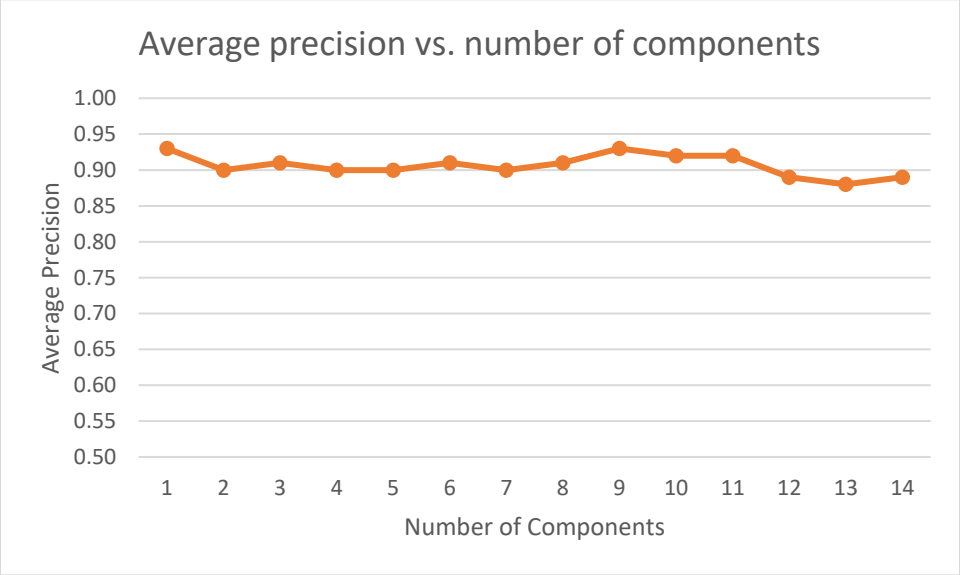
Philippe Sayegh

260637812

Comp 417 – Assignment 4

Question 1



We notice that the average precision stays about the same throughout the range of number of components tested here. The variations are so unpredictable and oscillating that they probably originate from some sort of computation artefact (rounding?) or noise.

We are working in case where we have 4 classes of terrain for the PCA to analyse. Converting these 4 observable variables/categories to more than 4 principal components seems redundant, and probably doesn't offer much of an advantage in terms of precision, if any.

Conversely, were we to increase the number of classes, increasing the number of principal components would probably yield better results.

Question 2

As I do not have any screen recording software on my Windows laptop (on which I worked on this assignment), I did not manage to make a video of the drone moving. However, I did include a few screenshots to show the progress of the drone, for the second algorithm (only one screenshot for the random algorithm, it's of much lower interest anyway).

I had to stop the algorithms before they had time to explore 200 to 400 distinct tiles, as the first algorithm would've taken much too long of a time to reach such results, and the difference between the results of both algorithms is already clearly visible by looking at the screenshots.

Algorithm 1 (Brownian) :

unique tiles visited: 40

visited tiles total: 463

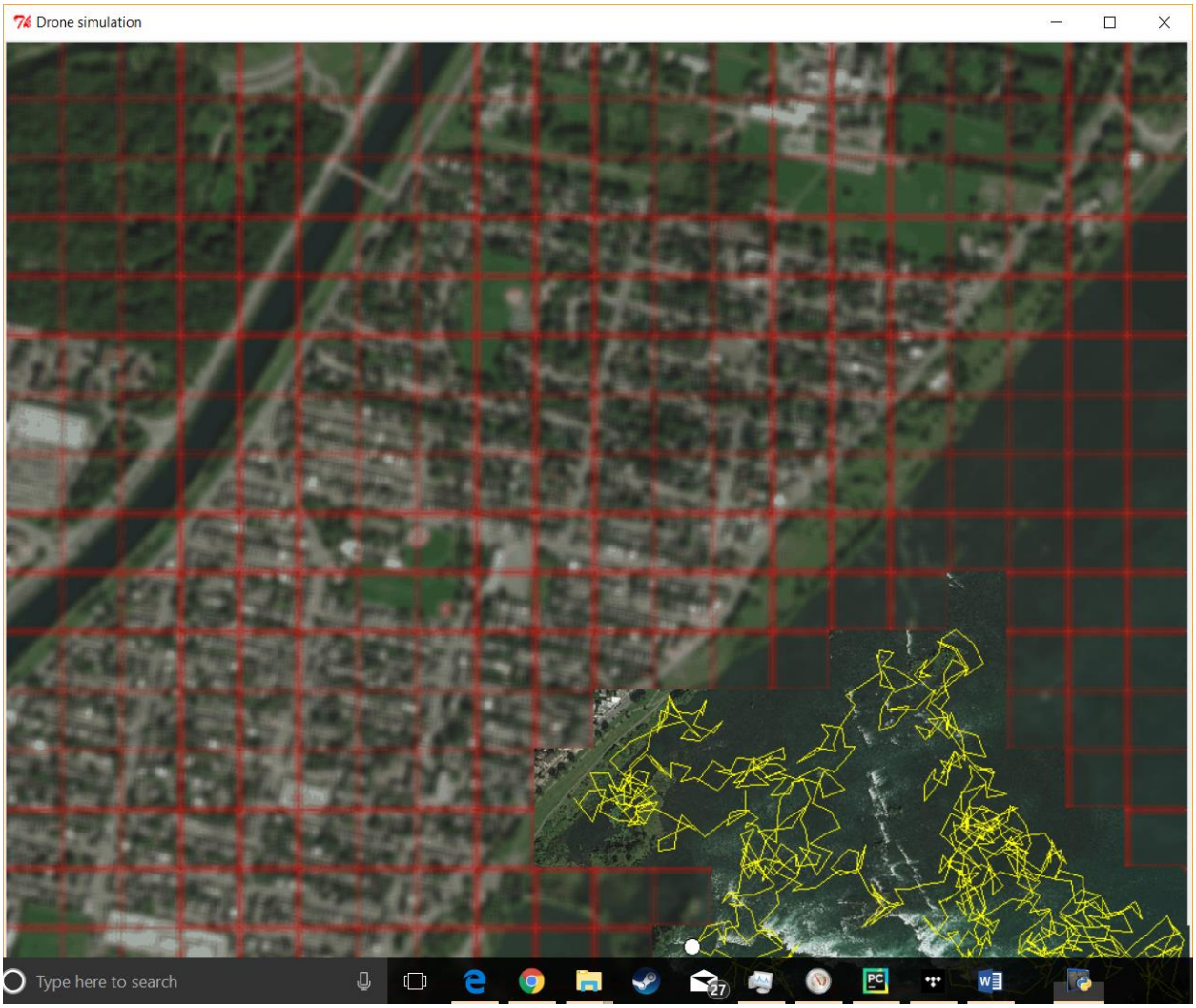
Ratio : 0.07

As far as I can tell, the only pro of this algorithm is its simplicity. It runs fast and is simple to understand, and it does, with time, explore the map. But that's pretty much it.

As it guides the drone completely at random, it does not work very efficiently towards the goal of covering the map.

It does avoid urban areas, for what that's worth.

Screenshot from algorithm 1:


Drone simulation
Type here to search

Algorithm 2:

unique tiles visited: 153

visited tiles total: 388

Ratio: 0.4

My second algorithm is a mix between a greedy procedure and a randomized one.

At each refresh, the drone checks if the tiles directly to its north, east, south and west are already explored, by checking its internal record of visited tiles (some sort of internal map).

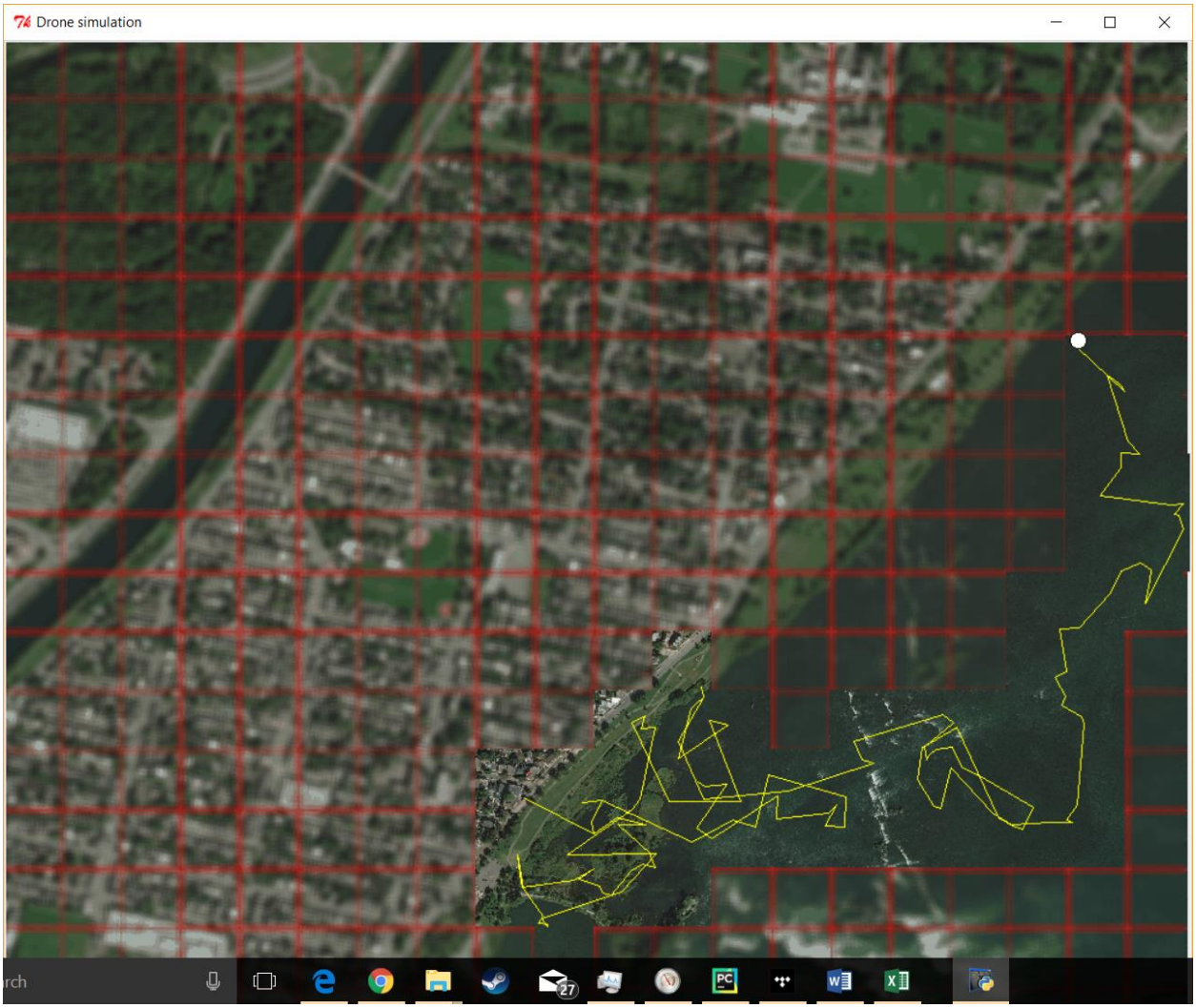
If there is at least one unexplored tile in its vicinity, it picks one at random and travels in the rough direction of that tile (the movement includes a random component).

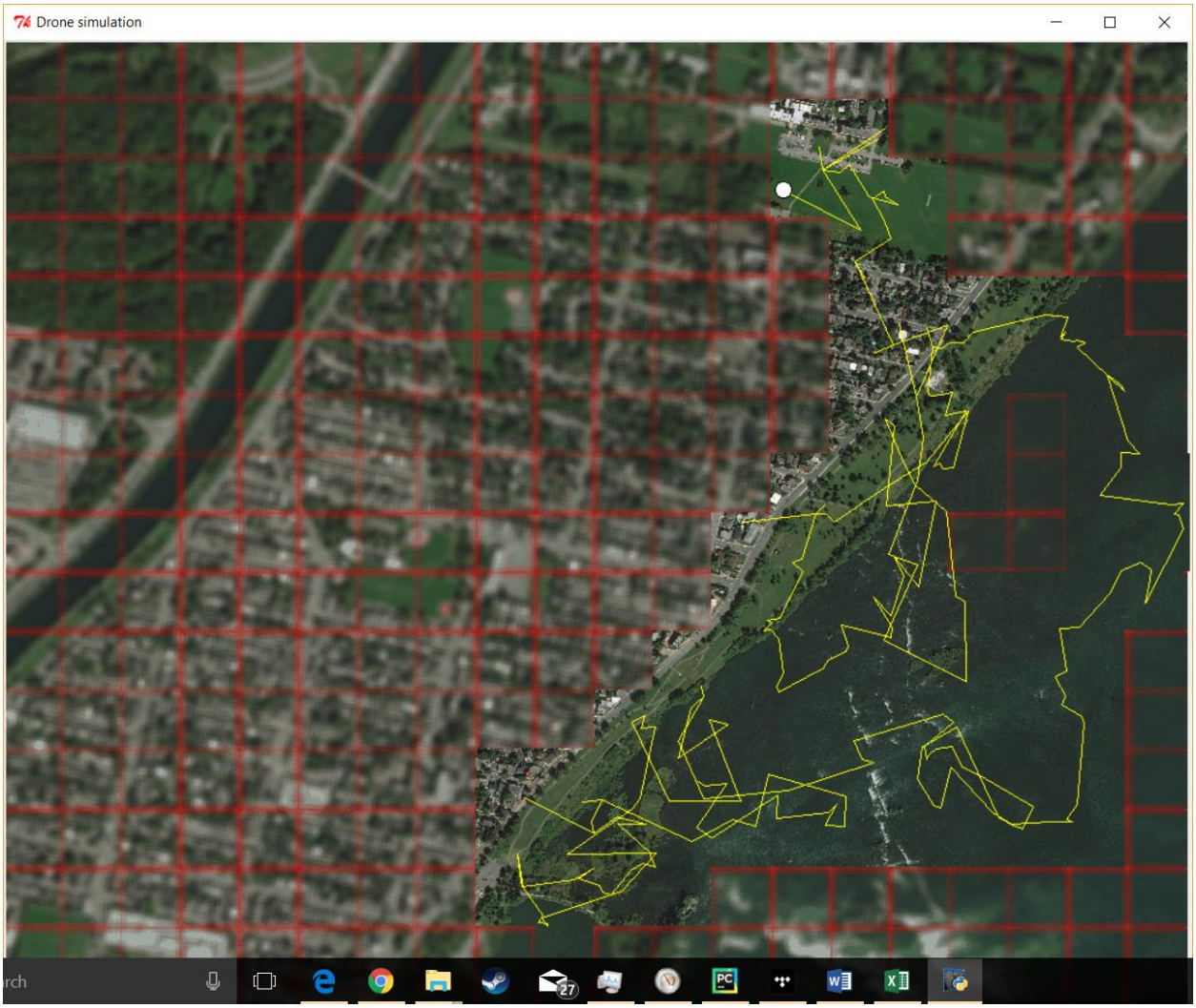
If no unexplored tiles are detected, the drone heads in a random direction by a random, but potentially much larger, step (up to 70.

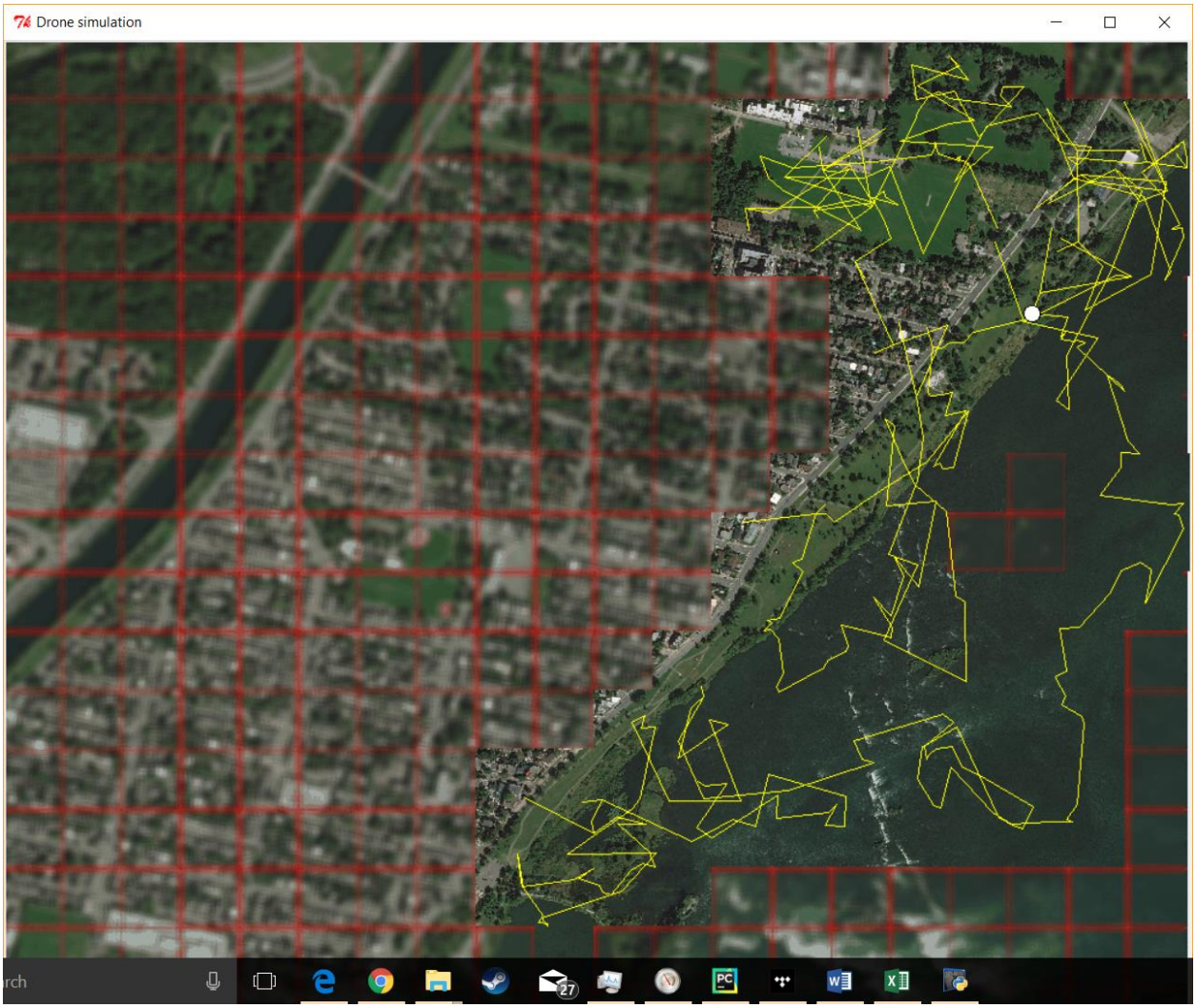
The pros of this algorithm are its greedy efficiency at hunting for new tiles to explore, …when it does. Due to the partial randomness of implied in the algorithm, the results vary and might be subpar.

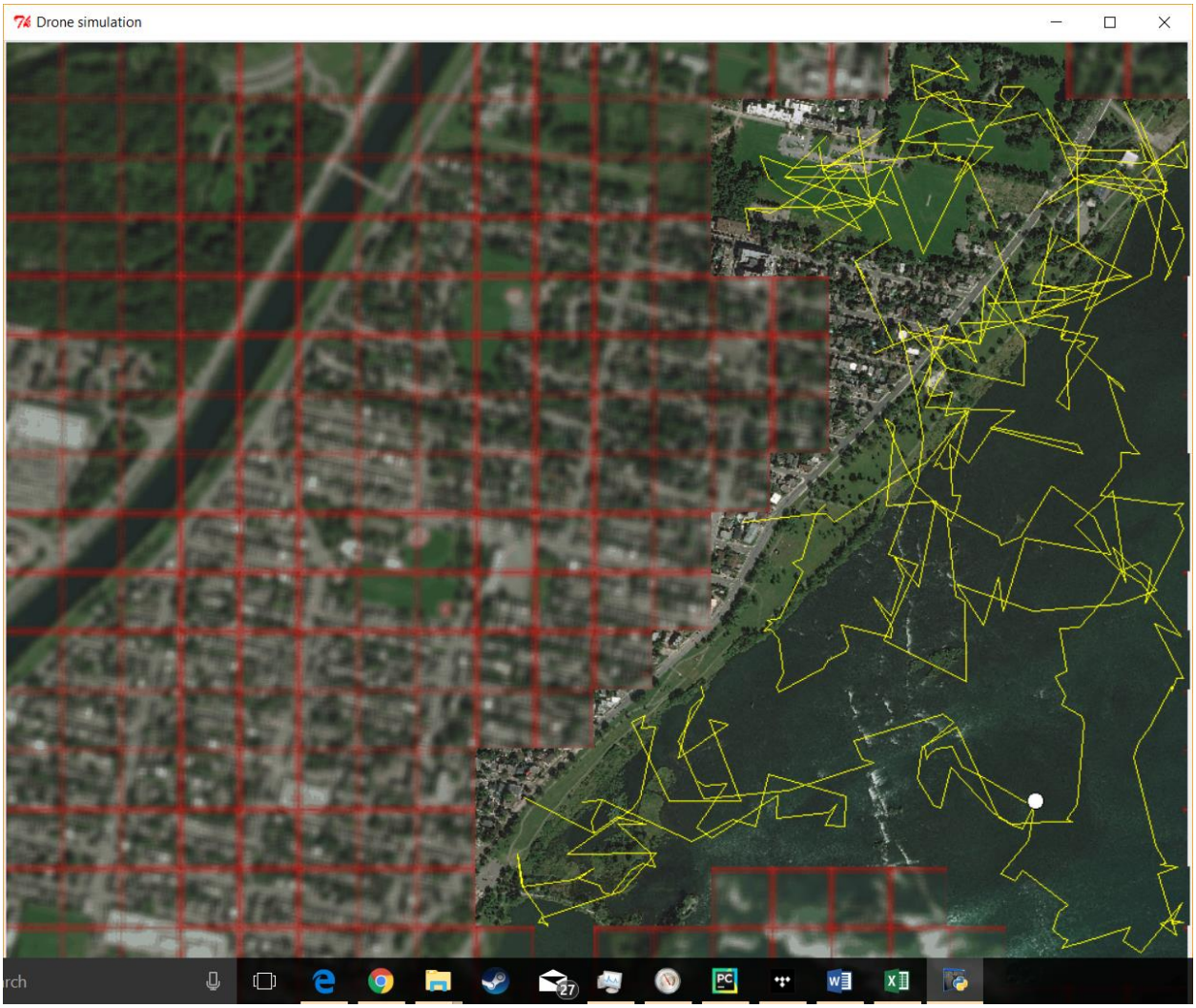
However, all in all, I noticed a definite improvement over the Brownian method. This algorithm yields significantly higher unique vs. total tiles ratios, on multiple trials.

Screenshots from algorithm 2:


Drone simulation


Drone simulation


Drone simulation


Drone simulation