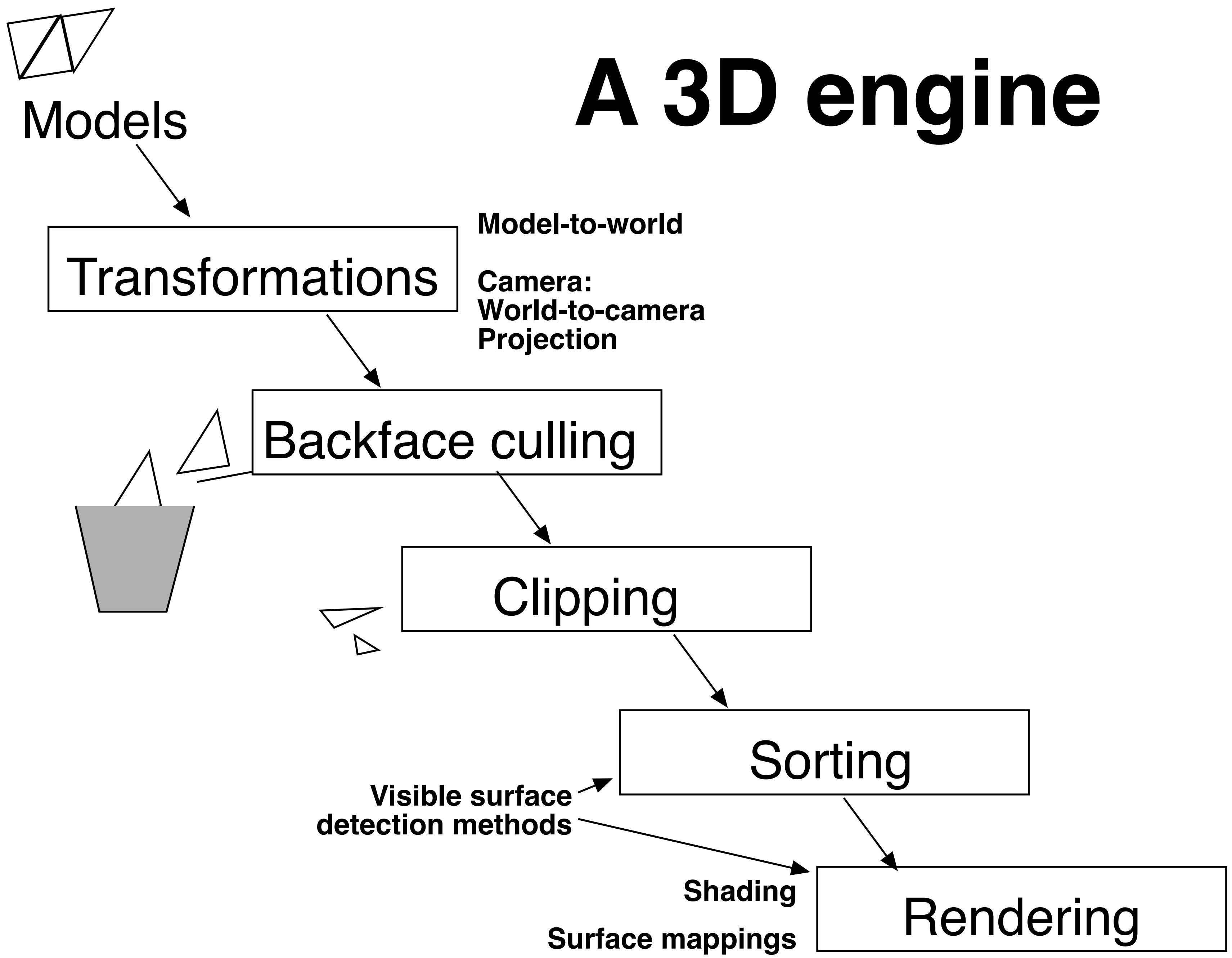
Not extremely different

- Models given with 3D coordinates

- Projection and ”look-at” matrices

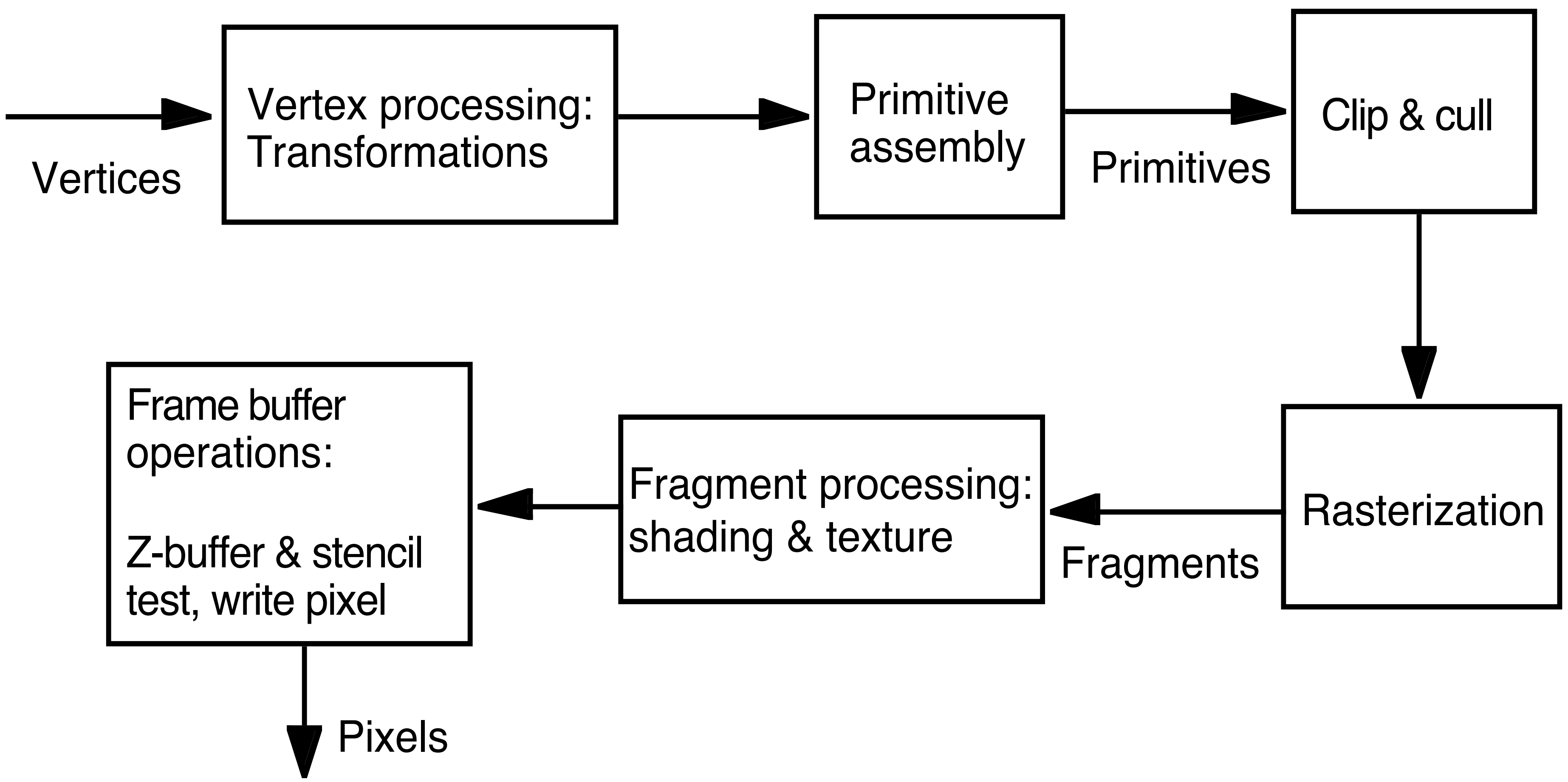
Color cube demo: Solid models more complex. Visible surface detection vital. (Future lecture.) Also color data per vertex.

A 3D engine



Models
Transformations
Model-to-world
Camera:
World-to-camera
Projection
Backface culling
Clipping
Sorting
Visible surface
detection methods
Rendering
Shading
Surface mappings

The OpenGL pipeline



Next lecture

The OpenGL pipeline

The OpenGL Shading Language

3D object repreentation

Illumination