Computer Graphics, Lab Assignment 9

Handed out: May 4, 2021

Due: 23:59, May 5, 2021 (NO SCORE for late submissions!)

- *Only accept answers submitted via git push to this course project for you at https://hconnect.hanyang.ac.kr (<Year>_<Course no.>_<Class code>/<Year>_<Course no.>_<Student ID>.git).*
- *Place your files under the directory structure <Assignment name>/<Problem no.>/<your file> just like the following example.*

```
+ 2021_ITE0000_2019000001
  + LabAssignment9/
    + 1/
      - 1.py
    + 2/
      - 2.py
    + 3/
      - 3.py
```

- *The submission time is determined not when the commit is made but when the git push is made.*

1. Write down a Python program to visualize ZXZ Euler angles.
   A. This is how ZXZ Euler angles works
      i. Rotate along Z-axis by α
      ii. Rotate along X-axis of the new frame by β
      iii. Rotate along Z-axis of the new frame by γ
   B. Start from 9-Orientation&Rotation practice code, implement ZXZ Euler angles and add code to change α, β, γ values in the following way.
      i. If you press or repeat a key, the value of α, β, γ should be changed as shown in the table:

Key	Transformation
A	Increase α by 10°
Z	Decrease α by 10°
S	Increase β by 10°
X	Decrease β by 10°
D	Increase γ by 10°
C	Decrease γ by 10°
V	Initialize orientation

C. Hint: You do not need to store a composed rotation matrix as a global variable. You can just store α, β, γ as global variables.

D. Set the window title to **your student ID** and the window size to (480,480).

E. Expected result: Uploaded LabAssignment9-1.mp4

F. Files to submit: A Python source file (Name the file whatever you want (in English). Extension should be .py)

2. Write down a Python program to compare 4 orientation interpolation methods.

   A. First, implement following functions:

   B. exp & log functions

      i. exp(rv)

         1. Converts a rotation vector to a rotation matrix
         2. You can use Rodrigues' rotation formula or the method in 9-Orientation& Rotation
         3. Returns a rotation matrix

      ii. log(R)

         1. Converts a rotation matrix to a rotation vector
         2. You can use the method in 9-Orientation& Rotation slides.
         3. Returns a rotation vector (the length of the vector is the rotation angle)

   C. Interpolation functions:

      i. **slerp(R1, R2, t)** - slerp

         1. R1 & R2: rotation matrices for start & end orientations

ii. **interpolateRotVec(rv1, rv2, t)** - interpolate each element of two vectors

    1. rv1 & rv2: rotation vectors for start & end orientations

iii. **interpolateZYXEuler(euler1, euler2, t)** - interpolate each element of two euler angle tuples

    1. euler1 & euler2: tuples of ZYX Euler angles for start & end orientations (euler1[0]: xang, euler1[1]: yang, euler1[2]: zang)

iv. **interpolateRotMat(R1, R2, t)** - interpolate each element of two matrices

    1. R1 & R2: rotation matrices for start & end orientations

D. For all interpolation functions:

  i. All interpolation functions return a rotation matrix

  ii. The parameter t ranges from 0.0 to 1.0

E. Start from the uploaded code skeleton (LabAssignment9-2-code-skeleton.py).

F. You will need to use

  i. The given lerp() for interpolateRotVec(), interpolateZYXEuler(), interpolateRotMat()

  ii. The given ZYXEulerToRotMat() for interpolateZYXEuler()

  iii. Your exp(), log() implementation for slerp(), interpolateRotVec()

G. Program usage (already implemented in the code skeleton):

  i. When the program is run, only slerp() result is visible

  ii. A key: Toggle slerp() result

  iii. S key: Toggle interpolateRotVec() result

  iv. D key: Toggle interpolateZYXEuler() result

  v. F key: Toggle interpolateRotMat() result

  vi. Z key: Hide all results

  vii. X key: Show all results

H. Set the window title to **your student ID** and the window size to (480,480).

I. Expected result: Uploaded LabAssignment9-2.mp4

J. Files to submit: A Python source file (Name the file whatever you want (in English). Extension should be .py)