Assignment 3: Computing viewsheds

Name: Son D. Ngo
Date: 10/7/2016

How to compile and run your code:

- Run this program with 4 arguments on the command line:
  - *input file*
  - *output file (path)*
  - *x-coord of source*
  - *y-coord of source*

- Running locally:
  g++ -Wall -o flow flow.cpp grid.cpp
  ./viewshed [grid-file-to-read] [file-to-write] x-coord y-coord

- Running on dover:
  g++ -std=c++11 -Wall -o flow flow.cpp grid.cpp
  ./viewshed [grid-file-to-read] [file-to-write] x-coord y-coord

List of files in your project with a brief description of what they do:

- grid.cpp: Class file for Grid class, containing functions that can be operated on grid, such as *multiply, computing FD, FA and viewshed*.
- grid.h: Header file for Grid class.
- viewshed.cpp: Main file that sets up the procedure of reading ascii file, and then compute viewshed. The functionality is implemented by grid.cpp.
- Snapshots of viewsheds for different ascii files.

Small tests

What small grids have you created to test your code? When you check in the code, please check these in as well. Describe the grids and describe if your code works or not.

I tested with the *box.asc* and *tester.asc* provided by professor Toma. The code works with these two tests as expected.

For *box.asc*, I tested with viewpoints (0,0) and (9,9) (middle point), other corner points (0, 19), (19, 20) and a random point (4,4). The grid in this test looks like a gird with 9 boxes of different sizes. The boundary between each box is of size 3 with height 5, while the height of points within the box are 0. If you stand in the box, you should only be able to see things in the box. If you stand on the boundary, you should be able to see all other points on the boundary, and some parts inside the box.

For *tester.asc*, I tested with viewpoints (0,0) and (3,3) (middle point), and (2,2). This grid in the test looks like a prison, where the point in the middle is surrounded by a wall of height 30, and a layer of height 20 outside that wall is yet surrounded by another wall of height 100. If you stand in the middle, you should be able to see both of the walls, but not the layer of height 20. Similarly, if you stand in the layer, you can't see anything in the middle.

Testing set1.asc

Using your code compute the viewshed on on set1.asc with vp=(100,100) and vp=(250,250).

Visualize the viewshed computed by your program with QGIS and attach a snapshot for these two viewpoints.

Figure 1. set1.asc at (100,100) (cropped)

Figure 2. set1.asc at (250,250) (cropped)

Does your code give the correct (looks visually identical) output on set1.asc with vp=(100,100) and vp=(250,250)? YES!

Testing larger datasets

Please run your code on the datasets below (pick a viewpoint approx. in the middle of the terrain) and record the running time. You can use your laptop, one of the public linux machines, or the desktops in the labs. In each case, visualize in QGIS and attach a snapshot of the viewshed computed by your code.

DATASET (size in points)	Time to run viewshed from vp = middle (cpu secs)	Middle Point
brunsdem (200K)	0.704188	(266, 193)
kaweah (1M)	2.00128	(712, 581)
sierra (6M)	56.0644	(1875, 1336)
portland_me (120M)	1292.12	(5346, 5994)
usadem2 (250M)	627.349	(12750, 5500)
eelriver (500M)	16870.2	(12535, 10587)
washington (1000M)	11495.4	(16727, 15993)

(snapshots for eelriver and washington are included in the svn folder)

Figure 6. brunsdem.asc at (266, 193)

Figure 7. kaweah.asc at (712, 581)

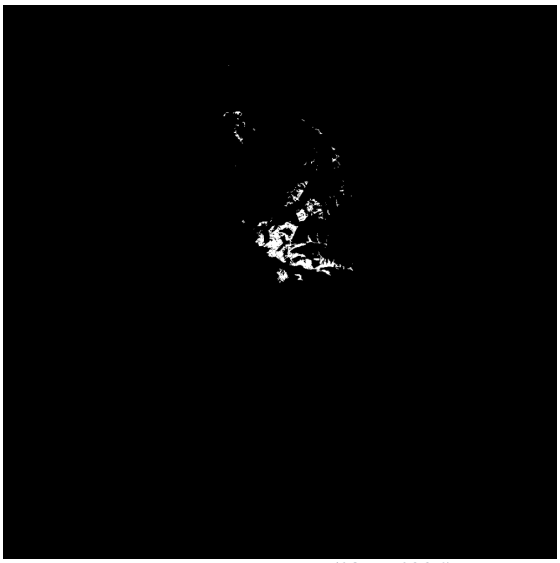

Figure 5. sierra.asc at (1875, 1336)

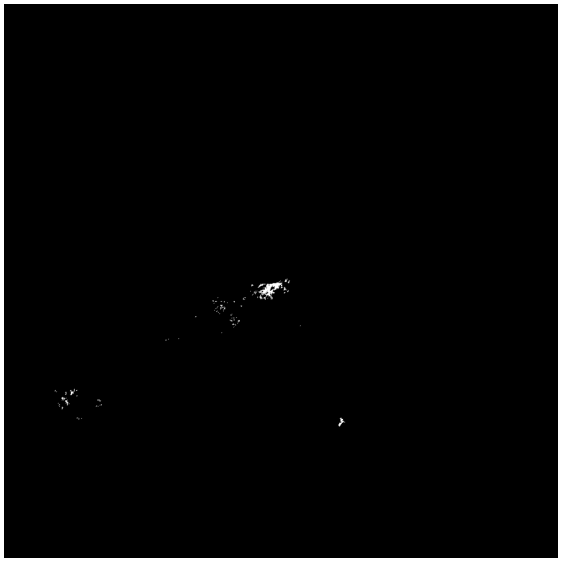

Figure 4. portland_me.asc at (5346, 5994)

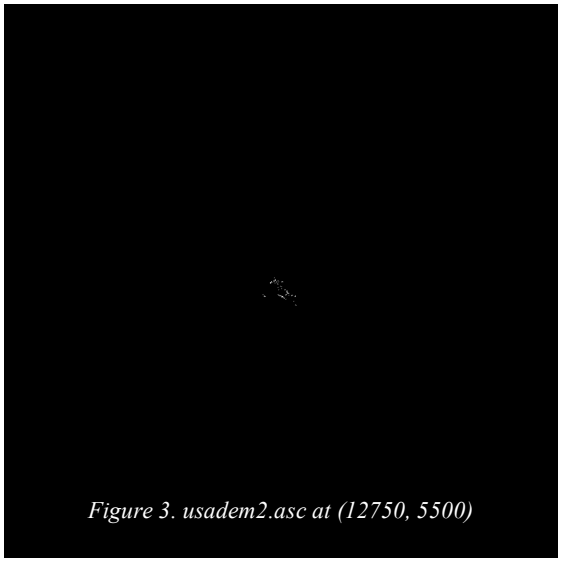

Figure 3. usadem2.asc at (12750, 5500)