Wildlife Drones and Flight Plans

DR BAM team

Project goals

- Empower rangers:
    - Where target animals are **likely** to be
    - Where **tracked** animals are likely to be **in the future**
    - Where **poachers** are likely to be
    - Track recent poaching incidents
    - Report poaching incidents
    - View drone routes

Project goals

- Generate flight plans which
  - Fly over as many high priority areas as possible
  - Are random enough so poachers cannot easily predict them
  - Are dependent on flight time of the drone
- Inform rangers of
  - Probable poaching areas
  - Areas where animals are likely to be
  - Live drone location relative to flight plans
  - Any reported poaching incidents

Project goals

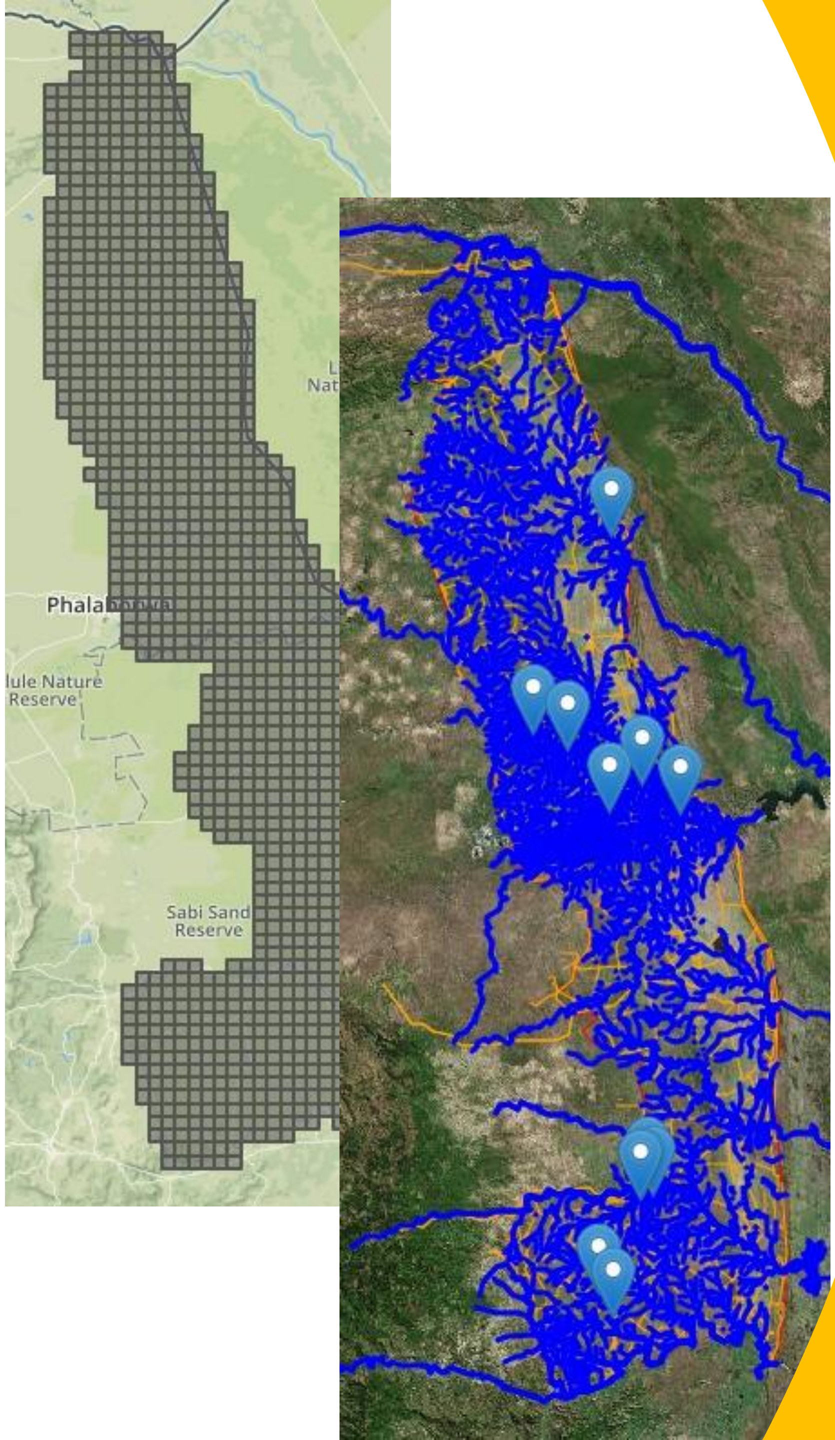
- The system should be applicable to most game reserves – we have stressed tested using Kruger National Park
- The system should be useful
- Configurable
- Front-end mobile and desktop application for use by pilots
- Back-end interface for use by administrators
  - Provide full control over system
- Server which handles all processing and storage

Map data

- Pulled from OpenStreetMaps
  - Rangers can update map data via OpenStreetMaps
  - Roads, residential areas, fenced areas, water
- Satellite imagery from Google Earth
  - 30cm accurate
  - Provides useful information not visible from vector maps
  - Overlay osm features on top of satellite to show names of places
- Elevation
  - 30m accurate from NASA SRTM Shuttle Radar Topography Mission

Map partitioning

- Divide reserve into cells
  - Cell size is configurable
  - Trade-off of performance vs accuracy (smaller = slower, more precise) – accuracy != precision
  - Drones patrol these areas (cells)
- Determine necessary properties for each cell
  - Distance to water: dams, rivers, weirs, lakes, streams (to water edge, not center)
  - Distance to roads
  - Distance to residential areas and fenced areas
  - Altitude, altitude difference (how slope-y)



Map calculations

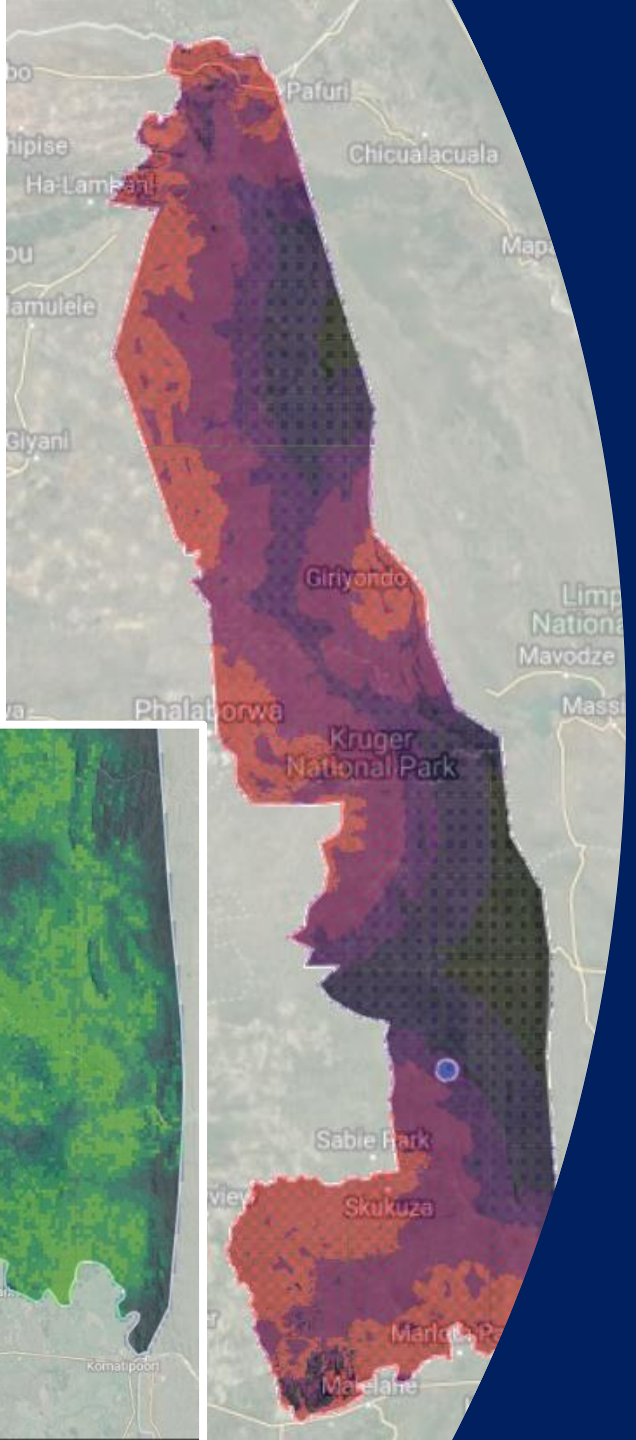
- OSM
  - 49 000 river points
  - 178 000 intermittent river points
  - 1 500 reserve boundary points
  - 127 000 road points
  - Residences, etc.

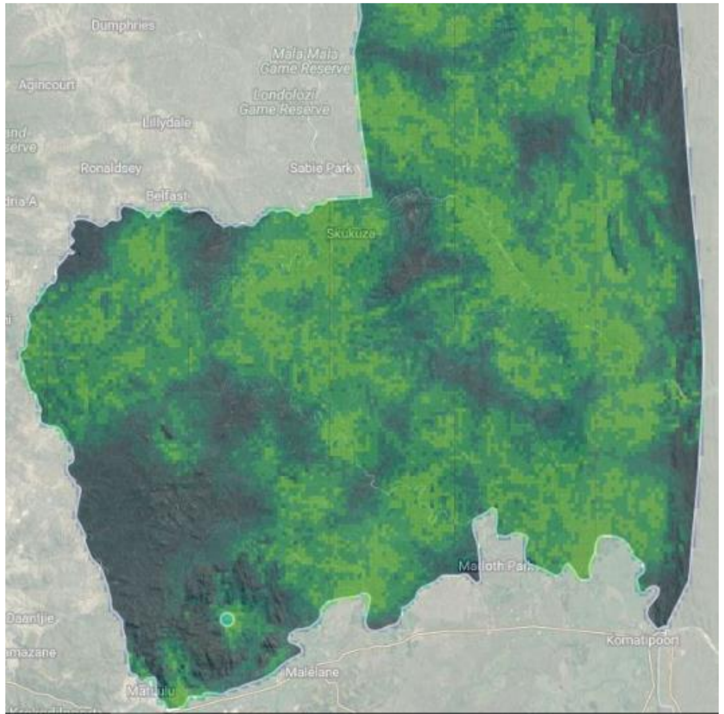
- Partitioned Kruger into 500x500m cells = 76 520 cells
- + 283 000 tracking points = +/- 350 000 calculations on +/- 350 000 map points

- Used a k-d tree to efficiently find in O(logn) time, rather than O(n^2)
- Few milliseconds vs few seconds -> minutes vs months (we saw a 19 000x speed up)

Finding where animals/ poachers might be

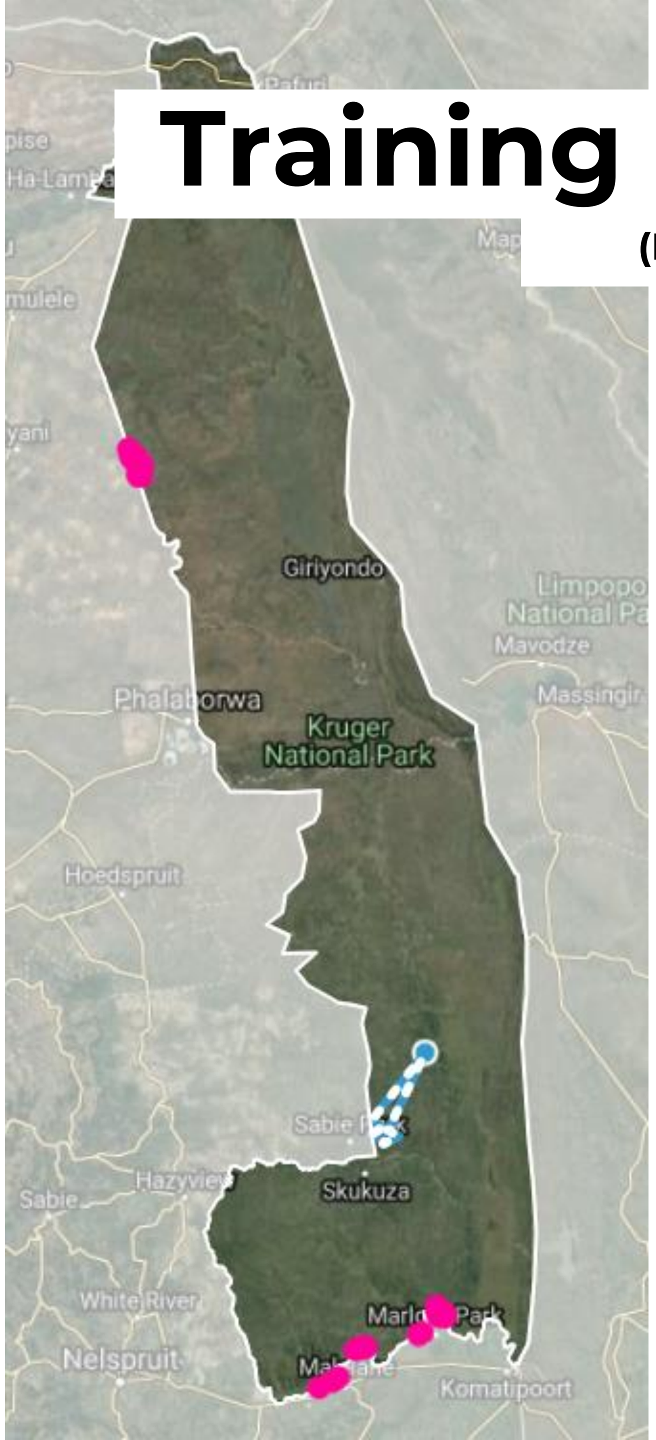
- Determine what types of cells animals and poachers like
- Custom K nearest neighbours algorithm
- Effective lazy unsupervised learning
- Compares geographical similarity between cell and animal tracking points and poaching incidents

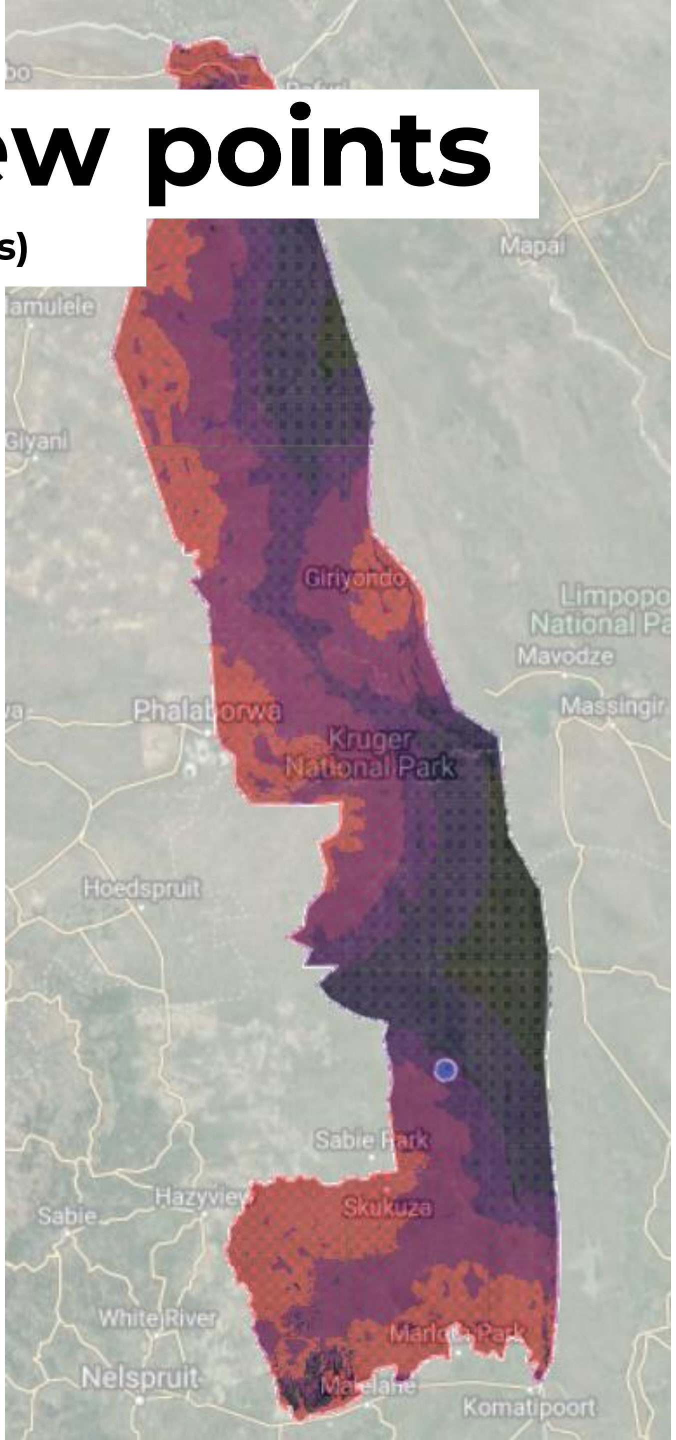




Training based on few points

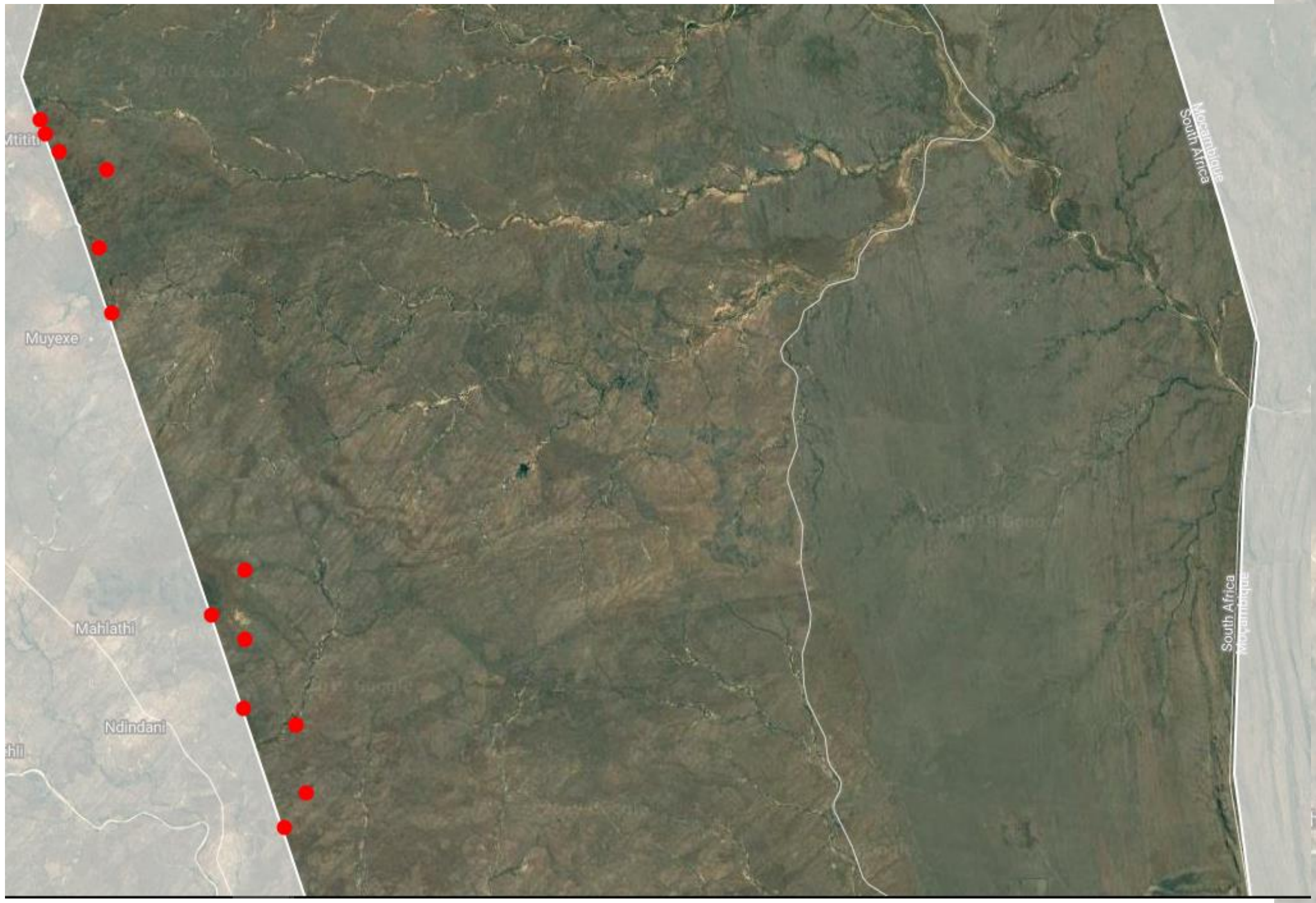
(Mocked poaching incident points)

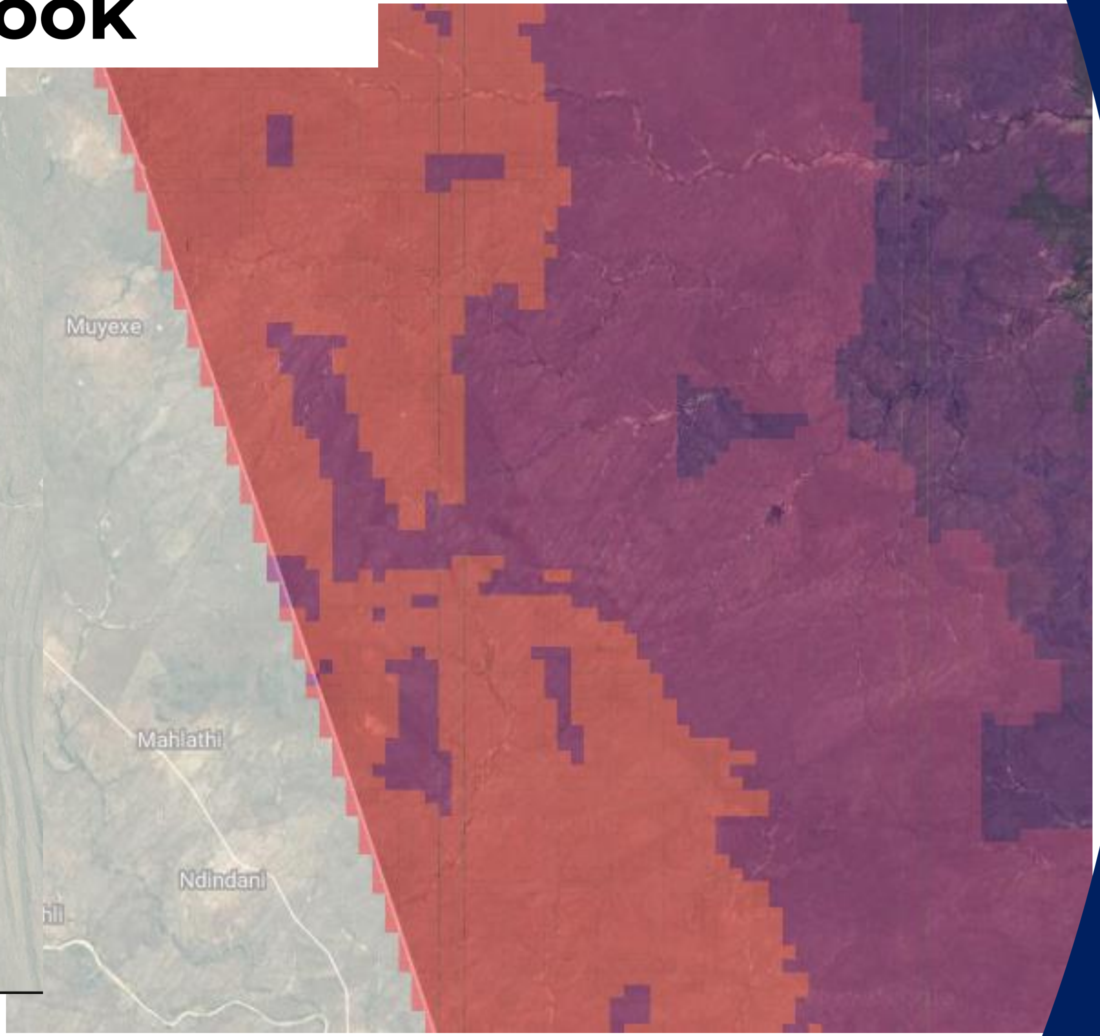




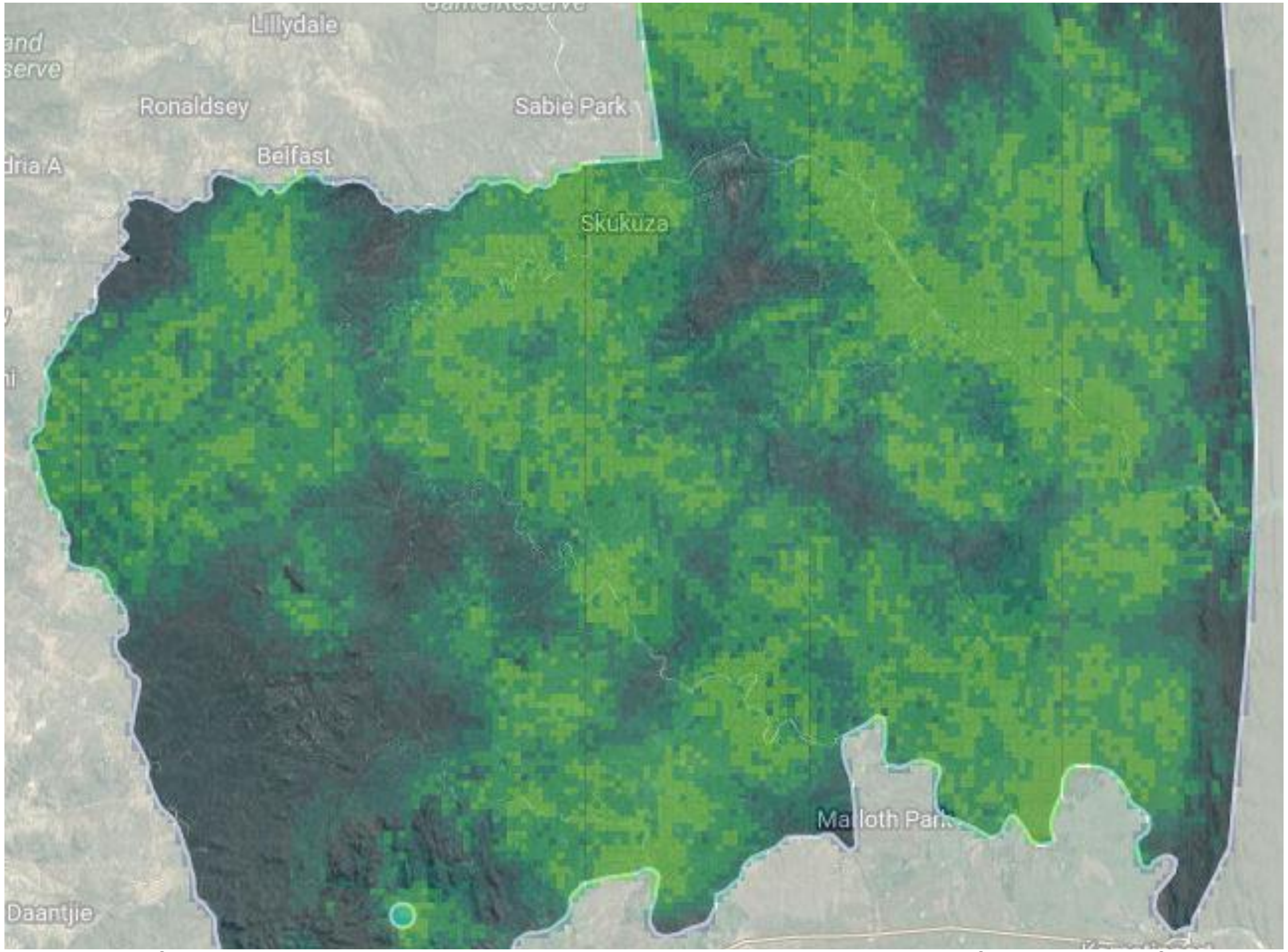
Training based on few points

Closer look





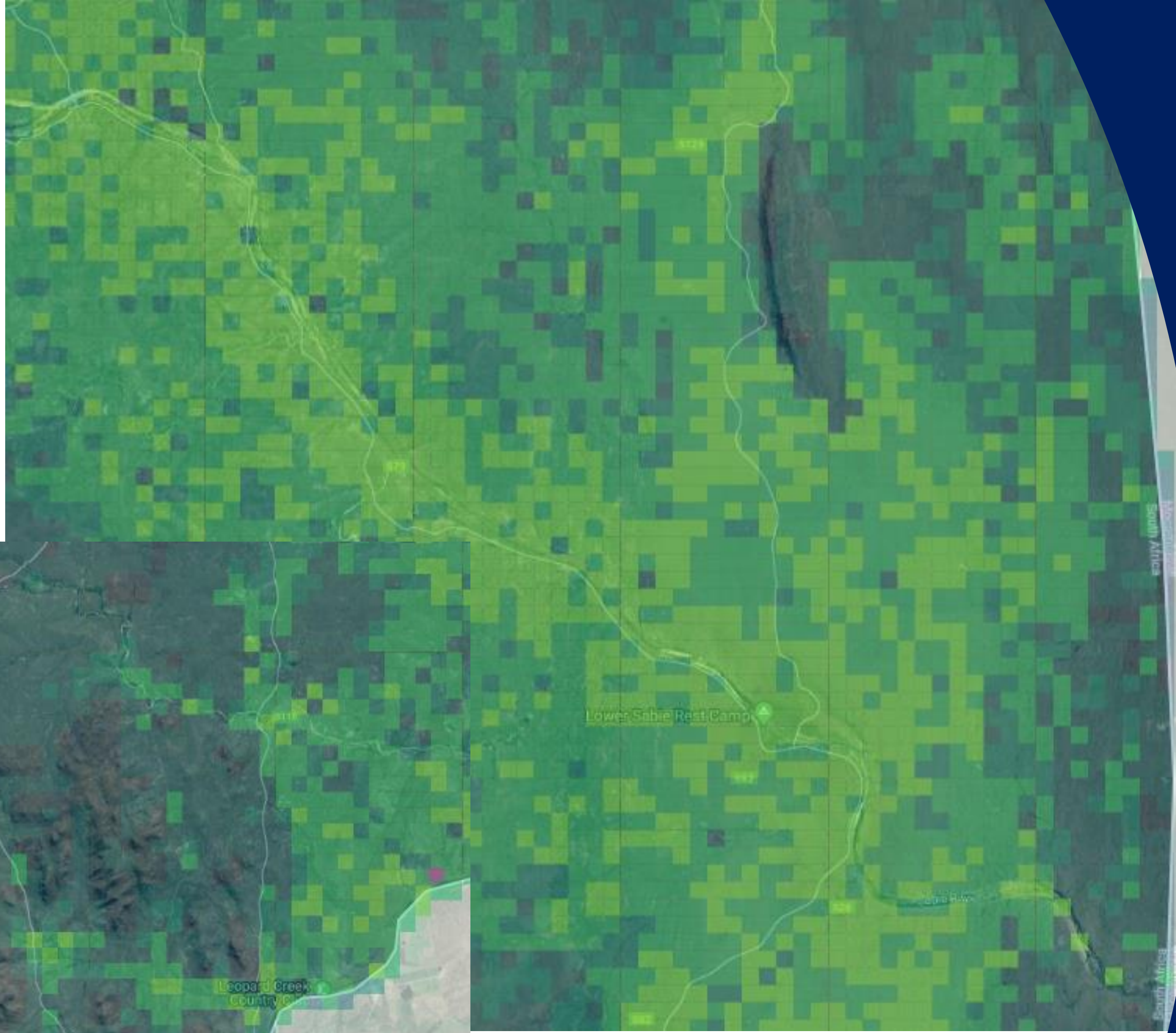
Trained on animal data

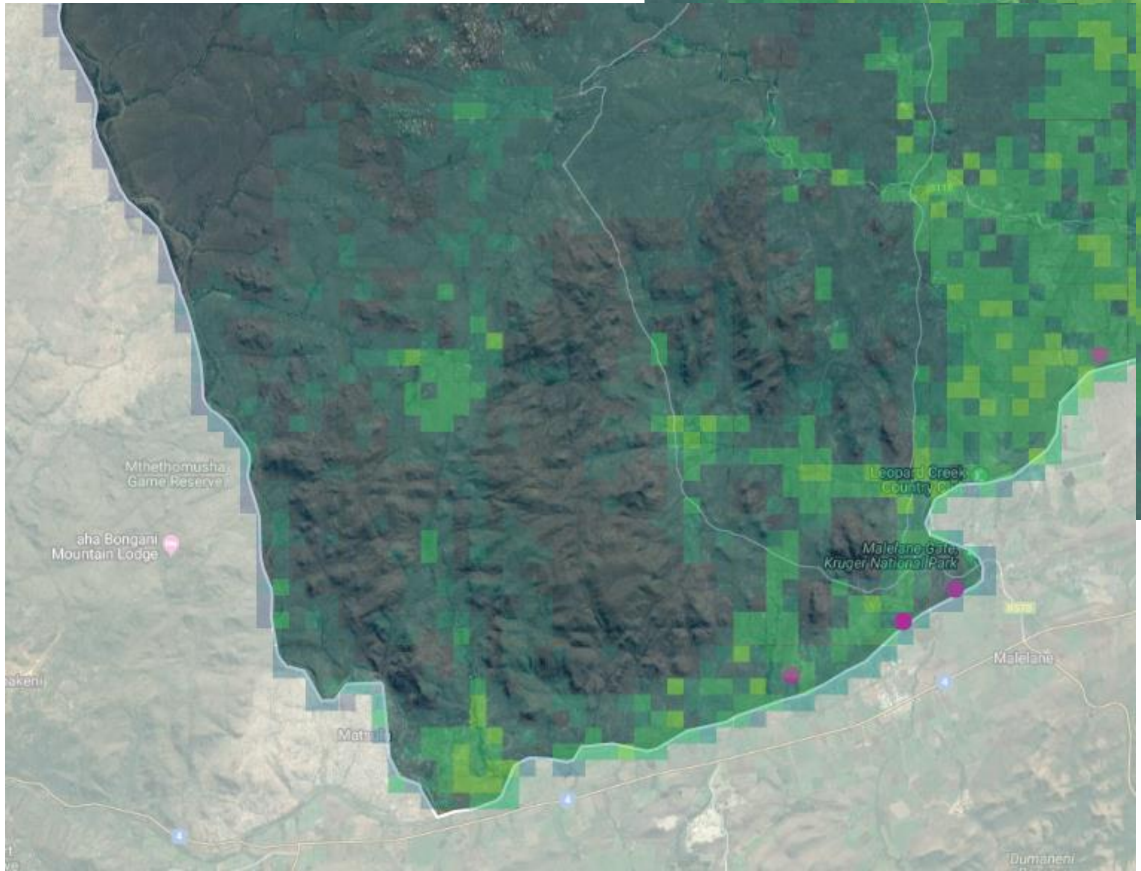


Train on 283 000 elephant tracking points (Movebank.org)

Findings

- Elephants like rivers and flat areas
- Not steep areas





Route generation

- We are breaking drone routes into two types:
  - “Static” routes – points do not move
  - “Dynamic” routes – intercepts points

- Static routing uses heuristic approach to shortest paths
  - Based on Clarke-Wright savings algorithm (vehicle routing problem)
  - Best pairs of points to visit are ranked and joined
  - Ranking based on distance savings, value of visiting node
  - Pairs are discarded if infeasible (flight time/distance is exceeded)

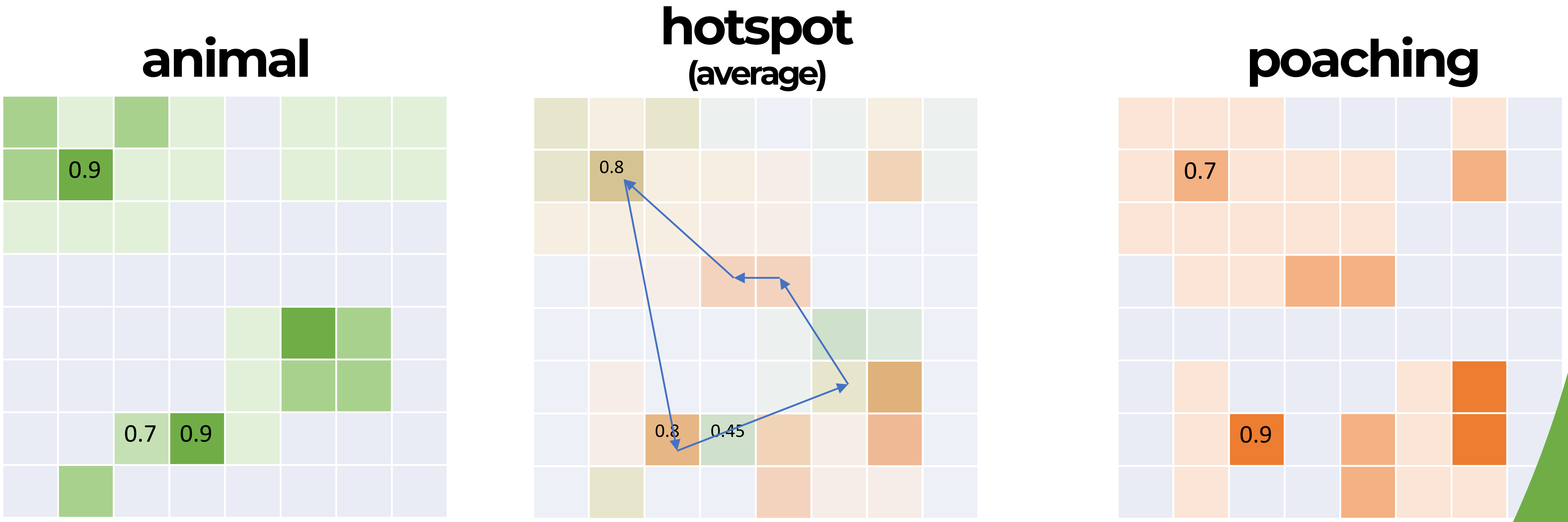
Static routes

- Savings algorithm
- Assume greatest savings (should) yield greatest number of nodes
- Works in polynomial time. Tested with 500+ nodes

- Savings = a + b – c
- But apply additional weights, like nodes’ value
- We also use asymmetry, adjacency and demand heuristics
- Two-opt algorithm is then applied to result to yield better results

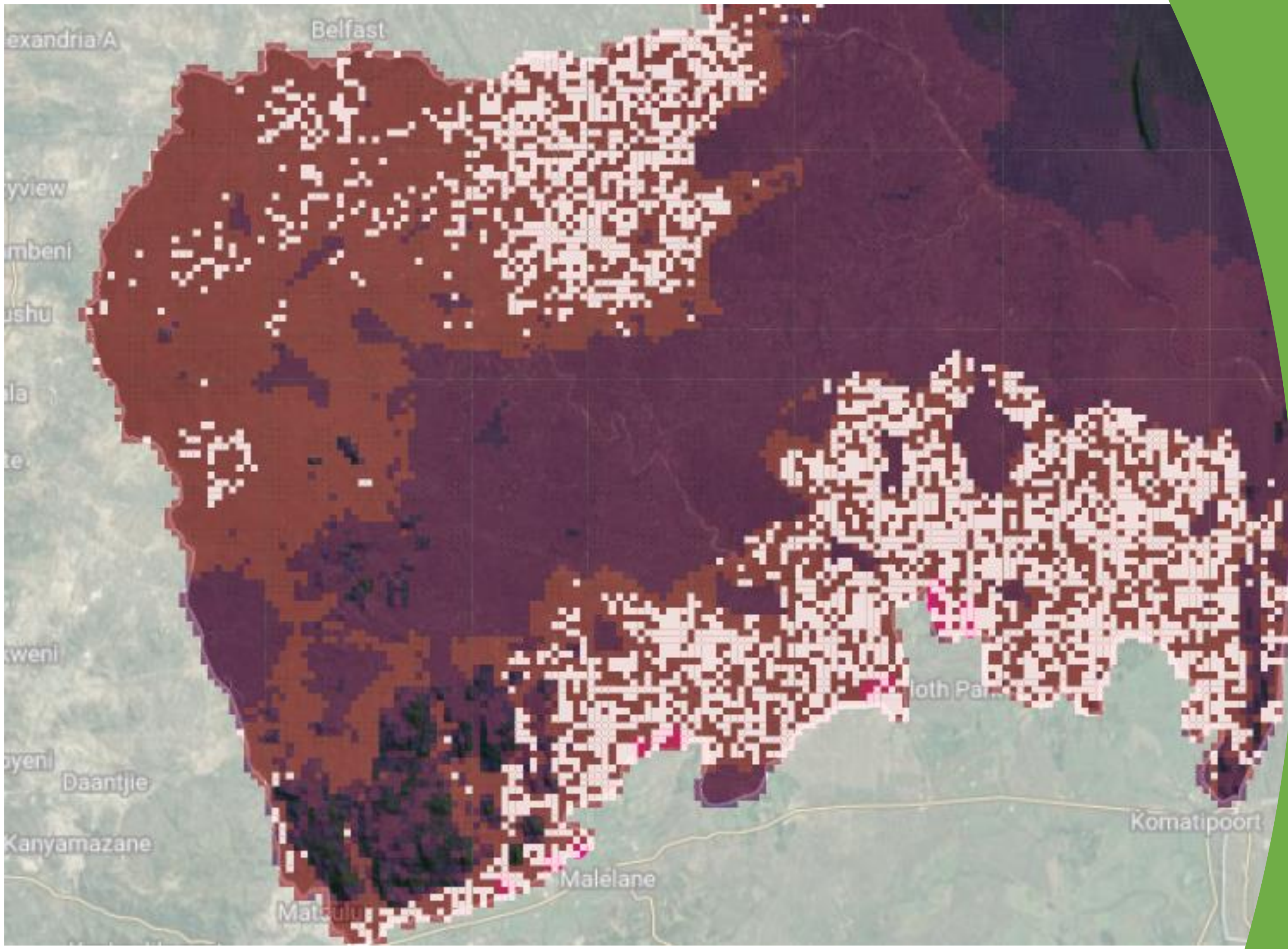
Static routes

- Visit cells which have a high priority of poaching and animals
- "Hotspots" are identified by taking weighted average



Hot spots

- Prominent outliers of high poaching + animal cell weight
- Cells which have not been visited recently are prioritised
- 5000 randomly selected every 2 hours (random routes)

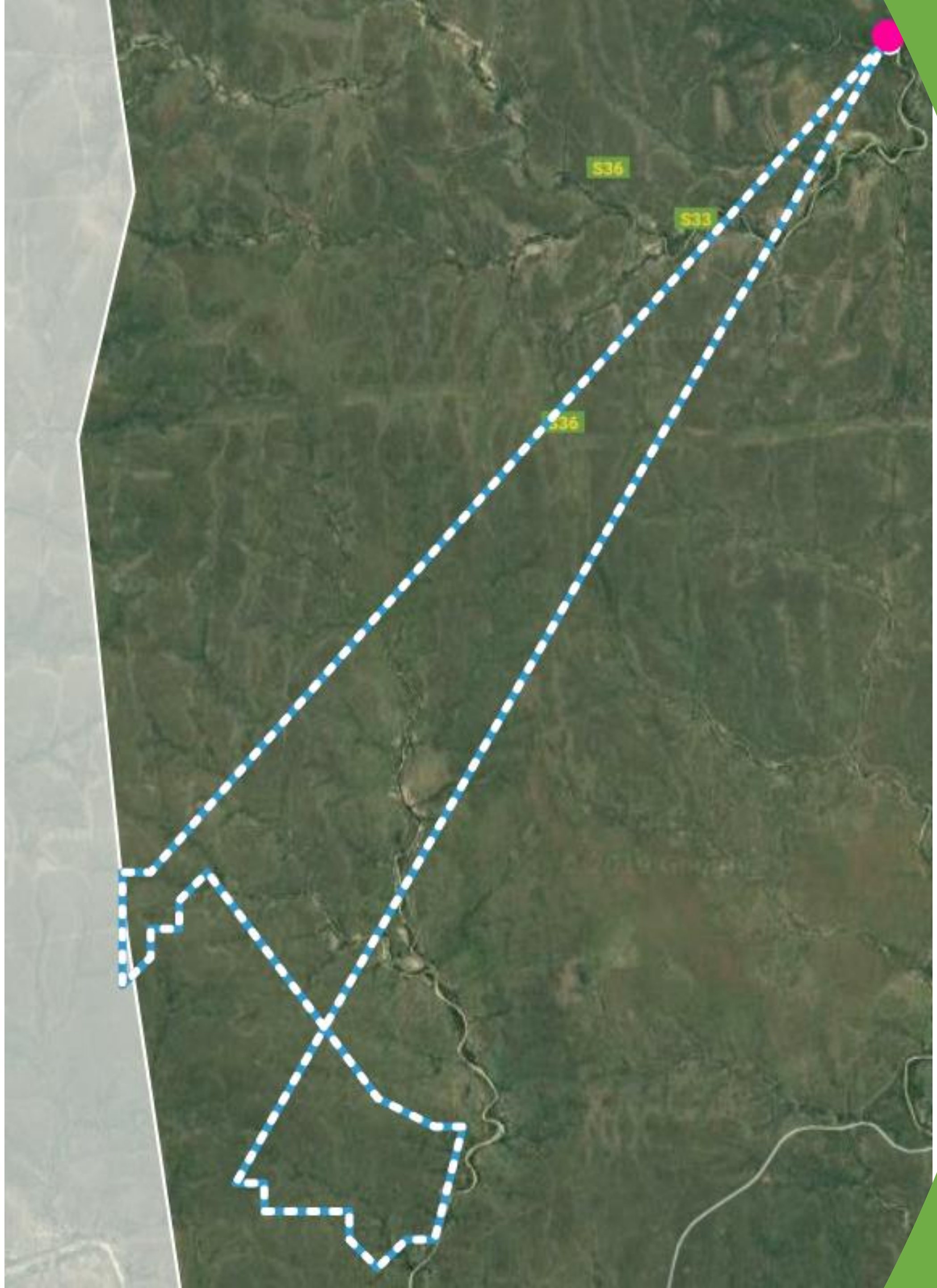


Static routes

- Vehicle Routing Problem NP-Complete
- Clarke Wright savings heuristic
- All routes are restricted to flight time

Static routes

- Weights using adjacency, symmetry, demand
  - Ran all possible combinations at 0.1 step count to find best weights
- Near-optimal, calculated in only a few seconds

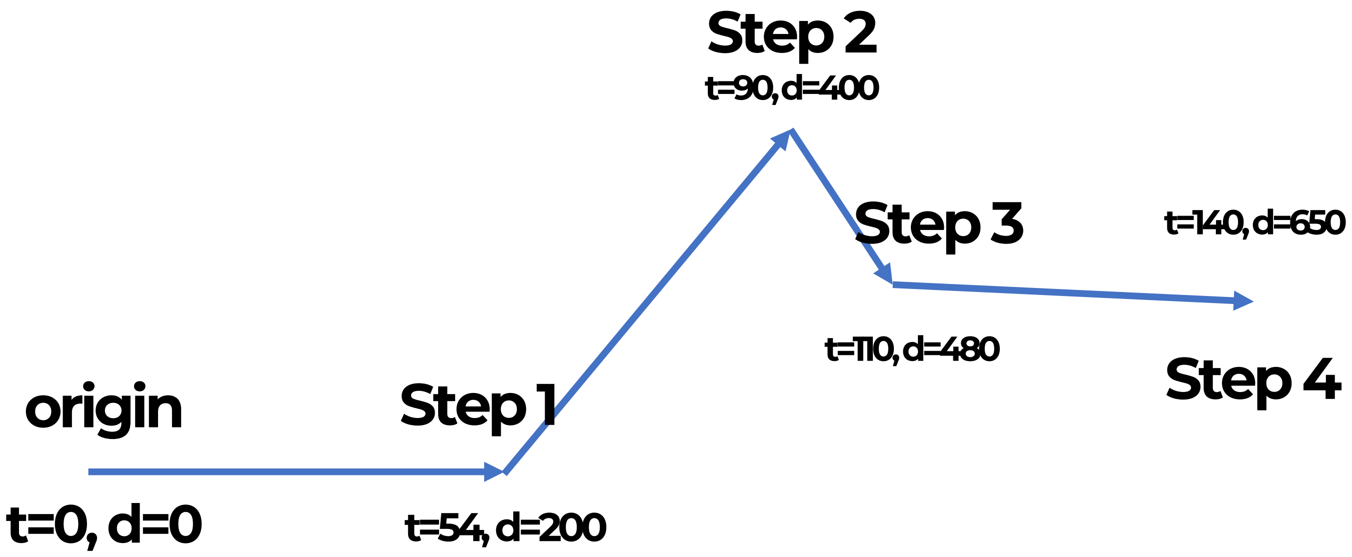


Dynamic routes

- Only use animal locations
- Attempt to intercept future location of individual animals
- We find the animal’s future position dependent on the drone flight duration

Dynamic routes

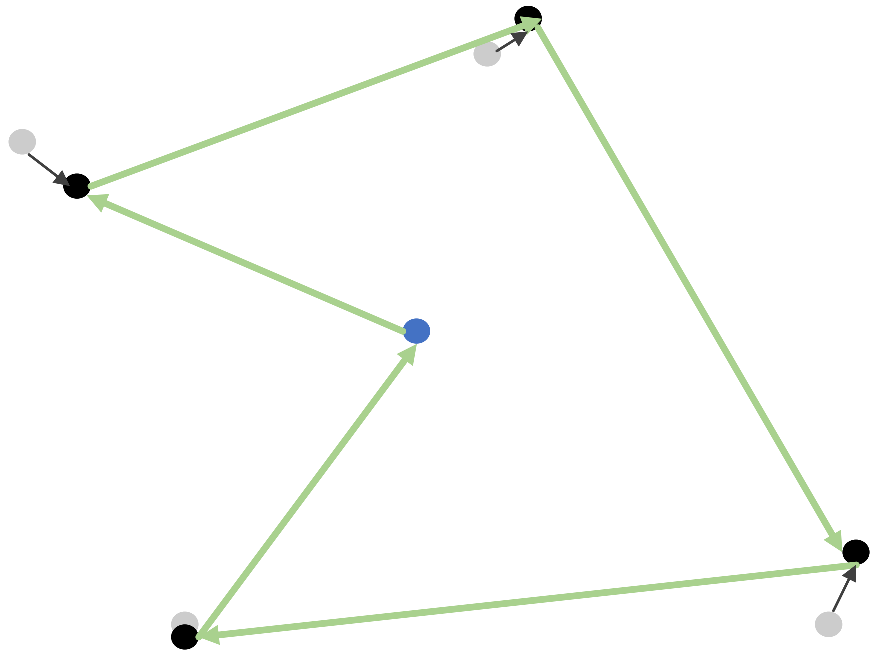
- Keep applying regressor to points until distance required reached
  - Distance = average speed of individual * flight time



- Points treated as functions
- $p = \left(d_x, d_y, t, v\right) \Rightarrow (x, y)$
- Coordinates returned based on drone position, speed and elapsed flight time
- Can find where the animal is along the path using variable “time”
(d = vt)

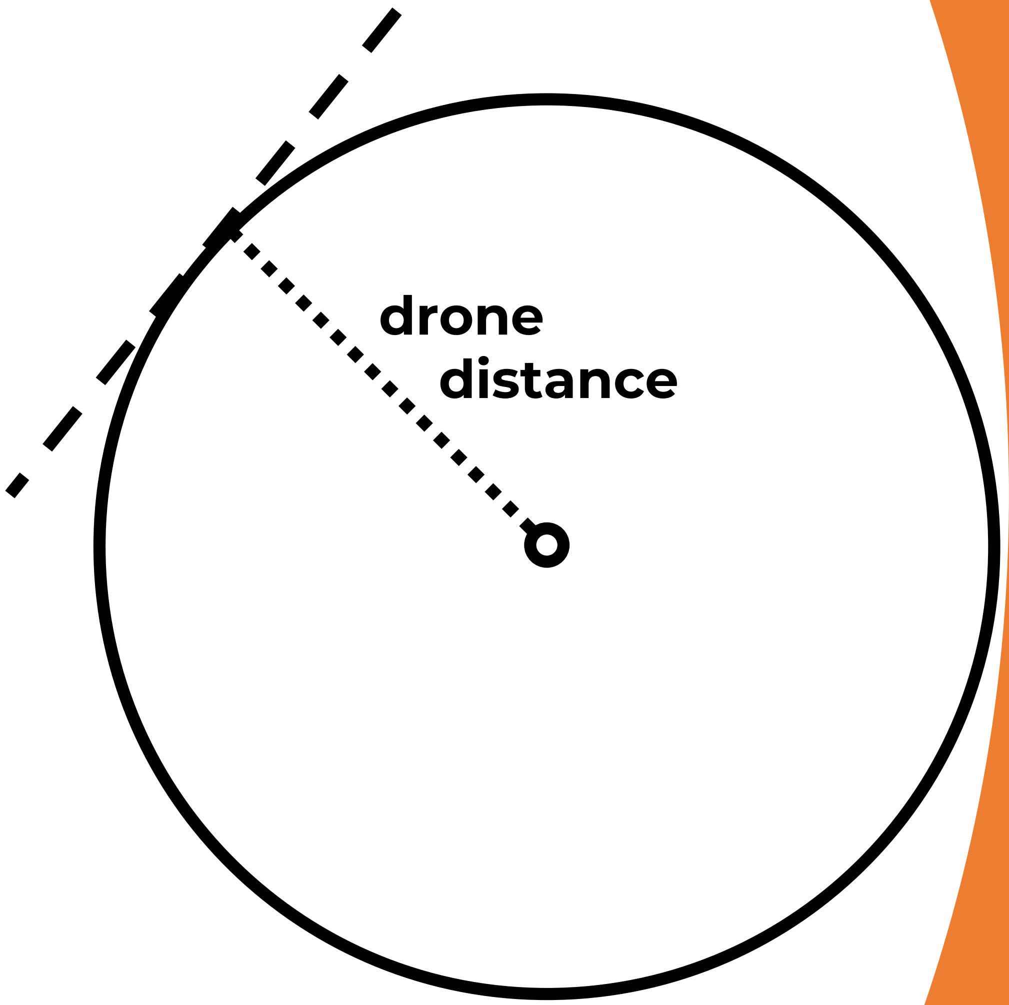
Dynamic routes

- Use predicted location at each step
- Keep calculating savings after each step
- Attempts all possible permutations
- Only works for few points (NP-hard)



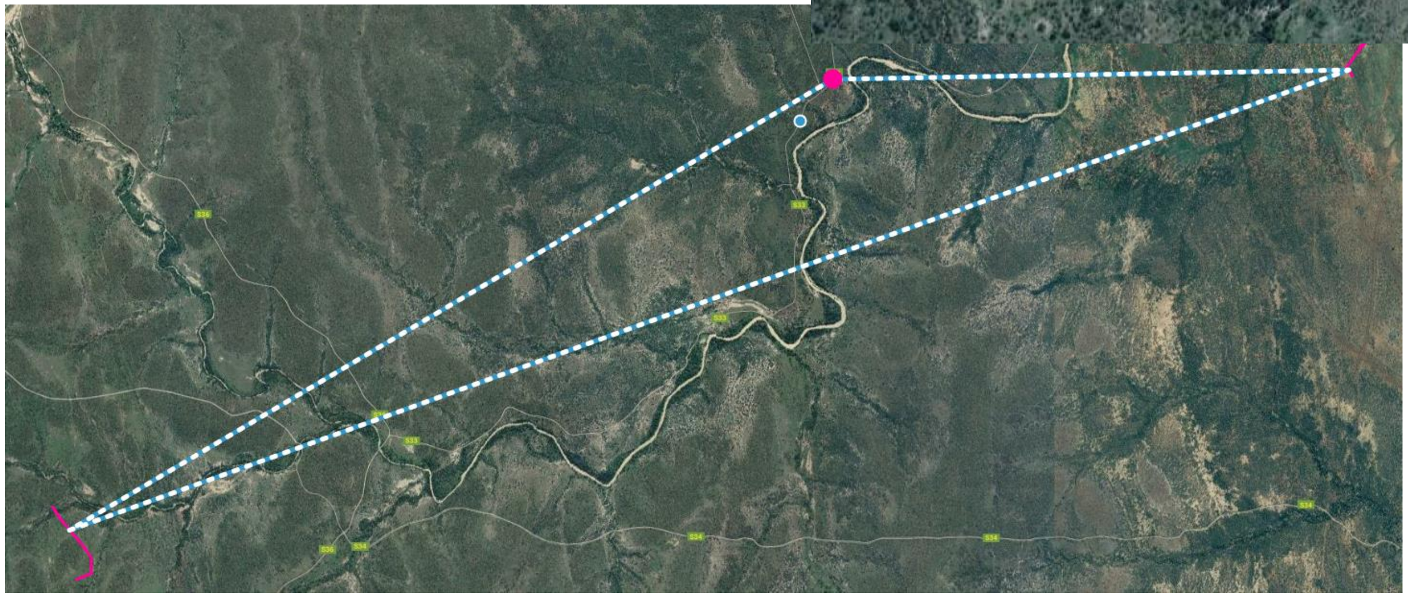
Dynamic routes

- Low number of nodes (<10) – we limit users to 5
- Intercepts paths
  - Grow circle of intercept from drone until circle meets line
- Attempts all possible combinations to find most optimal route



Dynamic routes

- Works for multi-line path segments
- Takes into account
  - Drone range
  - Drone speed
  - Animal speed
  - Animal path



Randomising routes

- Savings algorithm generates multiple routes
- Each returns to depot once fuel runs out
- If we start the routes at random edges, we get multiple random routes
- We then pick one of the routes at random
- Always take an efficient, feasible route
- Not able to predict which route might be taken

Conclusion

- Decisions based on data rather than human intuition:
  - Predict animal habitats
  - Predict poaching areas
  - Predict future animal locations
- Flight routes:
  - More likely to be closer to optimal than guessing
  - Can accurately intercept moving targets
- Expandable:
  - Can be integrated with other systems
  - Algorithms can be applied to other areas