

Advance Graphics — Homework #2

Image wrapping

Due 4/8/03

In this exercise you are asked to take digital pictures of one or more walls in buildings. Contact me if you don't have access to a digital camera. Using your own program or an existing program, find the pixel matrix of a rectangle showing the part of the wall you are facing (between the floor and the ceiling) between two vertical lines (orthogonal to the floor). These segments should be roughly of the same distance from the point of orthogonal projection of the camera onto the wall. We refer to the matrices you obtained in this way as the *wall-pictures*. Pick a wall that consists of enough structure, such as several paintings.

Next your program should produce a movie which shows a cube rotating in space. The facets of the cube consist of the wall-pictures. Of course, as the cube rotates. Assume that the viewer is located next to the cube, and the cube is relatively large, so there is a noticeable difference in the size of shapes on the walls between parts which are close to the viewer and parts which are further away. You might want to change (stretch or shrink) the wall-pictures to create the cube.

Your program should also let the viewer “fly” through a corridor whose wall texture consists of one or many copies of your wall-pictures. The path of the flight can be predetermined. (that is, the program should not be interactive, though of course it would be a plus).

Ideas about accelerating the program are most welcomed. However, you are not allowed to use z-buffer or structured solid texturing supported by OpenGL.

Please submit the source and executable of your program, and if possible also a video file (.mov, .mpeg, .avi) of the output of the program. Check the webpage of the course to look for updates and FAQ about this exercise.

Suggested order of development Use a standard image format (such as .jpg for example) to create the wall-pictures. Develop the correct geometric transformations. Start from linear motion of the viewer (moves along a straight line while his/her gaze is orthogonal to the wall) and then develop rotation. If you want to use a ray-shooting approach, develop a function that connects a ray and a wall-picture, and returns the pixel of the wall at which the ray hits (if exists).

Make sure to email me if you have difficulties — would try to help from abroad.

You can find useful examples at
www.cs.arizona.edu/classes/cs433/spring02/opengl/canvas.html. To capture a sequence of frames (under windows) please look at <http://www.techsmith.com/defaultflash.asp>.