NANYANG TECHNOLOGICAL UNIVERSITY

# CZ2003
# Computer Graphics & Visualization
# Revision of Part 2

**AY 2020/2021 Semester 1**

## Final Assignment

- Date/time: 19 Nov 2020 (Thu, week 14) / 16:00 – 17:00
- Detail:
  - see Announcement in course site
    - The Final Assignment can be done on your own computers in the same way as you worked on your labs. SW Lab 3 will be available.
    - Please note that if you miss the Final Assignment you will not be able to retake it.
  - See course site ➔Assignments ➔ Final Assignment
    - get familiar with detailed instruction in advance

NANYANG TECHNOLOGICAL UNIVERSITY



## Suggestions

- Review each module and pay attention to
  - important concepts
  - basic methods
  - important formulae
- Review & practice on
  - examples given in both TEL & review lectures
  - Tutorial questions
  - Lab assignments (1~5)

The following slides list some fundamental concepts and methods for the 2nd part of the course.

NANYANG TECHNOLOGICAL UNIVERSITY



## 1. 2D transformations

- Homogeneous coordinates

  (x, y) ⬄ (x, y, 1) = (k x, k y, k)

- 2D affine transformations

  x' = a x + b y + m

  y' = c x + d y + n or $\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & m \\ c & d & n \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$

Basic set: translation scaling rotation

$$Tr(m,n) = \begin{pmatrix} 1 & 0 & m \\ 0 & 1 & n \\ 0 & 0 & 1 \end{pmatrix} \quad S(sx,sy) = \begin{pmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad Rot(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

NANYANG TECHNOLOGICAL UNIVERSITY

## 2D transformations

**Others: *reflection*, …**

$$\mathrm{Ref}_o = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathrm{Ref}_{ox} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathrm{Ref}_{oy} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



## 2. 3D transformations

Affine transformation

$$\begin{aligned} x' &= ax + by + cz + l \\ y' &= dx + ey + fz + m \\ z' &= gx + hy + iz + n \end{aligned} \quad \text{or} \quad \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c & l \\ d & e & f & m \\ g & h & i & n \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Translation — Scaling — Rotation

$$\begin{pmatrix} 1 & 0 & 0 & m \\ 0 & 1 & 0 & n \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Rot_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad Rot_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad Rot_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



## 3D transformations

Reflection

$$\mathrm{Ref}_o = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathrm{Ref}_{ox} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathrm{Ref}_{oxy} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathrm{Ref}_{oy} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathrm{Ref}_{oyz} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathrm{Ref}_{oz} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathrm{Ref}_{ozx} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



## Affine transformations in VRML

- Transform {
  - translation dx dy dz
  - rotation ax ay az theta
  - scale sx sy sz
  - children [ …]
- }

where

- the rotation axis is from the origin (0,0,0) to point (ax,ay,az), and theta (in radian) is the rotation angle value;
- sx, sy, sz are the 3 scaling factors along x, y, z axes;
- dx, dy, dz are the translation amount along x, y, z axes.

The order is first scale, then rotation, and finally translation.

## 2D and 3D transformation problems

- Given two shapes A and B where B is obtained from A by an affine transformation, find the transformation matrix that defines the affine transformation.
- Find an overall transformation that is a composite of several relatively simple affine transformations (such as scaling, rotation, reflection, …).
  - Key: how to convert a "non-standard" transformation to a "standard" transformation

NANYANG TECHNOLOGICAL UNIVERSITY



## 2D and 3D transformation problems (cont)

- Matrix multiplication

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} = ?$$

- Matrix-vector multiplication

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = ?$$

NANYANG TECHNOLOGICAL UNIVERSITY





## 2D and 3D transformation problems (cont)

- Applications in rotational/translational sweeping
  - Use matrix/matrices for rotation
  - Use matrix for translation

NANYANG TECHNOLOGICAL UNIVERSITY



## 3. Motions and morphing

- Simulating three types of speed

| Uniform; | Acceleration; | Deceleration |
|---|---|---|
| $\tau = \frac{t-t_1}{t_2-t_1},\ t\in[t_1,t_2]$ | $\tau = 1-\cos\left(\frac{\pi}{2}\frac{t-t_1}{t_2-t_1}\right),\ t\in[t_1,t_2]$ | $\tau = \sin\left(\frac{\pi}{2}\frac{t-t_1}{t_2-t_1}\right),\ t\in[t_1,t_2]$ |

- Another way to describe animation
  - Use frame index k
- Methods for motions and morphing
  - Motion by path (three steps)
  - Linear interpolation of two items for morphing

    $v(\tau) = (1-\tau) A + \tau B, \quad 0 \le \tau \le 1$
  - Introducing time into rotational/translational sweeping

## Typical problems

- Simulate uniform speed, acceleration, deceleration

- Express an animation using time parameter
- Express an animation using frame index

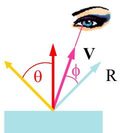
- Design an animation using “motion by path”
- Design an animation using linear interpolation
- Design an animation by introducing time into transformations
- Analyze an animation model
- …



## 4. Visual appearance (I)

- Color representation: (R,G,B)
- 3 light sources
- Phong illumination model
  - Ambient reflection
  - Diffuse reflection
  - Specular reflection

$$I = k_a I_a + \sum_{\substack{for\ each \\ light\ i}} k_d I_i \cos\theta_i + \sum_{\substack{for\ each \\ light\ s}} k_s I_i (\cos\phi_i)^n$$




- Computation
  - How to compute **unit** vectors L, N, V
  - Vector R = 2 (N •L) N – L



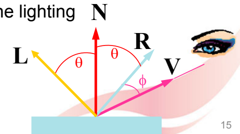
## Visual appearance (I)

- Lighting vector L: a vector from a point on the surface towards a light source
- Viewing vector V: a vector from a point on the surface towards the viewer
- Normal vector N: a vector that is perpendicular to the tangent plane of the surface

  For example, for an implicit surface f(x,y,z) = 0, its normal vector is

$$N(x,y,z) = \pm\left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \quad \frac{\partial f}{\partial z}\right]$$

- Reflected vector R: the image of the lighting vector L reflected off the surface






## Typical problems

- Use Phong illumination model to perform some analysis

- How to compute the reflected vector R
- How to compute the normal of a surface

- How to compute the diffuse reflection
- How to compute the specular reflection
- How to compute the overall illumination

- Design functions for r, g, b to define diffuseColor in FVRML

## 5. Visual appearance (II)

- Texture mapping
  - Texture (image) is used to change the color.
- Bump mapping
  - Bump map is used to change the normal of the surface.
- Displacement mapping
  - Geometric texture is used to change the surface.



## Visual appearance (II)

Texture domain (u,v) ⟺ Surface parameter domain (s,t) ⟺ Surface point (x,y,z)

Algorithm (parametric texture mapping)

Step 1: Parameterize texture with $(u,v)$ coordinates

Step 2: Parameterize the surface with $(s,t)$ coordinates

$x=x(s,t),\ y=y(s,t),\ z=z(s,t),\quad s\in[s_0,s_1],\ t\in[t_0,t_1]$

Step 3: Define a mapping between (u,v) and (s,t)

$$\frac{u-u_0}{u_1-u_0}=\frac{s-s_0}{s_1-s_0},\quad \frac{v-v_0}{v_1-v_0}=\frac{t-t_0}{t_1-t_0}$$

Step 4: (x,y,z) ➔(u,v) or (u,v) ➔(x,y,z)





## Typical problems

- How to perform forward mapping
- How to perform inverse mapping

- The concepts of the three surface mapping methods
- The properties of the three surface mapping methods
- Analyze the basic shape and the geometric texture

- Conduct texture mapping
- Conduct displacement mapping & its function-based extension