COSC422 Advanced Computer Graphics

Programming Exercise 13

Quaternions

Quaternions[8] are mathematical structures that have been extensively used in computer graphics applications, particularly in keyframe animation, for interpolating between three-dimensional rotations.

The program `RotnInterp.cpp` generates the display of an interpolated motion of a teapot between two orientations shown in Fig. 1. The display of the interpolation sequence can be repeated by pressing the space key.

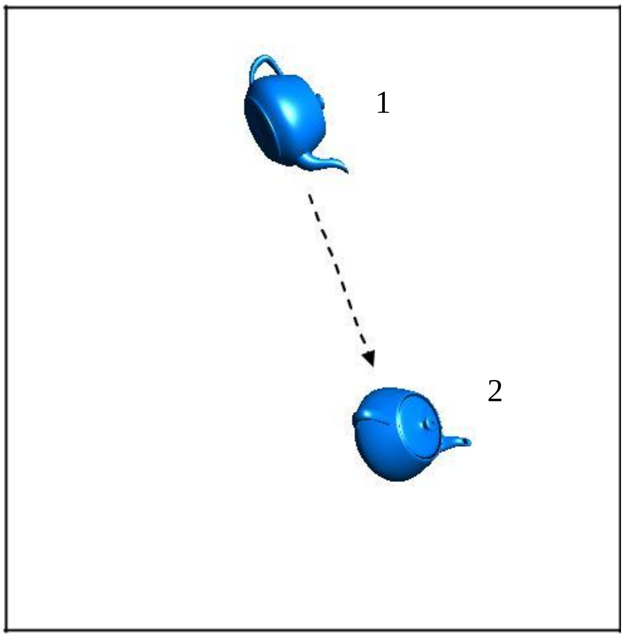

Fig. 1

The two configurations (or keyframes) of the teapot are defined by their positions and orientations in the `display()` function of the program. The orientation is defined in the angle-axis form.

```
//===== Configuration 1 ======
float angle1 = 120.0f;
glm::vec3 axis1 = glm::vec3(-0.8165, 0.40825, -0.40825);
glm::vec3 trans1 = glm::vec3(-2.0, 8.0, 0.0); //Position

//===== Configuration 2 ======
float angle2 = -50.0f;
glm::vec3 axis2 = glm::vec3(0.440225, -0.88045, 0.17609);
glm::vec3 trans2 = glm::vec3(3.0, -5.0, 0.0); //Position
```

The positional and the rotational parameters are then linearly interpolated using the `glm::mix()` function. The interpolation parameter *t* is varied from 0 to 1 in the timer callback function.

```
glm::vec3 trans = glm::mix(trans1, trans2, t); //Translation interpolation
float angle = glm::mix(angle1, angle2, t);  //Rotation angle interpolation
glm::vec3 axis = glm::mix(axis1, axis2, t); //Rotation axis interpolation
```

The interpolated parameters obtained as above are used to construct the transformation matrices. Notice that the display generated by the program shows a twisted motion of the teapot from the first to the second configuration.

We will use a quaternion interpolation between the two orientations and demonstrate the usefulness of quaternions in producing a smooth keyframe interpolation sequence. You will need to include the GLM quaternion header file `glm/gtc/quaternion.hpp`. The implementation uses a few functions of this library as outlined below.

Use the function `glm::angleAxis()` to convert angle-axis representations of the two given orientations of the teapot to quaternions:

```
glm::quat q1 = glm::angleAxis(angle1*CDR, axis1);
glm::quat q2 = glm::angleAxis(angle2*CDR, axis2);
```

Use the function `mix()` to perform a quaternion spherical linear interpolation (SLERP) between the two orientations.

```
glm::quat qt = glm::mix(q1, q2, t);
```

Use `mat4_cast()` to convert the interpolated quaternion to a rotation matrix.

```
glm::mat4 rotation = glm::mat4_cast(qt);  //rotation matrix
```

The above matrix replaces the previous rotation matrix specified using angle-axis transformation.

The program should now produce a smooth interpolation between the two configurations.

[8] COSC422 Lecture slides, “8 Quaternions”.