Mikhail Kouzminov

3/21/19

Implementing streaklines to show wind space traveled

Abstract:

Using wind data collected form 7 stations hourly across the Bay Area over a period of 12 hours, I implemented drawing streaklines, showing the front of how far the wind has traveled over a period of 12 hours. In a addition, I implemented the ability to view the previously calculated information – streamlines, a velocity plot, and individual station information separately, hourly over the course of those 12 hours.

Introduction:

Whereas streamlines show flow under a constant field, and pathlines show flow with a changing field, streaklines show how pathlines change over time, when released from a single given point. They can be used, in addition to pathlines , to see how something would travel if it were leaving a streak behind it. For example in the case of wind, a pathline would show the trail of smoke if a large amount were released at a given moment, assuming altitude were not a factor, and a streakline would show the trail of smoke if it were released constantly over a period of time. It is created by connecting the endpoints of the pathlines at any given moment.

In the case of this project, I had the overall goal of being able to visualize how wind changes over time, and I mostly succeeded.

Several changes have been made to previous work, however. Namely, the number of seed points for both streamlines and the velocity grid has been decreased in order to match up with the number of seed points used for streamlines. This is easily adjusted within the code.

Related Works:

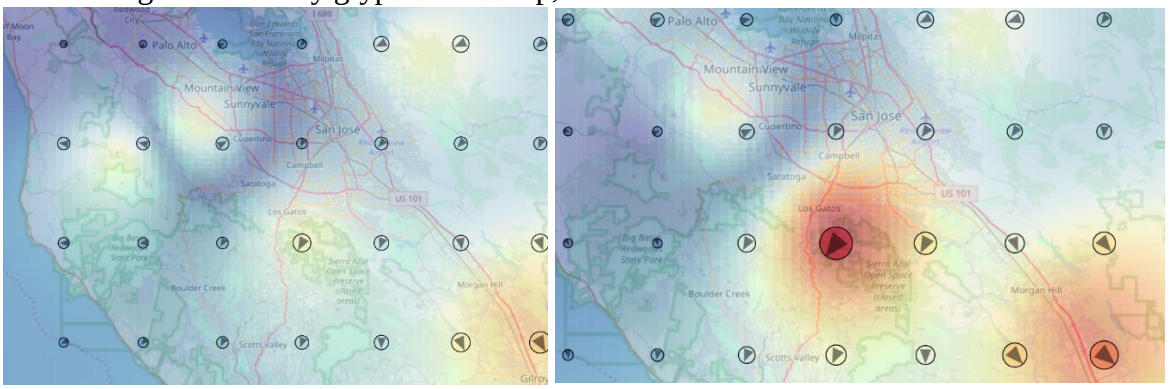
The webpage that was most helpful to use in understanding how various form of visualization, including streamlines, work was: https://www3.nd.edu/~cwang11/2dflowvis.html

My first project for this class would also qualify as a related work, in that this visualization is an extension of that one.

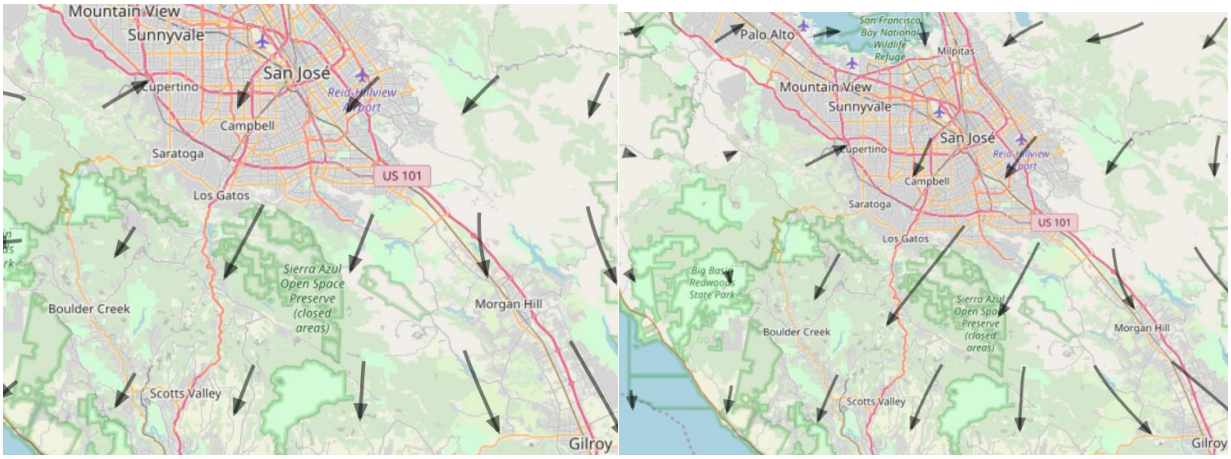
Technical Detail:

My extension of the project started out with allowing the display of multiple visualizations taken over hourly periods. For example, I extended my heatmap and streamlines visualizations to show multiple versions separated by time by moving a slider on the right side of the screen, as well as having a display of exactly what time it was taken. I kept my previous inputs of being able to switch heat map on and off with a button, as well as being able to switch visualization modes from showing stations, to showing velocity glyphs, to showing streamlines, to showing pathlines by pushing another button. While this makes it impossible to display more than one mode at a time, this would typically get too crowded quickly when attempted previously.

A few images of velocity glyphs / heatmap, to show difference over time:



1Taken at 7:00 vs taken at 6:00



2streamlines for the same

As shown, the 6:00 image has a lot more wind visible. In addition, as shown, there are 4 times fewer seed points than in project 1. This is to ensure that the very crowded combination of pathlines/ streaklines shown in later images don't get too crowded, and that the same points are always seeded.

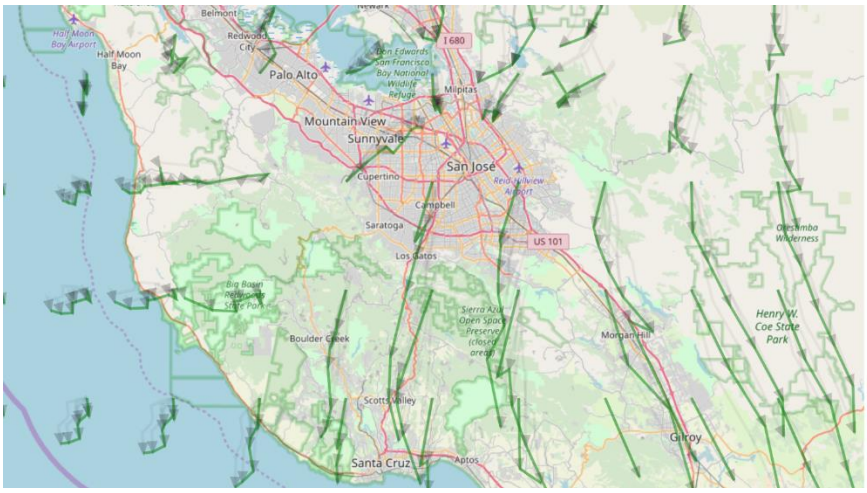
Other than that, as demonstrated in the above image, while wind speed was mostly stable, and didn't get past 6 mph on the day measurements were taken, the image show that there is a notable relative difference in wind strength and distance traveled. This is later demonstrated more thoroughly.

Drawing Streaklines:

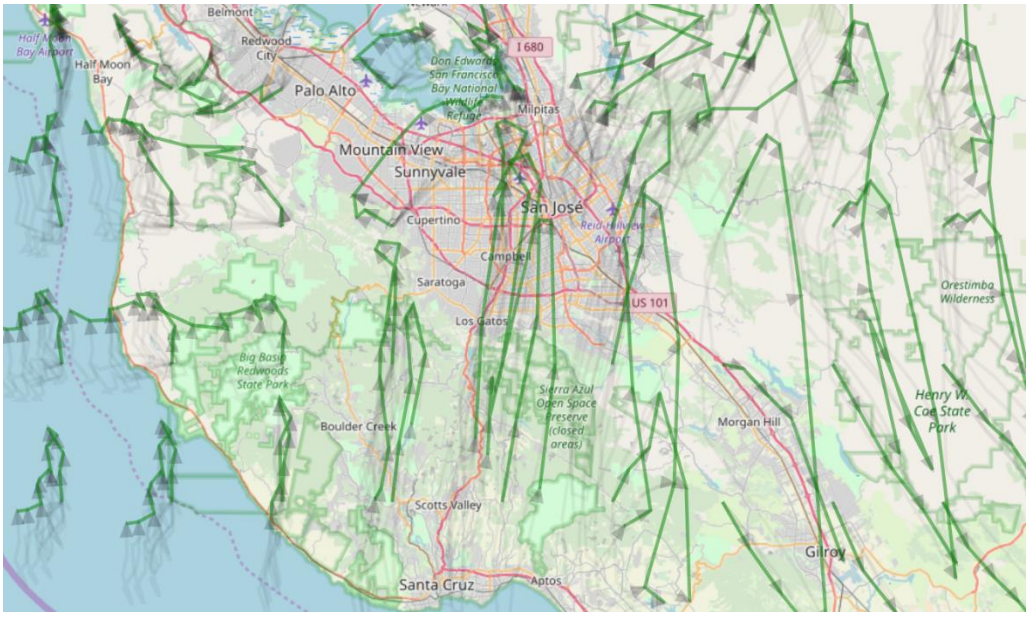
Unlike in the case of streamlines, distance traveled does matter with pathlines and streaklines, since we are taking values given every hour, and a wind shown going 10 mph, must travel 10 miles in a given direction within an hour. While I was unable to figure out the proper map for this, I measured the latitude, which averaged to between 22.35 and 23.04 on different sides of the screen. Longitude was a constant 74.63 miles per degree.

I also lightly showed the pathlines that go into making the streakline. They are the light black lines , while the streakline itself is painted dark green. Other than that, it is worth noticing that drawing this many streaklines, with the extra length from matching actual distance, and with extra lines from showing pathlines really messes with visualization. It can be difficult to see the fine details of the streaklines, especially when adjacent ones travel in the same direction and thereby overlap.

While they start out short, the streaklines tend to heavily overlap when over 4 hours long, and some lines were much longer than others – almost the size of the entire image. In particular , lines to the right side of the screen tend to have a longer size than those to the left. This is due to the lines on the left actually reversing movement direction partway through the process, and having existing lines get closer to old ones than they were before.



This is a large valley, so it is not immediately obvious that this many streaklines is problematic, but as they grow, this becomes very annoying. The streaklines become nearly impossible to differentiate between.



Do note, that a lot of these seem to not move due to staring off in one direction, then reversing and staying in mostly the same place a little later.

Having a few more spots to take measurements from would have been very useful for this project, since having only 7 or so spots leads to what is likely a much less accurate portrayal.

Results/ Other' impressions:

In addition to the above display of streaklines, and the improvement on prior portions of the project, there are not many actual results. This is likely due to having both a mix and lack of data, and only sampling over 12 hours, during which period overall differences wouldn't get too large. Some better choices might have been made in visualization, but those have been covered in the result for Assignment 1. Namely, that there is a lack of actual animation, and that there may not be enough steps in the gathered data.

The color choice might also be a little weird. While the green is visible, it may have been a better idea to make the streamlines black, and the lines leading to them green, instead of the other way around.

I couldn't figure out how to make the project actually animate, which might have made it look neater, but having a bar on the right seemed close enough to an animation to show how the streakline gradually transitions between hours.

As for showing different versions of the streamlines / velocity glyphs: It may have been a good idea to interpolate over time between the same velocity glyphs and streamlines. This would have led, however, to even more interpolation, in a program already lacking in accuracy due to it's small scale and large amount of data points. While such an interpolation might have been possible, and might have resulted in smoother pathlines/ streaklines, the version used has a lot less math used, and is easier to process and understand how the code works.

As for the code itself: sections of it can be a little confusing, particularly when it comes to calculating streaklines. There are a lot of points used, and since pathlines were not calculated separately and simply combined into one array, it can be a little confusing trying to figure out the logic. Nevertheless, it is fairly similar to the logic for calculating streamlines, just done over a larger period of time, leading to it being readable without too much explanation. As for organization: There is a fairly obvious order, with all display options being written in a logical order, and clearly separated in their description from one another.

Conclusion:

This project would have possibly looked a lot better as a smooth animation, and maybe interpolated over time to ensure the actual transition between different streakline lengths was smooth. Unfortunately, there was a lack of data available to show such a thing. Transitioning from lines of 1 line segment in length to lines of 12 this quickly leads to a bit of a mess. While reducing the number of seed points was helpful for clarity, it was not necessarily something to do to all visualization options. Having more seed points for streamlines / velocity glyphs would have helped give a more thorough picture in the case of needing one – though it is less necessary in the current case with the sparsity of available data. Green might also not be the best color for displaying streaklines.

Nevertheless, this project succeeds in what it set out to do: using streaklines to provide a way to visualize wind data changing over time. The final result leads to some interesting displays, due to the lack of movement and outright wind direction reversal on the west side of the map, and the acceleration of wind as movement south and through time happens leading to the large separation of pathline endings on the east side of the map. This sort of visualization seems to be very good at displaying the acceleration of wind, and not just the front of a released "net" of smoke from a point.

3East side vs west side of map (long v short streaklines)

This might also somehow indicate that there is a long valley with a wind corridor at the east side of the map, and be useful data when setting up wind farms and the like.

Reference Materials:

Website explaining types of visualization: https://www3.nd.edu/~cwang11/2dflowvis.html

Wikipedia article on Streaklines:
https://en.wikipedia.org/wiki/Streamlines,_streaklines,_and_pathlines

Calculator used in calculating distance between sides of map in miles for implementing streaklines:
http://boulter.com/gps/distance/

Discussion of proper way to do this (not used in final version):
https://gis.stackexchange.com/questions/142326/calculating-longitude-length-in-miles

Wind data taken from: https://www.windfinder.com/#10/37.2680/-121.6296

Other sources used for copying bits of code:

Main Source (from which I took most of my starting code, including the map/tiles system:

http://bl.ocks.org/mbostock/eb0c48375fcdcdc00c54a92724733d0d

Tutorials for markers (Borrowed a decent amount of code for making arrows):

http://tutorials.jenkov.com/svg/marker-element.html

http://bl.ocks.org/dustinlarimer/5888271

Wikipedia article on Shepard's Interpolation:

https://en.wikipedia.org/wiki/Inverse_distance_weighting

Wikipedia article on RK4/ Euler methods:

https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods