

CIS 410/510: Project #3

Due Jan 24, 2022

(which means submitted by 6am on January 25, 2022)

Worth 7% of your grade

Caution: please read step (4) closely. It is dense and has a lot of instructions. If you miss part of its meaning, you will likely get the wrong answer.

Steps:

1) Download skeleton file proj3.cxx and file data_proj3.vtk and put them in a new directory.

a. proj3.cxx has an empty implementation for EvaluateFieldAtLocation ... copy your implementation from proj2.

2) Re-use your CMakeLists.txt from the last project ... copy it to the new directory and rename as appropriate to reflect proj3.

3) Run cmake, compile the program, and run the program. It generates three outputs: "bluehot.png," "difference.png," and "hsv.png." If you look at them before proceeding to steps 4 and 5, all three images should be black. Your job is to make them generate the right colors.

4) Look for the comment "ITERATE OVER PIXELS." You need to implement code here. Your code should map the physical space $X=-9 \rightarrow +9$, $Y=-9 \rightarrow +9$ to an image of size n_x by n_y . Specifically, map $i=0, j=0$ to $X=-9, Y=-9$ and $i=n_x-1, j=n_y-1$ to $X=+9, Y=+9$. Each of the n_x times n_y pixels will get mapped to a two-dimensional location, and the locations between the extreme values (between $i,j=0, i=n_x-1, j=n_y-1$) should be placed evenly between -9 and $+9$. Once you know the two-dimensional location for a given pixel, interpolate the field to that location. Note that the scalar field has a minimum value of 1.2 and a maximum value of 5.02. You should normalize these values before you do your color mapping. For example, a field value of 3.11 (= 50% of the way between 1.2 and 5.02) should be normalized to 0.5. When you get to the color mapping portion, a value of 0.5 should be mapped to the middle of the color map. Once you have done that, the pre-existing code ("I TAKE OVER HERE") will call a function to map the field value to a scalar.

5) Implement the following functions (all functions defined in function header in the source code):

- a. ApplyBlueHotColorMap
- b. ApplyDifferenceColorMap
- c. ApplyHSVColorMap

6) Cross-reference with the correct images posted on the website using the differencer. I expect the images to be correct, and you should expect to receive less than half credit if it is not. HOWEVER: floating point precision is an issue, and we

may get a few pixels differences for each image. If you have <20 pixels different, then you are good to submit – the difference is almost certainly floating point issues. If you have >20 pixels different, then there is very likely a problem with your code. (Especially if there are many, many pixels that are different.)

7) Upload your source code when it is working. What to upload? – just proj3.cxx.

== More on differencer program ==

The process to generate the images for this project involves floating point math. While we all may be doing the same thing conceptually, we may use slightly different formulations to get there. Occasionally, this can lead to slightly different images. The rate at which these differences occurs is very low. For example, I ran an example where two different implementations differed on 4 out of 250,000 pixel calculations. But we will need to tolerate a little error.

I have provided a "differencer" program that will allow you to difference your output with the reference output. If the number of pixels different is <=20 pixels, then we will declare this correct (full credit). If it is more than that, then we will declare it incorrect (less than half credit).

I include an example of running differencer below. More information on how to set up differencer is on the class webpage.

```
# compares hsv.png generated by my proj3 program with the hsv.png file I
downloaded from the class webpage
% ~/differencer/differencer.app/Contents/MacOS/differencer hsv.png
~/Downloads/hsv.png
Both images have 250000 pixels, good.
Difference at column = 341, row = 54
  File hsv.png has 0, 255, 201
  File /Users/hank/Downloads/hsv.png has 0, 255, 202
Difference at column = 176, row = 329
  File hsv.png has 255, 0, 132
  File /Users/hank/Downloads/hsv.png has 255, 0, 131
Difference at column = 204, row = 342
  File hsv.png has 207, 0, 255
  File /Users/hank/Downloads/hsv.png has 208, 0, 255
Difference at column = 210, row = 385
  File hsv.png has 131, 0, 255
  File /Users/hank/Downloads/hsv.png has 132, 0, 255
The number of different pixels is 4
Writing file differenceMap.png
```