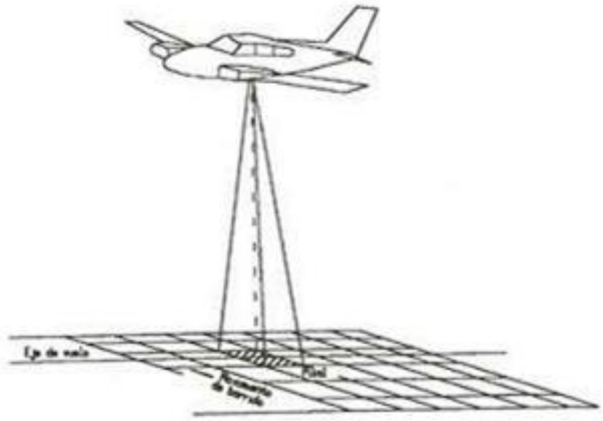
Project – Mapping Drone



Overview

The aim of our project is to build a map in a real-time manner, in other words, it is a scanner that will scan the area that the drone traverses above.

The information is stored as coordinates in a data structure, and the display will be a window that will show a map of the area around the drone, the map consists of tiles, where each tile is a picture taken from the drone.

The information that will be stored and viewed is: global position, an interactive flight map of the drone and a photographic map.

1. Global Position

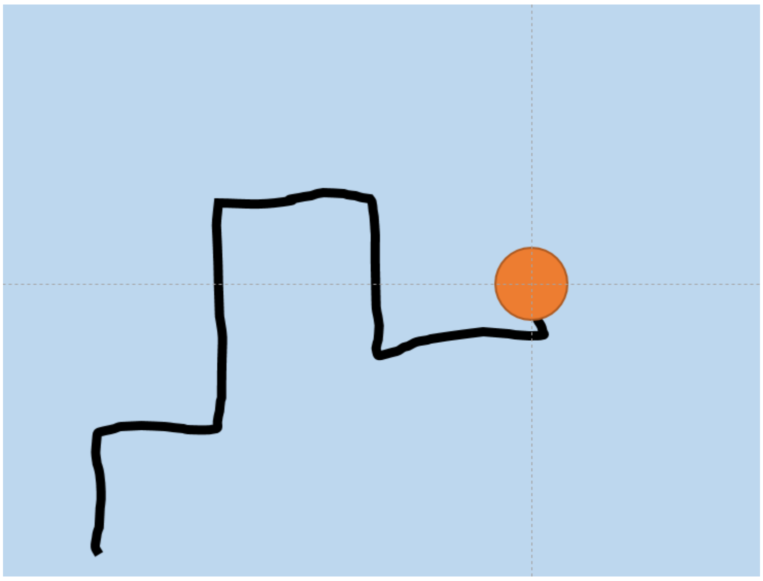
Currently the drone only knows its position in 3D relative to the current floor tile (Or another anchor point). The position is measured in centimeters. However, the drone does know when it steps from one tile to another, and it also knowns the direction of the step (left, right, up, down).

We used this knowledge to keep track of the current tile position, and once we know the current tile position in a global sense, which is achievable since we already know the coordinates of each point taken, and then we can find the global drone position.

2. Interactive Flight Map

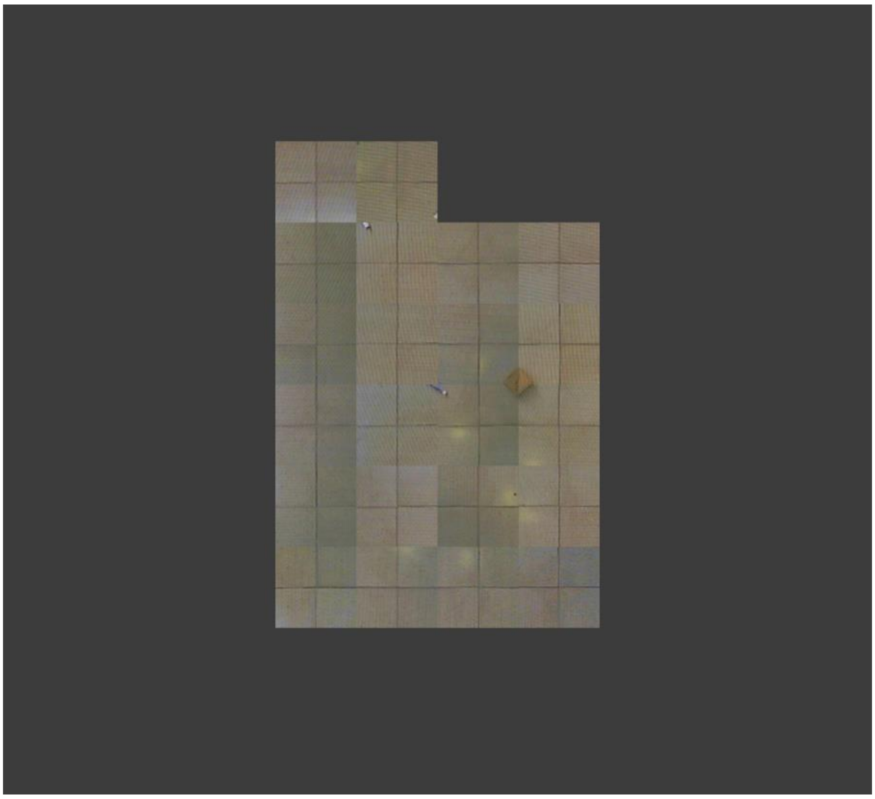
The Interactive flight map represents a map that will show the current position of the drone and the route taken by the drone prior to that specific point, it will be shown by a map that represents the drone's world, and a line that will show the traversal of the drone. The map will update on each frame.

This is done by receiving as parameters the dimensions of the world prior to the actual traversal of the drone (at the beginning of our program). This way we can build the map in a compatible way with the world. As for the line, since we can access the information of the current drone position, we can store in a real-time manner, every position the drone has been to. After storing this data in a data structure, the line will be a representation of each node in the data structure. What's already been traveled by the drone will always be stored in the data structure, and what's yet to come in terms of drone pathing, will be shown as a line once the information is received and will be stored too in the data structure.



3. Photographic Mapping

During the flight, the drone will take pictures of each tile, and use them to create a photographic map of the area. In each picture that it takes, it knows where the current tile is, so it also knows where to put the image in the map.



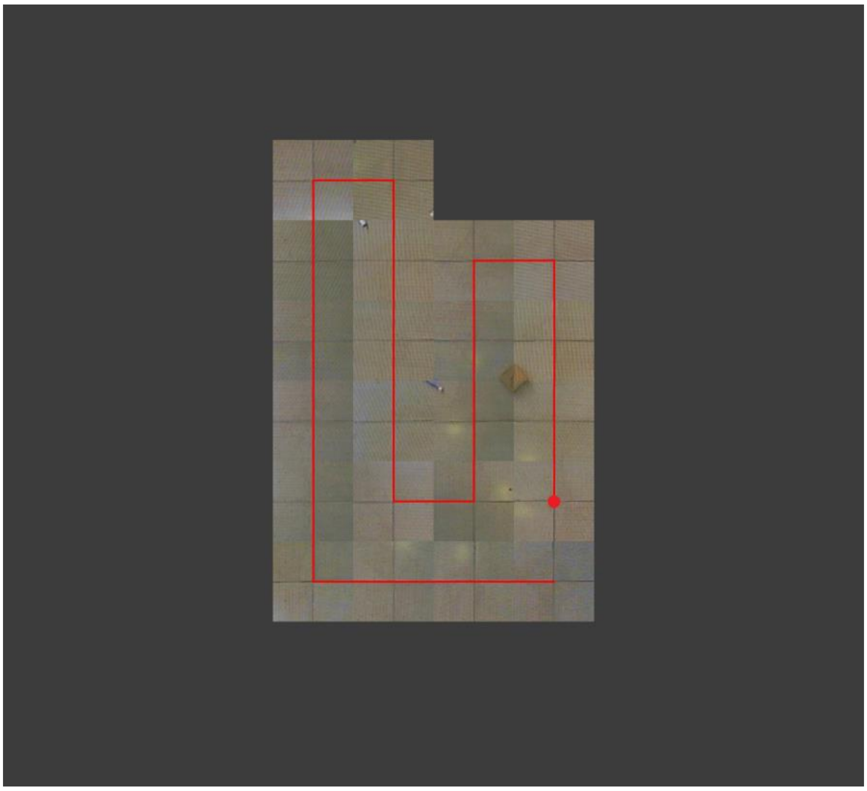
This illustration shows the map (the world), the grey color represents area in the map that the drone has not traveled to, and the pictures inside are the pictures taken by the drone camera while flying.

4. Live Photographic Map

When the implementation of the three previous steps is done, the live photographic map will be a combination of all of them. It will show:

- The position of the drone in real time
- A line that shows the historical path of the drone
- A photographic map of the area, updated in real time.

An Illustration:



The red dot signifies the point where the drone started flying.

Software Info:

Platform: Linux Fedora 24 (or higher)

Programming language: C++

Library used: OpenCV 3.1 (or higher)

Compilation Info:

Build:

```
$ cmake -G 'Unix Makefiles'
```

Compile:

```
$ make
```

Run:

```
$ ./mapper
```

Or, run with Sony camera:

```
$ ./mapper 1
```

Useful information for future relevant projects

Our project did not have a direct connection with drones, in fact, the project may be useful for virtually any moving object.

The reason behind that is that we are not manipulating the movement of the drone, but receiving information from it, and this information is used to accomplish our goals.

An addition to the project can be the stitching of the pictures taken by the drone into the map, there are various functions in the internet, few of them actually work. Reading the implementation of some is recommended.

For it to work, you need to always know which direction the drone is facing because the photo taken by the camera may be rotated relatively to the map, in addition, OpenCV has quite a bit of relevant code to use, but make sure you have an updated version of OpenCV since some of the stuff does not work on the old versions anymore.