CS 598 WSI

Lecture 11

10/02/2024

Ananya Kommalapati

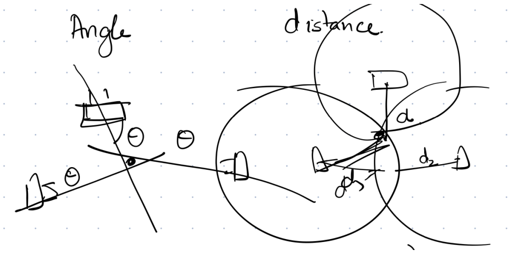
Topics:

Estimating Position

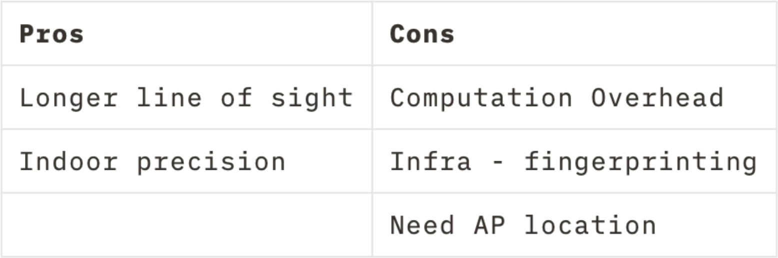
Velocity Estimation: Optical Flow & Doppler Shift

BatMobility

Estimating Position:

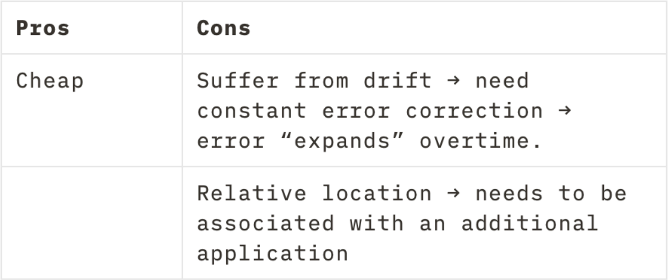
- Up until now, we have discussed 2 primary approaches of estimating position:
  - Angle:
    - Measure the angle from multiple vantage points; pick out the intersection.
  - Distance:
    - Measure the distance from multiple vantage points; again pick out the intersection.
  - 

- The tradeoffs of RF/RSSI-based methods are as follows:

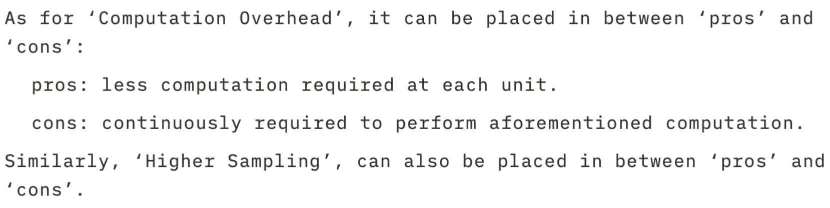


Pros	Cons
Longer line of sight	Computation Overhead
Indoor precision	Infra - fingerprinting
	Need AP location

- In addition to RF/RSSI-based approaches, we can also estimate position via IMU.
  - IMU: Inertial Measurement Unit
  - Some components involved: Accelerometer (acceleration), Gyroscope (angular velocity), Magnetometers (magnetic field -> global orientation).
- The tradeoffs of IMU-based methods are as follows:



Pros	Cons
Cheap	Suffer from drift → need constant error correction → error “expands” overtime.
	Relative location → needs to be associated with an additional application



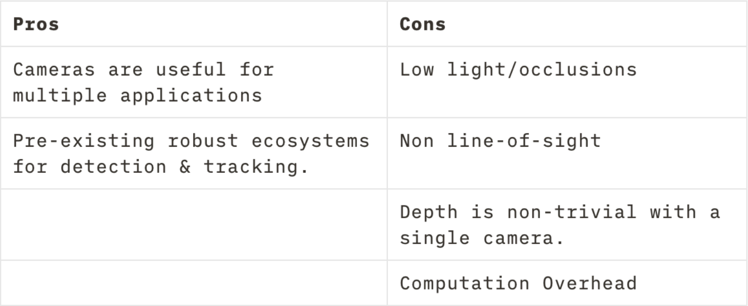
As for ‘Computation Overhead’, it can be placed in between ‘pros’ and ‘cons’:

pros: less computation required at each unit.

cons: continuously required to perform aforementioned computation.

Similarly, ‘Higher Sampling’, can also be placed in between ‘pros’ and ‘cons’.

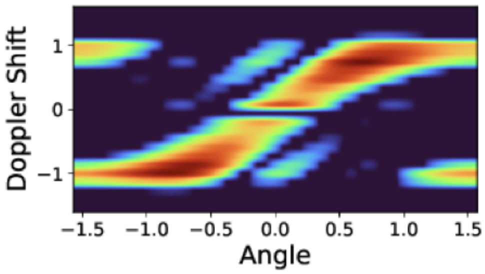
- Lastly, we discussed how vision-based approaches can also be leveraged to aid us in estimating positioning.
    - Cameras (localize) and LIDARs can be used to accurately predict where objects/subjects are.
- The pros and cons of vision-based approaches are as follows:



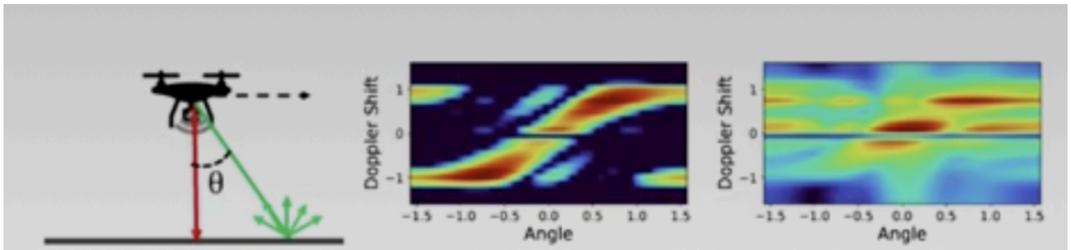
Pros	**Cons**
Cameras are useful for multiple applications	Low light/occlusions
Pre-existing robust ecosystems for detection & tracking.	Non line-of-sight
	Depth is non-trivial with a single camera.
	Computation Overhead

<u>Estimating Velocity:</u>

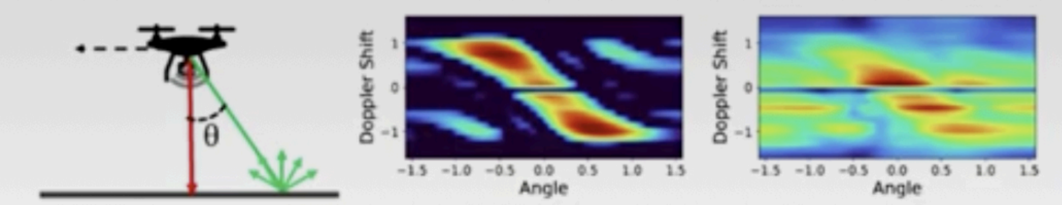
- If we already know position, then the derivative of the position over time can describe velocity.
- However . . . there are simpler/ “easier” representations that can be leveraged for velocity.
    - Accelerometer -> single integration of acceleration.
    - Side note: velocity is described with respect to relative motion.
- E.g.,
    - Assume I have a drone.
    - If we tell the drone to stand still, it will try to ensure that v’ = v (as close as possible) where v = target velocity = 0 and v’ = observed velocity.
        - However, we do need some sort of mechanism to actually keep track of v’.
- Hence, how do we actually get this feedback?
    - Outside-in:
        - Infrastructure-driven.
            - IR (infrared) - based systems.
            - Feedback from the IR sensors; used to localize drone position.
    - Inside-out:
        - Sensor(s) are on the drone.
            - Will measure the velocity by looking at the environment around it.

- Optical Flow:
  - A drone has a downward facing camera.
  - In order to determine the movement of the drone, environmental features are pinpointed and the movement of those features over time is extracted.
  - E.g.,
    - Assume I pinpoint a particular object in my environment (a box) and notice that the movement of the box is to the left.
    - In the case of our static environment, we can thus infer that the drone would have actually moved to the right.
  - Challenges of using Optical Flow:
    - Dark/bad lighting conditions.
    - Featureless.
    - Privacy challenges.
      - Using radar to provide velocity-based control -> BatMobility.
- Doppler Shift:
  - RF signals -> have some frequency f.
  - Observed frequency = $f + vf/c$ where vf/c will increase or decrease based on if the object is moving away or towards you.
  - An application where the Doppler Shift can be applied is as follows:
    - To catch speeding cars:
      - Obtain the frequency of the reflections -> measure the velocity from the received reflections -> compare velocity against the speed limit -> apprehend driver.
  - In the below image from BatMobility, we can gather that the shape represents the direction of motion whereas the amplitude represents the absolute velocity itself.
  - 

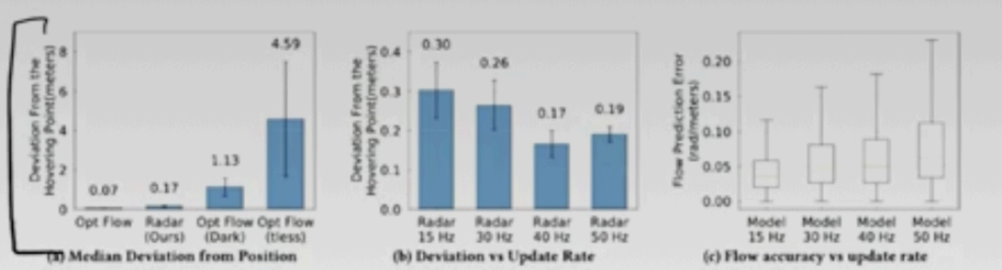
BatMobility:

- With drones moving parallel to the ground, the Doppler Shift is supposed to be zero.
- Surface-Parallel Doppler Shift:
  - Dispersion vs Reflection:
    - If the surface is non-smooth, it may cause issues with reflections.
      - If the jitter is much smaller than the wavelength, then we have a “smooth” surface.
        - Based on the Law of Reflection, if the law of reflection = law of incidence, then we will only get that one point of reflection.
      - However, if the jitter is the same order as the wavelength, then we see a rough surface.
        - Note: light is dispersive in nature.
        - In this paper, the frequencies are < 1cm (few mm) -> floor is dispersive.
          - We get reflections from multiple points.
- 

  (a) (left) Motion of the UAV, (center) Simulated doppler-angle plot, (right) Observed doppler-angle plot.

  (b) (left) Motion of the UAV, (center) Simulated doppler-angle plot, (right) Observed doppler-angle plot.
- The Doppler Shift resolution is inversely proportional to the total time considered for computation.

- Results:
  - When comparing BatMobility's performance to that of Optical Flow in determining how static the drone is, Optical Flow showcases slightly better numbers at first.
  - However, once we turn off the lights, Optical Flow's performance degrades whilst BatMobility's performance is still maintained.
  - And finally, once we eliminate texture/features, Optical Flow's performance becomes much worse whilst BatMobility's performance, once again, is maintained.
  - 

    (a) Median Deviation from Position (b) Deviation vs Update Rate (c) Flow accuracy vs update rate

    Figure 13: Loiter Test. (a) UAV equipped with BatMobility holds its position, but optical flow fails in dark and textureless conditions. (b) Higher update rates support better hovering performance, in spite of higher flow prediction errors shown in (c).
- Challenges/Limitations:
  - Relative position-based work -> we just know the velocity, and not the actual position itself.
  - BatMobility leverages a CNN -> so we must think of how we can instead train it for a broader set of settings as well.
  - Fitting the neural network on top of the board housed on the drone was a challenge.
    - The device was also required to run fast-enough/as per the drone's measurement frequency requirements.
    - Some optimizations to address these challenges are as follows:
      - Reduce resolution of measurements, interpolations, etc.