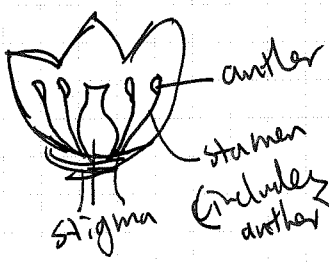
7/3/21

Background on Artificial Pollination



Pollination is transferring of pollen from anther to stigma

1. self pollination

desired genotype can be maintained, like in wheat, oats, etc.

2. cross pollination

increases genetic diversity to produce more resilient plants, however often requires more work to transfer pollen long distances

(natural) Pollination Agents

- wind - water
- insects ie

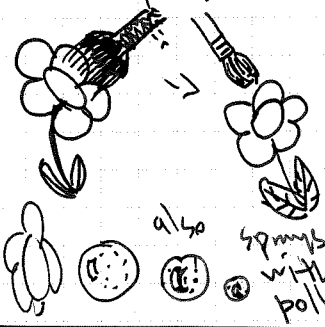
Insects like bees pollinate flowers when they collect nectar and pollen sticks to their body when is spread around.

- For many commercial crops, insect pollination is vital because it increases the quality and production greatly compared to self pollination or manual pollination.

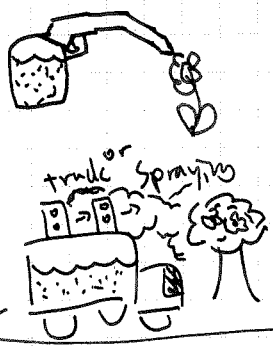
Artificial Pollination Methods:

- Hand Pollination

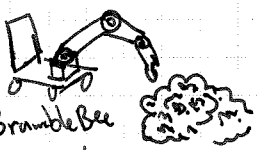
using a brush to manually distribute pollen



- spray pollination



- robotic pollination



uses a hairy attachment to pollinate like a bee

drone pollination
-sprays pollen

7/4/21

Ideas for autonomous artificial pollination

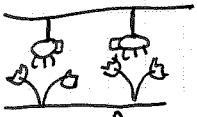
Beetle?

crawl up stem/branch onto flower

vision w/ camera to distinguish flowers

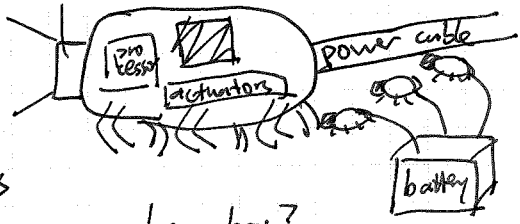
-hair on body like a bee, climbs into each flower?

how heavy would it be?
-what if plant is too weak, stem cannot hold the weight of the robot?



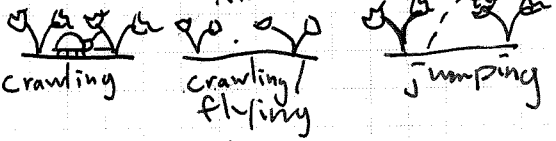
support from above?

Camera



how fly?
- microprocessor
- motors
- camera
- battery
  - or if connected to power source doesn't need onboard battery

plant to plant:



For a jumping insect-inspired robot,
1. jumps far but at the cost of control over where and how they land
2. impact could damage the robot (depends on mass)

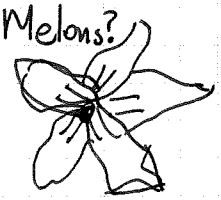
Maybe not feasible?
not that an insect robot isn't possible, just that robots at this size are not feasible with my resources/budget.

7/5/21

More ideas for autonomous pollination robots

Narrowing scope to plants that grow in a greenhouse, w/ male + female flowers (so the solution is not trivial)

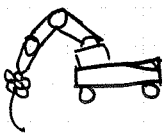
Melons?



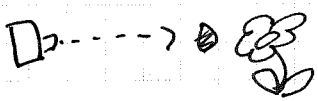
Female

Male

Wheeled platform with movable arm attachments



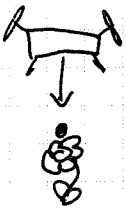
- sprayer for end-effector?
- launcher pollen packet



Maybe a folding structure that expands over the flower?

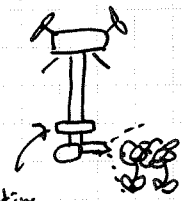
1

Dropped by drone?



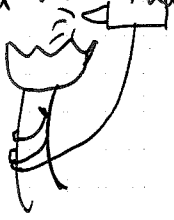
or

drone w/ payload



- drone does coarse positioning, payload does more precise

Also, for attaching to the plant (if that is even necessary) a wire that wraps around the stem to position the nozzle head



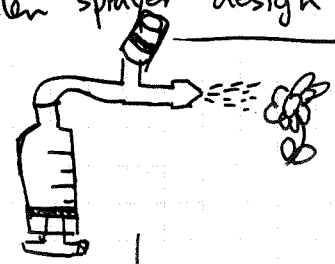
Pollination process broken up into 3 parts

1. pollen collection
2. pollen processing + transport
3. pollen delivery

3

7/10/21

Pollen sprayer design



solenoid pump

syringe as container, cheap and easy to use

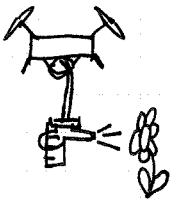
what about control system for sprayer head?

2 deg of freedom

Meeting w/ Prof. He:

Talked about details of drone + payload system

A drone with payload attached through a tether

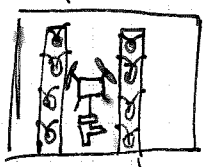


Because ground-based robots have trouble with flowers growing at different angles (not facing outward), a drone-based approach where the payload w/ sprayer is above the flower, and has no ~~limits~~ on constraints on position, might be more effective approach.

ground based robot

can't reach flowers on opp side

TOP view



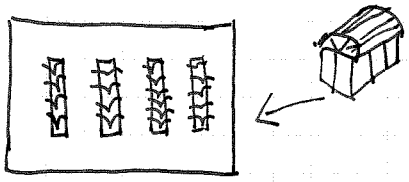
can reach flowers at any angle

7/10/21 cont.

Scenario and Concept of operations

Enclosed, rectangular greenhouse (melon plants?)

↑ plants have flowers that can be either male or female



0. (forgot) Mix pollen mixture, load into payload w/ syringe

1. The drone (without payload maybe?) moves around the area to do a survey of the location using a camera system
   a) get the location of flowers
   b) map out male/female flowers (is this possible?)
   c) plan path with SLAM etc.

2. The payload is attached to the drone using a tether

3. The drone moves along the planned path, hovers above each flower, descends to an appropriate height for the payload

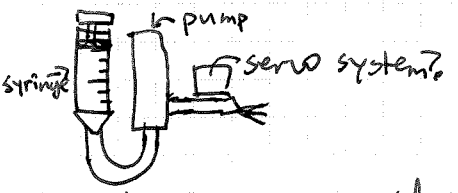
4. The payload-mounted sprayer system uses a small camera that is mounted above the nozzle to accurately position the nozzle relative to the flower.

5. After spraying, the drone moves upward to prevent collision then moves to the next planned flower.

7/16/21

Plan for sprayer:

buy nozzle, pump, nozzle, tubing



the syringe should be mounted above base plate, so ~~the~~ it doesn't block or destroy flowes.

Also, 3D design for sprayer system v0:

INSERT PHOTO

For the drone, moving it to next month, this month will be about the payload.

Also tried to install ROS on rasp pi 4 that I have, did not work. Would a fresh install work better?

I designed a connector from the flexible tubing to the nozzle screw on connection. I couldn't find the size or threading from the company's website, so I did some research and found that the pipe was likely using an NPT threading, so I went on McMasterCarr's website and downloaded the right .SAT files and put it together. (Fusion doesn't support NPT threading)

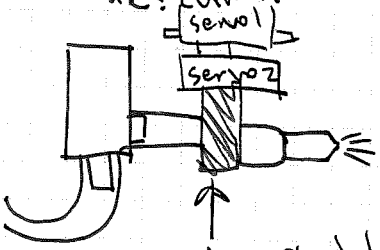
← holds flexible tubing in place with barbs

7/21/22

Looked at specifics of SLAM w/ ROS on rasp pi.

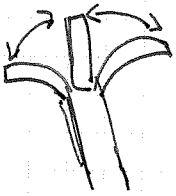
I have ROS Noetic installed on rpi, but one package for camera control I wanted to use is only compatible w/ Ubuntu 16.04 & ROS Kinetic. I flashed an SD card with a pre-built image from Ubiquity robotics. Set up VNC to use with my PC. Unfortunately camera display still not working. There's data transfered but the ROS image display isn't working. Decided to go with Orb-slam2 as a first test, once I get the camera working. It seems common for this kind of application.

Hardware: current design



still need to work on servo movement

tube flexibility is a constraint. The angle at which it buckles is the limit of range of motion. Although the servo is 120° so it shouldn't be too big of a problem.



7/26/21

Meeting w/ Prof. He:

Reevaluation of scope and research question of project. For more meaningful research, using more novel/newer ideas like deep learning for robotic control. Rescoping project.

1. Deep Learning / image processing hybrid approach

With DL, the robotic system can learn and adapt to the given environment with greater efficiency and possibly less effort than a hand-engineered approach. However, basic tasks like flower labeling might be more resource efficient with traditional image processing methods.

2. Low-cost self-developed end-effector

3. Integration of end-effector and payload w/ drone-based system

4. Full integration and testing in real-life greenhouse environment.

I also worked on the CAD sprayer design this week.

TODO:

- [x] timeline to OCT 15
- [x] CAD model finished
- [x] Ros camera access

7/27/21

I decided to use ROS Noetic w/ Raspbian, so I wiped the 128 sd card and put it back in my camera.

Ros notes:

On Raspi setup a .bashrc file to activate ROS in each new terminal:

[source /opt/ros/noetic/setup.bash]

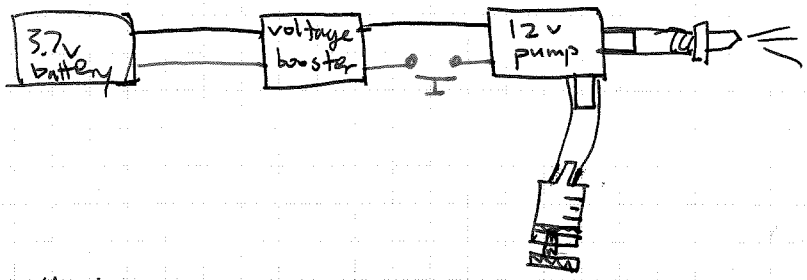
I had an error with the Ubiquity Robotics raspicam_node package, but when I tried to uninstall it I think I broke the entire ROS installation, so I need to fix everything. If it doesn't work out, could build a module manually?

Note: Robostack, the conda compatible ros installation, can be started by activating the correct env, "robostack".

7/30/21

Constructed proof-of concept for sprayer system

Sprayer system v0?



I soldered the wires together, used a 2000 mAh battery, but one thing is that the pump is operating opp. direction to my design, will have to change the wires.

With the pushbutton, it's clear that the velocity that the water is spraying out at is a little high. (velocity? pressure?) But the water is spraying out too fast. It needs finer control, which is accomplished with the Resp Pi, but also a motor control board

TODO:

- [x] add motor control board to BOM
- [x] add servos to design

8/2/21

Continued to try to get ROS camera access working. Still having issues on the raspbian noetic installation. After looking for a solution for several days, I think that based on the problems that other users had, it's not feasible to use ROS Noetic + Raspbian with the packages I want to use.

I installed Noetic from source, which means I can't apt-get install other packages, I have to also compile them from source. So, I will have to use Ubuntu, either 16 or 20.

I installed Ubuntu Server 20.04 on an SD card and set up wifi from the terminal. Then I installed a graphical interface (GNOME first, but decided Lubuntu would be better w/ the limited resources of the RPI. I set up VNC server and was able to connect with my PC.

8/4/21

Fixed error with VNC setup on the pi where it didn't autologin, so it couldn't start RealVNC until I manually entered the login info. Since I'm using Lubuntu, I thought it would use SDDM (SDMM?), but it actually uses lightdm. I set everything up, so now I can log into VNC and it will show the desktop without needing to touch the pi. I did have to use the dummy desktop solution, which I think is a little sketchy but it seems to work.

After a lot of work, raspberry pi image capture and video capture w/ command line is working.

raspistill -o image.jpg

raspivid -o vid.h264

While trying to install the Ubiquity Robotics raspicam_node, could not install from the binary so the only option is to install from source. There's a lot of problems, and after compiling the code it still doesn't work correctly. It seems like it never installs the package, so ROS can't find it. Trying a different solution that someone posted on a forum, but it's taking a long time, so I'll continue it.

- [x] ~~Uninstall ROS package raspicam_node~~
different solution used

Now I opened VNC and the pi has a broken ~~login~~ boot. I can't access the graphical interface at all. Not sure why it happened, but lightdm must have broke somehow.

Notes: terminal ~~at boot~~: Ctrl + ALT + F1 echo $XDG_CURRENT_DESKTOP
gui ~~at boot~~: Ctrl + Alt + F7

I reinstalled lightdm and switched to gdm3, and it's finally able to connect to VNC (although I can't access it directly now) 12/ Not sure why? Anyways, I'm now using LXQT?

8/5/21

I tried to run the compilation and install, and again it crashed and I'm unable to access pi through VNC. Not only this, I also can't type anything in the pi, not sure why. I think I may have to reimage the card if this continues.

I emailed Prof. He about my problem, and he said to reimage the card, due to corruption most likely.

Setting up ubuntu again, this time I'll keep the steps in order.

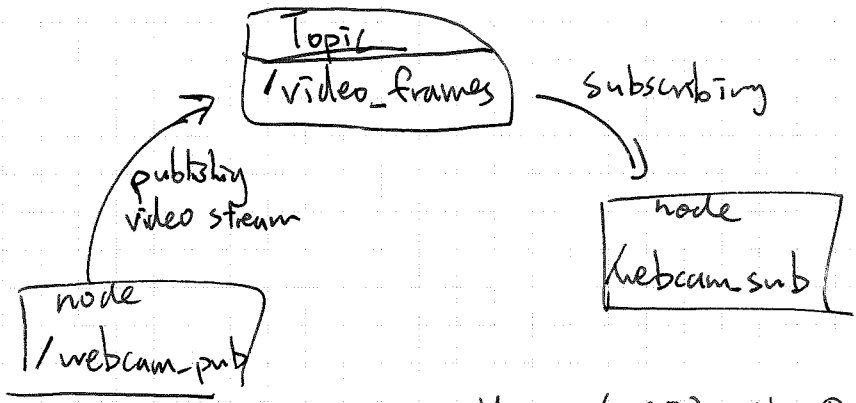
8/6/21

For my new ubuntu installation, I'm going to use the default Ubuntu Gnome gui, but I'm using the minimal desktop installation so I don't have all the bloatware

This time, I recorded all the steps I took in a google doc. I also made backup images just in case.

I did get ~~the~~ a ros camera node working, this one is based on opencv. I don't have everything setup, but it's already able to integrate w/ ros. And best of all, it's using the usb-cam directly without needing the rpi menu and all the pulse packages.

Ros structure



Learned about ROS package building w/ CPP with Prof Ile.

For Wednesday:

finish up ROS example

going to learn 3D modeling in ROS + Gazebo
robot

set up a Ubuntu 20.04 virtualbox VM w/ 8 gb ram to run Gazebo on a native linux system

8/11/21

Continuing with virtualbox machine. One thing I can't figure out is Virtualbox's clipboard sharing. It's not completely necessary, just annoying.

Update: I got vb working correctly, I just didn't load the guest additions disc onto my VM!

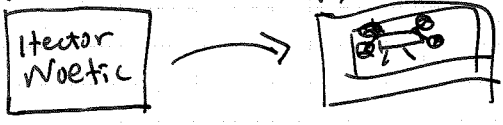
8/13/21

Met with Prof. He

On Ubuntu VM, installed Hector-Quadrotor package with a tutorial for ROS Noetic version, since Hector only has a Kinetic version.

I set everything up and was able to run the demo, and use teleop controls.

The construction



ROS app, took only the simulation_ws catkin workspace

One thing is that even with the GPU, the VM runs slowly. I looked online and apparently VMs have trouble with 3D acceleration because it's not direct. Still should be okc.

Next steps: ~~SLAM~~ with Hector Quadcopter
Finish Payload Model

8/14/21

Meeting w/ Prof Ite:

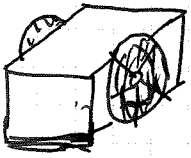
Intro to RViz and Robot Simulation

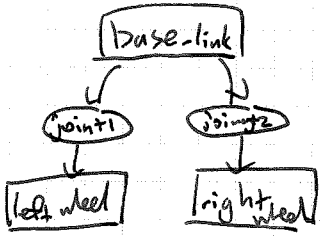
URDF (Universal Robot Description Format)

- building example of 2-wheel differential drive robot

parts - chassis/body
- wheels

links attached by joints





Problem w/ rviz launch, joint state publisher not working. Not sure why!

8/18/21

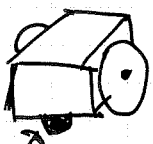
Meeting w/ Prof Ite!

Discussed updated research plan:

ROS-based Drone Simulation
↓
SLAM methods
↓
Deep Learning

Working title: Design & simulation of DL powered SLAM Algorithms for Auto.. Drones in Precision Artificial Pollination

Worked more on robot simulation, Prof Ite found a way to get it working, added kinematics + collision model



added a caster wheel ("wheel") it's part of the main box

To specify kinematics, using 3D inertia tensors for each part (but I don't calculate it)

urdf → sdf

Then, visualization & simulation in gazebo

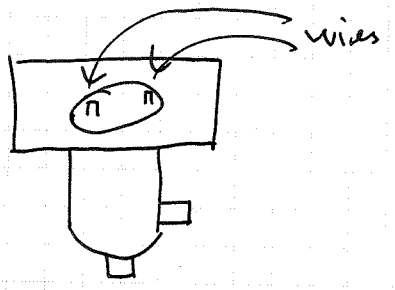
able to see everything, didn't get to moving robot in gazebo

For next time,
- gazebo sim. movement & adding motors to ddrobot

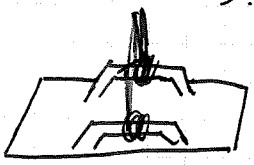
☒ Read over research papers
☒ ~~make plan in Asana~~
not in Asana, google sheets

8/19/21

Worked on CAD model, ran into a design decision, where and how to attach the tether?



with handles?



easiest, going to use this for now

3D printing basepiece tomorrow

8/22/21

Meeting w/ Prof He,

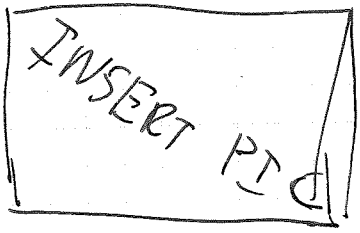
went over schedule to finalize it, project plan until Nov.

talked about payload model completion, I'm almost done w/ base design

Introduction to SLAM theory, I'm reading research papers on visual slam mostly

Notes on Sprayer System:

printed base out, some notes:

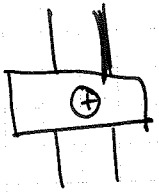


1. handles too thin, snapped off

2. servo fits, need to add screw hole

3. holes for syringe / motor need to be bigger

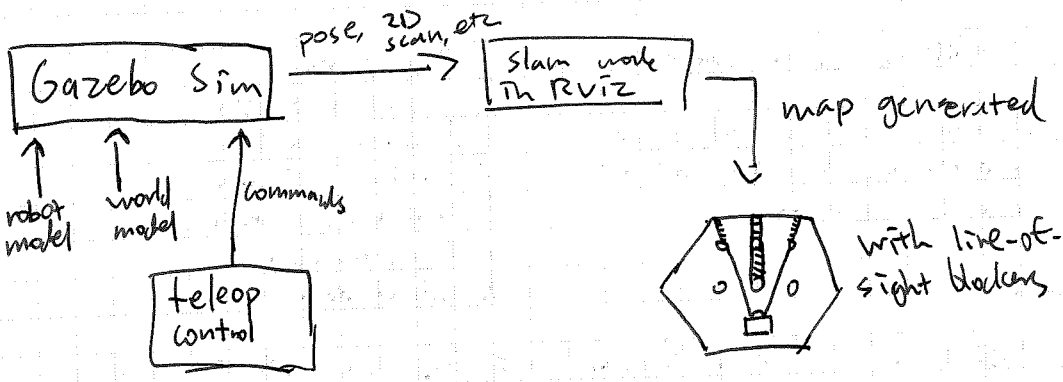
4. think of way to hold syringe/pumps in place, maybe using screws?



or

motor mount

Figured out SLAM w/ turtlebot3 in Gazebo



Next step is to implement this with my robot.

9/4/21

Notes on VSLAM/MonoSLAM

traditional Vslam uses relies on

1. Initialization
2. Feature Detection
3. Feature Matching
4. Outlier Rejection
5. Motion Estimation
6. Optimization
7. relocalization

A little confused still about feature-based SLAM, how are features defined?

using DNN (Deep Neural Network) to estimate camera motion

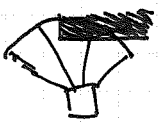
LIFT used to extract features from images and use them in trad. VSLAM pipeline w/ ORB-SLAM

MonoSLAM - extended Kalman Filter (EKF) estimates camera motion and 3D structure of env

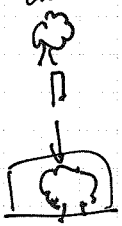
ORB-SLAM - 3 threads: Tracking, Local Mapping, Loop Closure

DeepVo - RNN estimates camera pose from CNN feature extractor

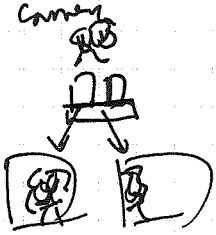
laser range finder



monocular camera



Stereo camera



How does Graph-based SLAM work?

also, particle filter?

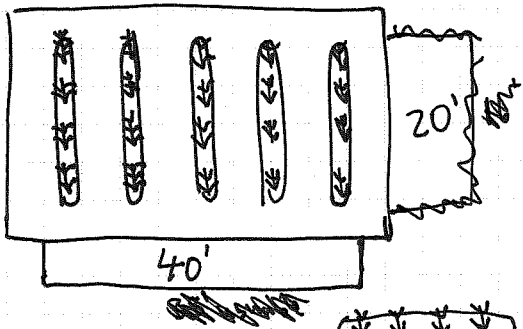
for camera-based SLAM, a sparse, landmark-based approach is used

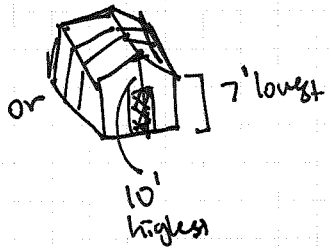
Single camera techniques:

1. use sequence of images to estimate camera movement and 3D position
   - structure from motion
   - offline (due to needing
2. Incrementally incorporate images
   - can be online

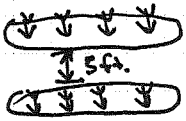
Meeting Notes:

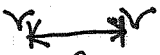
Parameters in plants/environment that could affect slam





5 ft between rows





8 ft. tall

To create the gazebo world, maybe use blender to create stl models? Maybe using a script w/ L-systems to generate individual plants?

TODO:
- [x] create gazebo world
- [x] document and explore tomato growing patterns
- [ ] print 3D models

don't even know what I mean?

9/10/21

This week I met w/ Prof. Dodds, HMC CS independent study, so I'll work on this project in an official class as well.

Prof. Dodds 2-week recommendations:

- [x] ORB-SLAM2 working in sim
- [ ] use photos of flowers instead of models
  - in process, (10/23) change of plans

Also met w/ Prof. He,

- [x] Hector_Quadrotor - upload my setup to GitHub.
- [ ] Having trouble w/ world model gazebo

Also during meeting, got it working by converting the model into .dae file format using ubuntu cmd `ctmconv`
One problem - the scale is totally wrong, I'm planning on using a 3D editor to correct it, like Blender.

- [x] correct scale of greenhouse model

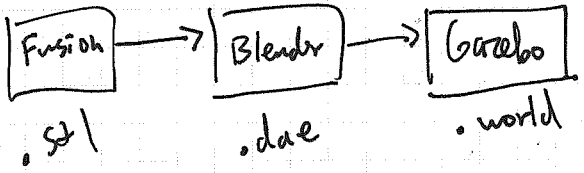
Working on ORB-SLAM2 ros integration today, still working on it.

9/12/21

Meeting w/ Prof. Ite,

got orb-slam2 working with a turtlebot-3 sim, can take image stream though image-view and orb-slam node running. One problem - the initialization requires 300 points to get started, so it was constantly showing message "Map point vector is empty!" Would it work better with

Also have my greenhouse model converted into DAE format



Could test orb-slam2 w/ real camera since I have one. Very easy to connect camera to virtualbox.

9/17/21

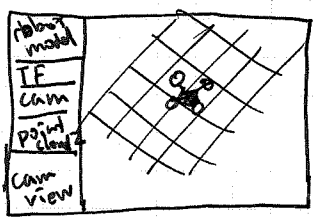
Working on Orb-slam2, trying the webcam approach.

connected camera with Vbox guest addons, then

lsusb — shows all devices connected through usb

cheese doesn't work, hope that's ok!

For Hector Quadrotor, connected it to Rviz



camera works! (although the TF links aren't doing so good, it only works when it's frame is the camera.)

Orb-slam node is actually connected and receiving data. In order to do this, I went into the ros-orb-slam2 code for slam-mono and changed

In MonoNode.cc the ~~node~~ topic it subscribes to to

↳ /front_cam/camera/image

from

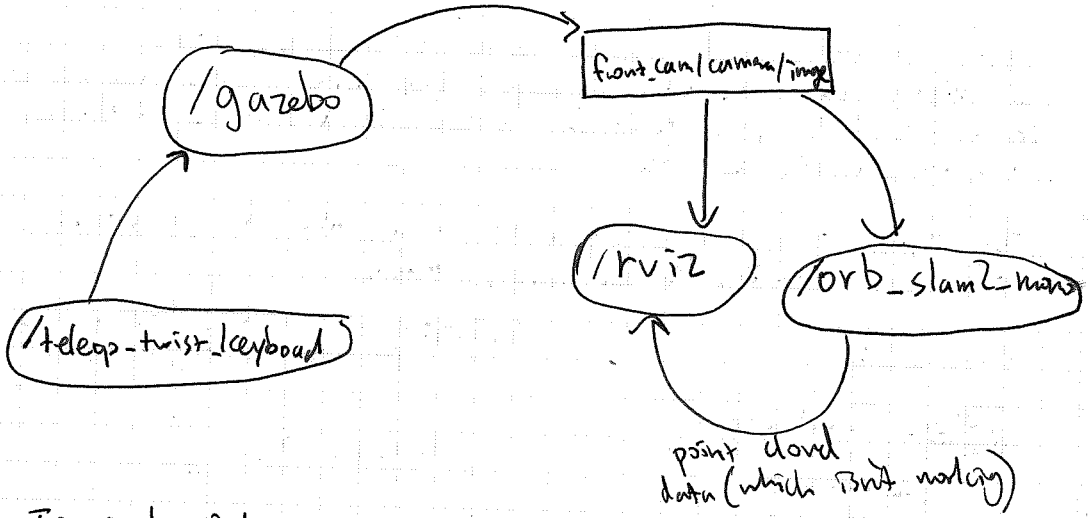
/camera/image_raw

Orb-slam complains that map point vector is empty, I think that it might actually be a config issue. I want to verify this by

- ☐ connect to real usb-cam
- ☐ use custom camera config

I also started the gazebo world with the greenhouse model. But it's too big now! I think it's a scaling issue.

Hector Quadrotor ROS diagram!



Terminal windows

Hector Quadrotor Sim	Twist Controller	rviz	orb slam 2	rqt graph

— just for debugging

9/22/21

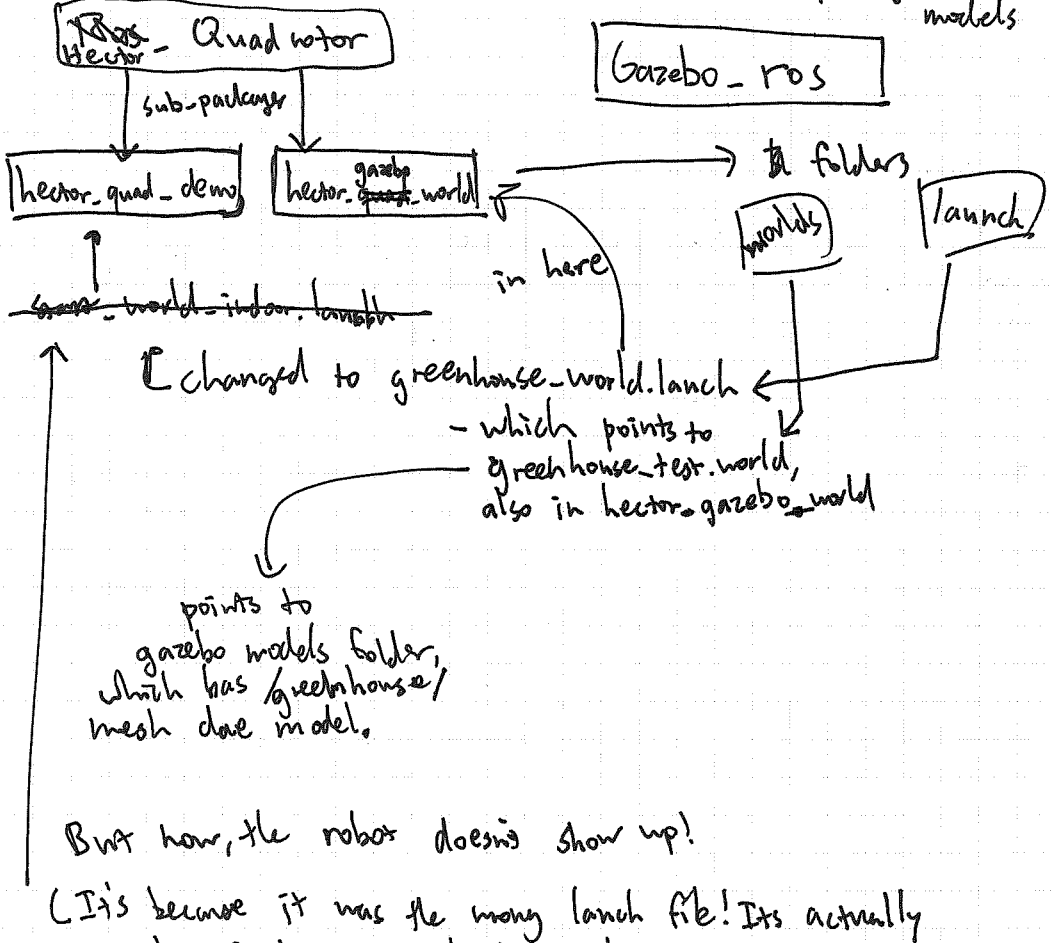
Meeting w/ Prof He:

Before meeting, fixed scaling issue w/ greenhouse model
I figured out that in Blender, it was using meters and scaled it 100? 1000? percent. So I changed it and it was the right size in Gazebo as well.

Also trying to get hector_quadrotor working with my greenhouse world. Not working, its an empty world.

Packages:

Update: now working! Problem was it wasn't pointing to Gazebo models



But how, the robot doesn't show up!

(It's because it was the wrong launch file! Its actually greenhouse_~~test~~gazebo.launch
a clone of indoor-slam-gazebo.launch

Running w/ teleop node, logitech controller

it's not working rn, not sure why

- [x] get logitech teleop working
- [x] Prof Dodds 2-week suggestions
- [x] fix the model's starting location

9/23/21

Testing controller:

Installed jstest-gtk and was able to test controller (its fine)

looked into hector_quadrotor_teleop package where
logitech_gamepad.launch is located

because jstest-gtk confirmed that the controller was on /dev/input/js2, changed it to that value instead of js0.

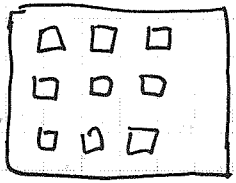
9/24/21

Meeting with Prof. He

Deadlines - 1st draft outline of paper by ~~10/1/21~~ 10/12/21

ML for precision pollination

Also created a lower poly version of the greenhouse,



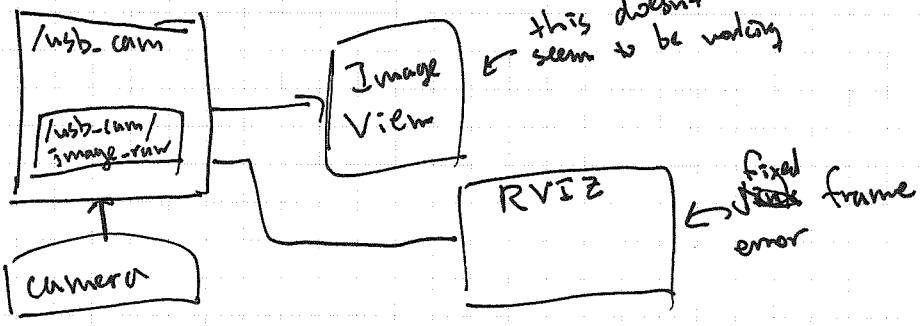
instead of cylinders I'm using squares, hopefully it reduces complexity

But, during testing, there was the same issue. But why does it still move so slowly? The demo outdoor world moves so much faster, so I looked at the code and found that there were a lot of changes to the physics simulation details, like a faster physics solver.
Changed rolling_landscape_120m.world in hector_gazebo_world to be my greenhouse model world, and it works super well. It's fast, I can see the laser slam working, and I can fly the drone.

saved as a snapshot

- [x] ~~Gazebo~~ w/ more optimization ←
- [x] get orb-slam working w/ webcam

right now, still seeing green screen!



9/27/21

Worked on setting up 3D acceleration in virtualbox, which really speeds things up a lot, like opening and moving around a gazebo world runs at 60fps+. However, when I open Gazebo through ROS with roslaunch, I get a black screen where the viewport which leads me to believe that it's some sort of graphics issue but not a simulation physics processing issue.

Wondering if it would be easier to just dual boot and get it working that way? After seeing how fast it ran, how could I go back to before?

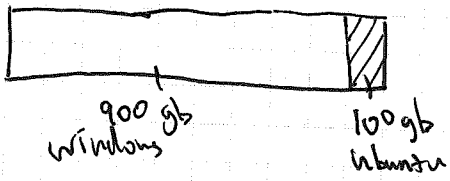
Also asked Mr. Hamilton and Ms. DeRemer if to write the recommendation forms, although I'm a little worried. They don't have any connection to my project, so they don't know if it's good or not. Will that affect anything?

10/1/21

Set up dual booting with Windows 10 and Ubuntu 20.04.

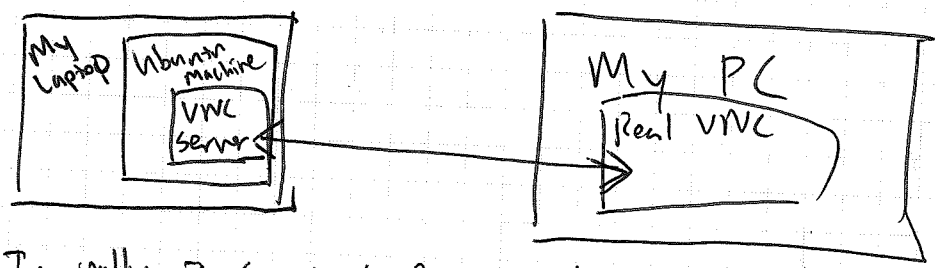
Steps:

- with a usb, burn a ubuntu ISO image onto the drive.
- insert usb into my laptop and boot into BIOS
- boot into ubuntu and select install with windows
  - user : rachel
    pass : pass



Dual booting wasn't too hard, but I was unable to use the display.ing I generated from the VM and moving it onto the ubuntu machine. Maybe I'll just re-install some things?

Started a log for the dual-boot ubuntu installation, installed everything, my setup now is:



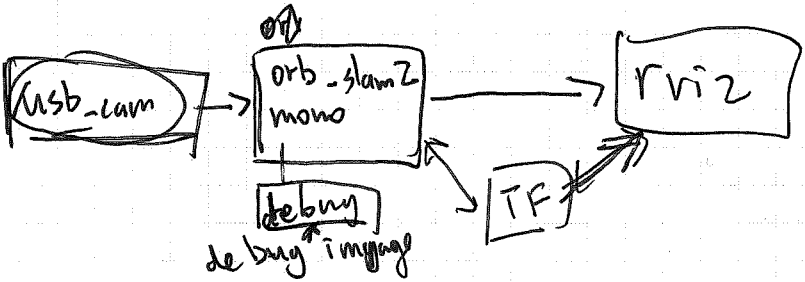
It really is so much faster with native ubuntu.

Calibrated usb-cam with a paper checkerboard pattern that I printed out, it was very successful and worked.

But, after a lot of work (refer to log), it's still not working properly with ORB-SLAM! I'll look into it more since it's already late and the ubuntu installation is heavy trouble w/ VNC.

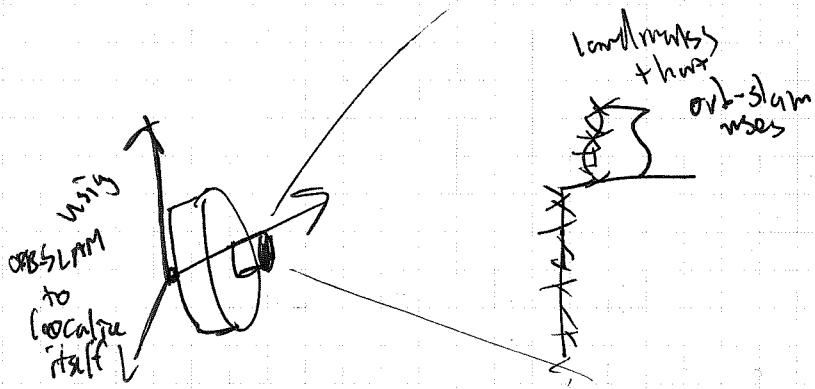
10/3/21

Got ORB-SLAM working! With real-life camera and viz details in log, but here's a rosgraph:



Also worked on trying to save a map of the trajectory, did not work!

- ☐ view and load map for orb-slam2
- ☑ get ORB-SLAM2 working with virtual camera in Gazebo



1 thing I noticed: the point cloud generated is incredibly inaccurate. Is that just par for the course with mono slam and cheap webcams?

10/6/21

Working on getting orb-slam2 working with gazebo:

1st Part: getting camera connected

/front_cam/camera/camera_info

Made change in orb-slam code launch files and got it working with the orb-slam setup I had last time, but it breaks when integrated with the hector outdoor terrain demo rviz. Why?

Also, it only really worked on high-~~poly~~ complexity terrain/environments otherwise it can't initialize which means I need to update my ~~world~~ greenhouse world model!

turns out I have this problem just in general ...

Plan for next meeting/this week: have

- [x] update greenhouse model to ~~be~~ more complex plants
- [ ] figure out gazebo + rviz problem with orb slam
- [x] ~~start~~ filling in forms for STS application
- [x] ask school for officially sponsoring me for STS
- [x] get navigation working, probably with Hector-SLAM stack

10/9/21

Unique contribution
- agricultural robot
- simulation-based greenhouse model with ROS + Gazebo
- deep learning for sub-centimeter precision pollination

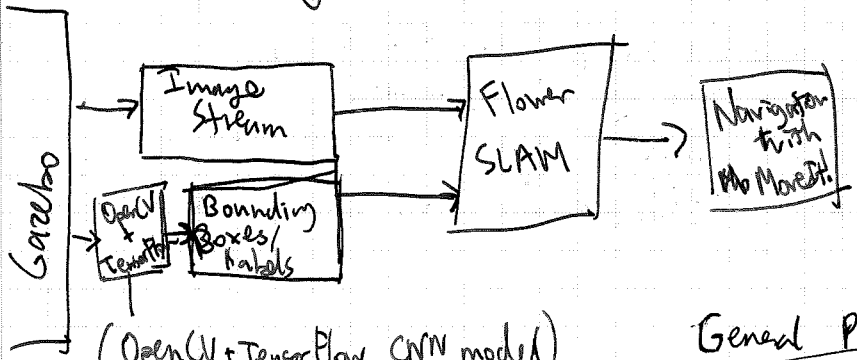
For the greenhouse world, my plan is to use a tomato flower dataset and place them onto the plants.

Could use DL w/ OpenCV + tensorflow to detect flowers within each photo stream

found a dataset with images of tomato plants in all stages of growth from flower to fruit. Could better replicate real-life greenhouses, which have multiple stages of tomato growth.

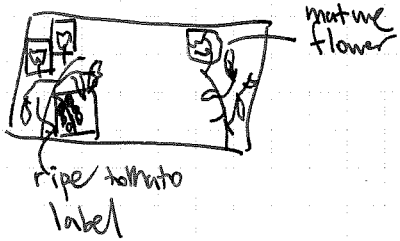
Also added collision to greenhouse world (finally)

Idea for adding on to SLAM: multimodal



(OpenCV + TensorFlow CNN model)

ex)



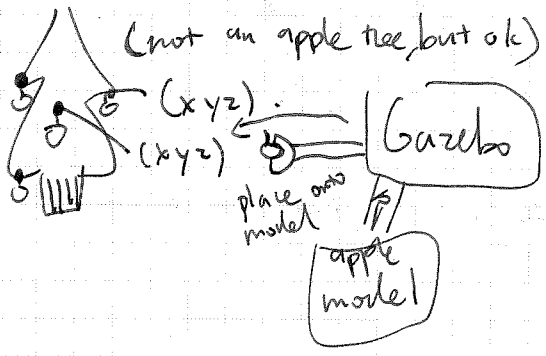
General Plan (dataset)

1. Integrate flowers w/ greenhouse model
2. DL CNN model with classify flowers
3. Integration of CNN model w/ SLAM
4. Navigation w/ MoveIt!
5. Demo!

creating a new SLAM algorithm (likely based on ORB-SLAM)

I also saw a GitHub repo with Hector integration with MoveIt. It's using Ros Melodic and Gazebo 9, so it is old and likely parts of it don't work, but I think it's a good place to start.

The more interesting thing is that in the process of their project, they ~~picked~~ programmatically generated apples on a given tree model by picking vertices from the tree randomly and then finding the location of the point, then using a gazebo plugin to spawn an apple there. That's a possible approach that I could use too!



TODO for next meeting

- [ ] try and get at least a few pictures of flowers onto greenhouse plants
- [x] get a basic demo, doesn't even have to be hector working with ROS Navigation stack
- [x] write a framework and short abstract for research paper

10/12/21

From flower dataset, found 18 flower images, and I'm thinking I want to ~~classify flower~~ train a ML model to classify flower/no flower images from dataset.

But, could I use a clustering algorithm instead? Might be good.

DeepCluster vs. SCAN

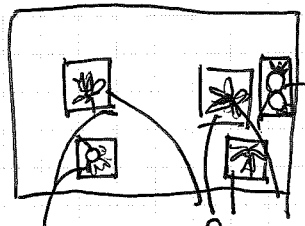
Tomatoes in different stages of growth

1. 2. 3. 4. 5.

mature flower is yellow

mature tomato is red

But there's multiple flowers/fruits in each picture. How to separate them? How to classify them?



what about clusters?

an unripe fruit

flowers ready for pollination

Also started going over research report, abstract and structure of report

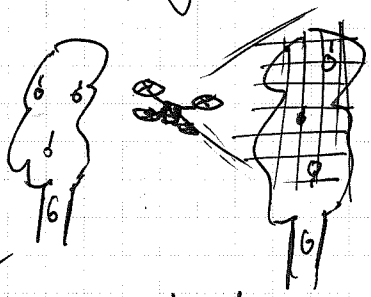
10/17/21

Meeting with Prof. Ite

Going over a mostly-finished draft of research report, changed structure a little, and fleshed out abstract more.

☒ finish a 1st draft by Wednesday

Also, got started on getting a MoveIt! w/ Hector Quadrotor demo I found on Github working. So far, the package builds without errors (despite being for ROS Melodic & Gazebo 9), but when the demo is launched returns some errors. However, it still runs just not the trajectory part. My best guess is MoveIt isn't working so well w/ the versions of software that I have, since if mapping w/ SLAM is accomplished then pretty much everything else is working.



pretty sure it uses LIDAR, scans into a voxel grid, but could work with dmy good maps, right?

☑ get MoveIt! w/ Hector quadrotor working

10/22/21

Plan for next 2 ½ weeks:

1. MoveIt! w/ Hector Quadrotor Navigation

2. View/Load map generated by ORB-SLAM2
   - feed into navigation stack

3. Flower Detection w/ DL-based object recognition
   - Tensorflow Obj detection models (Resnet, Inception, Mobilenet)

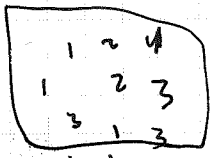
4. Update Greenhouse model w/ flower images
   - either manually make models and spawn them in randomly

ex)

model 1

model 2

model 3



orchard w/ random trees

- or, use programatically generated tomato plants like in the apple orchard Github repo.

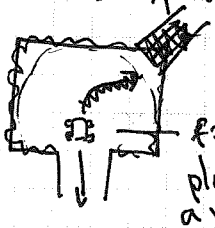
10/23/21

Finish 1st draft of research report, working on revising and adding more parts.

MoveIt! Demo - I wasn't very smart last time, forgot to run the command to start MoveIt.

It works! But, when I made it work in the Greenhouse world environment instead of the orchard, it pauses because of an issue with finding frontiers.

Frontier_Exploration in ROS:



find the "frontiers" places the robot has not found a wall, then explore the empty spaces.

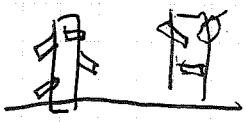
The problem with the Hector-MoveIt demo is that it just stops when no frontier is located. Why not keep exploring? I think I need to find a different navigation algorithm to incorporate into the demo.

- ☐ fix frontier issue (maybe)
- ☐ get MoveIt integrated w/ ORB-SLAM2 (hard)

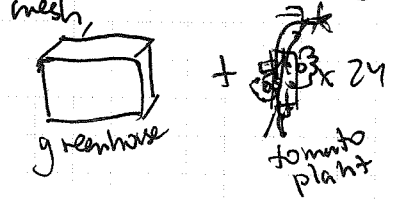
Also, the Github repo for Hector-MoveIt was super helpful. It ~~also~~ ~~helps to include~~ works out of the box, but also includes a cool script that generates apples on tree ~~pos~~ randomly (method already described before). But, this I can incorporate into the design of the greenhouse plants

☐ Change gazebo world to use individual plant models

meaning instead of 1 greenhouse



☐ run the random vertices program w/ my plant models.



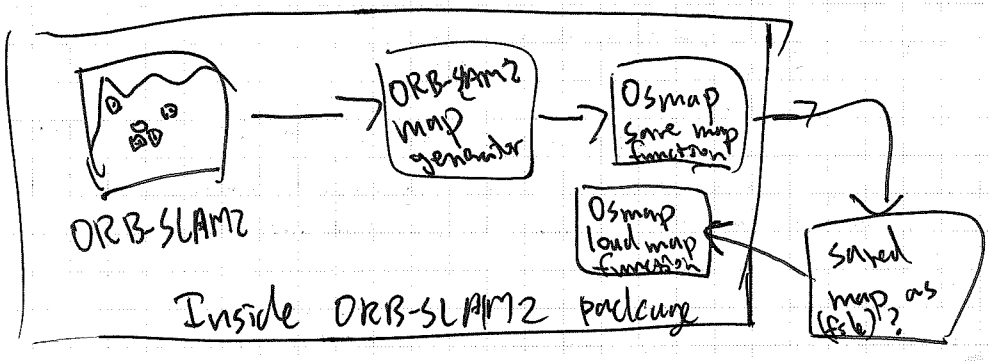
Also started the process of setting up OsMap, a serialization add-on to ORB-SLAM2.

ORB-SLAM2 doesn't come w/ a map saver or loader. It generates a map, but I can't see it or save it. This is where Osmap comes in.

Osmap uses a pretty complicated serialization protocol to save the maps and then load them again

I followed their tutorial, but it's a little vague. "Write the code to call save and load" — where? I think I'll work more on this next meeting, since I don't even know what's going on w/ this.

D Osmap serialization of ORB-SLAM2 generated map.



10/30/21

With a draft already converted into LaTeX, covered which parts I need to work more on.

Related Work:
- background of bee pollination
- Deep Learning for robotics
- simulation systems

- saw a WPI student report on Autonomous Drone Pollination, pretty similar to the problem I'm tackling, but the differences are

1. Simulation vs. Real-Life
2. Tomato vs. sunflower
3. Deep Learning vs. Trad. color-based flower detector
4. SLAM localization vs. GPS
5. sprayer vs. ~~[illegible]~~ makeup brush

Also, it seems that their project, while performing well in flower detection, lacked the precision ~~[illegible]~~ for their end-effector to be able to pollinate any of their flowers.

However, their report is super helpful in providing an example of how to write background/related work sections.

Concept of Operations
- add diagram
- move to inside approach

Software
- add more about details, diagrams, etc. (like Hector)

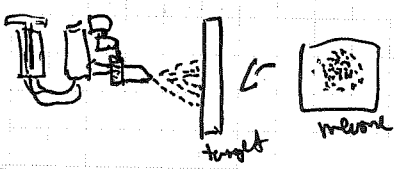
~~Body~~
Background
System design
-models ← actually parts of my actual results

Simulation
- simulation
- flower detection
- navigation

add Problem statement after related work

Validation/Testing
- successful location of flows
- successful pollination
- speed of pollination?

① testing of real-life sprayer system prototype



Outline:

Intro

Related Work
- divided into sections
1. artificial poll.
2. V-SLAM
3. End Effector
4. Robotic Poll.
5. Deep-learning obj-detect (Deep-Envir)

~~Background~~ ~~Role~~ ← ~~add~~ Major Contributions

Problem Statement
- what is the problem? Why am I studying? Objectives

Background - set of definitions
1. Pollination
2. Autonomy
3. Object-Detection
4. V-SLAM
5. Validation Metrics
6. Assumptions

~~Approach~~ Technical Approach

1. Overview
- what did I do w/ the problem I defined?
- "autonomous drone powered by ....."
- simulation-based approach

2. Design of Drone
~~Drone autonomy~~
- hw/sw config

3. Autonomy (of drone)
- Localization and Mapping
- V SLAM
- TF OBJ Detection
- Navigation

4. End-Effector
- diagrams.

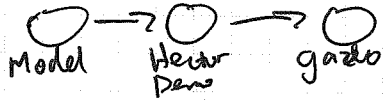
5. Environment
- design of greenhouse - real greenhouse
- simulated design of greenhouse
- includes model

Concept of Operations?

proposed pollen collection system
↓
an assumption
↓
can be adapted to other pollinating scenarios

~~5. Simulation~~ above Implementation

6. Implementation
   - design of systems to make everything fits together



7. Simulation
   - experimental setup
   - demo???
   - some source codes

Flower Detection
- V-slam w/ DL algorithm

Results

- hit rate ← may need to define metrics in earlier section?
- complete system
- Demo.
- end to end scenario simulation

Conclusions

Future Work

Acknowledgement

References

11/2/21

Include more detail in research report, add more diagrams

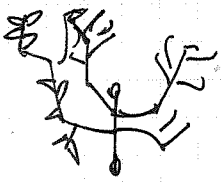
1. Use MoveIt! to move through waypoints in greenhouse
2. Use SLAM to generate map through multiple passes of drive through greenhouse
3. Convert located flowers to MoveIt
4. Move to each waypoint and "pollinate" flower

and TF Obj Detection

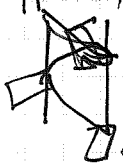
Designing a tomato plant

Technically, tomato plants don't have one central stalk

sympodial



Typically

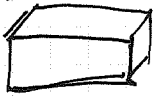


spiral leaf arrangement

137.5 apart

SO ignoring some

I'm using 3 files to generate the greenhouse:

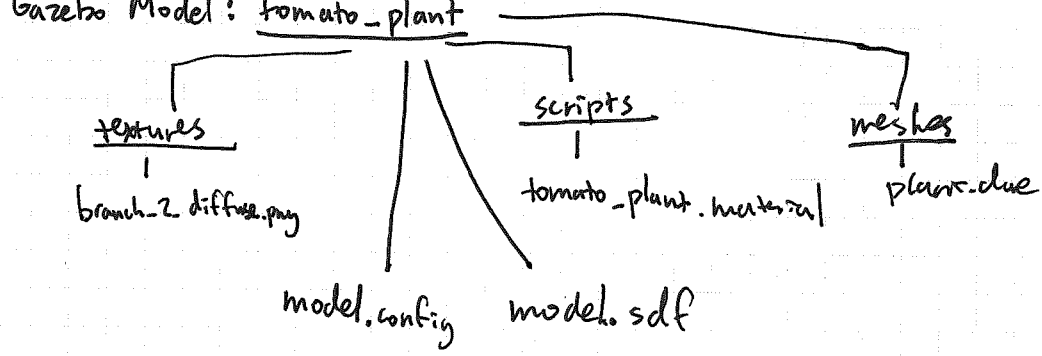
1. (empty) greenhouse model 
2. Main stem || —— 1.5 in diameter, 6ft tall
3. branching stem(s) ▯ ◇ —— 0.7 in diameter, 1-1.5ft long

First, randomly generate branching stems on main stem, then randomly generate flowers on the branches

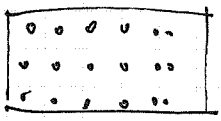
But for tomorrow, I'm going to just ignore the branching part and use the premade model I made.

11/3/21

To make the greenhouse thing work, created a new Gazebo model based on the pine_tree example model



Place a population of tomato plants into Gazebo world using Gazebo's built-in <population> tag. However, one issue is that I cannot randomly rotate the models, they are all uniform.



So, I might need to use a Gazebo plugin to use Python as a RNG / positioning tool. And what about the cpp plugin I wanted to use? I still haven't figured it out.

But, with my current model of the greenhouse, I have been able to get the flower texture onto an image of a plant. Exciting!

flower texture.jpg

Also worked on getting ORB-SLAM2 integrated. Right now, I don't want to use the Kinect camera, so I have to use a different model, which requires some modification of the code and drone model

11/5/21

Got ORB-SLAM working with Move-it
had to change some configuration files

Using the Kinetic Camera
RGB and Depth sensor

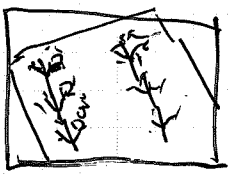
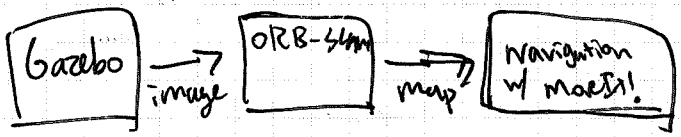


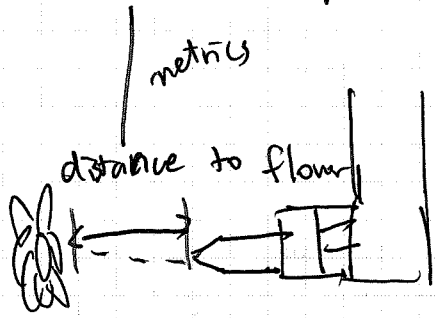
Image view
from ORB-SLAM-Debug

The next step is to actually integrate them together
so that it is one pipeline



Next steps:

- ☐ get map from ORB SLAM and use in moveit
- ☐ navigate to waypoints w/ map
- ☐ validate payload precision



or see if the flower is in a cone of possible moves the ~~payload~~ end-effector can make ~120°

11/6/21

Got the end-effector working with the Hector URDF file, originally tried to use a link and connect it to the base_link. That didn't work, got a seg fault, so I just attached them together in an stl.

But I just smushed them together overlapping because I didn't want the payload to stick out, or the drone wouldn't balance.

Spent some time trying to get ORB-SLAM2 map, which is in the format of a 3D pointcloud. Using this, I need to either convert into a better map format like Octomap or Octomap, and then give that to moveit, or another thing I could do is to feed the sparse pointcloud directly into moveit. I should take that approach first.
But, I'm having different issues

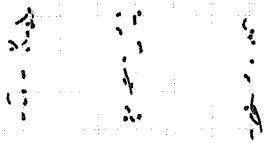
1. Remember the strange TF error from Hector before? I've found that it's caused by the fact that hector uses ground_truth_tf, and it's overlapping w/ ORB-SLAM's tf, I think? It must be a TF error. But I don't know how to fix it. It's just working now though, which leads me to

2. Maybe related to the TF error, the pointcloud, even when I used the RGBD camera, is completely incomprehensible. Something's wrong with the point transforms, because none of them make sense. I'll work on this.

11/7/21

The TF error went away by itself. Sometimes it just happens, but I'm hoping it won't be too bad.

I used ORB-SLAM2 to generate a point cloud, and it seems like this time it worked better.



This three distinct columns, but the entire point cloud seems to be continually centered around the origin, not the translating based on the robot. So something is wrong w/ TF

~~However~~, I also tried to integrate it w/ MoveIt!

In sensors_kinect.yaml, set point_cloud_topic to /orb_slam2_rgbd/map_points
This makes ~~the~~ moveit use the generated map from ORB-SLAM as a pointcloud to navigate.

However, when moveit converts the pointcloud into an occupancy_grid format, the scattered points that ORB-SLAM tends to produce get turned into an absolute mess. In order to fix this, I think I might need to use an intermediate mapping stage

3D pointcloud → slicmap/octomap → MoveIt!

This way, the resulting map will actually be coherent.
However, I don't have much time, so I plan on using the depth camera for now. This will let me implement the end-to-end much faster. Plus, it's already very accurate.

So, besides this SLAM stuff: navigation

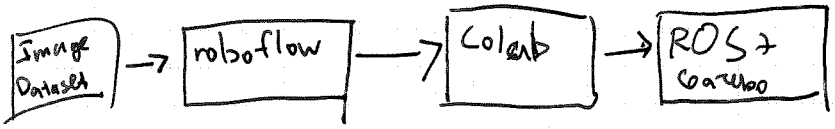
- navigate to waypoints with MoveIt!

I also got started with the flower detection model. I'm using Yolo v3 for unsupervised training. I found a Github repo that I'm trying to get working.

D train flower detection model on flower image dataset

11/8/21

Changed my mind, decided to use ~~the~~ Yolo V5, the newest version, ~~the~~ with Colab to train the dataset. I'm using roboflow to get my data into the correct format.



11/10/21

Turned in STS application.