

Charlie Weiss and Matthew Beaudouin-Lafon

We want to make a Skydio knock-off using Neatos (a Groundio, if you will). The goal is to identify a person and move the robot to keep that person in the frame, then follow that person at a reasonable distance. It loves you.

Inspiration:

- Does Skydio count?
- <https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/>
- <http://faculty.ucmerced.edu/mhyang/papers/cvpr09a.pdf>
 - Comparison algorithms? OAB1, MILTrack, FragTrack
 - FragTrack: <https://ieeexplore.ieee.org/document/1640835>
 -

Training ideas/Getting data:

- *End to End* (Image in, Velocity command Out) Piloting: Take camera data in conjunction with controller data. One person walks around, the other person pilots the Neato to face them.
 - Things to avoid
 - Accidentally training with non-movement (Could hold down a certain key on the keyboard while taking good data and filter out all the other stuff)
- *Object tracking* (Image in, Person velocity out): Train with image data that marks a person in a frame compared to the next frame and matches the change with their 2D velocity.
 - Tricky things
 - Obtaining person velocity to go with images (take accelerometer data from phone?)

Learning Algorithms

- End to End: Similar to the AC114, just throw the current and previous image at a neural net and spit out command velocity.
- Object Tracking - OpenCV API
 - BOOSTING, MIL, KCF, TLD, MEDIANFLOW, GOTURN, MOSSE, CSRT

Learning Orientation

- Top down! We both want something to show.

MVP:

- Detect a person! Turn toward that person! Even if they move! **WOW!**

Stretch goal:

- Combine our algorithm's output with teleop to have the robot move in the user's desired direction while keeping the user in frame. Note: this might not be mathematically possible due to the neato's forward and backward motion.
 - Ex: I see this person, but I'm in front of them, and I want to get to their left without losing track of them.
 - **Solution:** Servo mechanism on camera #putaservo on it