A4—Shadow Mapping & Projective Texture Mapping

Due: Tuesday October 13, at noon

1. Enhance your graphics pipeline with shadow mapping

Given a 3D scene S modeled with triangles, a point light source defined by a 3D point L, and an output view modeled with a planar pinhole camera PPC, render S from PPC with hard shadows cast by L.

1. Enhance your graphics pipeline with projective texture mapping

Given a 3D scene S modeled with triangles, a projector modeled with a planar pinhole camera PPC0, an image to be projected I, and an output view modeled with a planar pinhole camera PPC1, render S from PPC1 with I projected onto S from PPC0. In other words, use the color of the projected image for a pixel that captures a surface point that is seen by the projector, and use the scene color for the other pixels.

1. Demonstrate the new capabilities of your graphics pipeline

Most of the implementation is in DrawTriangle function in FrameBuffer.cpp

You can switch scenes by define or undefine SHOWSHADOWMAPPING in Define.h

Video is inside output folder.

Please build and run in Release x64 for best performance

* 1. Shadow mapping in a scene with one planar receiver (i.e. object in shadow), one complex receiver, a complex blocker (i.e. object casting shadow), and a moving light. Complex means not a planar surface, e.g. a teapot, bunny, etc.
  2. Projective texture mapping in a complex environment (e.g. the auditorium scene), with a moving projector.
  3. Make a 20s 30Hz 720p video to illustrate shadow mapping and projective texture mapping; the video should have audio narration.

1. Extra credit
   1. Projecting an image with transparent pixels (e.g. project text) 1%. Not implemented
   2. Four light sources that start at the same point and then move away from each other, casting 4 shadows 2%. See Video
   3. Invisibility effect 5%. Given the geometry of a scene S, the geometry and the trajectory of a moving object O, a projector P and the position of an audience approximated with a 3D point A, compute the image P has to project to hide O from A. Make a 10s video that shows simultaneously what the audience would see without the effect, what they see with the effect, and what the projector projects; use a 3-way screen split. See Video
2. Turn in via blackboard one zip archive that contains
   1. Source code
   2. Executable
   3. Video file

© Popescu 2020, popescu@purdue.edu