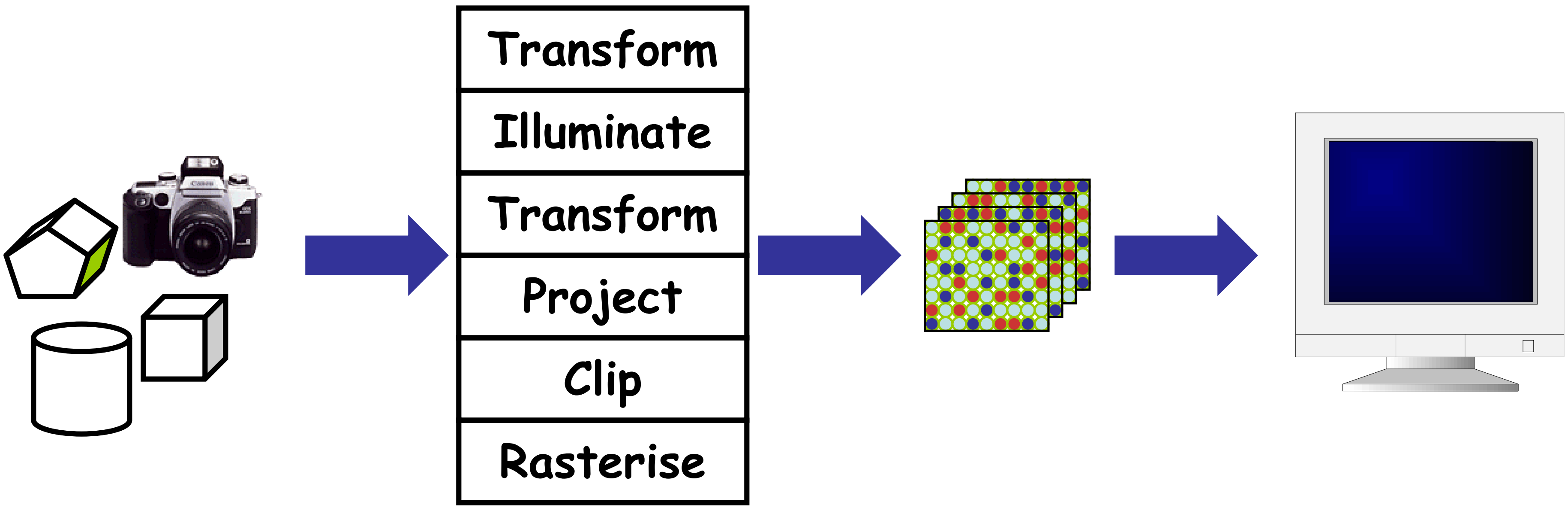
3D Graphics Programming Tools

The rendering pipeline
(Revision 1)

(Go to **www.menti.com** and use the code **1179 3402** to ask me questions)

Queen Mary
University of London

Rendering 3D scenes



model & camera parameters

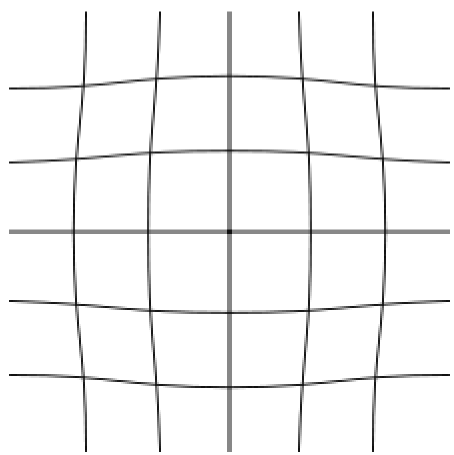
rendering pipeline

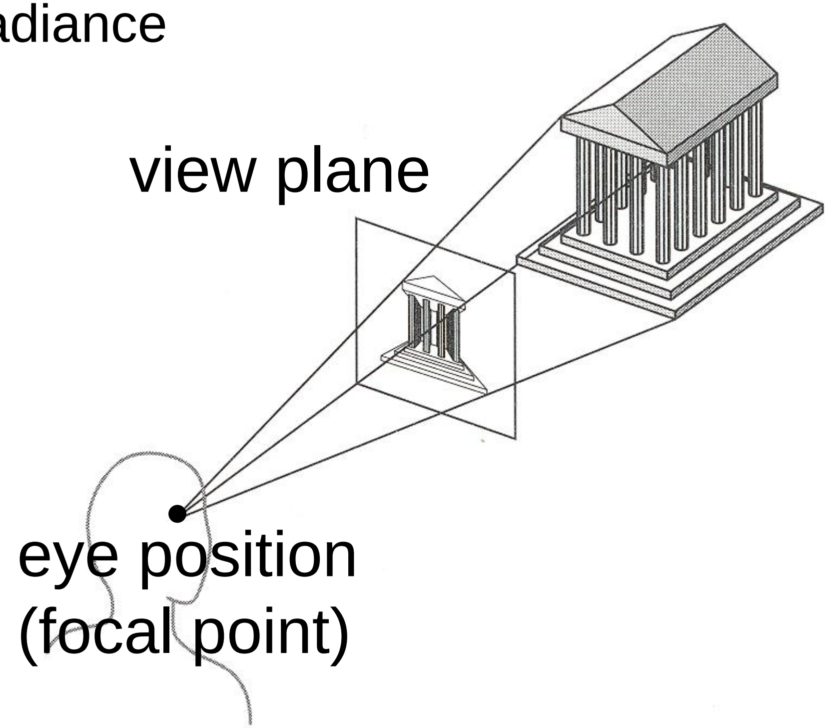
framebuffer

display

Camera models

- Most common model: pin-hole camera
  - All captured light rays arrive along paths toward a focal point without lens distortion, everything is in focus
  - Sensor response proportional to radiance
  - Note: other models consider:
    - » depth of field
    - » motion blur
    - » lens distortion

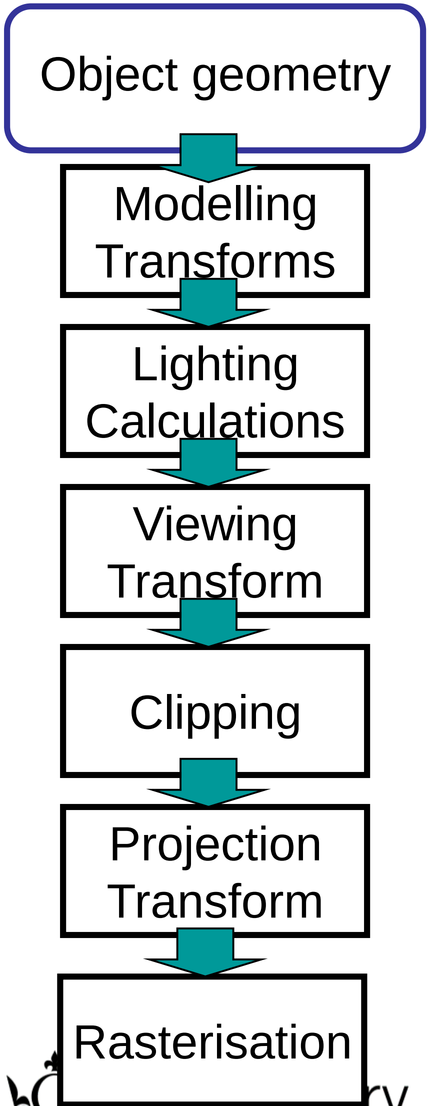




Queen Mary
University of London

The rendering pipeline

- Move models
- Illuminate
- Move camera
- Project to display
- Clip
- Rasterise



Queen Mary
University of London

The rendering pipeline: 3D

Object geometry

Modelling Transforms

Lighting Calculations

Viewing Transform

Clipping

Projection Transform

Rasterisation

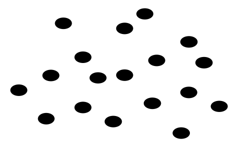
To create 3D models for objects

Queen Mary
University of London

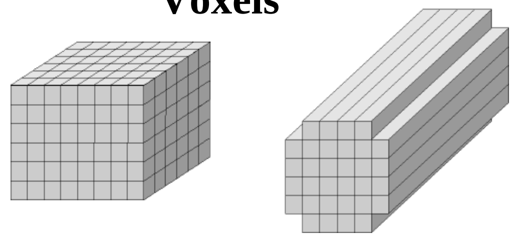
Geometric Modelling – Summary

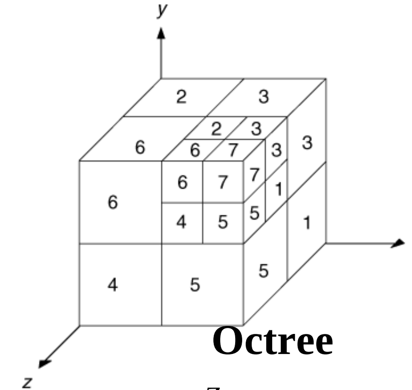
Point-based, Surface-based, Constructive

Point cloud



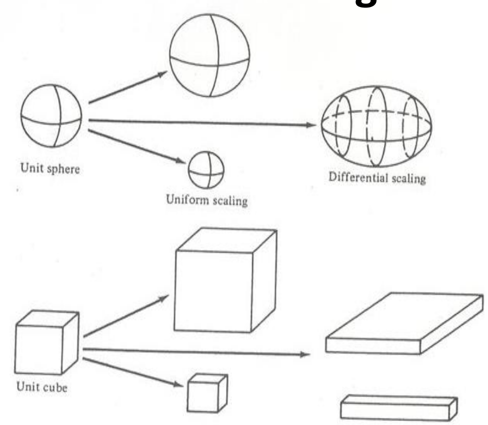
Voxels



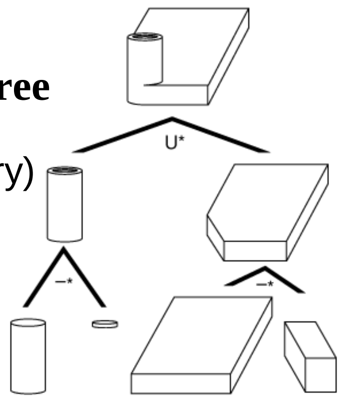


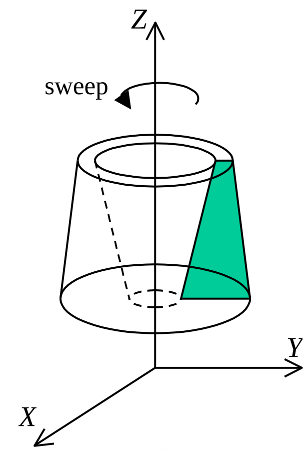
Octree

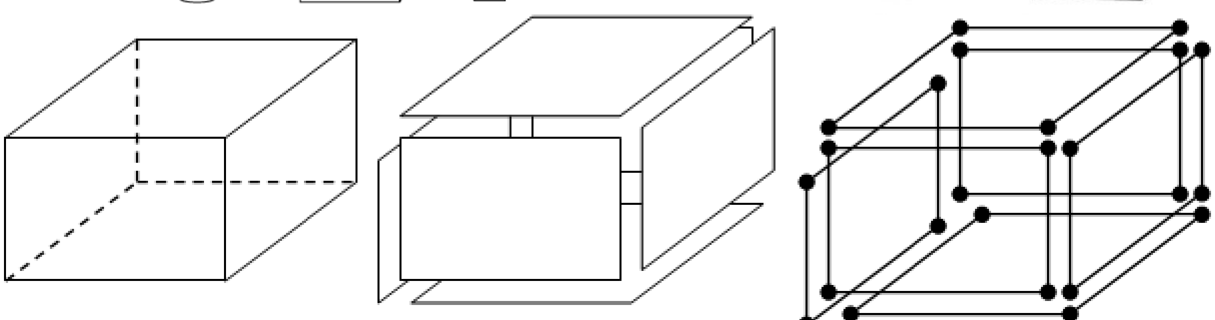
Primitive instancing



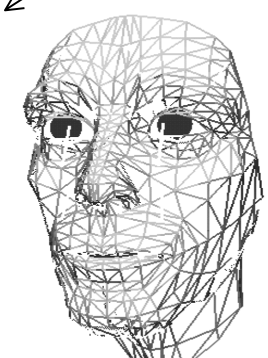
CSG Tree
(Constructive
solid geometry)







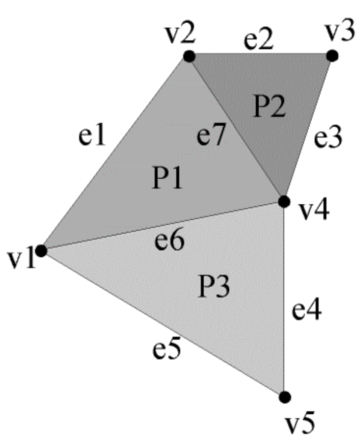
Boundary representations (b-reps)

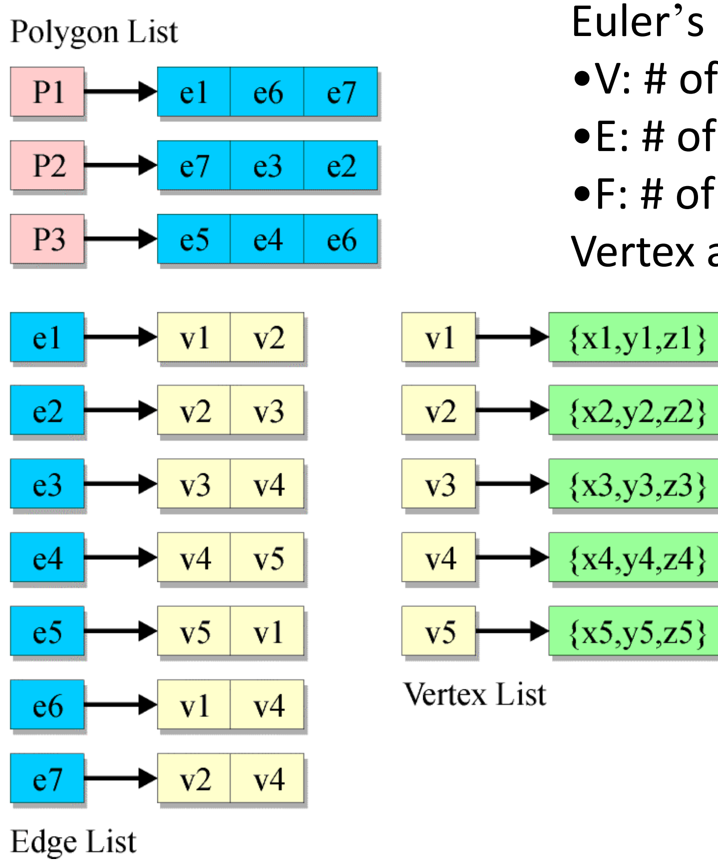


Polygon Mesh

Representing polygon meshes

- Vertex list → locations of the vertices, geometric info
- Edge list → indexes into end vertices of edges, topological info
- Face list → indexes into vertices and normal lists, topological info
- Normal list → directions of the normal vectors, orientation info





Euler's Formula: $V – E + F = 2$

- V: # of vertices
- E: # of edges
- F: # of faces

Vertex and face lists are enough

Queen Mary
University of London

The rendering pipeline: 3D

- Object geometry
- Modelling Transforms
- Lighting Calculations
- Viewing Transform
- Clipping
- Projection Transform
- Rasterisation

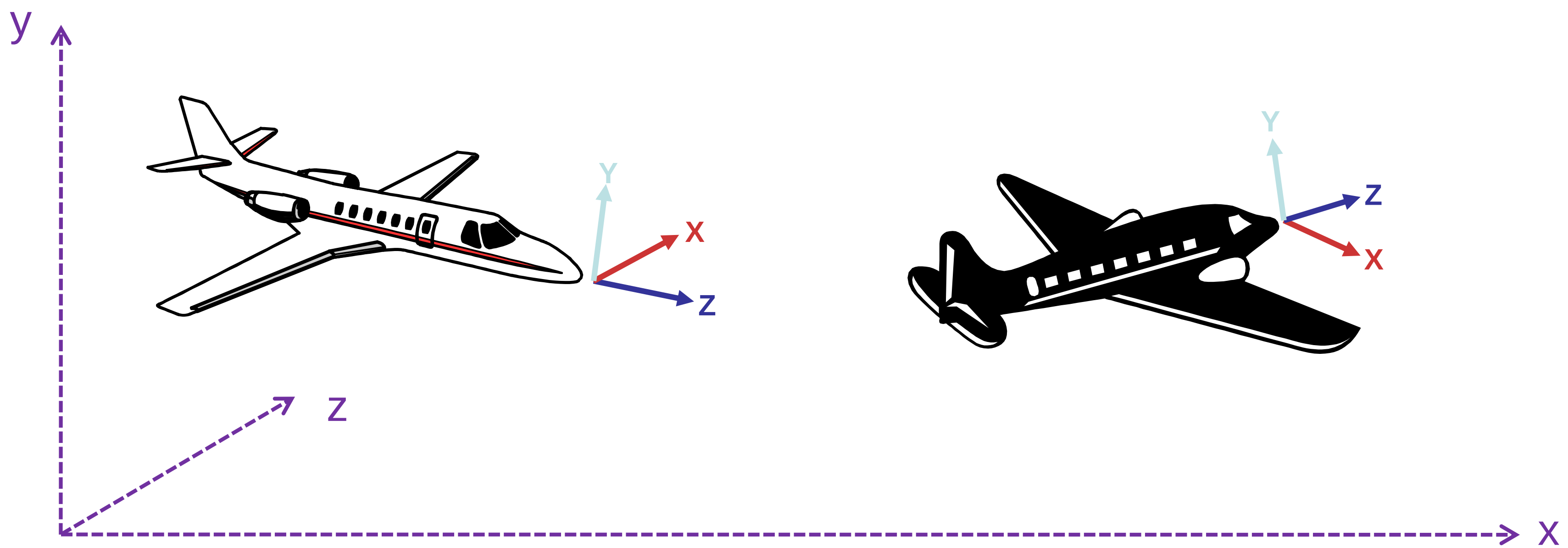
Transformation: Position models in WCS
Result: all vertices of scene in shared 3D “world” coordinate system (WCS)

Transformations

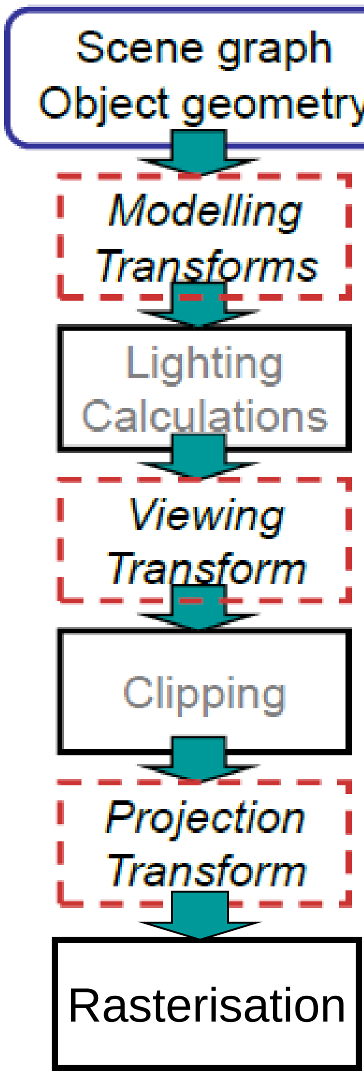
- Transformations → used in three ways
  - modelling transforms
    - position models
  - viewing transforms
    - position the camera
  - projection transforms
    - change the type of camera

Modelling Transformations

- Modelling transforms
  - Size, place, scale, and rotate objects and parts of the model with respect to each other
  - Object coordinates → world coordinates



Geometric Transformations



Transformation : converting to a representation in a different coordinate system.

From object coordinates to world coordinates

From world coordinates to view reference (or camera or eye) coordinates

From camera coordinates to window (or screen) coordinates

Queen Mary
University of London

Geometric Transformations – 2D

- **For points written in homogeneous coordinates:** $\boldsymbol{v} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

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

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

- **Rotation:** $\boldsymbol{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \boldsymbol{v}' = \boldsymbol{R}(\theta)\boldsymbol{v}$

Geometric Transformations – 3D

- **For points in homogeneous coordinates:**
  (x, y, z) → (x, y, z, 1) → (wx, wy, wz, w)

$$\boldsymbol{v} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \qquad \boldsymbol{v}' = \begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix}$$

- **Translation:**

$$T(dx, dy, dz) = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **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:**

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

about x-axis

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

about y-axis

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

about z-axis

The rendering pipeline: 3D

Object geometry

Modelling Transforms

Lighting Calculations

Viewing Transform

Clipping

Projection Transform

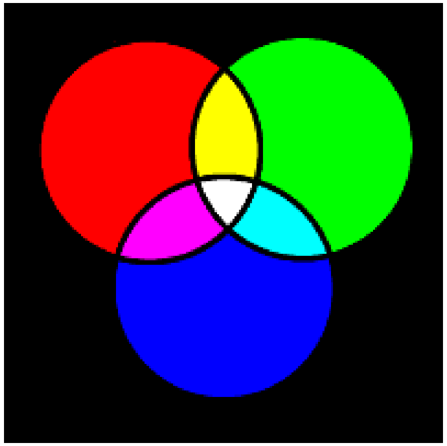
Rasterisation

Result: all geometric primitives are illuminated

Queen Mary
University of London

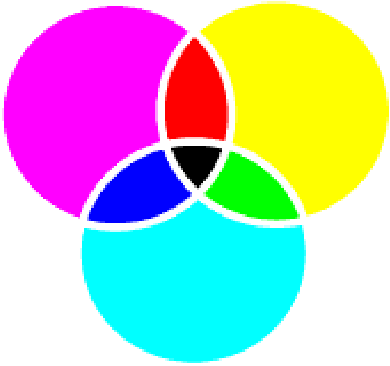
Colour Models

Additive (RGB)



$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

Subtractive (CMY)



The CMYK model

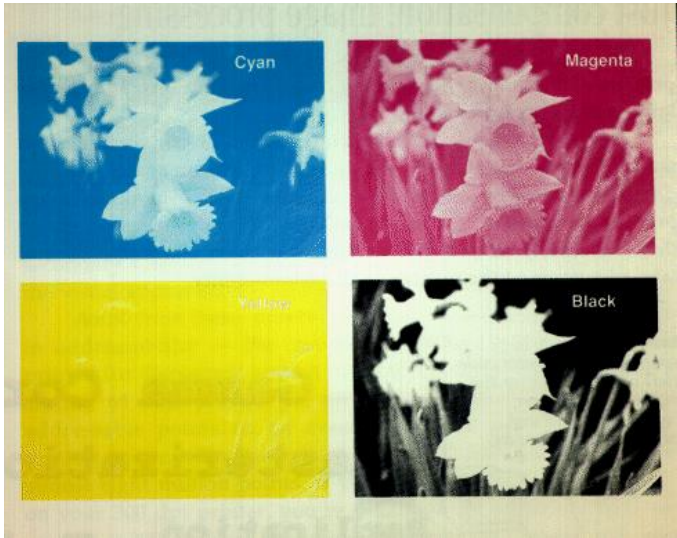
- Cyan, Magenta, Yellow & Black
- Used in printers
- Richer black
- Black ink is cheap

$$K := \min(C, M, Y)$$

$$C := C - K$$

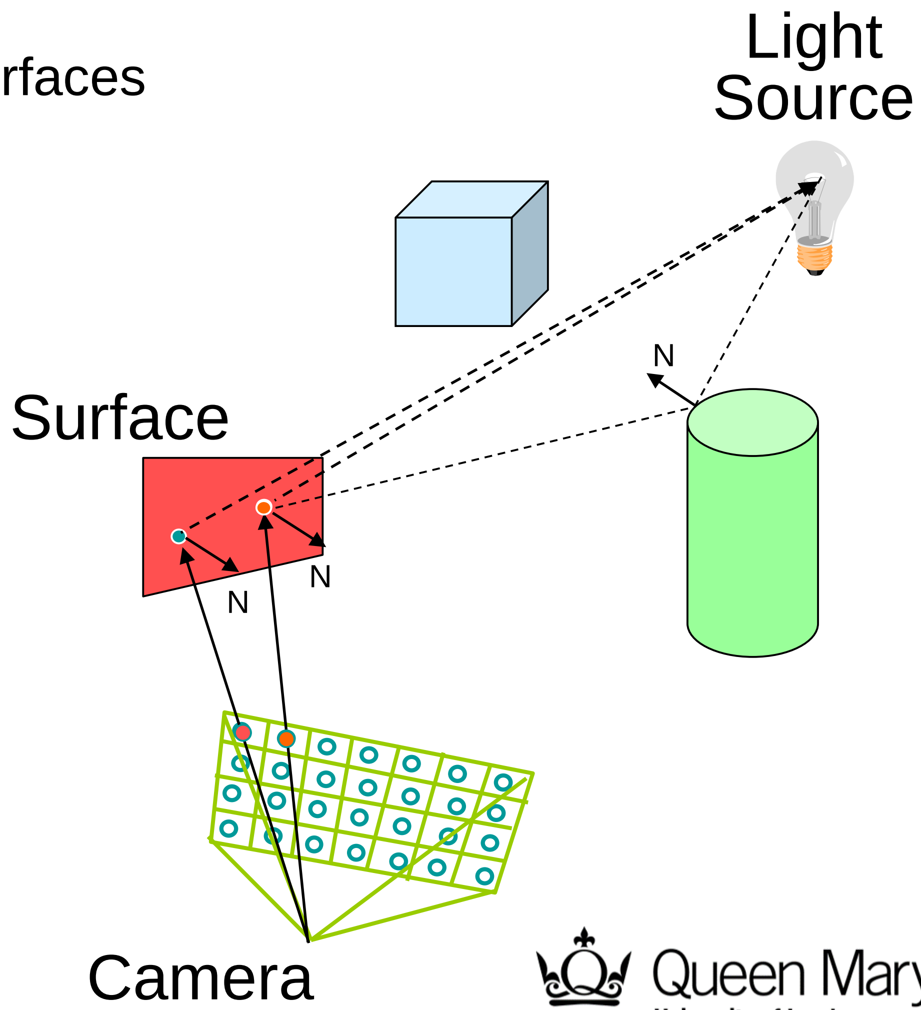
$$M := M - K$$

$$Y := Y - K$$



Lighting simulation

- Lighting parameters
  - Emission at light sources
  - Reflectance/Scattering at surfaces
  - Reception at the camera
- Direct illumination
  - Ambient
  - Diffusive
  - Specular
- Global/Indirect illumination
  - Radiosity
  - Ray tracing



Queen Mary
University of London

Lighting modelling: 3 contributions

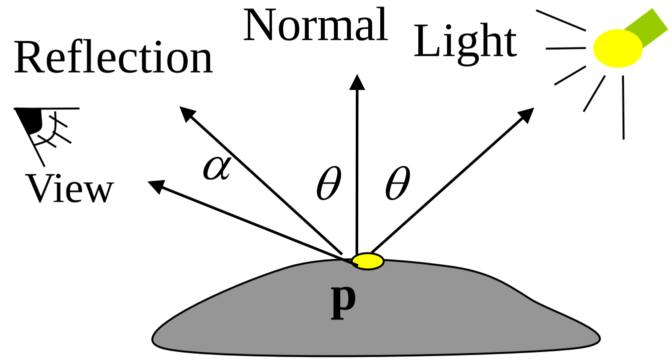
$$I = I_a + f_{att}(I_d + I_s) = I_a \cdot k_a + f_{att} I_l (k_d \cdot \cos\theta + k_s \cdot \cos^n \alpha)$$

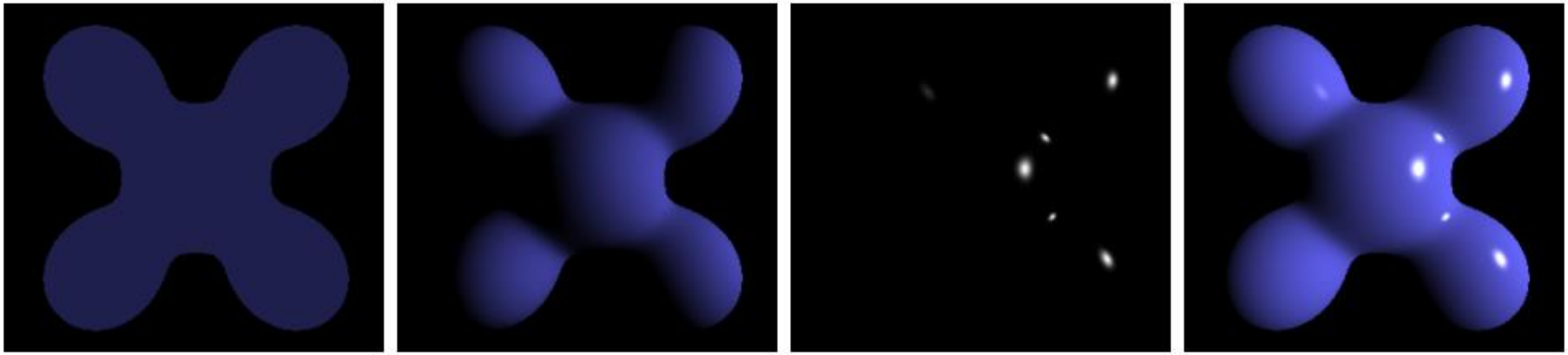
specular reflection

diffusive reflection

attenuation factor

ambient refection





Ambient + **Diffuse** + **Specular** **Combined illumination**

Multiple Colour Light Sources

$$I_{\lambda} = I_{a\lambda} \cdot k_a + \sum_j f_{att_j} I_{l\lambda_j} (k_d \cdot \cos\theta_j + k_s \cdot \cos^n \alpha_j)$$

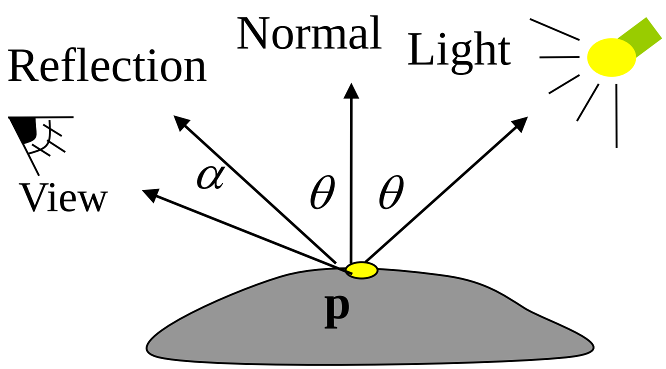
$$= I_{a\lambda} \cdot k_a + \sum_j f_{att_j} I_{l\lambda_j} \left(k_d \cdot (\boldsymbol{N} \cdot \boldsymbol{L}_j) + k_s \cdot (\boldsymbol{R}_j \cdot \boldsymbol{V})^n \right)$$

view vector

reflection direction

lighting vector

normal of surface



Householder reflection:

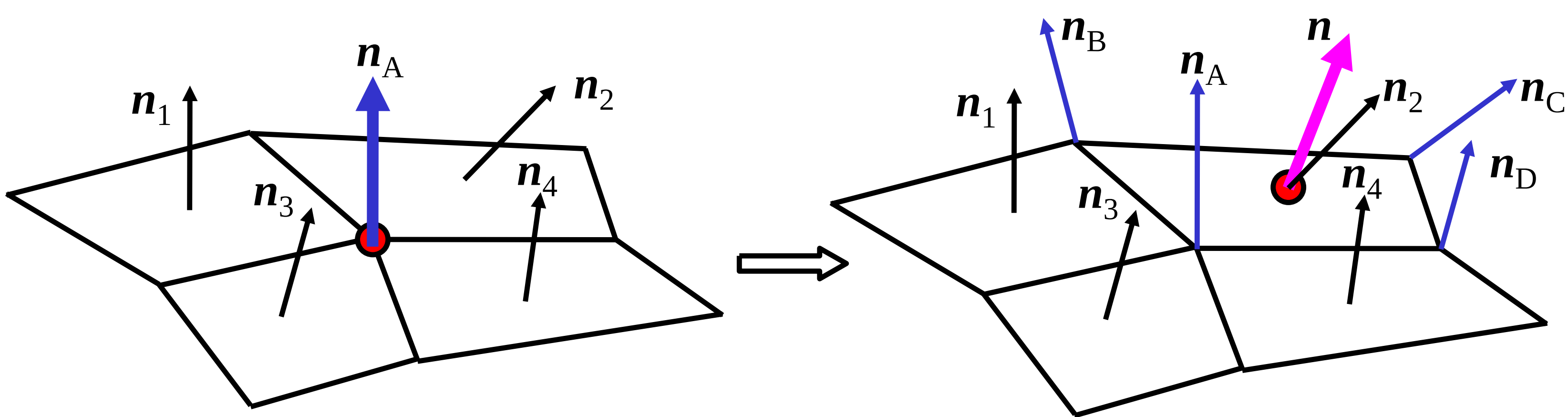
$$\overrightarrow{\boldsymbol{R}} = \boldsymbol{H} \cdot \overrightarrow{\boldsymbol{L}} \quad where \quad \boldsymbol{H} = \boldsymbol{I} - 2\overrightarrow{\boldsymbol{N}} \cdot \overrightarrow{\boldsymbol{N}}^T$$

Queen Mary
University of London

Shading models

Interpolation:

- Flat (Constant) Shading (**Nearest Neighbour Interpolation** of the illumination) (Mach band effect, specular highlights)
- Gouraud shading (Linear Interpolation of the **illumination**)
- Phong shading (Linear Interpolation of the **normal vectors**)



$\boldsymbol{n}_A = (\boldsymbol{n}_1 + \boldsymbol{n}_2 + \boldsymbol{n}_3 + \boldsymbol{n}_4) / 4$

$\boldsymbol{n} = a\boldsymbol{n}_A + b\boldsymbol{n}_B + c\boldsymbol{n}_C + d\boldsymbol{n}_D$

Cross product of two edge vectors for the face normal
Interpolation with face normals for the vertex normal
Interpolation with vertex normals for all the normals across faces

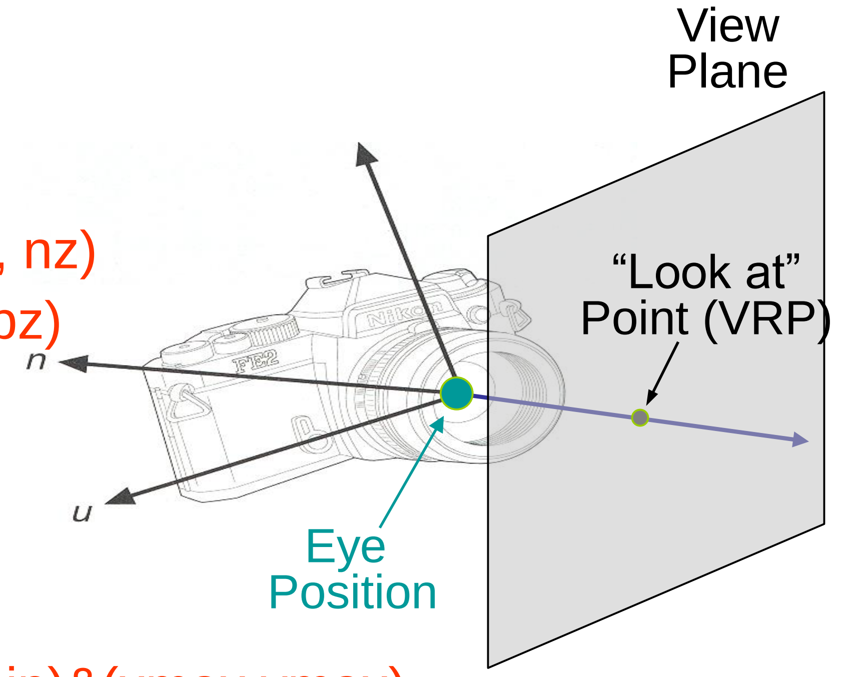
The rendering pipeline: 3D

- Object geometry
- Modelling Transforms
- Lighting Calculations
- Viewing Transform
- Clipping
- Projection Transform
- Rasterisation

Result: scene vertices in 3D “view” or “camera” coordinate system
(World Coordinates => View Reference Coordinates)

Camera parameters

- Position
  - COP(eye position) (px, py, pz)
- Orientation
  - VPN(or view direction) (nx, ny, nz)
  - View up direction (upx, upy, upz)
- View window (Aperture)
  - Field of view (xfov, yfov)
  - Or window size (width, height)
  - Or window rectangle (xmin,ymin)&(xmax,ymax)



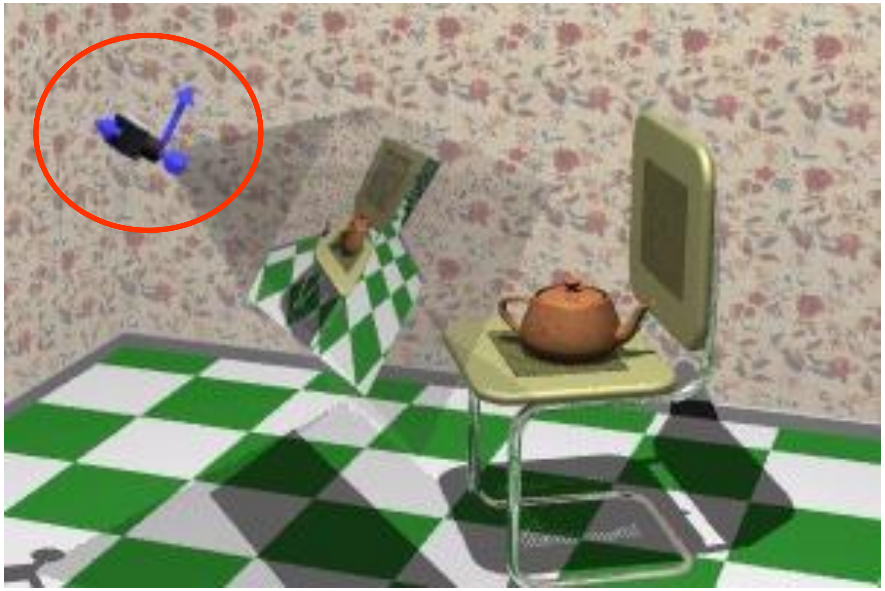
Queen Mary
University of London

Viewing Transformations

- Viewing transform
  - rotate & translate the world to position directly in front of the camera
    - typically place camera at origin
    - typically looking down along n axis
  - world coordinates → view reference coordinates

Viewing transformations

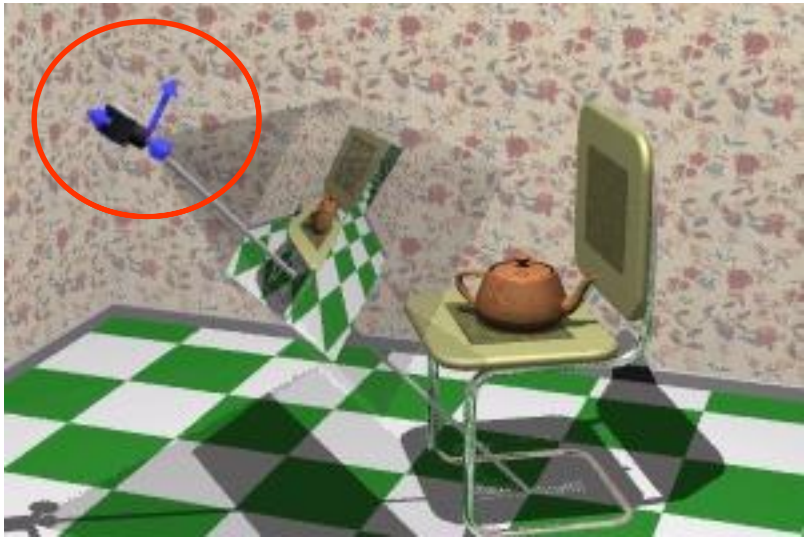
- Create a camera-centered view
  - camera is at origin
  - camera is looking along negative n-axis
  - camera's 'up' is aligned with v-axis



2 basic steps

- Translate to align origins

$$T = \begin{bmatrix} 1 & 0 & 0 & -VRP_x \\ 0 & 1 & 0 & -VRP_y \\ 0 & 0 & 1 & -VRP_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



- Align the two coordinate frames by rotation

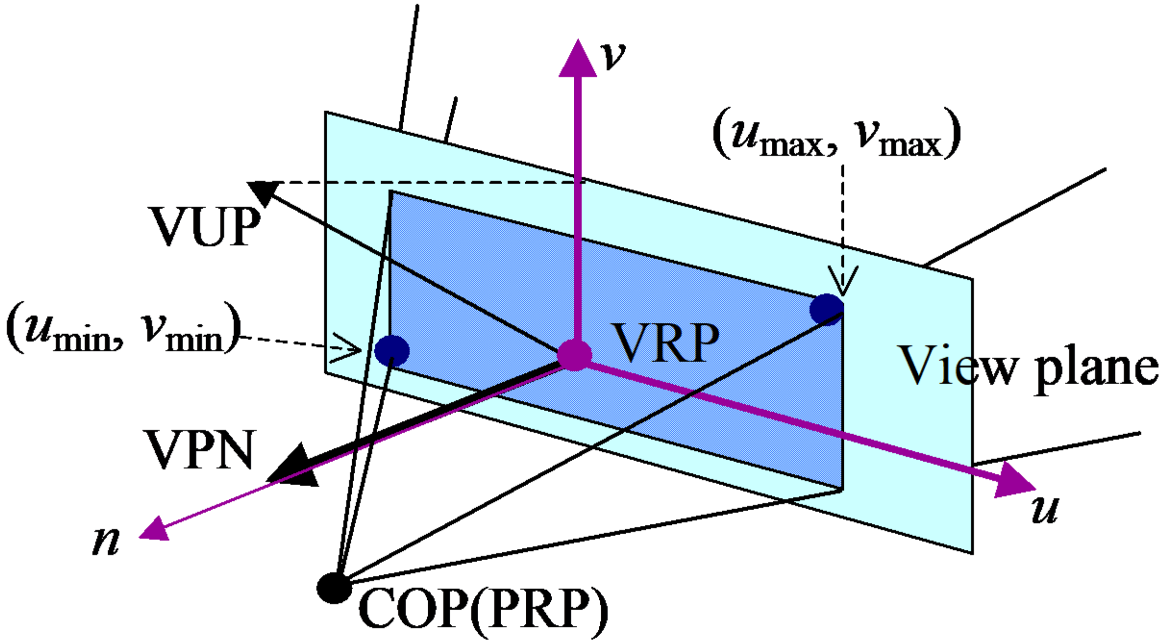
$$R = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Queen Mary
University of London

Creating camera coordinate space

View Reference Coordinate (VRC) System

- **View reference point (VRP, as the origin)**
- **Three orthogonal axes:**
  - **- VPN is one axis (*n*-axis).**
  - **- The second axis (*v*-axis): projection of *view-up vector* (VUP) onto the view plane.**
  - **- The third axis (*u*-axis) can be easily found in the right-handed coordinate system.**



$$n = \frac{\mathrm{VPN}}{|\mathrm{VPN}|}$$

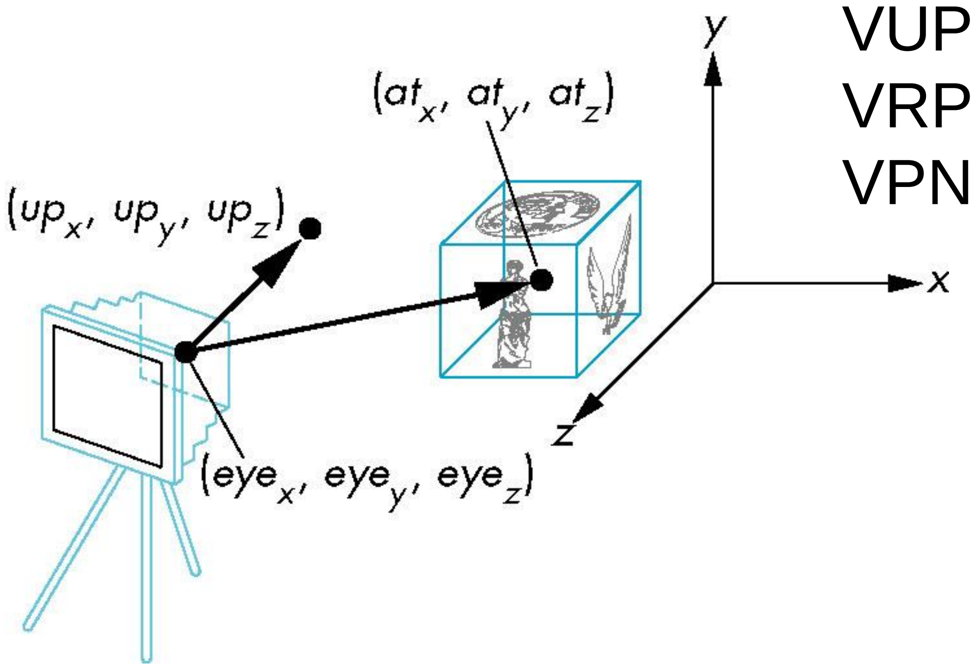
$$u = \frac{\mathrm{VUP} \times \mathrm{VPN}}{|\mathrm{VUP} \times \mathrm{VPN}|}$$

$$v = n \times u$$

Queen Mary
University of London

Use eye to lookat

- Specify
  - the eye point → a point the camera is located in world space
  - the lookat point → a point in world space that we wish to become the center of view
  - the up vector → a vector in world space that we wish to point up in camera image



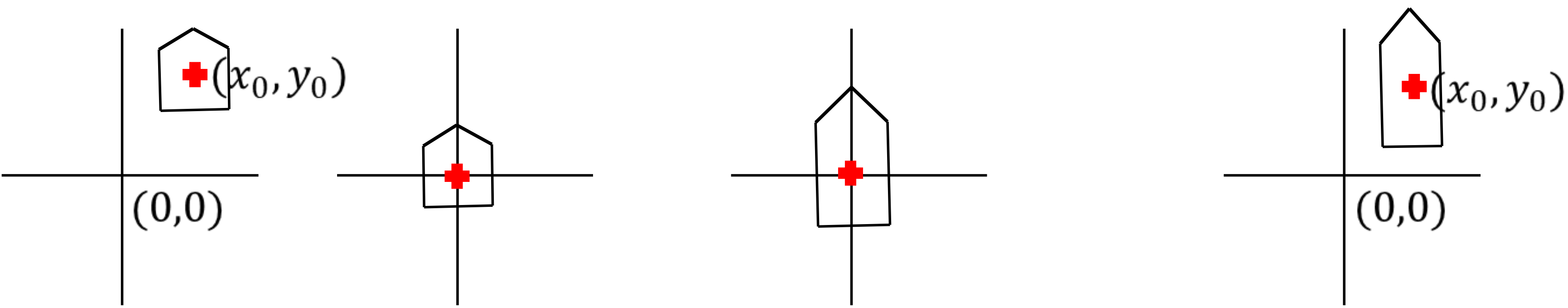
VUP = up vector
VRP = eye point
VPN = eye point – at point

Queen Mary
University of London

Composition of 2D Transformations

Scaling about a fixed point (x_0, y_0)

$House\ (H)$ $\quad$ $\boldsymbol{T}(dx, dy)H$ $\quad$ $\boldsymbol{S}(S_x, S_y)\boldsymbol{T}(dx, dy)H$ $\quad$ $\boldsymbol{T}(-dx, -dy)\boldsymbol{S}(S_x, S_y)\boldsymbol{T}(dx, dy)H$

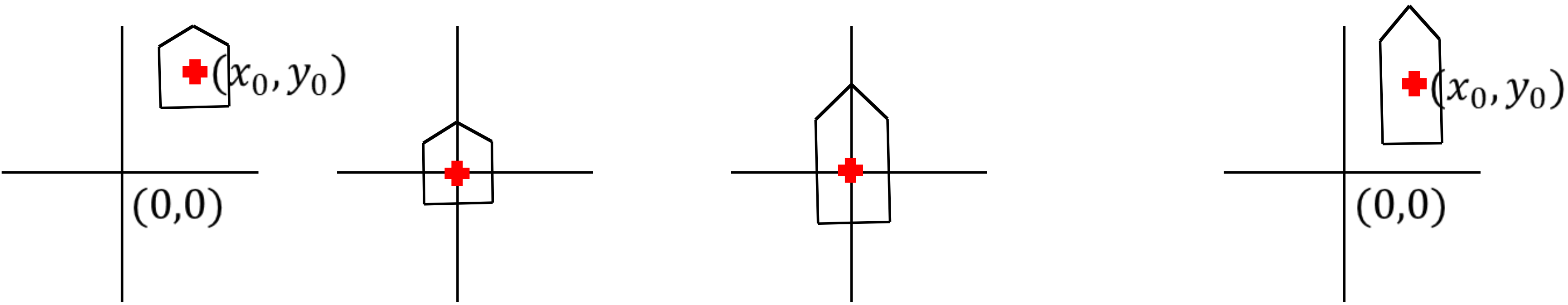


Queen Mary
University of London

Composition of 2D Transformations

Scaling about a fixed point (x_0, y_0)

$House\ (H)$ $\quad$ $\boldsymbol{T}(dx, dy)H$ $\quad$ $\boldsymbol{S}(S_x, S_y)\boldsymbol{T}(dx, dy)H$ $\quad$ $\boldsymbol{T}(-dx, -dy)\boldsymbol{S}(S_x, S_y)\boldsymbol{T}(dx, dy)H$

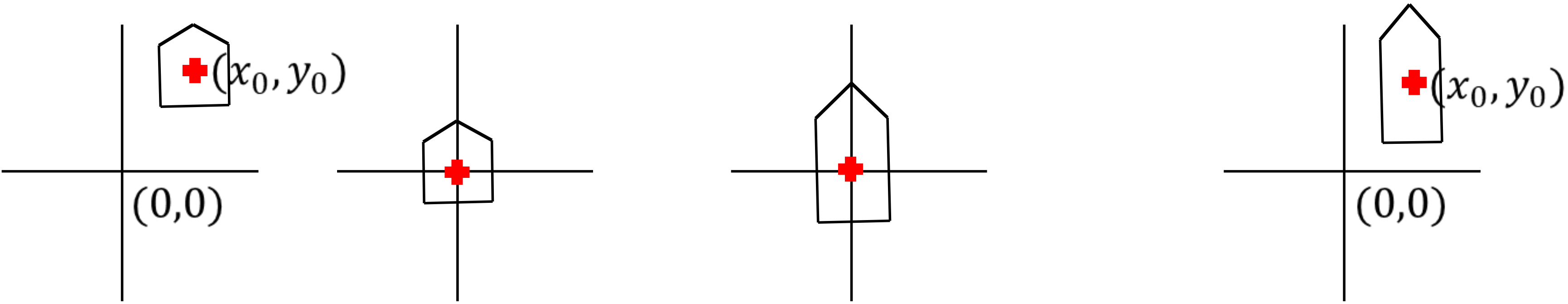


$$T(dx, dy) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}, S(S_x, S_y) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, T(-dx, -dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Composition of 2D Transformations

Scaling about a fixed point (x_0, y_0)

$House\ (H)$ $\quad$ $\boldsymbol{T}(dx,dy)H$ $\quad$ $\boldsymbol{S}(S_x,S_y)\boldsymbol{T}(dx,dy)H$ $\quad$ $\boldsymbol{T}(-dx,-dy)\boldsymbol{S}(S_x,S_y)\boldsymbol{T}(dx,dy)H$



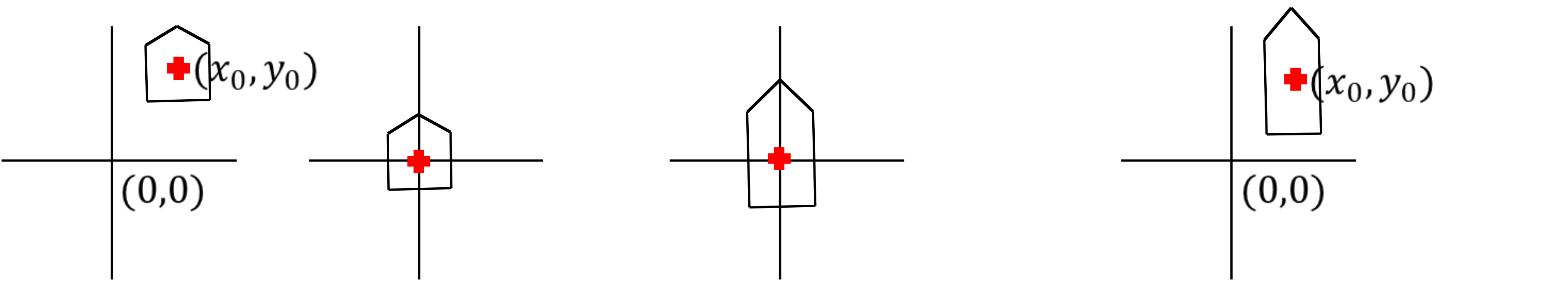
$$T(dx,dy) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}, S(S_x,S_y) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, T(-dx,-dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(-dx,-dy)S(S_x,S_y)T(dx,dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & (1-S_x)x_0 \\ 0 & S_y & (1-S_y)y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Composition of 2D Transformations

Scaling about a fixed point (x_0, y_0)

$House\ (H)$ $\quad$ $\boldsymbol{T}(dx, dy)H$ $\quad$ $\boldsymbol{S}(S_x, S_y)\boldsymbol{T}(dx, dy)H$ $\quad$ $\boldsymbol{T}(-dx, -dy)\boldsymbol{S}(S_x, S_y)\boldsymbol{T}(dx, dy)H$



$$T(dx, dy) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}, S(S_x, S_y) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, T(-dx, -dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

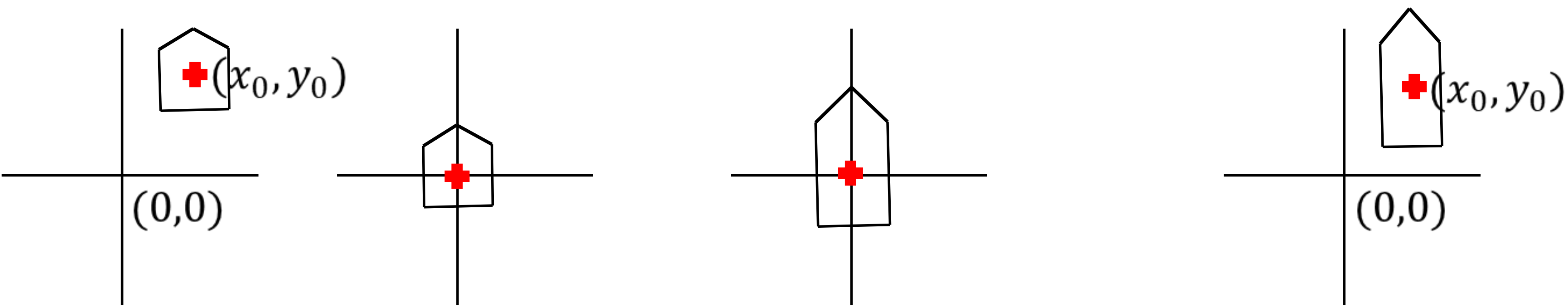
$$T(-dx, -dy)S(S_x, S_y)T(dx, dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & (1 - S_x)x_0 \\ 0 & S_y & (1 - S_y)y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

If $(x_0, y_0) = (1, 1.5)$ and $(S_x, S_y) = (0.9, 2)$,

Composition of 2D Transformations

Scaling about a fixed point (x_0, y_0)

$House\ (H)$ $\quad$ $\boldsymbol{T}(dx, dy)H$ $\quad$ $\boldsymbol{S}(S_x, S_y)\boldsymbol{T}(dx, dy)H$ $\quad$ $\boldsymbol{T}(-dx,-dy)\boldsymbol{S}(S_x, S_y)\boldsymbol{T}(dx, dy)H$



$$T(dx, dy) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}, S(S_x, S_y) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, T(-dx, -dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(-dx, -dy)S(S_x, S_y)T(dx, dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & (1 - S_x)x_0 \\ 0 & S_y & (1 - S_y)y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

If $(x_0, y_0) = (1,1.5)$ and $(S_x, S_y) = (0.9,2)$,

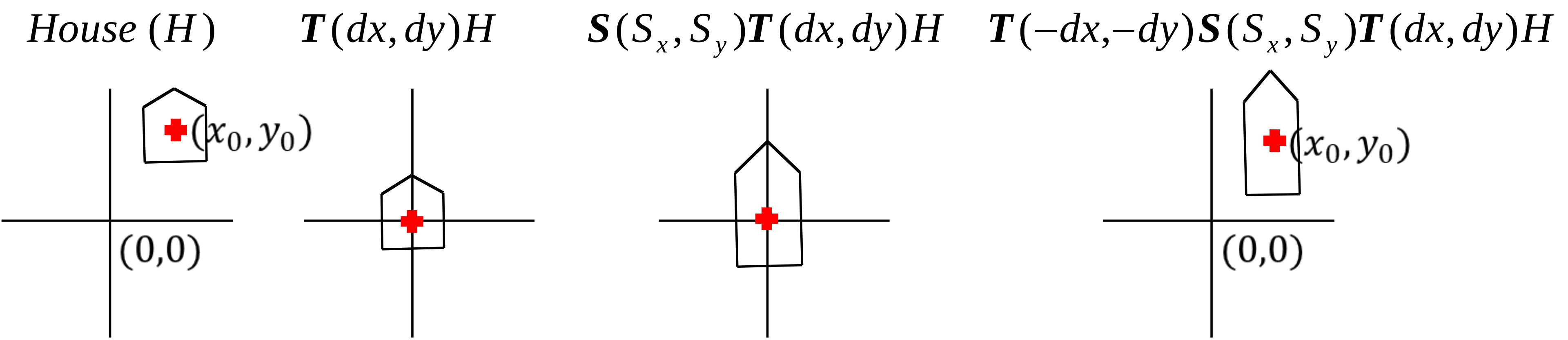
$$T(1,1.5)S(0.9,2)T(-1,-1.5) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1.5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1.5 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.9 & 0 & (1 - 0.9) \times 1 \\ 0 & 2 & (1 - 2) \times 1.5 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.9 & 0 & 0.1 \\ 0 & 2 & -1.5 \\ 0 & 0 & 1 \end{bmatrix}$$

Queen Mary University of London

Composition of 2D Transformations

Scaling about a fixed point (x_0, y_0)



$$T(dx, dy) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}, S(S_x, S_y) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, T(-dx, -dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(-dx, -dy)S(S_x, S_y)T(dx, dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & (1-S_x)x_0 \\ 0 & S_y & (1-S_y)y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

You can verify with (x_0, y_0) and (S_x, S_y),

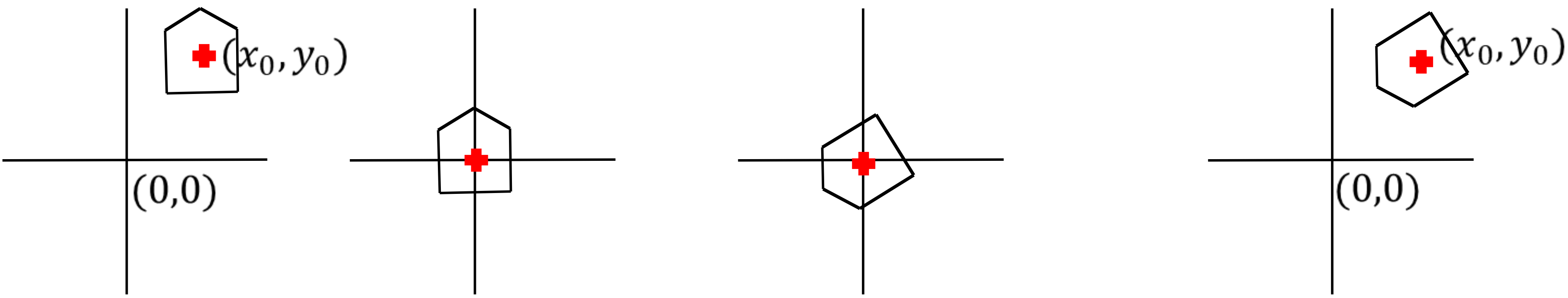
$$T(x_0, y_0)S(S_x, S_y)T(-x_0, -y_0) \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & (1-S_x)x_0 \\ 0 & S_y & (1-S_y)y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

Composition of 2D Transformations

Rotation about a fixed point (x_0, y_0)

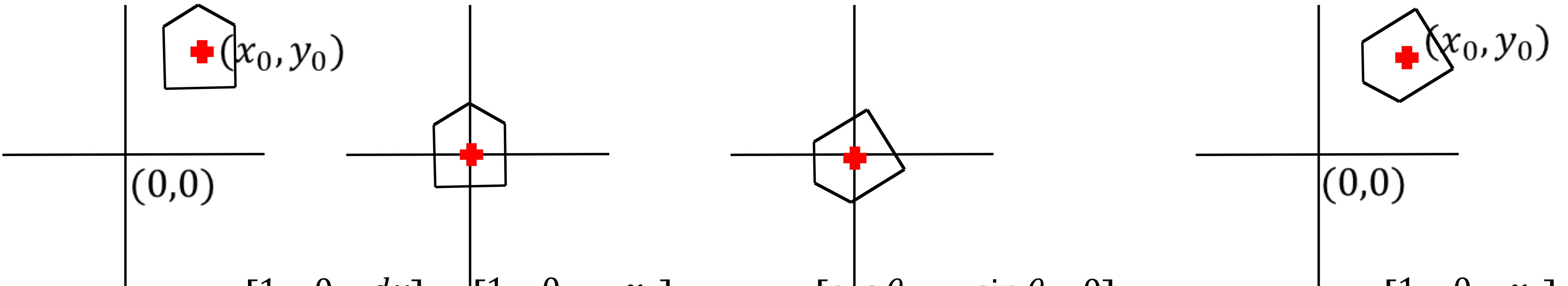
House (H) $\quad\quad$ $\boldsymbol{T}(dx, dy)H$ $\quad\quad$ $\boldsymbol{R}(\theta)\boldsymbol{T}(dx, dy)H$ $\quad\quad$ $\boldsymbol{T}(-dx,-dy)\boldsymbol{R}(\theta)\boldsymbol{T}(dx, dy)H$



Composition of 2D Transformations

Rotation about a fixed point (x_0, y_0)

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

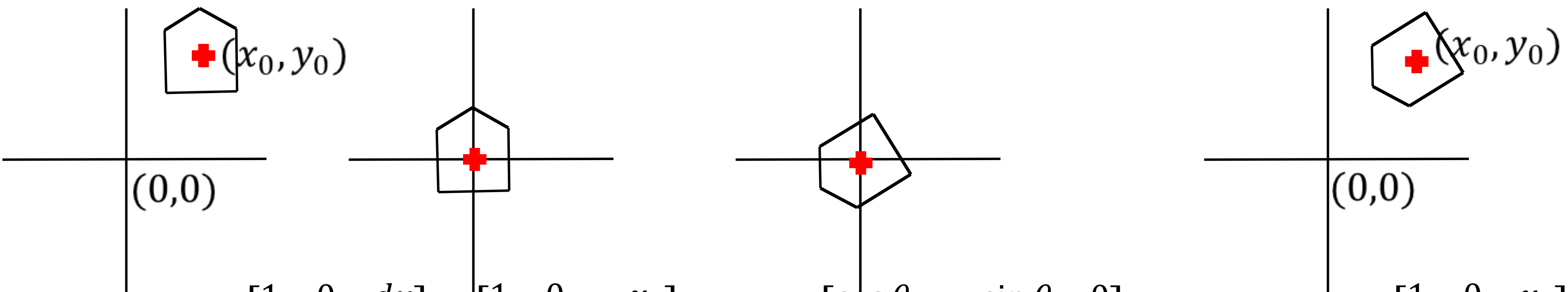


$$T(dx, dy) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}, R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, T(-dx, -dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Composition of 2D Transformations

Rotation about a fixed point (x_0, y_0)

House (H) $\qquad$ $\boldsymbol{T}(dx, dy)H$ $\qquad$ $\boldsymbol{R}(\theta)\boldsymbol{T}(dx, dy)H$ $\qquad$ $\boldsymbol{T}(-dx, -dy)\boldsymbol{R}(\theta)\boldsymbol{T}(dx, dy)H$



$$T(dx, dy) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}, R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, T(-dx, -dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(-dx, -dy)R(\theta)T(dx, dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

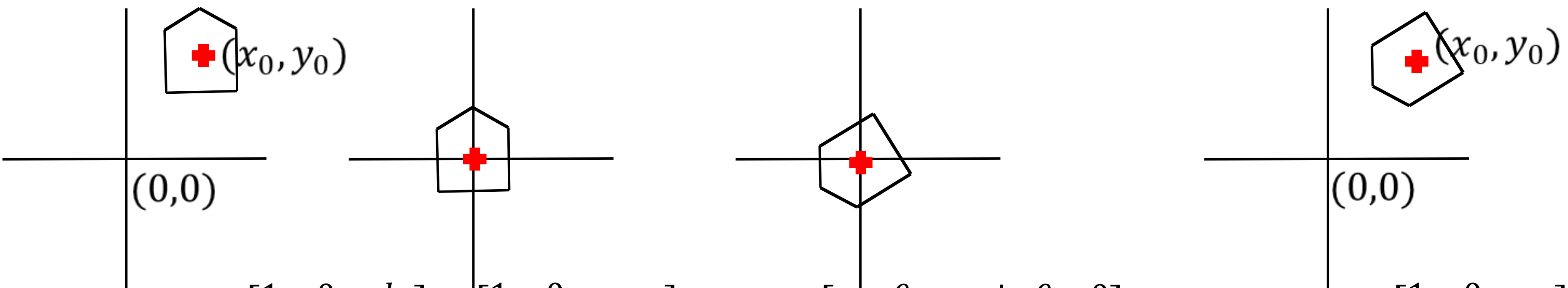
$$= \begin{bmatrix} \cos\theta & -\sin\theta & x_0 - (x_0\cos\theta - y_0\sin\theta) \\ \sin\theta & \cos\theta & y_0 - (x_0\sin\theta + y_0\cos\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

Queen Mary University of London

Composition of 2D Transformations

Rotation about a fixed point (x_0, y_0)

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



$$T(dx, dy) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}, R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, T(-dx, -dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(-dx, -dy)R(\theta)T(dx, dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & x_0 - (x_0\cos\theta - y_0\sin\theta) \\ \sin\theta & \cos\theta & y_0 - (x_0\sin\theta + y_0\cos\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

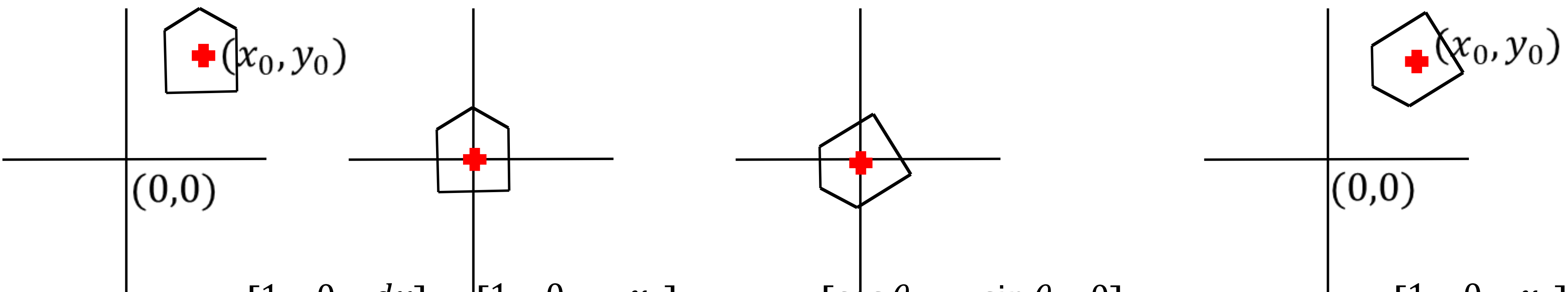
You can verify with (x_0, y_0),

$$T(-dx, -dy)R(\theta)T(dx, dy)\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & x_0 - (x_0\cos\theta - y_0\sin\theta) \\ \sin\theta & \cos\theta & y_0 - (x_0\sin\theta + y_0\cos\theta) \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

Composition of 2D Transformations

Rotation about a fixed point (x_0, y_0)

$House\ (H)$ $\qquad$ $\boldsymbol{T}(dx, dy)H$ $\qquad$ $\boldsymbol{R}(\theta)\boldsymbol{T}(dx, dy)H$ $\qquad$ $\boldsymbol{T}(-dx, -dy)\boldsymbol{R}(\theta)\boldsymbol{T}(dx, dy)H$



$$T(dx, dy) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}, R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, T(-dx, -dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(-dx, -dy)R(\theta)T(dx, dy) = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & x_0 - (x_0\cos\theta - y_0\sin\theta) \\ \sin\theta & \cos\theta & y_0 - (x_0\sin\theta + y_0\cos\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

If $(x_0, y_0) = (1,2)$, $\theta = 45^o$, $T(1,2)R(45^o)T(-1,-2) =$

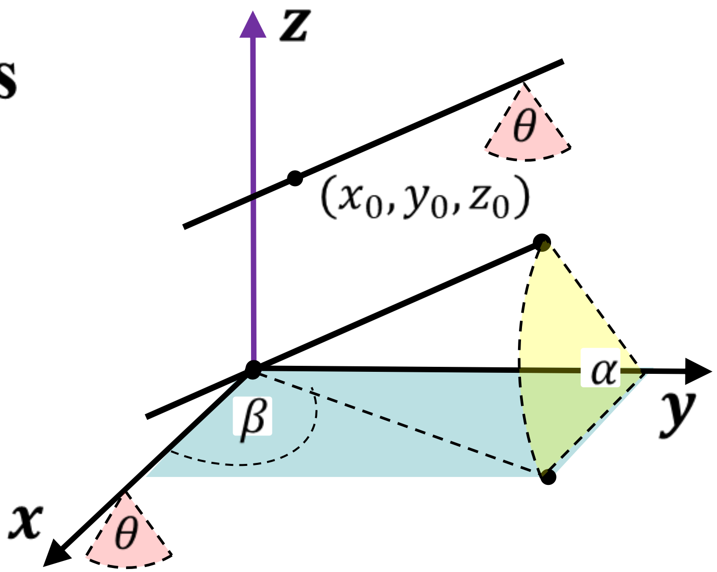
$$\begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 1 - (\sqrt{2}/2 - 2\sqrt{2}/2) \\ \sqrt{2}/2 & \sqrt{2}/2 & 2 - (\sqrt{2}/2 + 2\sqrt{2}/2) \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 1 + \sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 & 2 - 3\sqrt{2}/2 \\ 0 & 0 & 1 \end{bmatrix}$$

Composition of 3D Transformations

- **Rotation about an arbitrary axis**

  The axis is a line through a point (x_0, y_0, z_0)

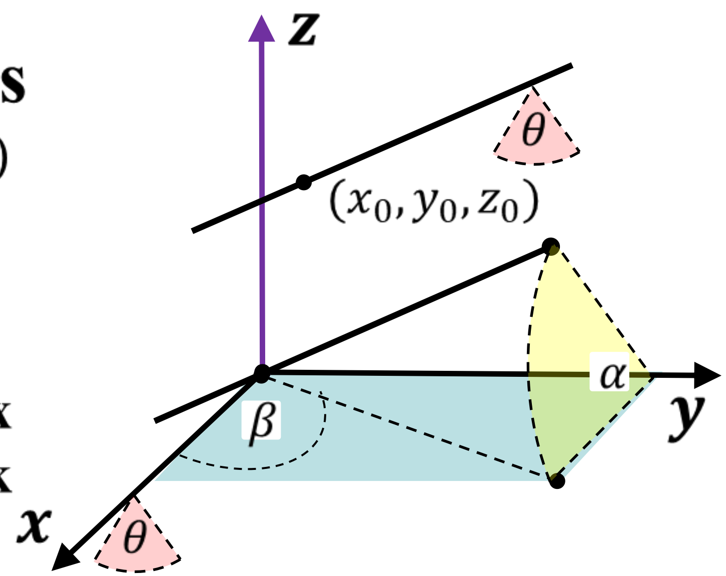
  if the rotation angles are given



Composition of 3D Transformations

- **Rotation about an arbitrary axis**

  The axis is a line through a point (x_0, y_0, z_0)



- **Step1. Translate the object to the origin**
- **Step2. Rotate to align the axis with x-axis**
- **Step3. Perform the specified rotation about x**
- **Step4. Inverse rotations to turn the axis back**
- **Step5. Inverse translation to move back**

$$R_v(\theta) = \underbrace{T(x_0, y_0, z_0)}_{\textbf{Step5}} \cdot \underbrace{R_y(-\alpha) \cdot R_z(-\beta)}_{\textbf{Step4}} \cdot \underbrace{R_x(\theta)}_{\textbf{Step3}} \cdot \underbrace{R_z(\beta) \cdot R_y(\alpha)}_{\textbf{Step2}} \cdot \underbrace{T(-x_0, -y_0, -z_0)}_{\textbf{Step1}}$$

Queen Mary
University of London

3D Graphics Programming Tools

The rendering pipeline
(Revision 1)

(Go to **www.menti.com** and use the code **1179 3402** to ask me questions)

Queen Mary
University of London