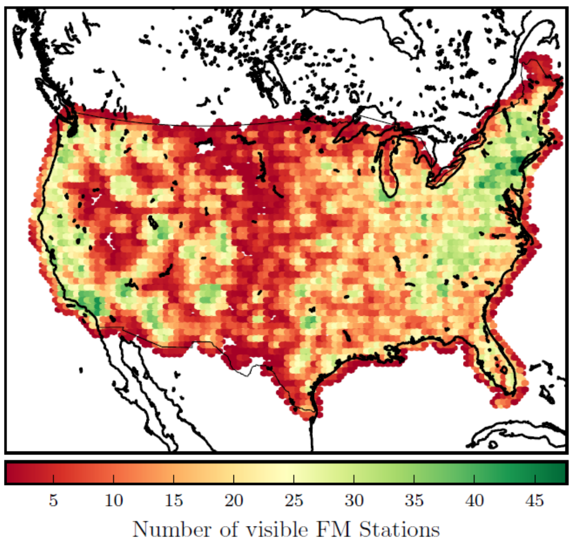
Cheap Passive Approximate localization

Using FM Radio



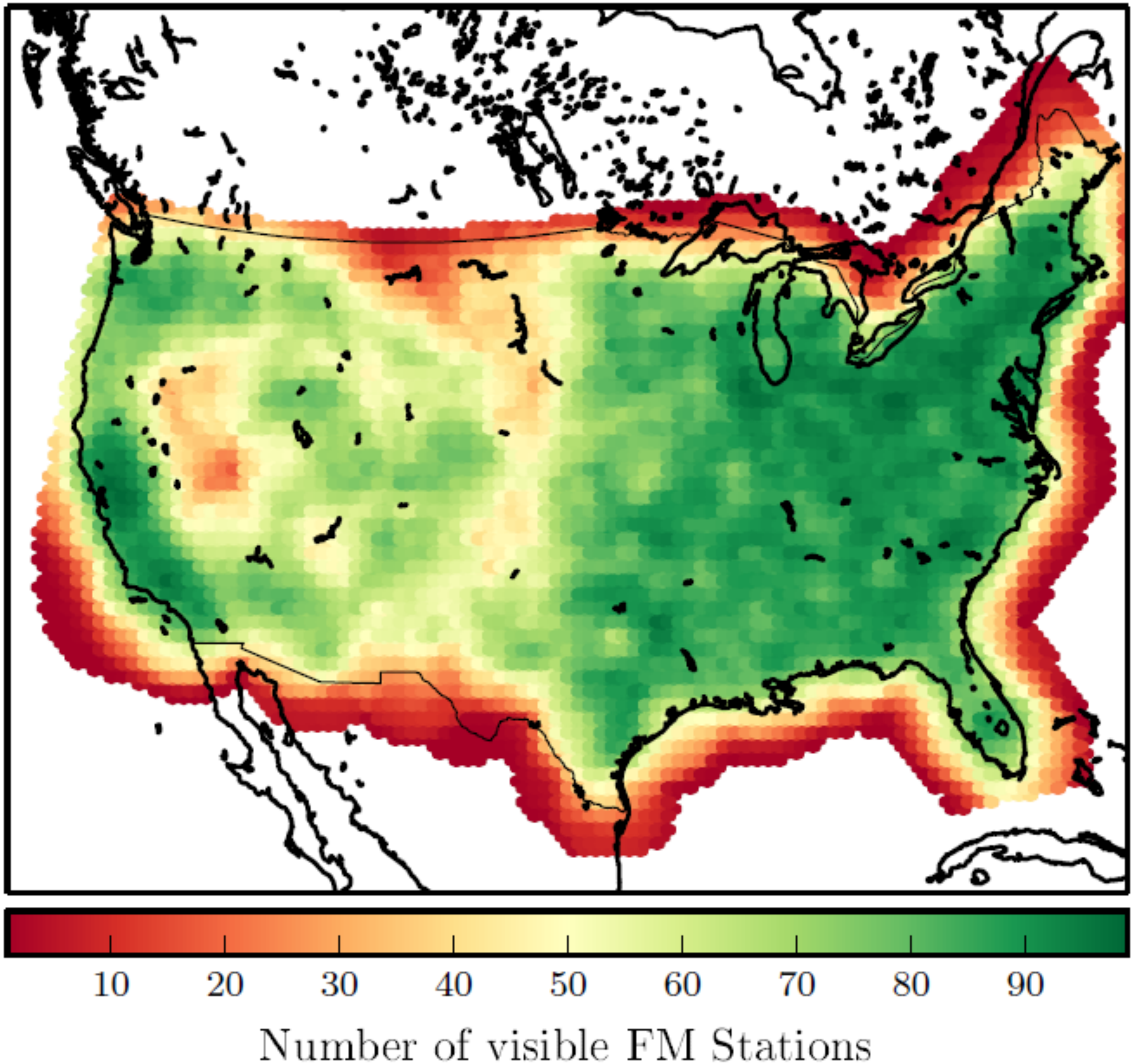
Joint work with:

Andreas Adolfsson, Intelligent Robotics Inc
Tathagata Mukherjee, Intelligent Robotics Inc
Eduardo Pasiliao, AFRL

Piyush Kumar
Webpage: compgeom.com/~piyush

Large Scale Localization using just RSS

Receiving Music : A good antenna or height makes a difference



> 40dbU

Localization

- Defn: To confine or restrict to a particular locality

- But, I have a Cell Phone!
  - Unavailable GPS
  - What if you are indoors?
  - What if power was a concern?
    - (Compared to Wifi/GPS)

- Important for many different applications including communication & navigation in GPS Denied environments

= 500m of no GPS

But why FM?

Technology	Accuracy	Coverage	Power consumption	Cost of infrastructure	Note
Wi-Fi	medium	low	high	low	Low cost if the infrastructure is already available; however, initial deployment is expensive.
Cellular	low	medium	high	low[4]	Subject to environmental influence; low accuracy with standard hardware.
Bluetooth	medium	low	high[1]	high[5]	High localization latency.
RFID	high	low	low/high[2]	low/high[2]	Sporadic location updates.
Powerline	medium	low	not reported	high	Requires specialized hardware.
DECT	medium	medium/ low[3]	low [76]	low	Mobile device requires special (expensive) hardware.
FM (outdoor)	low	high	low [77]	low	Receivers are readily available in mobile devices.
FM (indoor)	?				

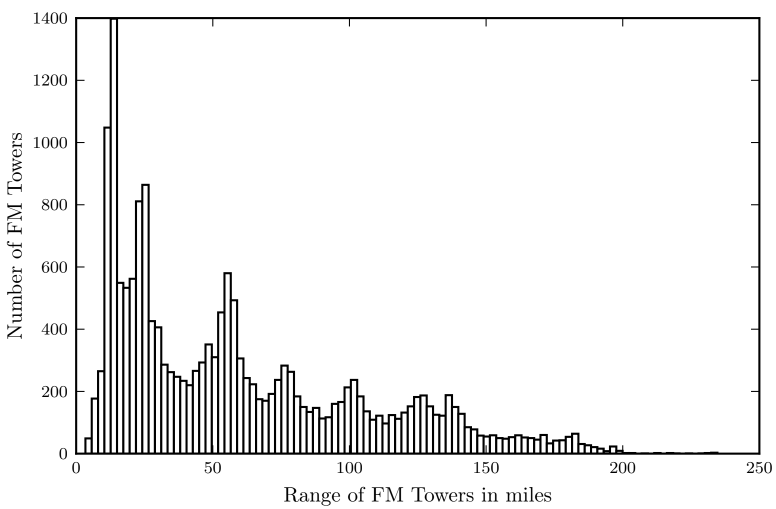
Others: ADS-B, TV, ATC, …

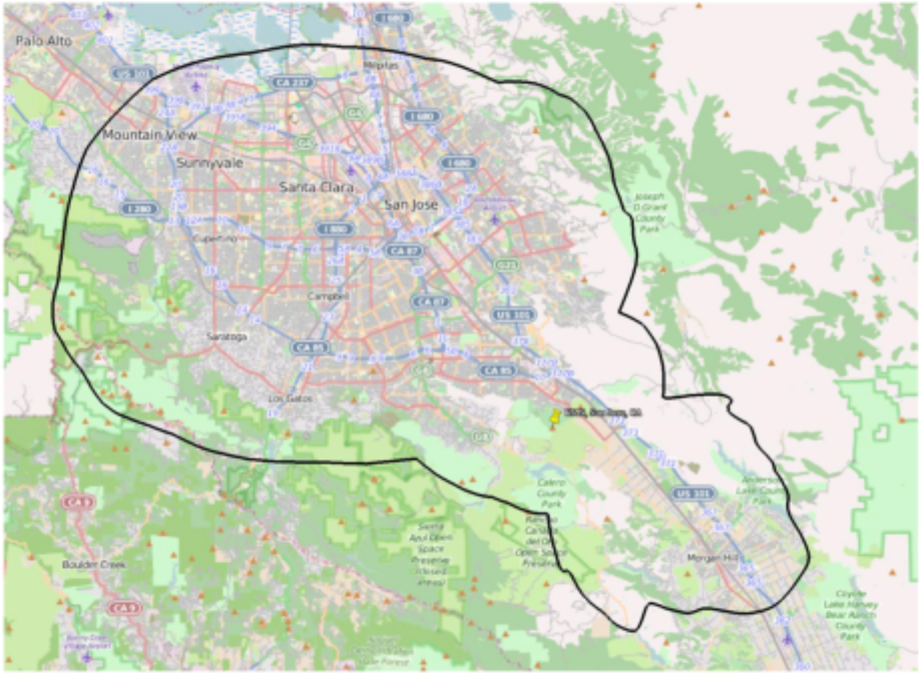
Ours: Two level localization system

FM Localization

FM Broadcast Signal

- FM broadcast band:
  - Large coverage
  - Reliable
  - VHF: 88MHz-108Mhz, BW: 200kHz
  - Less sensitive to weather condition and indoor limitation than GPS





- KSJS (FM 90.5) 60dBu polygon in San Jose, CA

FM Localization Methods

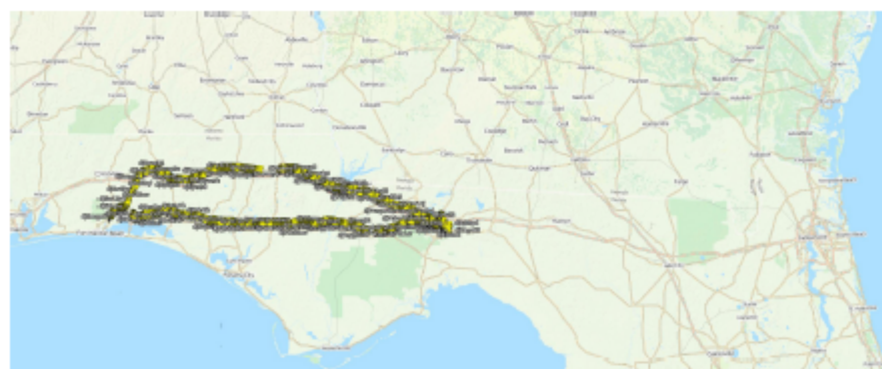
- RF based localization techniques
  - Algorithm: Beacon based, Anchor based, Time of Arrival, Time Difference of Arrival, Angle of Arrival, Doppler
  - Fingerprinting
  - We just use RSS to localize

RSS Based Localization

- Uses RTLSDR for prototyping.
- Capable of scaling to entire Planet/US
- Simple and scalable algorithm
- Improves Localization, both indoors and outdoors
- Easy to make distributed
- Works without line of sight
- No Sync required

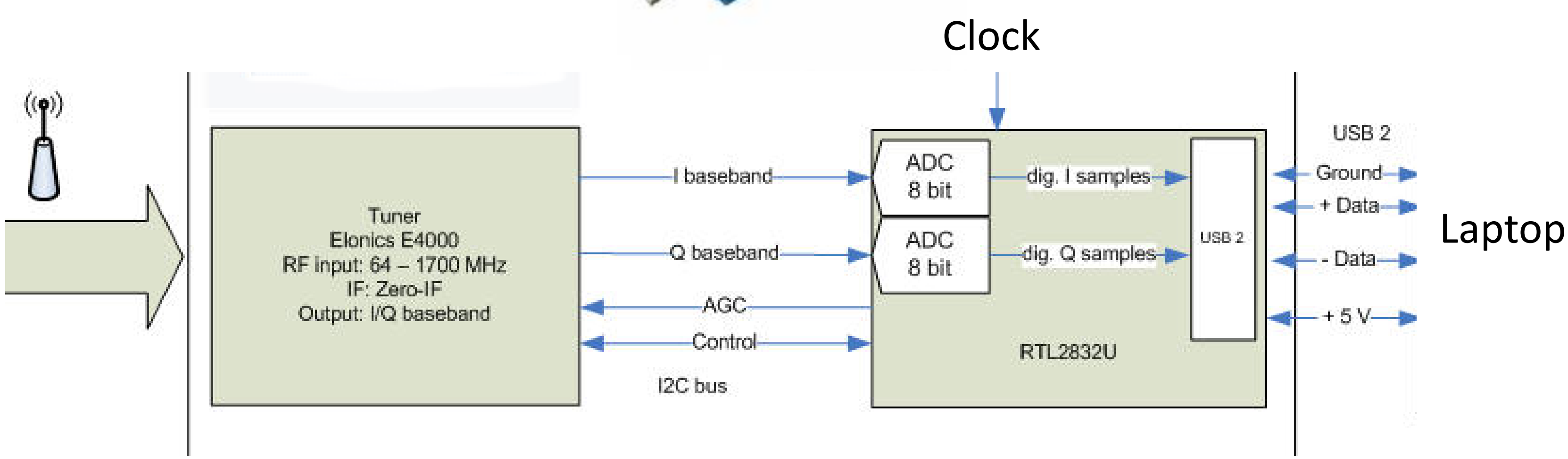
Test Data

- Drove multiple cars: 350+ Miles, multiple days
- Measured FM Signals at approximately 1000 locations



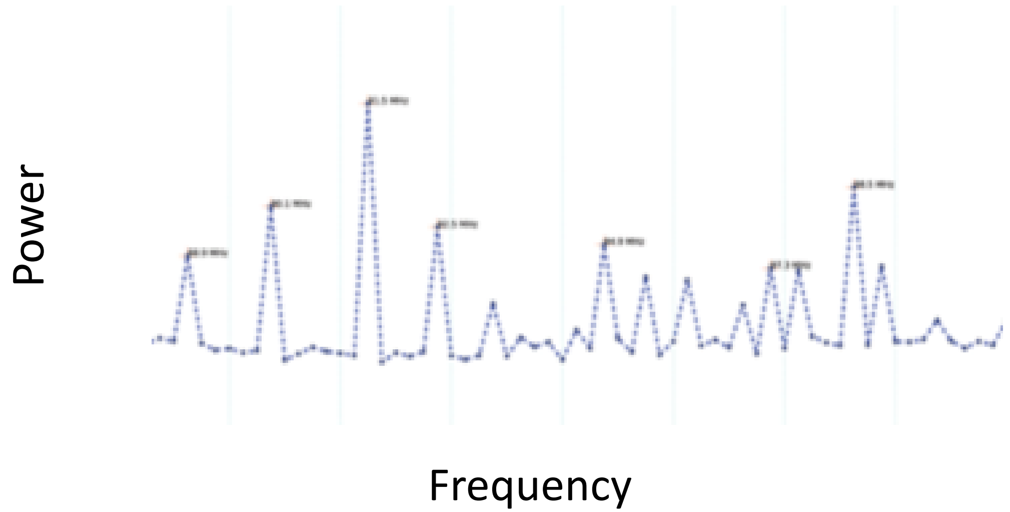
Data Acquisition

- Cheapest RTL Software Defined Radio



Data Acquisition

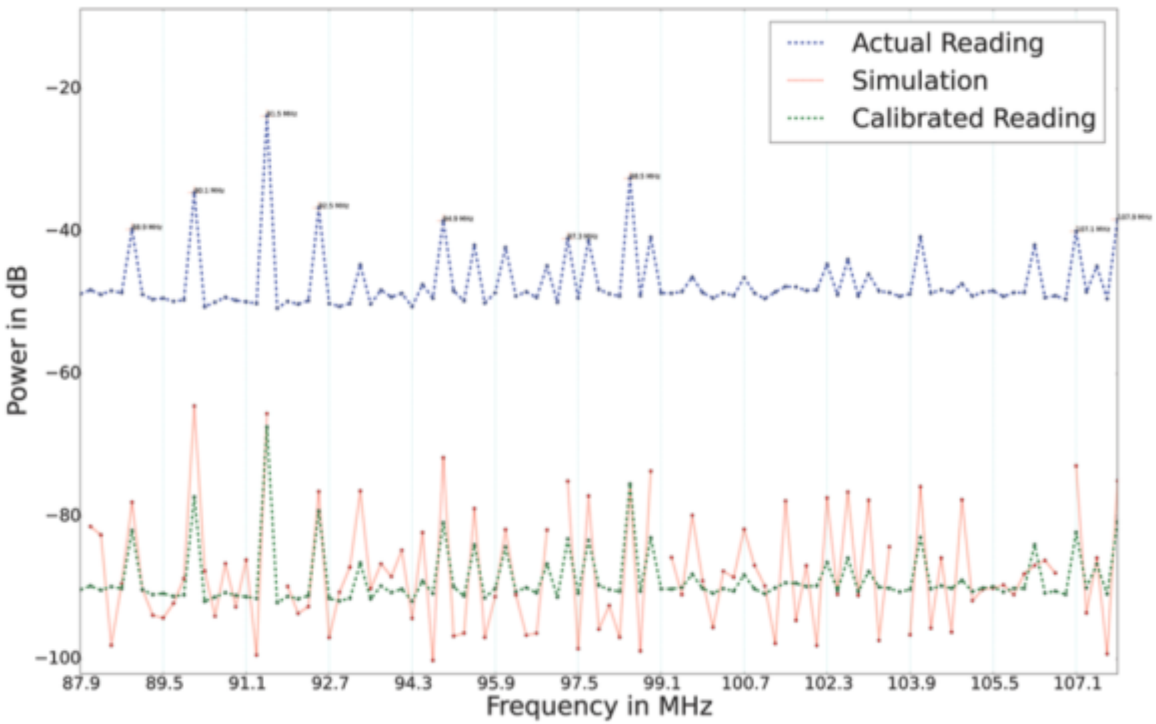
- Cheapest RTL Software Defined Radio
- Power vs Frequency plots



FM Localization

System Overview

- Preprocessing Phase
  - Map Generation
- Query Phase
  - Peak Finding
  - Subset Filtering
  - Nearest Neighbors



FM Localization

Preprocessing Phase

- **Goal**
  - Predicts the estimated power at a point based on the prior knowledge of nearby FM station.
- **Map Generation**
  - 40dBu coverage
  - Entire US with approximately 2.4 mile x 2.4 mile grid
  - Power spectrum in each grid

Algorithm 1 Outputs a model of the simulated RSSI

Require: τ: list of FM tower tuple (t, r, P)
Require: t: FM tower
Require: r: radius of influence of t
Require: P: μ dbu polygon of t

```
function SIMULATE(τ)
    D ← GenGeoHash()                         ▷ Generate all geohashes
    for x ∈ D do
        HT[x] ← [−∞]*101
    end for
    for tp ∈ τ do
        (t, r, P) ← (tp[1], tp[2], tp[3])
        for x ∈ GetPixels(t, r) do
            v ← HT[x]
            p_x ← RayIntersect(P,Location(t), x)
            pd ← ||p_x −Location(t)||
            xd ← ||x−Location(t)||
            arg = (Power(P),pd,xd,v[FreqIndex(t)])
            v[FreqIndex(t)] ← CalculatePower(arg)
        end for
    end for
    return HT
end function
```

FM Localization

Preprocessing Phase

- **Goal**
  - Predicts the estimated power at a point based on the prior knowledge of nearby FM station.

- **Map Generation**
  - 40dBu coverage
  - Entire US with approximately 2.4 mile x 2.4 mile grid
  - Power spectrum in each grid

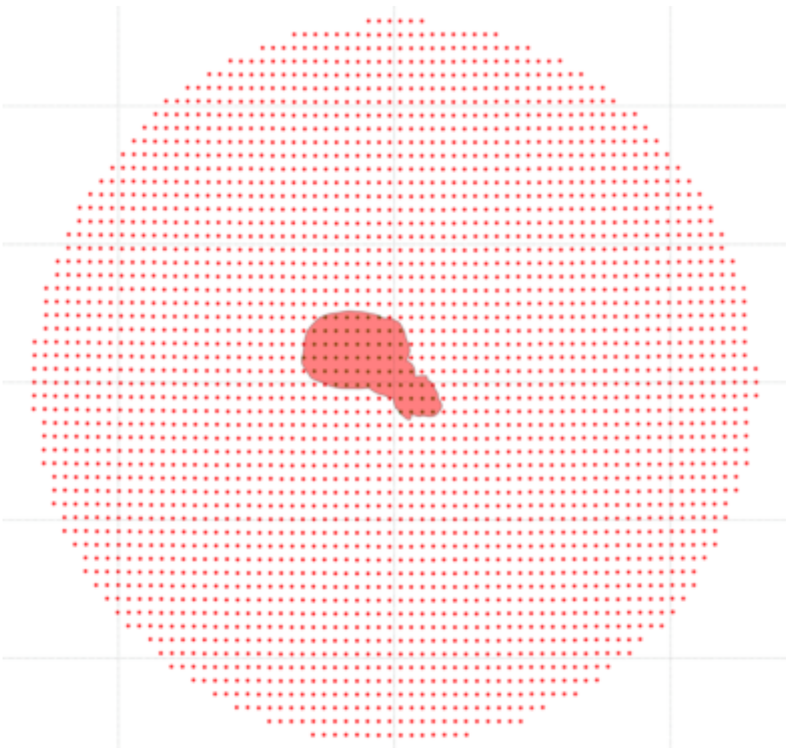
Algorithm 2 Outputs computed power at a given location

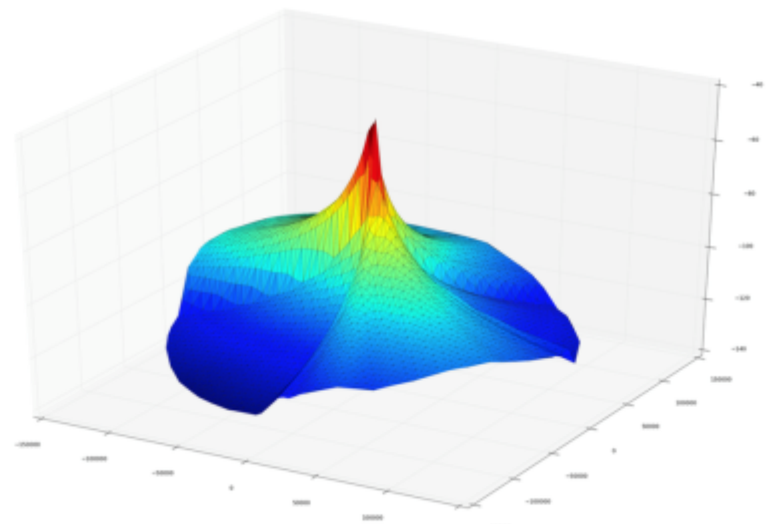
Require: τ: tuple containing following:
Require: PP: dbu polygon for a FM tower t
Require: $polydist$: distance of polygon intersection from center of tower
Require: $pixeldist$: distance of point from center of tower
Require: $Ppow$: the previous power at point

function CALCULATEPOWER(τ)
    if $pixeldist < 0.1$ **then**
        $pixeldist \leftarrow 0.1$
    end if
    dbu $\leftarrow PP - 40 \log \frac{pixeldist}{polydist}$
    if $Ppow == -\infty$ **then return** dbu
    end if
    $Npow \leftarrow$ Aggregate(dbu,Ppow)
    return $Npow$
end function

FM Localization

Preprocessing Phase





FM Localization

Peak Finding Phase

- Power spectrum at one location
  - Looking for "spikes" along the spectrum
  - Compare adjacent signal strength with a threshold
  - Return the channel frequencies with peak

Algorithm 3 Outputs peaks in power spectrum

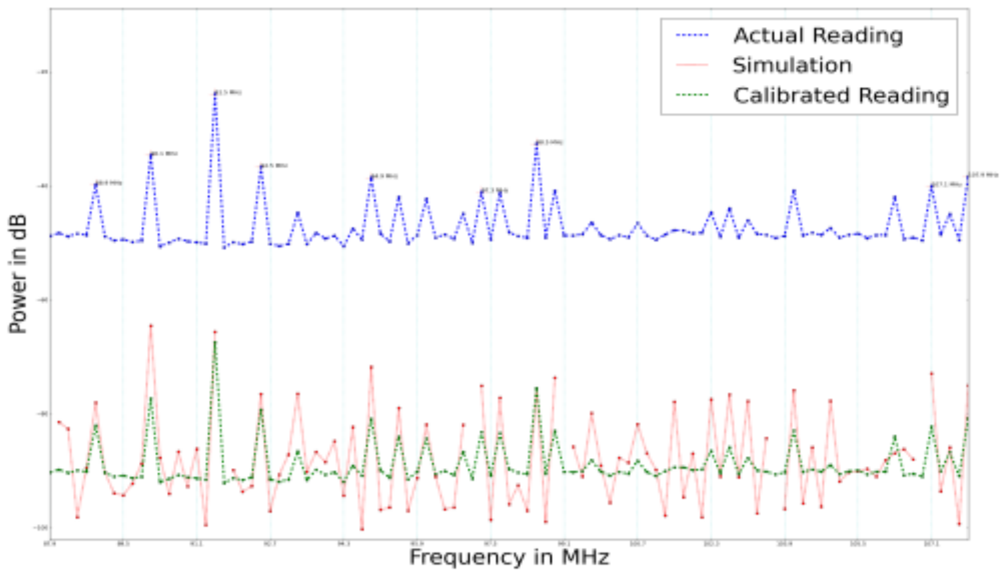
Require: Ψ: observation of power spectrum

function FINDPEAKS(Ψ)
    p $\leftarrow$ [] $\triangleright$ For peaks
    $k \leftarrow$ cutoff1
    $\nu \leftarrow$ cutoff2
    for $(i, pw) \in$ Enumerate(Ψ) **do**
        $lg \leftarrow min\{pw - \Psi[i-1], pw - \Psi[i+1]\}$
        if $lg > \nu$ **then**
            c $\leftarrow min\ lg$
            p.add((pw, lg, i))
        end if
    end for
    $p \leftarrow$ MinK(p, k)
    return p
end function

FM Localization

Peak Finding Phase

- Power spectrum at one location
  - Looking for “spikes” along the spectrum
  - Compare adjacent signal strength with a threshold
  - Return the channel frequencies with peak



FM Localization

Subset Filtering Phase

- Subset Filtering: Search Space Reduction
  - Goal: reduce initial search area down to few hundred square miles
  - Given a set V, for any query vector q $\in$ {0,1}, detects if any vector p $\in$ V such that q is a subset of p

---

Algorithm 4 Outputs geohashes matching observed peaks

Require: $peaks$: observed power spectrum
Require: SS: hash table: bit vectors to geohashes

function SUBSETFILTER($peaks, SS$)
    $subset \leftarrow \emptyset$
    for $i \in SS.keys()$ **do**
        if $peaks \wedge i == peaks$ **then**
            $subset \leftarrow subset \cup SS[i]$
        end if
    end for
    return $subset$
end function

FM Localization

Query Phase

- Query Phase: Actual localization algorithm
  - Acquires the power spectrum
  - Finding the peaks in the acquired power spectrum
  - Invoke Subset Filter
  - Get a minimum value, which indicate the distance between two spectrums, restricted on a given set of peaks P.
  - Return predicted location from Geohash

Algorithm 5 Outputs approximate localization

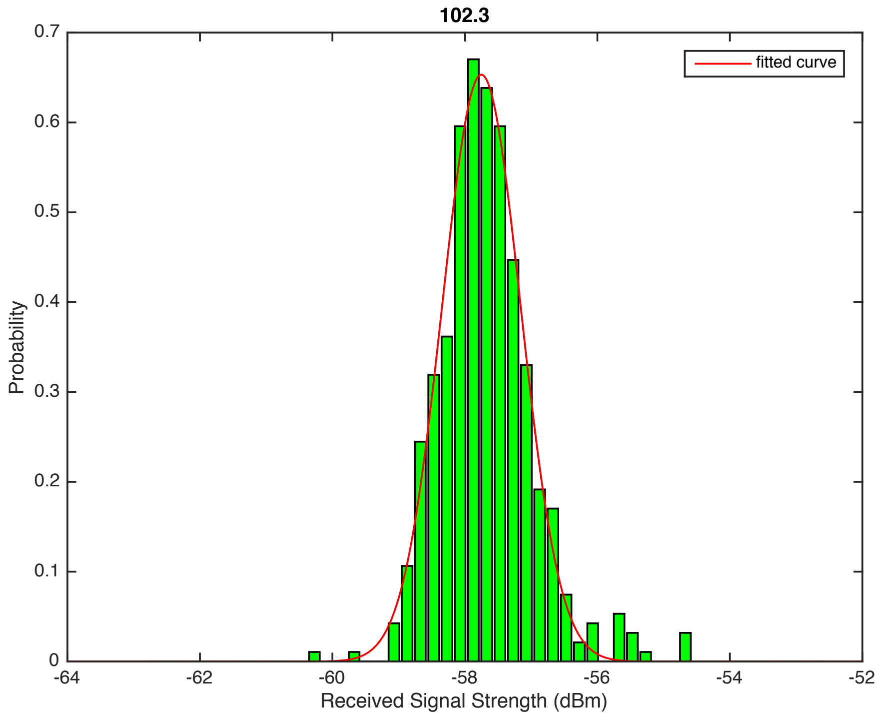
Require: $\mathcal{HT}$: model obtained from simulation
Require: $\mathcal{SS}$: model for subset filter

function LOCALIZE($\mathcal{HT}, \mathcal{SS}$)
  $\Psi \leftarrow$ AcquireRTLPower()
  $pks \leftarrow$ FindPeaks(Ψ)
  $\delta \leftarrow$ SubsetFilter($pks, \mathcal{SS}$)
  $o \leftarrow argmin_{i \in \delta}$ EuclideanDistance(Power(i),Calibrate(Ψ))
  return Geohash.Decode(o)
end function

FM Localization

Euclidian Metric / Calibration challenges

- Variability (Both Tx and Rx)
    - Time
    - Temperature
    - Humidity
    - Experimental Error



Why min distance?

$$\hat{L} = \underset{i \in [1 \ldots M]}{argmax} P(D_i | X)$$

D_i = i-th Result of the subset query
X = Peaks at the location of interest
M = Number of matches from subset query
L = Most probable location

$$P(D_i|X) = \frac{P(X|D_i)P(D_i)}{P(X)} \propto P(X|D_i)$$

FM Localization

Why the Euclidian Metric?

$$P(X|D_i) = \prod_{w \in C_X} \{ \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(S_w - \mu(\widehat{D_{wi}}))^2}{2\sigma^2}} \}$$

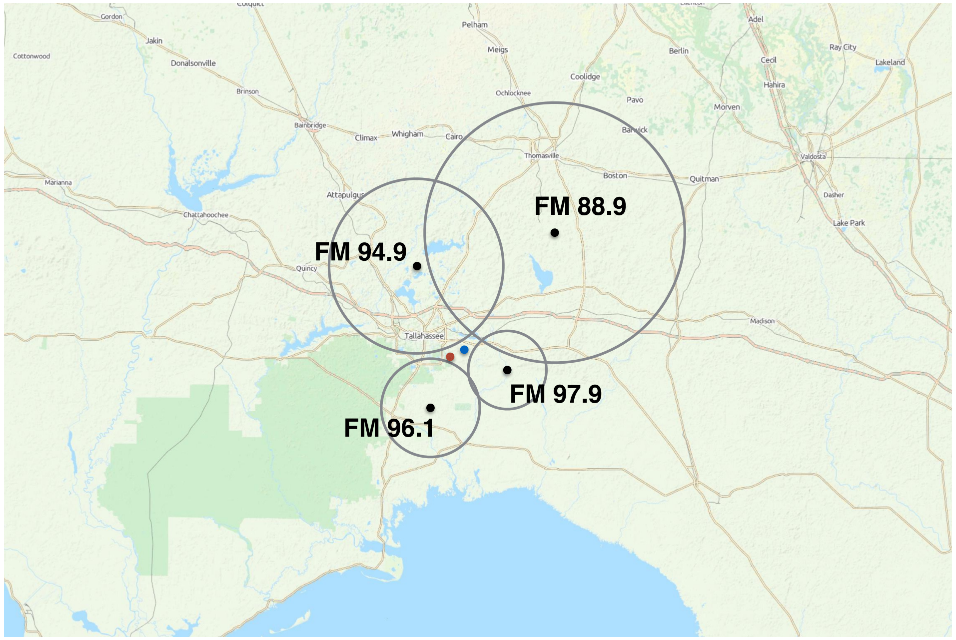
$$\hat{L} = \underset{i}{argmax} \{ l \cdot \ln \frac{1}{\sigma\sqrt{2\pi}} - \sum_{w \in C_X} \frac{(S_w - \mu(\widehat{D_{wi}}))^2}{2\sigma^2} \}$$

$$\hat{L} = \underset{i}{argmin} \sum_{w \in C_X} (S_w - \mu(\widehat{D_{wi}}))^2$$

FM Localization

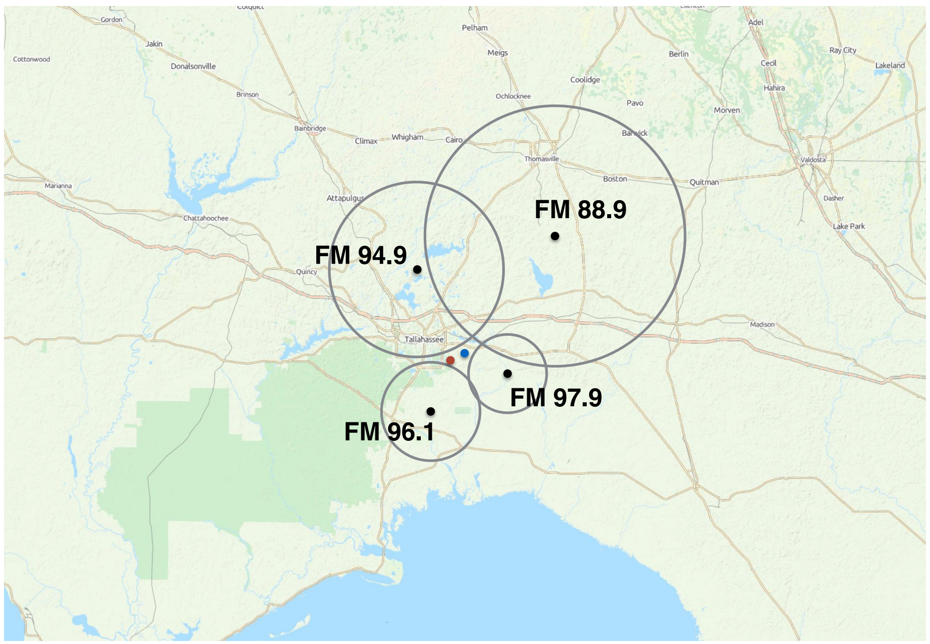
Friis Model

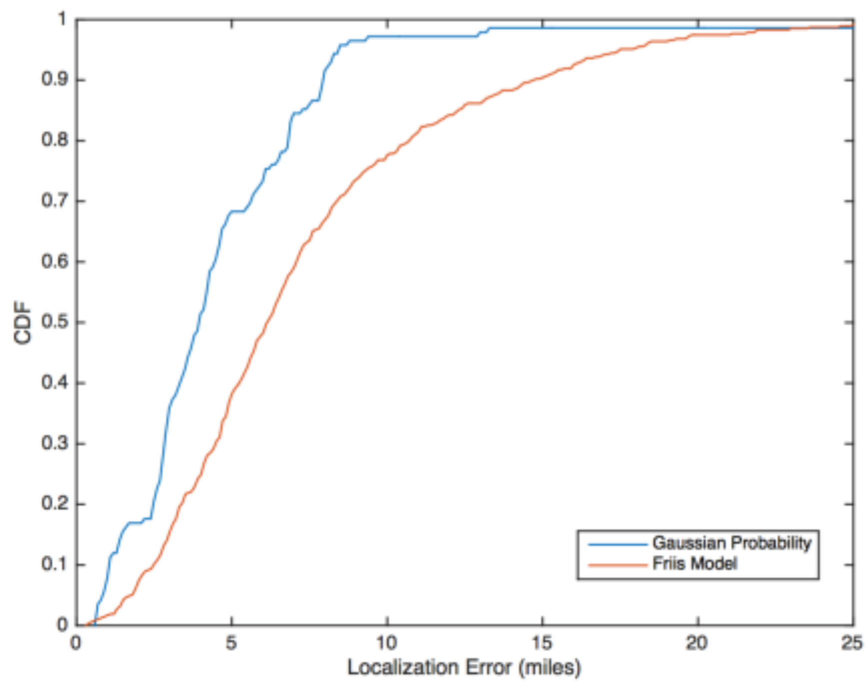
- Friis Model:
  - Direct receive signal strength calculation
  - 1700 measurements for loss factor in Tallahassee, FL
  - Assuming isotropic transmission, ignoring multipath/terrain effect
  - Trilateration fitting for circle



FM Localization

Friis Model





FM Localization

Results

- Euclidian algorithm with Gaussian probability has the minimum error compares to Friis Model and Kendall-Tau Model

Table 1: *Comparison of two different metrics used for Localization in Miles.*

Metric	Mean Error	Std. Dev.	Median	Max
Eucledian	4.98	3.93	3.98	37.14
Kendall-Tau	849.76	586.07	702.23	2373.92

Improving accuracy in the air

https://youtu.be/DYP22RmxbQ8

Autonomous Data Collection

DJI S1000+ Frame+ motors+ ESC

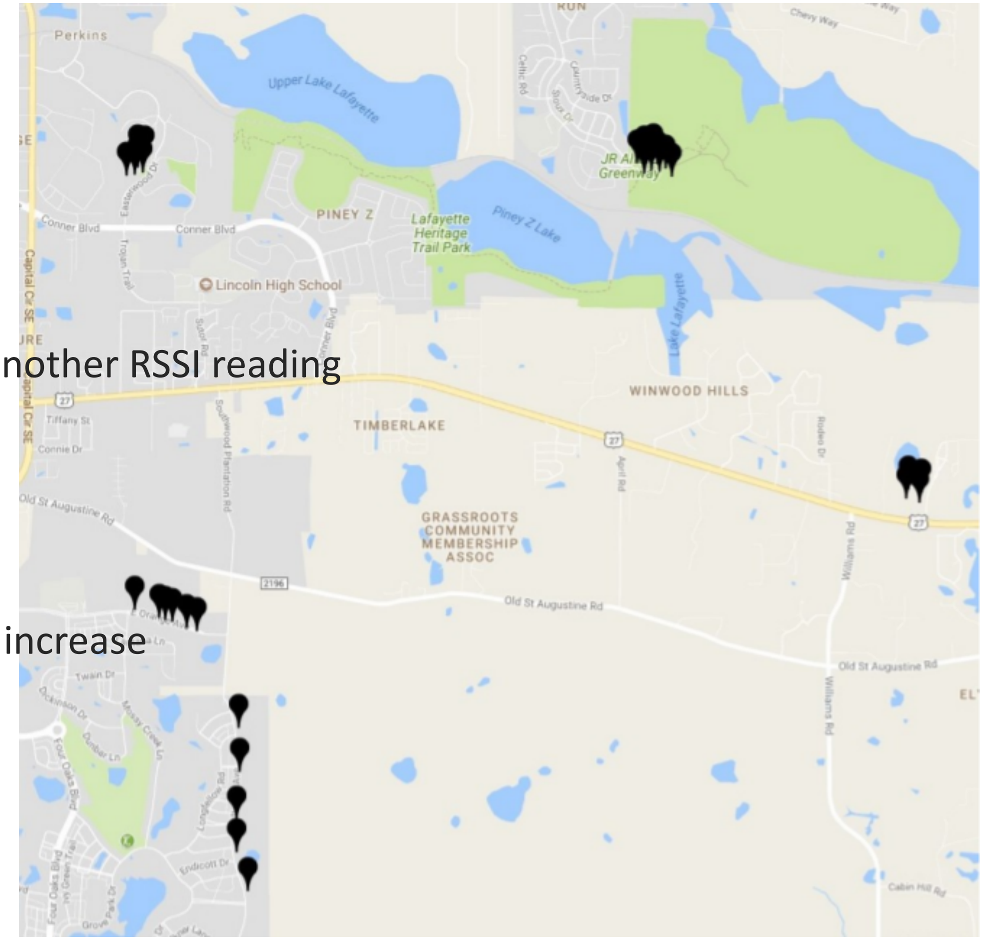
Pixhawk autopilot + PX4 firmware

- RTK GPS module
- FM Antenna
- i7 NUC computer
- RTL-SDR + Ettus B210
- Bluetooth Speaker
- Logitech c920 Camera

Autonomous Data Collection

After drone is armed:

- Checks GPS accuracy
- Collects RSSI reading on ground
- Lifts to 120 meters in the air
- Remains stationary in air while collecting another RSSI reading
- Lands



Collected 30 Data points

- Accuracy of results improve as data points increase

Data Processing

We first use the previous algorithm to get our error to 5 miles.

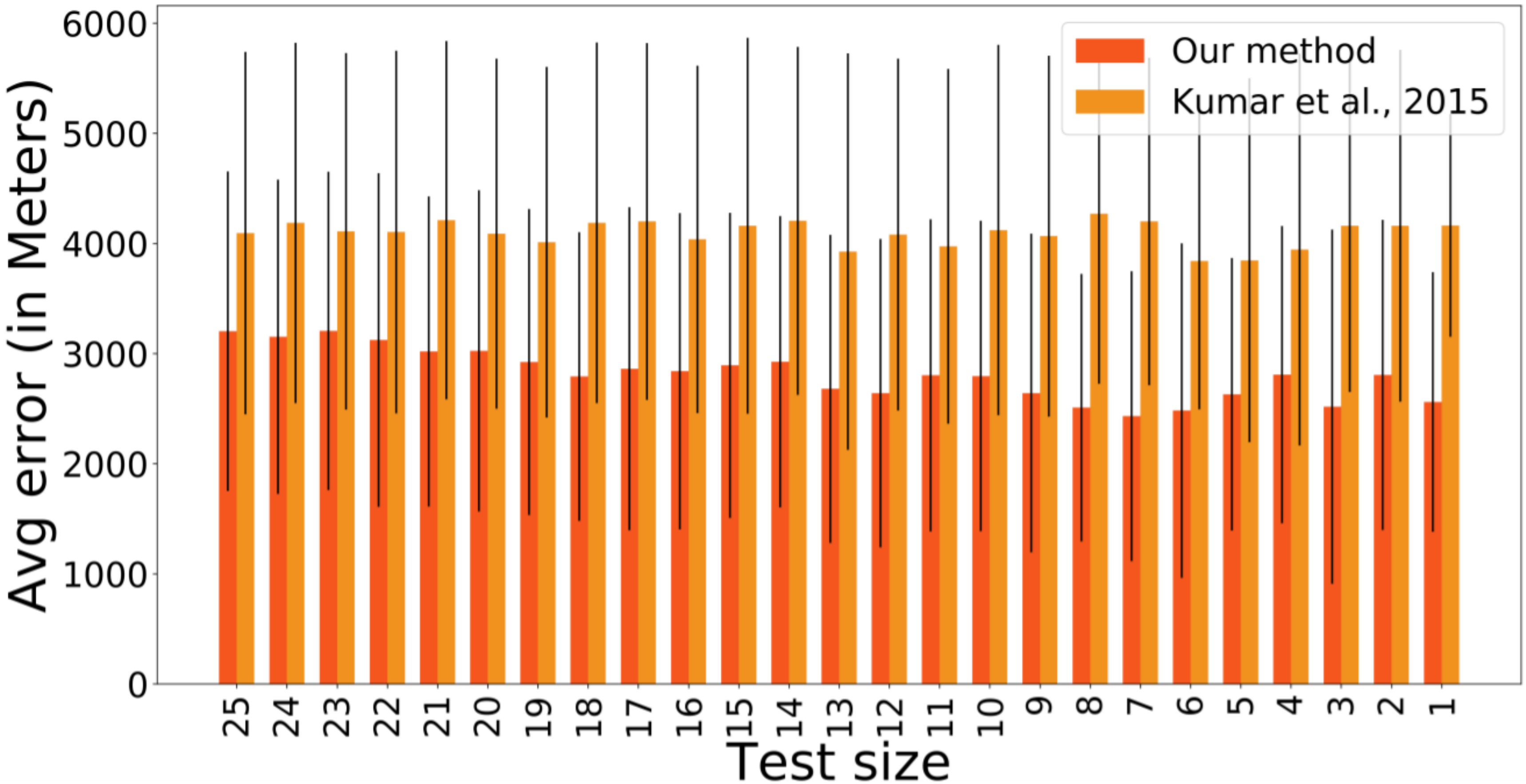
Our model learns to estimate the distance to the transmitters from a given location using:

- the transmitted power
- the received power at the location
- the height of the receiver
- the height above average terrain (HAAT) of the transmitter

We chose a Random Forest regression model, using supervised learning techniques to estimate this distance to each transmitter.

random forest / Neural Net / Support Vector Machine

Aerial Results



Min Error: 172 meters, Average Error: 3000 meters.

Acknowledgements

- AFRL
- CompGeom Inc.
- Intelligent Robotics Inc.
- Ettus Research
- Florida State University

Questions

Future Work

FM Localization and Robotics

- Improve FM localization accuracy:
  - TDoA and AoA with directional antenna
  - Simulated Database improvement: Splat! Simulation or RadioMap (DARPA)
  - Alternate modalities: ADS-B, Iridium Satellite Constellation

- Computer Vision for localization and collision avoidance