ΗΥ416
ΓΡΑΦΙΚΑ ΥΠΟΛΟΓΙΣΤΩΝ

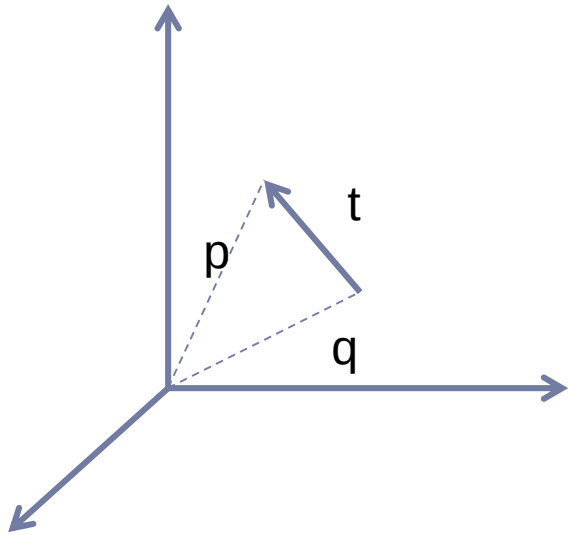
Γεωμετρικοί Μετασχηματισμοί

Π. ΤΣΟΜΠΑΝΟΠΟΥΛΟΥ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Transforming Tangents

$$\mathbf{t} = \mathbf{p} - \mathbf{q}$$

$$\begin{aligned}\mathbf{t}' &= \mathbf{p}' - \mathbf{q}' \\ &= M\mathbf{p} - M\mathbf{q} \\ &= M(\mathbf{p} - \mathbf{q}) \\ &= M\mathbf{t}\end{aligned}$$



Transforming Normals

- Dot product as matrix multiplication

$$\underline{v} \cdot \underline{w} = v^T * w = (v_x, v_y, v_z) * \begin{pmatrix} w_x \\ w_y \\ w_z \end{pmatrix}$$

Matrix multiplication

Dot product

- Normal N on a plane
  - For any vector T in the plane: $\underline{N}^T * \underline{T} = 0$
  - Find transformation M' for normal vector, such that

$$(\mathbf{M}' * \underline{N})^T * (\mathbf{M} * \underline{T}) = 0 = \underline{N}^T * (\mathbf{M}'^T * \mathbf{M}) * \underline{T}$$

$$\mathbf{M}'^T * \mathbf{M} = 1$$

$$\mathbf{M}' = (\mathbf{M}^{-1})^T$$

Transforming Normals

- Remember:

  Normals are transformed by

  the transpose of the inverse

  of the 4x4 transformation matrix of points and vectors

- No problem with orthogonal transformations
  - E.g. rotation, uniform scaling
    - $M^{-1} = M^T$
    - $M^{-1T} = M^{TT} = M$

Transforming Normals

$$\mathbf{n}^T \mathbf{t} = 0$$

$$\mathbf{n}'^T \mathbf{t}' = 0$$

$$\mathbf{n}'^T M\mathbf{t} = 0$$

$$\mathbf{n}'^T M\mathbf{t} = \mathbf{n}^T \mathbf{t}$$

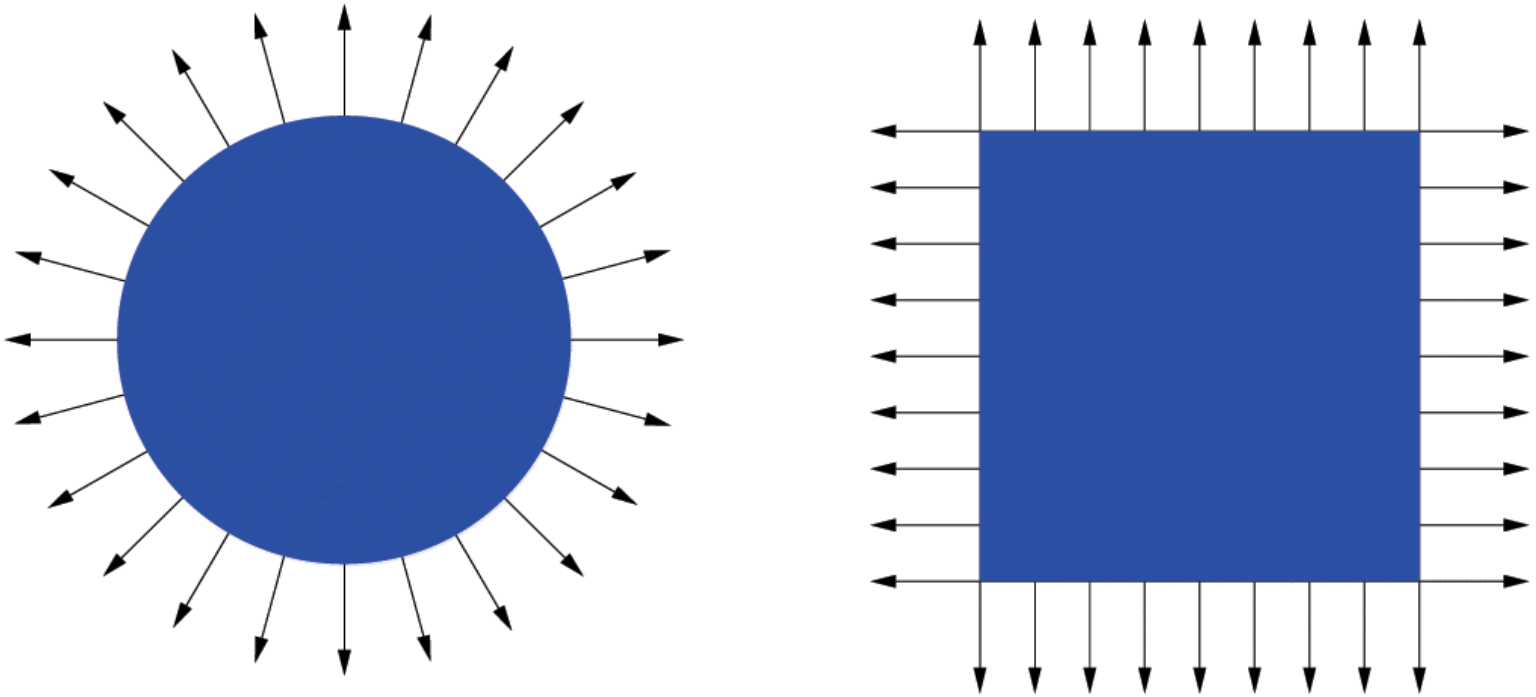
$$\mathbf{n}'^T M = \mathbf{n}^T$$

$$M^T \mathbf{n}' = \mathbf{n}$$

$$\mathbf{n}' = M^{T\,-1}\mathbf{n} = M^{-1^T}\mathbf{n}$$

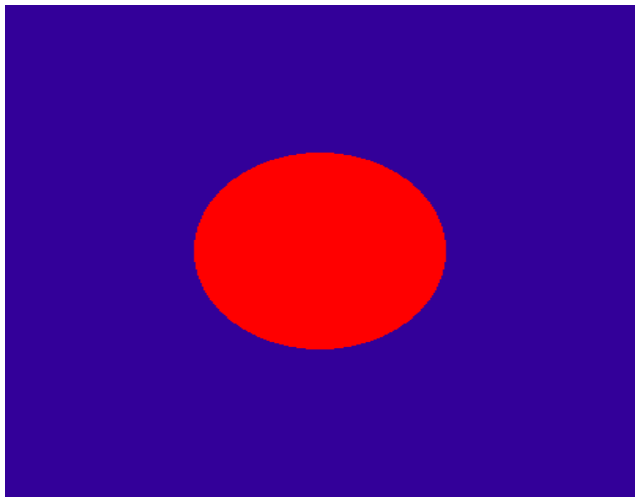
Surface Normal

- Surface Normal: unit vector that is locally perpendicular to the surface

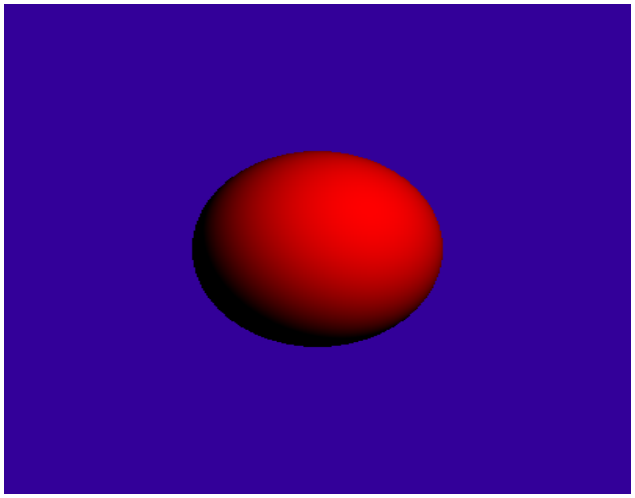


Why is the Normal important?

- It's used for shading — makes things look 3D!

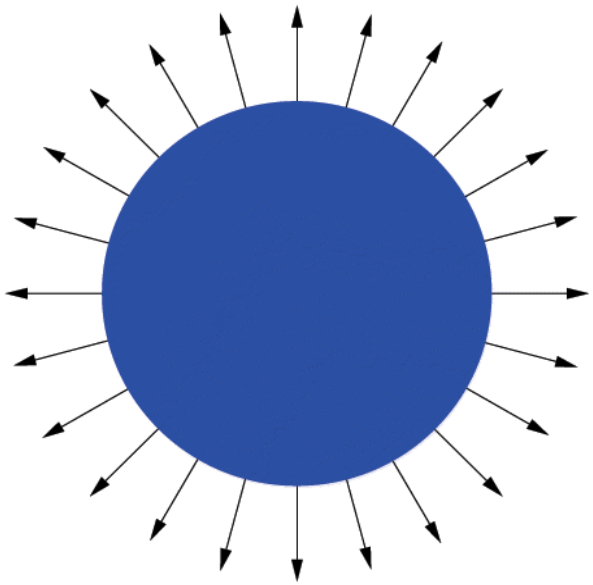


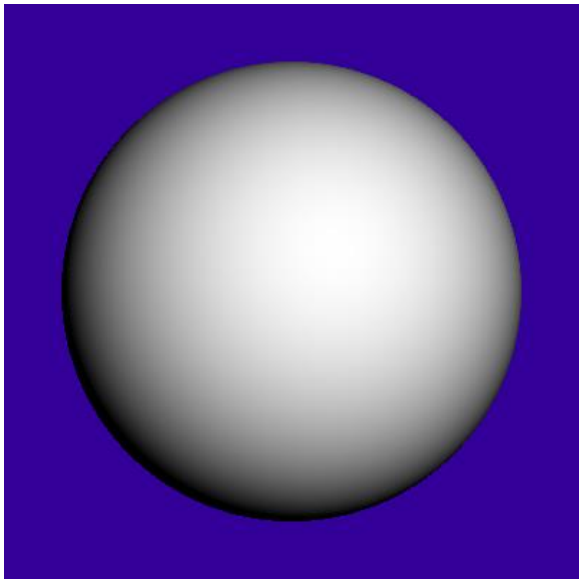
object color only

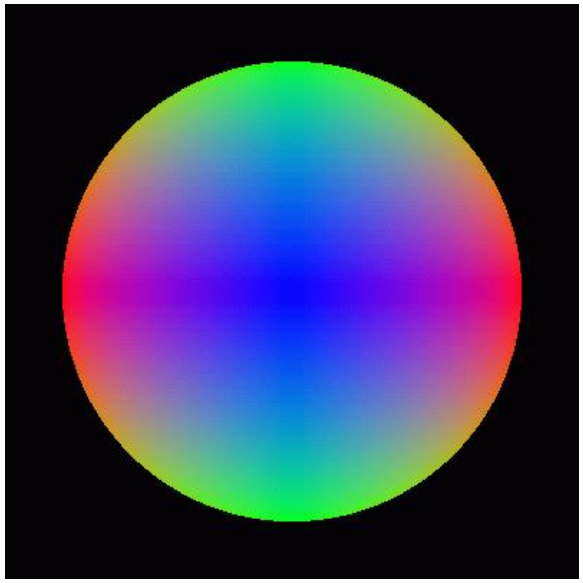


Diffuse Shading

Visualization of Surface Normal

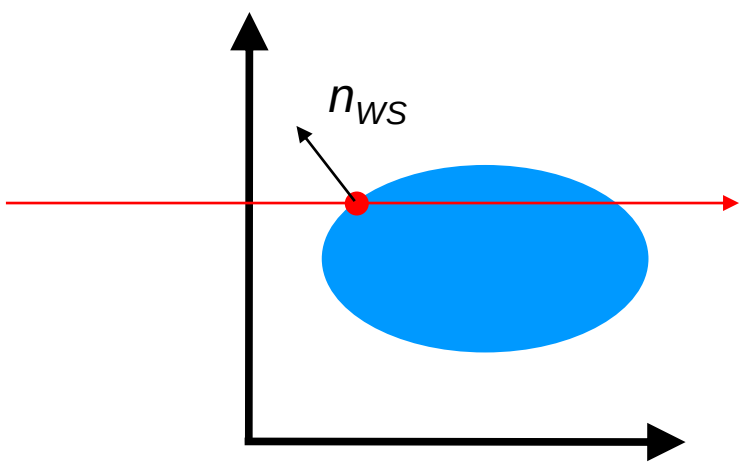




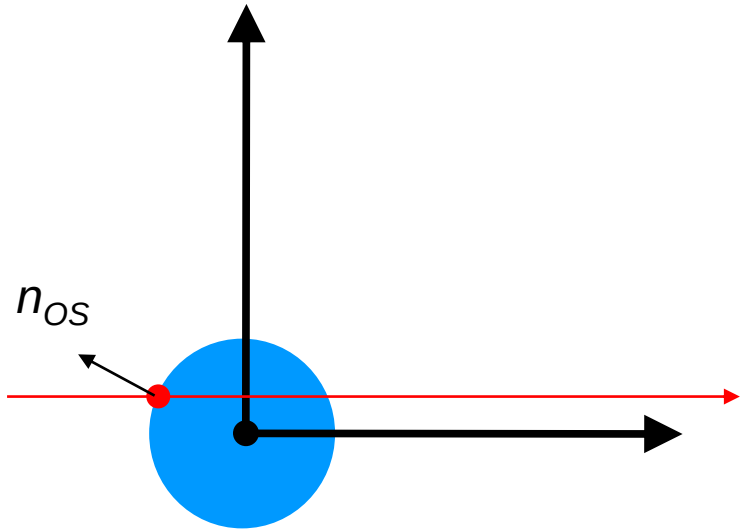


± x = Red
± y = Green
± z = Blue

How do we transform normals?



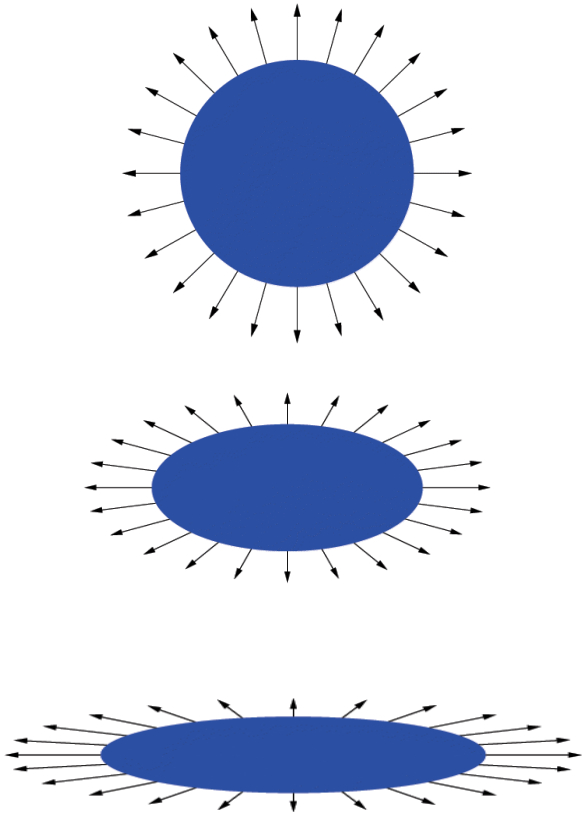
World Space



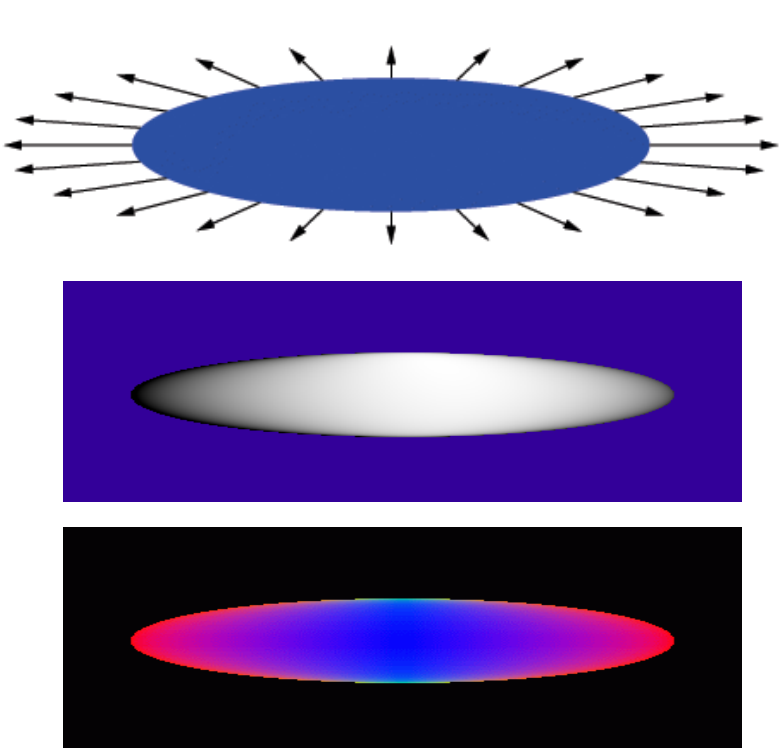
Object Space

Transform the Normal like the Ray?

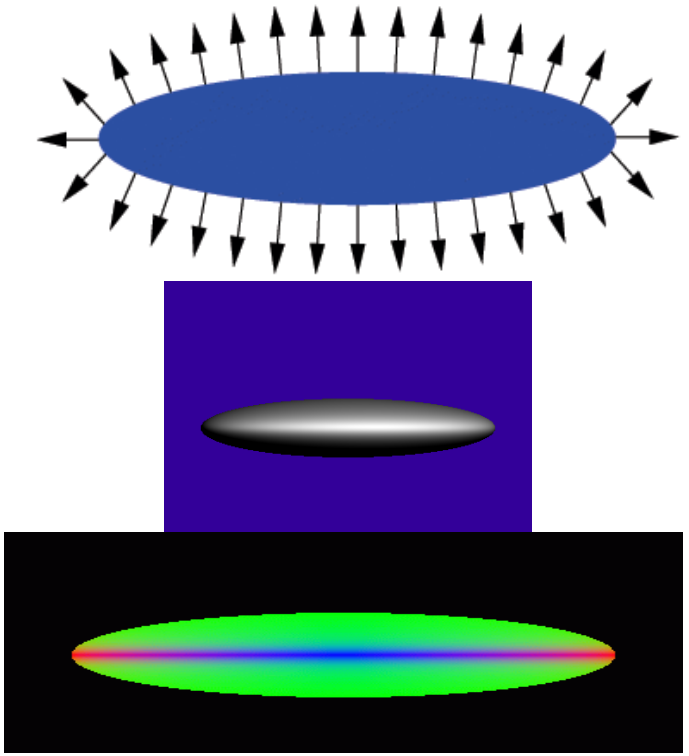
- translation?
- rotation?
- isotropic scale?
- scale?
- reflection?
- shear?
- perspective?



More Normal Visualizations



Incorrect Normal Transformation



Correct Normal Transformation

Transforming Normals

$$\mathbf{n}^T \mathbf{t} = 0$$

$$\mathbf{n}'^T \mathbf{t}' = 0$$

$$\mathbf{n}'^T M\mathbf{t} = 0$$

$$\mathbf{n}'^T M\mathbf{t} = \mathbf{n}^T \mathbf{t}$$

$$\mathbf{n}'^T M = \mathbf{n}^T$$

$$M^T \mathbf{n}' = \mathbf{n}$$

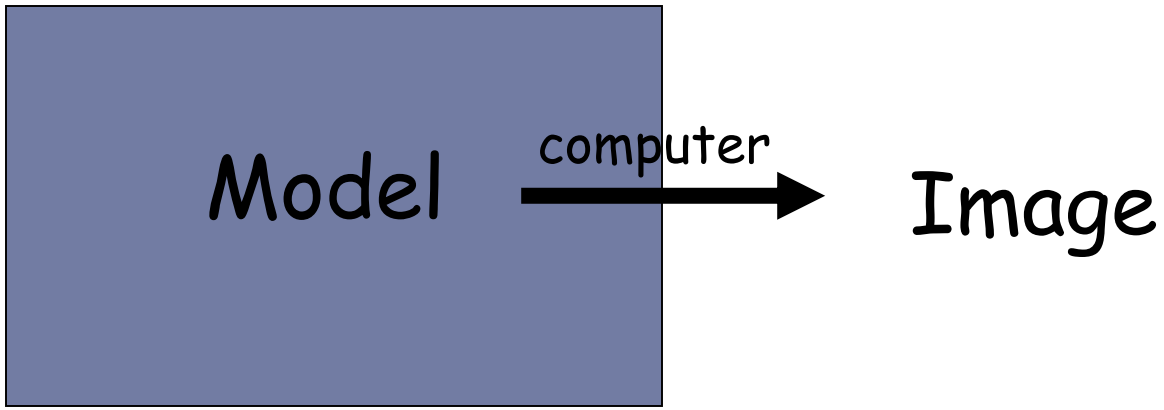
$$\mathbf{n}' = M^{T\,-1}\mathbf{n} = M^{-1^T}\mathbf{n}$$

Quick Recap

- Computer Graphics is using a computer to generate an image from a representation.

Modeling

- What we have been studying so far is the mathematics behind the creation and manipulation of the 3D representation of the object.



What have we seen so far?

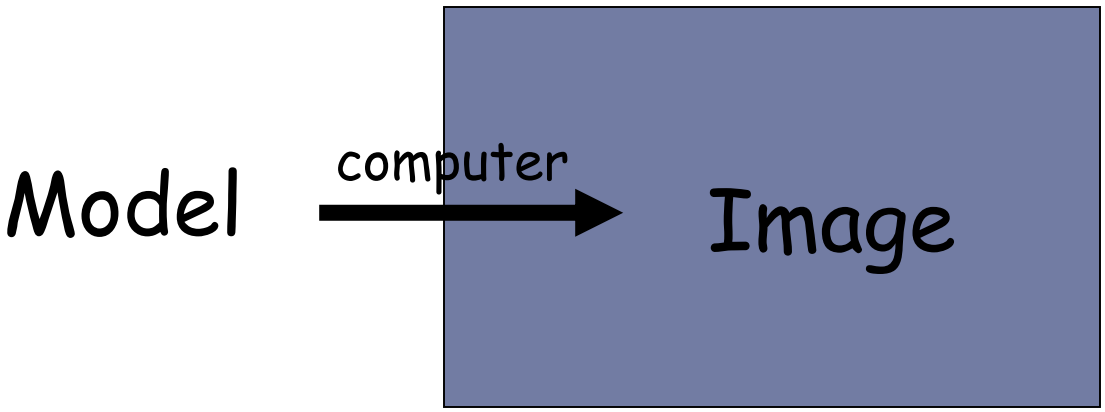
- Basic representations (point, vector)
- Basic operations on points and vectors (dot product, cross products, etc.)
- Transformation – manipulative operators on the basic representation (translate, rotate, deformations) – 4x4 matrices to "encode" all these.

Why do we need this?

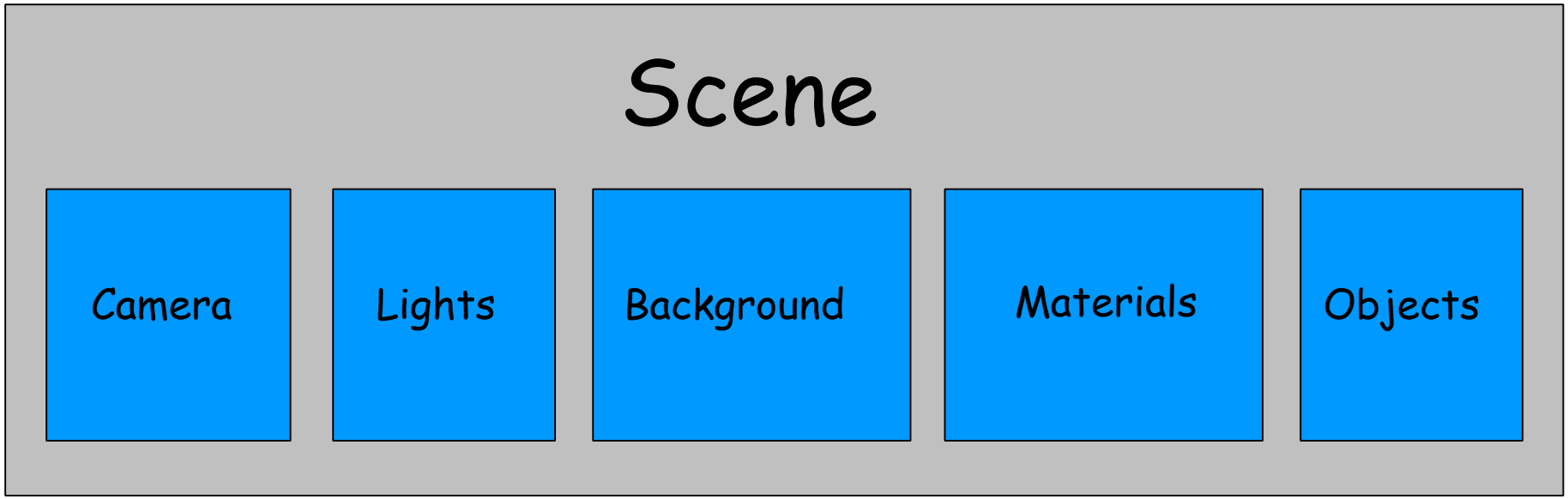
- In order to generate a picture from a model, we need to be able to not only specify a model (representation) but also manipulate the model in order to create more interesting images.

Overview

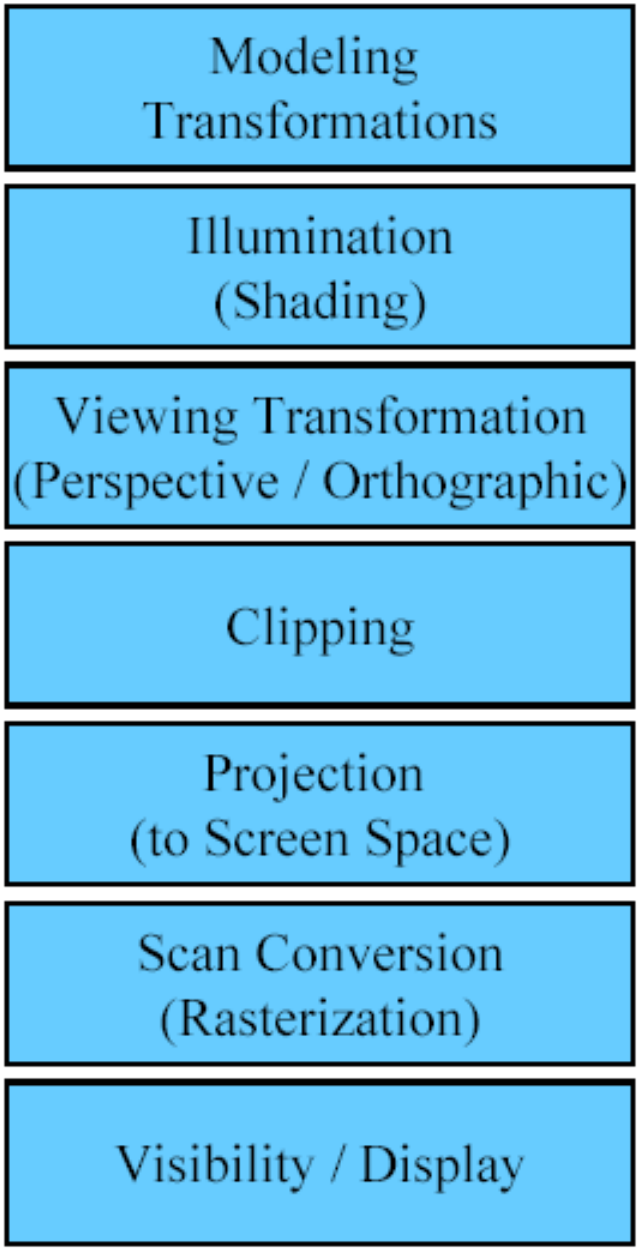
- The next set of slides will deal with the other half of the process.
- From a model, how do we generate an image



Scene Description

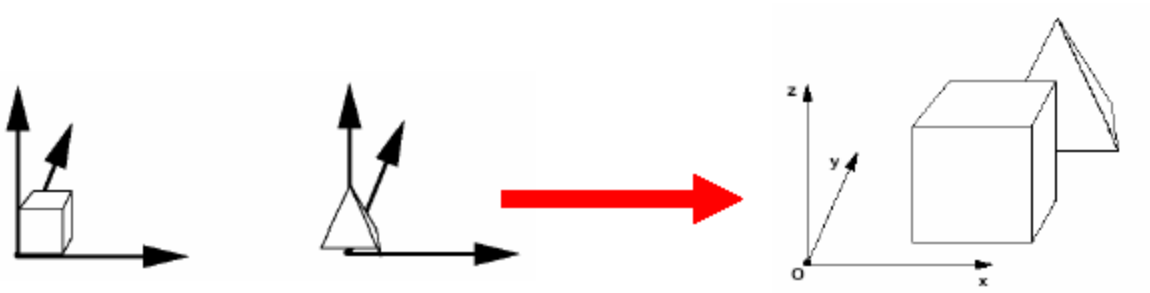


Graphics Pipeline



Graphics Pipeline

- Modeling transforms orient the models within a common coordinate frame (world space)



Object space

World space

Modeling Transformations

Illumination (Shading)

Viewing Transformation (Perspective / Orthographic)

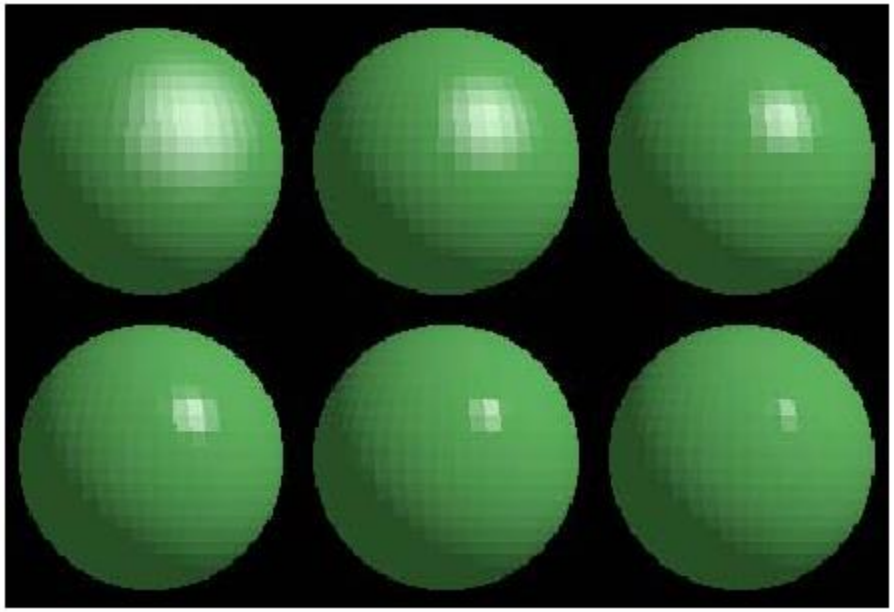
Clipping

Projection (to Screen Space)

Scan Conversion (Rasterization)

Visibility / Display

Graphics Pipeline



Modeling Transformations

Illumination (Shading)

Viewing Transformation (Perspective / Orthographic)

Clipping

Projection (to Screen Space)

Scan Conversion (Rasterization)

Visibility / Display

Graphics Pipeline

- Maps world space to eye space
- Viewing position is transformed to origin & direction is oriented along some axis (usually *z*)

Modeling Transformations

Illumination (Shading)

Viewing Transformation (Perspective / Orthographic)

Clipping

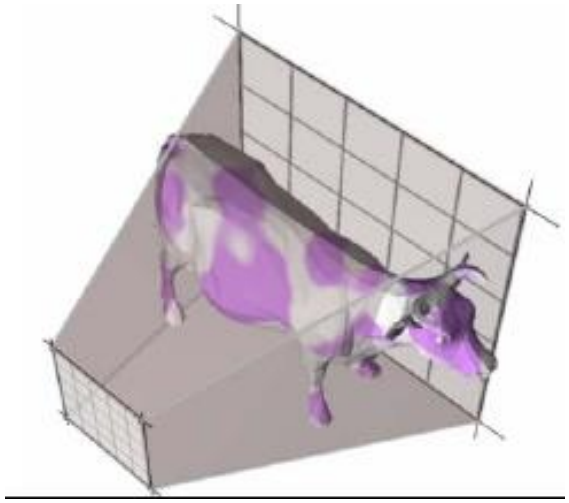
Projection (to Screen Space)

Scan Conversion (Rasterization)

Visibility / Display

Graphics Pipeline

- Transform to Normalized Device Coordinates (NDC)
- Portions of the object outside the view volume (view frustum) are removed



Modeling Transformations

Illumination (Shading)

Viewing Transformation (Perspective / Orthographic)

Clipping

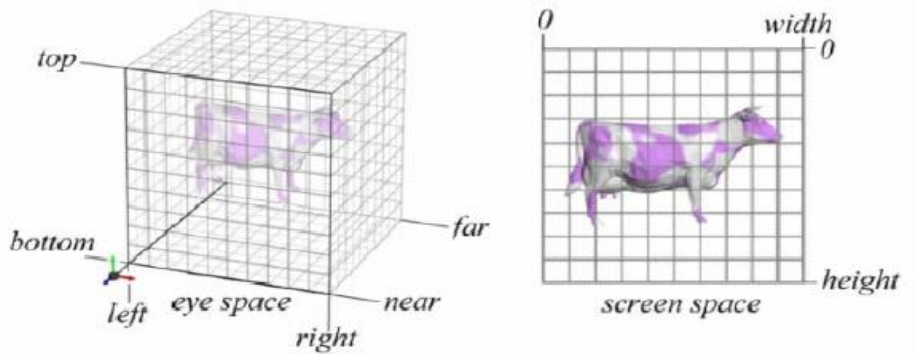
Projection (to Screen Space)

Scan Conversion (Rasterization)

Visibility / Display

Graphics Pipeline

- The objects are projected to the 2D image place (screen space)



Modeling Transformations

Illumination (Shading)

Viewing Transformation (Perspective / Orthographic)

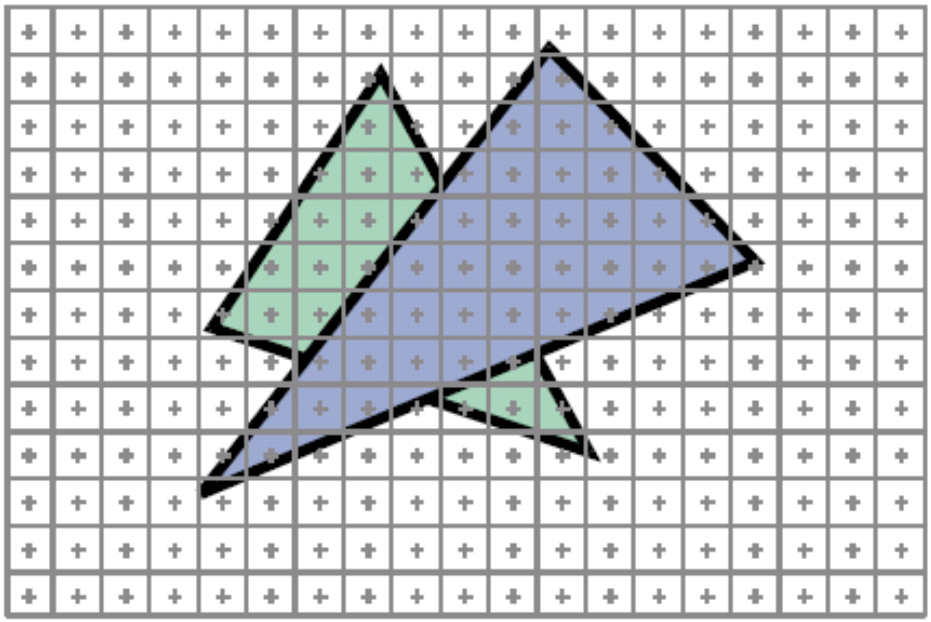
Clipping

Projection (to Screen Space)

Scan Conversion (Rasterization)

Visibility / Display

Graphics Pipeline



Modeling Transformations

Illumination (Shading)

Viewing Transformation (Perspective / Orthographic)

Clipping

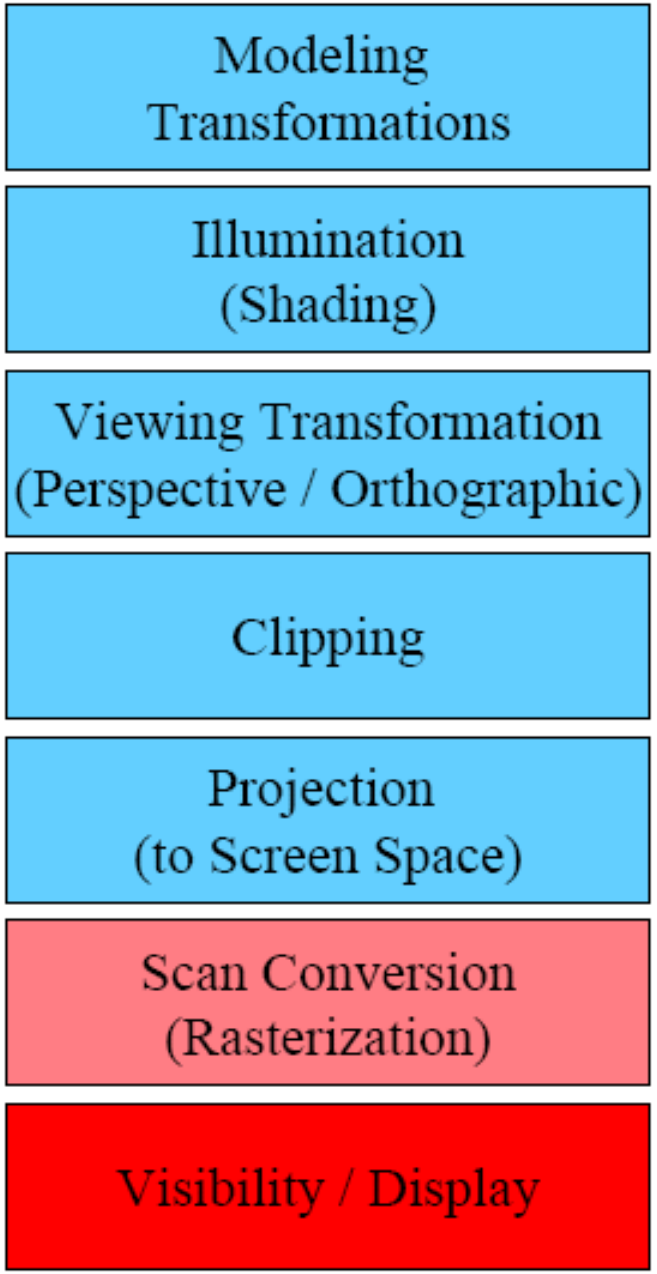
Projection (to Screen Space)

Scan Conversion (Rasterization)

Visibility / Display

Graphics Pipeline

- Z-buffer - Each pixel remembers the closest object (depth buffer)



Graphics Pipeline

- Almost every step in the graphics pipeline involves a change of coordinate system. Transformations are central to understanding 3D computer graphics.

Ερωτήσεις

- Ιστοσελίδα μαθήματος (ενεργοποιημένη) :
  http://support.inf.uth.gr/courses/CE416/

- E-mail λίστα του μαθήματος:
  ce416@inf-server.inf.uth.gr

- Π. Τσομπανοπούλου, Ε3-12, yota@uth.gr