|  |  |
| --- | --- |
| COMP 4560 |  |

Assignment #3 – due week of April 3

This is a group assignment (maximum three students per group)

The objective of this exercise is to:

* Learn how human sees 3D objects
* Set up the calculations for the synthetic camera transformation and subsequent perspective transformation
* Apply the calculations for a 3D object using two different views
* Use the views in an anaglyph picture to view the 3D object

We are 3D creatures, living in a 3D world but our eyes can show us only two dimensions. The depth that we all think we can see is merely a trick that our brains have learned; a byproduct of evolution putting our eyes on the front of our faces. To prove this, close one eye and try to play tennis. The miracle of our depth perception comes from our brain's ability to put together two 2D images in such a way as to extrapolate depth. This is called stereoscopic vision. It works like this. Because your eyes are separated on your face, each retina produces a slightly different image. That difference in images is a direct result of the depth of the objects that we are looking at.

|  |  |
| --- | --- |
| Image result for left eye right eye view | http://scecinfo.usc.edu/geowall/images/stereo01.jpg |

3D stereoscopic imaging is as simple as producing two slightly different images - the same as your eyes would produce - and then showing each eye only one of those images. This can be done with light-refraction, color-filtering, or light polarization. In this lab, we are going to use color-filtering (Anaglyph) method. Anaglyph 3D is the name given to the stereoscopic 3D effect achieved by means of encoding each eye's image using filters of different (usually chromatically opposite) colors, typically red and cyan. Anaglyph 3D images contain two differently filtered colored images, one for each eye. When viewed through the "color-coded" "anaglyph glasses", each of the two images reaches the eye it's intended for, revealing an integrated stereoscopic image. In simpler words, if you create an image with red and cyan colors, the eye red filter only sees the red colored shapes, and the eye with cyan filter only sees the cyan colored shapes. Anaglyph glasses are shown in the figure below and will be provided by the lab instructor.

|  |  |
| --- | --- |
| Image result for anaglyph glasses | Image result for anaglyph glasses |

An example of anaglyph image is shown below. Use the provided anaglyph glasses to see the image.



The scene to be viewed in this lab is the shape sketched in the figure below. In all, 33 vertices are required to describe this shape, and they are labeled as ‘a’, ‘b’, through ‘ag’ as shown. (Note that the vertex ag does not actually appear on the picture and it is just used for calculation purposes.)

fig1b

This assignment will be summarized to:

1. Using Excel to finding the perspective transformation of the sketch for each eye separately
2. Rendering the images separately in Red and Cyan colors in the programming language of your choice
3. Use anaglyph glasses to test the 3D image

Here are the different steps for this assignment:

**Step 1:** Download Assignment 3\_Data.xlsx file from D2L. It contains the dimensions of the object with which you will be working. All of the other cells in this coordinate table are linked to the B30, C24, and D7 cells by formulas. For this assignment you can start with B30=80, C24=100, and D7=70. The unit is millimeters. However, you will be asked to change the values in the marking session. **For this reason it is important that throughout this assignment, you use formulas and cell references wherever possible.** This should determine the coordinates of all of the vertices in your basic shape. Note how the relative dimensions and overall position of this basic shape can be changed by just changing the entries in the cells with a colored background.

**Work vertically down the spreadsheet. Each of the following steps should appear with the heading “Step \_\_”, and the values clearly labelled. Line up the vectors in such a way that it is easy to use cell references to compute values.**

**Step 2:** Consider uvn-coordinate origin at and is pointing at (0, 0, 0). You can start with millimeters, but you may be asked to change during the marking session. Find the vector and normalize this vector to get a unit vector . **You may check your work with a hand calculator, but no numbers calculated by hand should go into the spreadsheet!**

**Step 3:** Find vector that represents “more or less up” in uv-plane. Normalize to get the unit vector .

**Step 4:** Calculate .

**Step 5:** Set up the 4 x 4 translation matrix **T**-r and the matrix **M**T. The elements of these matrices must be created by formulas linking their values to the values in the cells containing the components of the , , and vectors, and the coordinates of the uvn-coordinate origin, as necessary. Form the product **A**WV = **T**-r•**M**T, using Excel’s mmult() function. While you’re at it, set up the matrix **M** and form the product **M•M**T, to confirm that you get a 4 x 4 identity matrix. If that doesn’t happen, it means you have an error in at least one of your vectors , , or .

**Step 6:** Form the matrix product **X•A**WV = **X**uvn, where **X** is the 33 x 4 matrix of (x, y, z)-coordinates of the vertices in the shape. This gives **X**uvn, a 33 x 4 matrix of coordinates of these thirty-three points relative to the **uvn** coordinate system constructed before.

**Step 7:** **The Perspective View for two different eyes**

In lecture notes, for simplicity, we let that is, the eye is just behind the origin of the uv-plane (refer to the left image below). However, in reality you have two eyes with a distance from each other and they both can’t be at . Assuming that the observer locate the two eyes in uv-plane and with symmetry with respect to v-axis, we can let , but the value depends on the distance between the eyes (refer to the right image below). Basically, we can assume to be half of the distance between the two eyes. Based on surveys, the average distance between the two eyes in people are between 62-64 millimeters. In this exercise, to be consistent, we assume it to be 64 millimeters. We may change this value to get better 3D view, when testing it next week.

|  |  |
| --- | --- |
|  |  |

So, we need a little theory to be able to find matrix P for the cases with non-zero . Note that is a positive value for the observer’s right eye, and a negative value for the left eye. Similar to lecture notes, we consider an arbitrary point on the object – let’s call it **p**=(pu, pv, pn). Then the line of sight extending from the viewer’s eye to the point **p** is the line extending from **e** to **p**. As we vary **p**, our line of sight changes.

For each point **p**, we will consider the point at which the line from **e** to **p** intersects the uv-plane. This point of intersection will be the point **u\*v\***. The parametric equation for this line is:



Solving for the value for which the n-component is zero:



Then



So, we have formulas for u\* and v\*, to give the image of the object as it actually appears to the viewer. However, this is not a very computationally-friendly form for u\*v\* coordinates. As before, what we’d really like is a way to convert points on the object (pu, pv, pn) to points on the u\*v\* plane using matrices; that is, we’d like a matrix P such that

[pu pv pn 1]P=[u\* v\* n\* 1]

Since we wish to express the point (pu, pv, pn) in homogeneous coordinates in such a way that obtaining u\* and v\* does not require us to do any division, it makes sense to make



Which makes:



Now we can obtain the transformation matrix P.



or equivalently,



Set up the uvn-coordinates in the center of the screen and assume that the observer’s two eyes as (-32, 0, ‑1000) and (+32, 0, ‑1000). This means that the observer eyes have 1000 millimeters distance from the computer screen in *n*-direction, and are at the origin level in *v*-direction. Construct the 4 x 4 perspective transformation matrix discussed above separately for each eye. Then, for each eye, multiply this matrix onto the uvn-coordinate matrix of the points to produce the intermediate 33 x 4 matrix of perspective coordinates. From this create the 33 x 4 matrix of u\*v\*n\*-coordinates in which the homogeneous coordinate of each point has been restored to 1.

**Step 8:** Use any suitable programming environment you wish (eg. Javascript, C#, JAVA, C++, VB, etc) to plot the perspective views. You should find the length of each pixel on the lab monitors to convert the numbers in millimeters to pixels. Consider the origin of the u\*v\*-plane in the center of the monitor and convert all the locations. Note that you will need to load two sets of data: one for the vertices (in u\*v\*-coordinates), and one containing information about which vertices are joined. This is similar to the “points” and “lines” files you loaded for your term project, so you may wish to “borrow” some of the code provided for that assignment. Use red color for one of the views (the one with red lens) and Cyan color for the other one. The canvas should be black. Doubled-check the lines with the anaglyph glass provided. You should be able to see only the red lines using red lens and only cyan lines using cyan lens. If both are visible to the observer, play with the color of the lines to make sure each eye only sees its own perspective image. Make sure that only the lines visible to observer’s eyes are plotted (the line on the back should not be plotted.

**Step 9:** Use anaglyph glasses to view the 3D object.

**To increase the quality of the 3D view you may:**

1. Use white color for the red-cyan intersections
2. Play with the colors to make sure only one image is visible to each eye
3. Play with the considered eye distances to make the image suitable for your eyes

Deliverables:

1. In the Dropbox folder, submit an excel file showing all the calculations**. To make it easier to mark, put heavy borders around A-matrix, P-matrix, coordinate values in uvn-coordinate system, and u\*v\* values.** (Only one student in each group needs to make a submission.) In addition, submit a single .txt file that contains the names and student IDs of everyone working on this assignment.
2. Demo of the 3D view **on the lab computers** during the week of April 3.

**References:**

[1] Geowall at The Southern California Earthquake Center, <http://scecinfo.usc.edu/geowall/>

[2] "Anaglyph 3D." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 16 Sep. 2017. Web. 10 Nov. 2017

[3] Gordon, C. C., Blackwell, C. L., Bradtmiller, B., Parham, J. L., Barrientos, P., Paquette, S. P., Corner, B. D., Carosn, J. M., Venezia, J. C., Rockwell, B. M., Murcher, M., & Kristensen, S. (2014). 2012 Anthropometric Survey of U.S. Army Personnel: Methods and Summary Statistics. Technical Report NATICK/15-007. Natick MA: U.S. Army Natick Soldier Research, Development and Engineering Center.