WolfSim: An Agent-based Simulation for Improving the Tracking and Monitoring of Wolf Packs by Selecting the Optimal Tracking Systems

Group 16

Caleb Harris

Johnie Sublett

GitHub Link: https://github.gatech.edu/charris92/Wolfsim

Table of Contents

Table of Figures

Abstract

Our team has been tasked by the Alaska Department of Fish and Game, working in conjunction with the Alaska Department of Transportation and the Anchorage Port Authority, to assess various tracking options for wolf populations inside an area of interest surrounding Anchorage, including the majority of the Alaska Range and segments of the Wrangell and Chugach Mountains. More specifically, this area spans a rectangular area 200 miles north, east, and west of the city of Anchorage, as well as 100 miles to the South. The main objective is to assess wolf-tracking methods under varying environmental factors and communications constraints to better assist in these conservation efforts. A simulation environment is created to evaluate the current tracking methods that use radio collars, satellite collars, or aerial spotting. Each method has its advantages and disadvantages that should be taken into consideration such as range, error, and consistency.

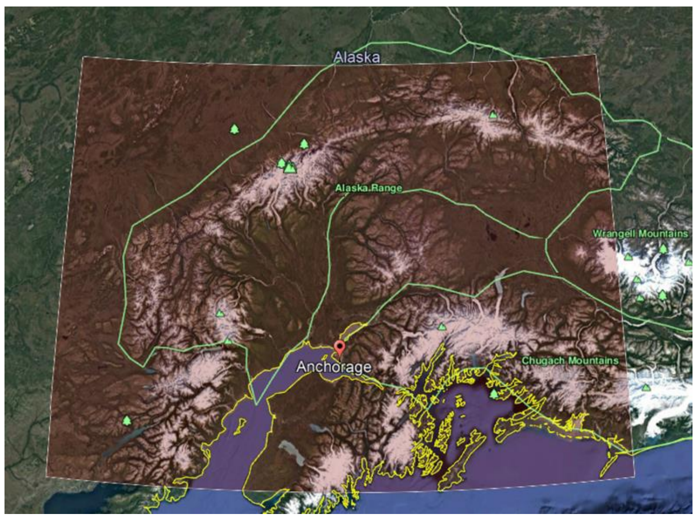


Figure 1 Map of the area of interest.

The tracking of wolves has been evaluated for many years, often using datasets from researchers such as in the work by Jedrzejewski et al [4]. Additionally, the documentation of wolf movement for researchers has been provided by sources such as the International Wolf Center, which can be seen in the references [1] [2] [5]. However, further analysis is needed to explore the different techniques, and potential future techniques, that could improve the conservation and understanding of these animals. Therefore, the simulation environment created in this works seeks to improve the capabilities of researchers that study wolves in this way by evaluating the detection and tracking methods. In addition to the goals of the wolf researchers, the Department of Fish and Game has two aims: conservation planning and conflict reduction. Territories for individual wolfpacks in Alaska can cover nearly 2500 mi^2[7]; by better understanding the historical distribution and frequency of pack activities, state-owned land can be better earmarked and prioritized for conservation designations. Real-time tracking may also have its benefits, as potential conflict between wolves and recreational or commercial activity in the area can be predicted and remediated. The agent-based simulation environment developed in this work involves assessing the effectiveness various technologies in tracking a single pack of wolves in the region of interest. Available data and documented behavior from relevant sources are used to model the movement of the wolves around the environment from feeding zone to feeding zone. The metrics of interest are the total tracking error for each of the tracking techniques, the overall health of the wolves based on tuning parameters, and the emergent behaviors of the animals based on the distribution of terrain and vegetation.

Project Description

System of Interest

The simulation models a region where a pack of wolves feed and explore. Their movement is modeled after available data, and the tracking of the wolves is limited by distance and environment conditions. The communication system capabilities and tracking uplinks will be considered. A series of assumptions are made based on available literature and expert knowledge.

- The region of interest is an area of land in Alaska where wolves roam and eat.
- The wolves move together in a pack but make decisions on moving as individuals without communication (decentralized movement).
- The locations of the wolves are detectable by one of the three tracking technologies.

For tracking platforms, several land and air-based options are being considered, including resources leveraged from local, state, and federal resources. With extensive experience in coastal tracking, the Marine Exchange of Alaska (MXAK) has offered several of their maritime Automatic Identification System (AIS) and Marine Safety Stations to be repurposed as stationary transmitters and receivers in the Alaska Range, as well as integration with their existing coastal tracking network.

Our first air-based tracking option leverages both state and federal assets, namely US Army aviation assets and Alaska Department of Transportation Division of Statewide Aviation resources. More specifically, military aircraft on routine training flights between Fort Wainwright, Fairbanks, AK, and Joint Base Elmendorf-Richardson, Anchorage, AK, as well as state-operated supply planes on their routine trips from Anchorage to Nome, AK and Juneau, AK, will be outfitted with tracking receivers to carry on their routine flights.

Another option considered for Wolf tracking is satellite service providers, with Ground Control Alaska as a prime candidate wither their Anchorage-centric service coverage. However, due to the bulkier collars and dense landcover in our area-of-interest, as well as the high orbital inclination required for tracking providers, limited effective coverage may result.

Finally, as a more traditional approach, we consider direct tracking measures through contracting with Native Range Capture Services, our current Wolf darting and collaring partner in the region. Their fleet includes several Robinson R44 helicopters; with their 344-mile range, these aircraft can reach most hunting areas in the Anchorage wilderness area and provide a more targeted approach to wolf-tracking measures.

Figure 2: Tracking Techniques and Technology

Figure 3: Area of Interest and Detection Sources

Literature Review

Wolf Behavior

In the paper by Muro et al [3], the movement of wolves around prey is defined by a set of simple rules where each individual wants to move toward prey until reaching a safe distance, where then the wolf moves away from the other wolves in the area. Therefore, the model of wolf movement can be defined as an emergent behavior where individual agents are given behaviors and global communication is not required. It has been documented in reports that wolves have attacked their own during territorial disputes. The story of wolf 381 in [5] details how the seven-year-old wolf is eventually attacked and killed by another wolf. This is detected by the radio collar when the wolf does not move for greater than 4 hours. In addition, [5] details that a percentage of wolves are killed every year over territorial conflicts in what is called “intraspecific strife”. From the tracking research by Jedrzejewski et al in [4], we learn that the movement of adult wolves differ by gender, time of year, environmental conditions, or mating. The average daily range was 21.4 km^2, which was about 9% of the total territory, however the mean straight-line distance between these locations was only 4.4 km.

Wolf Tracking

The online documentation by the International Wolf Center in [1] details how the radio-tracking of wolves has been used in the past to provide information on the movement, home territory, and dispersion of the wolves. Advanced methods by Downs et al in [6] demonstrate how the home range estimates of the wolves can best be modeled by utilizing kernel density estimates on the detections of the wolves to create a continuous probability density surface. The use of a Kernel Density Estimation with Delaunay Triangulation for approximating movements could be a powerful method to improve the tracking of the wolf agents.

Conceptual Model

The simulation environment is built around an agent-based simulation which defines a set of agents, an environment, and agent behaviors. We use available data such as technical specifications for wolf tracking collars, layouts of habitats and movement of wolf packs [1] [2], power and communication constraints/limitations for GPS, radio, and cellular tracking systems.   investigate the up-time and coverage of different wolf-tracking systems that use collars on the wolves. Additional analysis will analyze the reliability to maintain tracking of the entire wolfpack when there is limited or missing data.

Movement

We implement day-long time steps to move and the movement is based on feeding zones. Feeding zones are a function of vegetation in the environment. The wolf moves towards a target with probability of 44%, move towards average position of group with 55%, and move randomly with 1%. This technique was inspired by the individual behaviors from [3]. The total distance traveled is around 4 km per day [4].

Wolf pathing behavior is modeled through the generation of a sparse undirected graph, with nodes representing positions a wolf can inhabit, and edges representing adjacent spaces to move. Weighting for each edge is based on the change in elevation between the adjacent spots in the graph, and the average forest cover of the movement area. To then determine the path from the current wolf location to the food source, Dijkstra's algorithm or a similar shortest path algorithm implementation can be used to find efficient paths between feeding locations. In this way, the wolves will prefer to move along elevation contour lines and across open wilderness, not unlike how wolves, bears, and other Alaskan wildlife will repeatedly follow consistent traverses and game trails as they navigate their habitats.

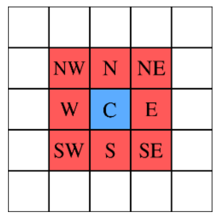


Figure 4: Wolf neighbor positions

The path for a wolf to track is found using Dijkstra's Algorithm [8]. Dijkstra's algorithm finds the optimal path between two points using an efficient search method that guarantees completeness and optimality. As shown in Figure 5, networkx is used to generate a lattice grid graph, or more simply a graph where edges between nodes only occur between adjacent grid spaces. The weights for those edges are then applied using the following equation:

$$Weight = |\Delta elevation| + k * (treecover)$$

```
def compute_updated_target_pathing(model, plot=False):
    agents_pos = [agent.pos for agent in model.schedule.agents if agent.alive]
    current_waypoint = model.path[0]
    feeding_site = model.target
    distance = np.linalg.norm(np.average(np.array(agents_pos), axis=0) -
np.array(current_waypoint))
    if distance <= 5.0 :
        if len(model.path) == 1: # reached target position
            model.feeding = model.feeding + 1
            if model.feeding >= 5:
                new_target = np.random.randint(0,len(model.sites)-1)
                source = np.average(np.array(agents_pos), axis=0).astype(np.int)
                source = tuple(source)
                target = model.sites[new_target]
                # Find shortest path for wolfpack travel using
                # new_path =
nx.algorithms.shortest_paths.generic.shortest_path(model.G, source=source,
target=target,
                #
weight='weight')
                new_path =
nx.algorithms.shortest_paths.weighted.dijkstra_path(model.G, source=source,
target=target,

weight='weight')
                if plot:
                    x, y = zip(*new_path)
                    print(y)
                    plt.scatter(y, x, marker='o')
                    plt.gca().invert_yaxis()
                    plt.imshow(model.tree, cmap='summer', interpolation='nearest',
alpha=.5)
                    plt.imshow(model.grid_elevation, cmap='hot',
interpolation='nearest')
                    plt.show()
            else:
                new_target = feeding_site
                new_path = model.path
        else:
            model.path.pop(0)
            new_path = model.path
            new_target = feeding_site
    else:
        model.feeding = 0
        new_target = feeding_site
        new_path = model.path

    return new_target, new_path
```

Figure 5 Path-finding code

The resulting weighting concept is shown in Figure 6, with squares with minimum elevation change and no vegetation present having the lower weight.

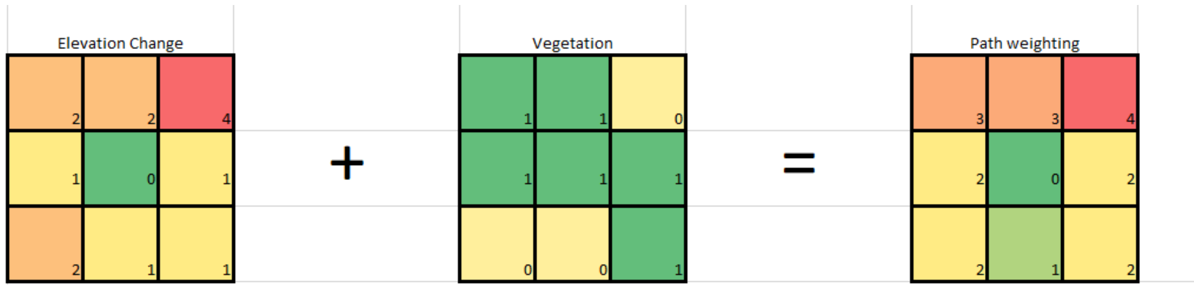


Figure 6 Wolf Path Weighting

The graph structure is generated for the entirety of the area of interest, pulling elevation and tree cover for all spaces in the grid, which are then represented by nodes in the lattice graph; this data can be seen in Figure 7, with the left figure showing forested areas, while especially mountainous areas are shown brighter in the right figure. An additional penalty function is added areas within the nearby bay and inlets, as otherwise the weighting would prefer swimming across hundreds of miles of flat ocean - a behavior not seen in wolf populations.

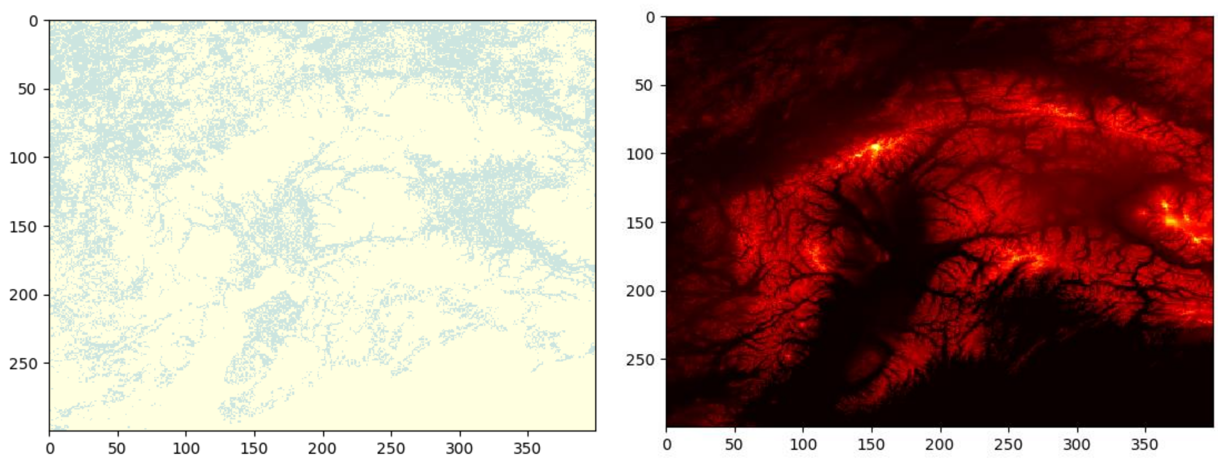


Figure 7 Tree Cover and Elevation Datasets

Overall, the pathing search algorithm tends to match travel through areas with near-constant elevation and minimal treecover, as further discussed in the following Input Validation section.

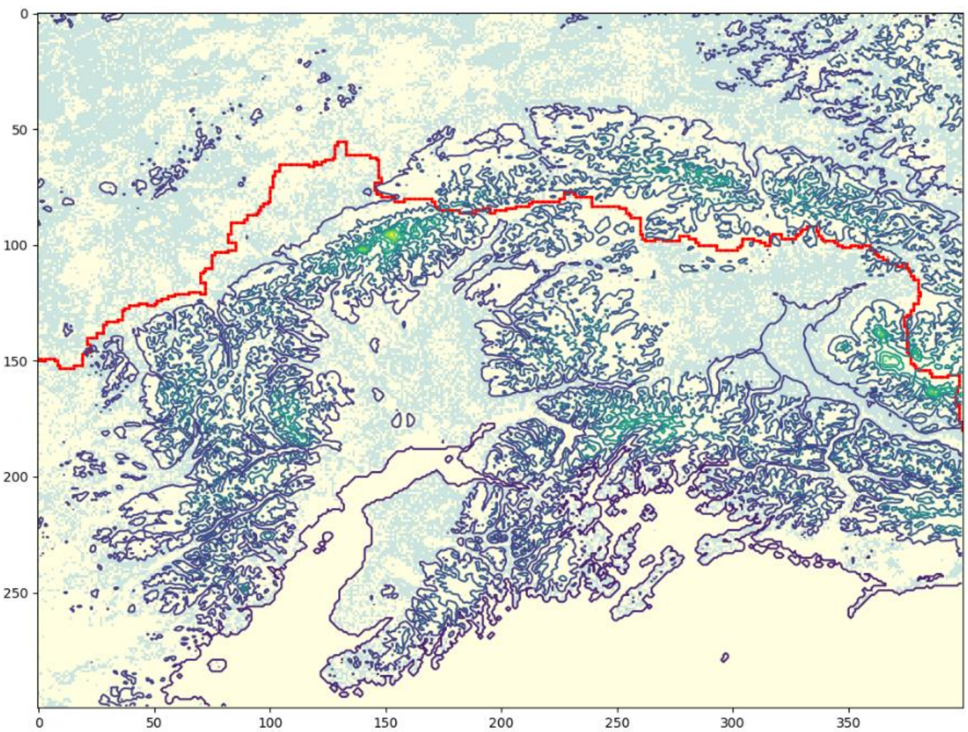


Figure 8 Wolf Pathing Follows Contours

Detection

For each simulation run, a select number of wolves in the model pack will be affixed with either radio or GPS tracking collars. A selected tracking system is employed for the simulation and assessed on its ability to maintain a positional inference on the pack through successful tracking pings. Collared wolves can die or their collars become damaged, especially for the bulkier GPS collars used for the satellite system. Elevation and forest cover can also affect the probability of a successful tracking ping for most systems.

The aviation and tracking station assets can ping any collars within their respective tracking ranges; however, the probability of a successful tracking ping is modeled to be inversely proportional to distance between the tracking asset and the collared wolf, with elevation and forest cover also contributing to tracking effectiveness. Although the MXAK stations are static deployments, the air assets are able to move across the simulation area and provide detection opportunities over larger ranges. Commercial and DOD flights with affixed ride-along receivers provide detection swaths along standard supply and training flightpaths. For the more targeted detection scenario, the Native Range Capture Services' R44 helicopters are deployed and fly to one or more known wolf feeding areas within the area-of-interest. Finally, satellite coverage allows for continuous tracking opportunities across the entire range, albeit limited by larger power requirements, more damage-prone collars, and greater sensitivity to environmental interference.

Tracking

Tracking uses the average position to store throughout the simulation time steps and then post data collection, we use a Kernel Density Estimation method as in [6]. Kernel Density Estimation is a non-parametric method to estimate the distribution of a random variable. In this case, we are using the random samples of agent position to generate the wolf pack home range.

The kernel density estimator for a point, x, is defined as in Equation 1, where K is the kernel function and h is the bandwidth. An example of this technique is seen in [6] and in Figure 9.

$$f_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \quad (1)$$

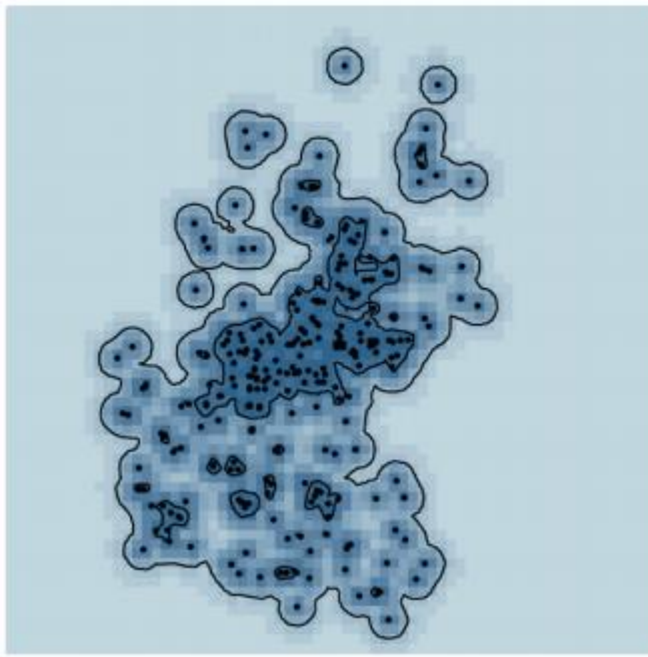


Figure 9: Kernel Density Estimation for Animal Home Range [6]

Simulation Fundamentals

Our simulation is first initialized with the geographical area-of-interest as potential location mesh. Each of these habitat locations has attributes for its elevation and forest cover. These parameters are used in building a movement graph, as well as determining the effectiveness of tracking systems in later timesteps. The wolf pack and food sources are initialized, and the wolves generate their first path to food sources. Air, land, or space-based detection assets are also initialized.

During the first timestep, the wolf pack makes its move towards the currently targeted source of food. Wolves or collars may also die at this point. Any airborne tracking assets update their position as well. Any tracking systems with collared, living wolves within their detection range will then attempt to ping those collars. The locations of successful pings, as well as the true location of all living wolves, is stored for the timestamp. The next timestamp can then occur.

Input Analysis

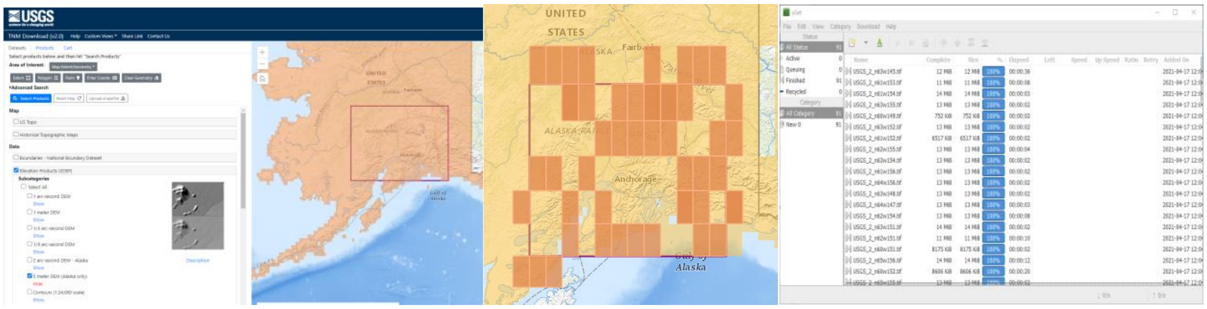


Figure 10 UGSG LandSat Data Collection process (Map extent, Elevation product search, UGet download interface)

All elevation and landcover data was collected from the USGS NationalMap interface.[10] The area of interest is first selected, and then the 3-meter DEM elevation product is searched in the corresponding map area and added to the sourcing cart, which generates an image listing all desired GeoTiFFs in the search area. This document-list text file can then be used with UGet download management software to acquire the 78 data files.

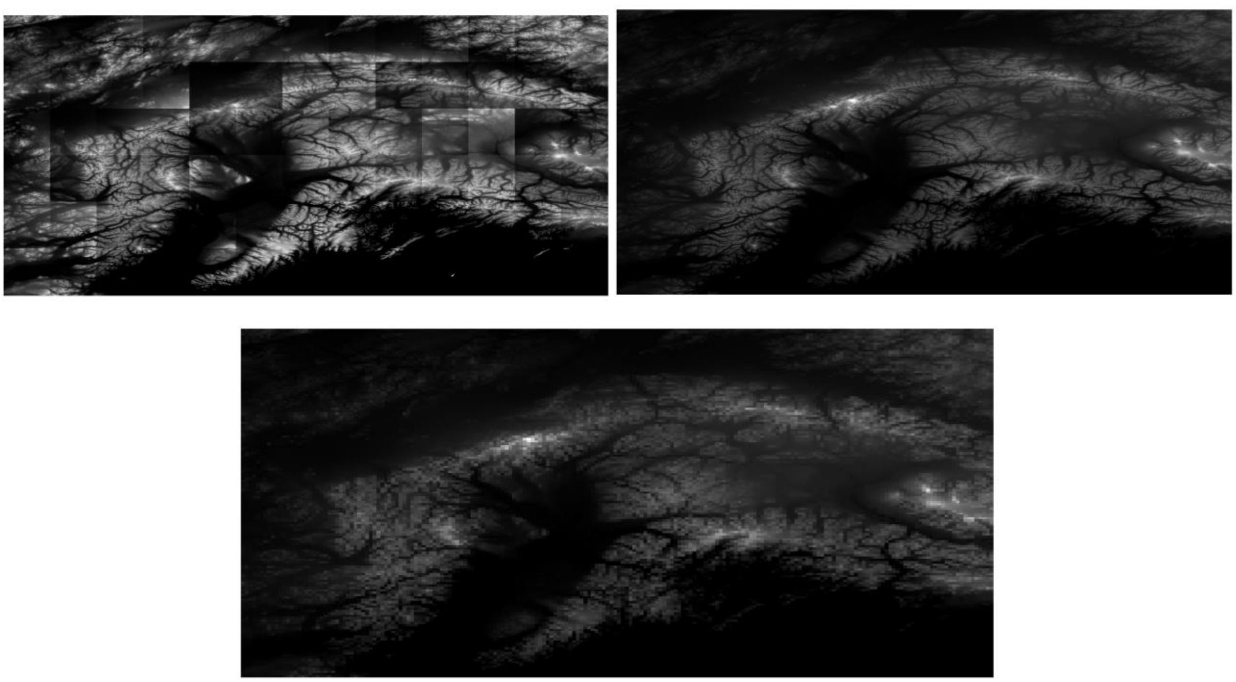


Figure 11 GIS Merged Dataset, vectorization, and downscaled rasterization

However, the elevation data for the interest area is separated into 78 individual files, and has significant border artifacts, as shown in Figure 11. Using QGIS, these 78 individual images were merged, as shown on the right. Then, a rasterization calculation was made on the GIS shape file, which converted the vector shape to an array of interpreted elevation values, with dimensions matching a two-mile grid within the 300x400 simulation area.

```
ncols        200
nrows        150
xllcorner    -156.003330000000
yllcorner    58.996670000000
dx           0.065033300000
dy           0.040044400000
NODATA_value -32768
260 495 352 506 411 421 611 781 577 303 217 250 231 272 551 762 749 859 378 202 129
```

Figure 12: Elevation File Headers and Portion of Elevation Data

For the final QGIS step, the rasterization is converted to an Arc/Info ASCII Grid, as shown in Figure 12.

Verification and Validation

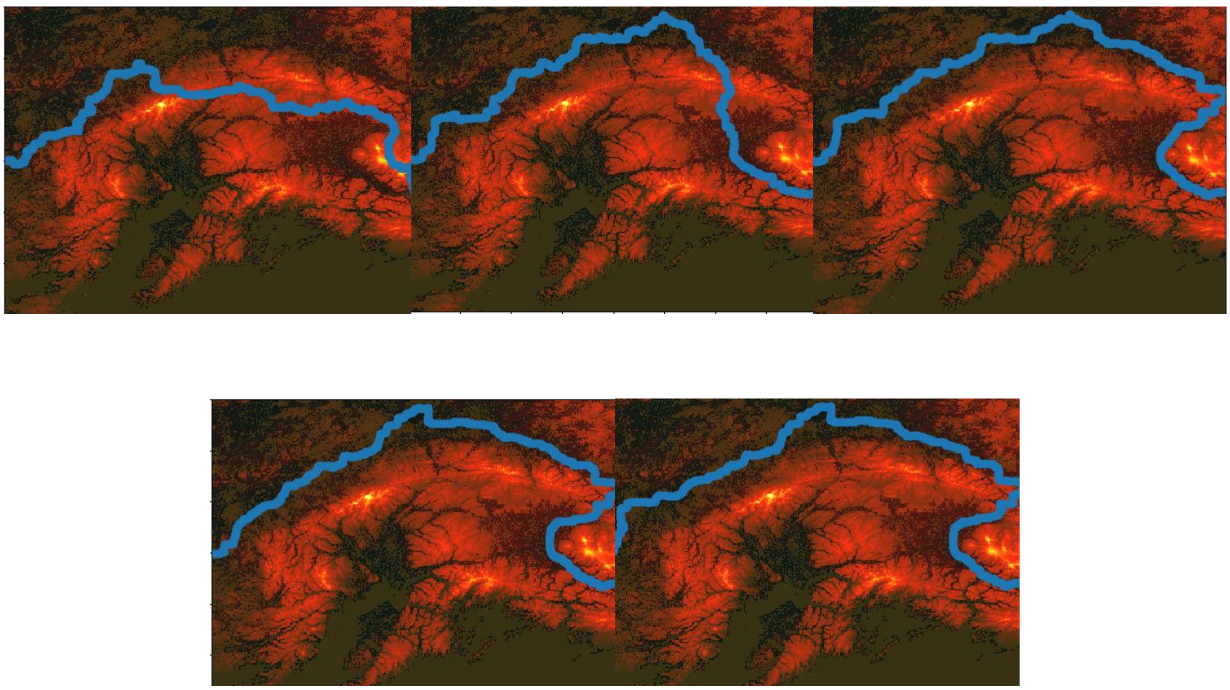


Figure 13: Evolution of wolf pathing (Top left: Greater Tree avoidance; Bottom Right: Greater Elevation-change avoidance)

Using the two-stage weighting scheme also allows for the tailoring of wolf pathing behavior to real-world data. For example, using wolf den locations collected from Denali National Park and Preserve[9] shows how closely pathing choices coincide with known habitats. As shown in Figure 14, wolves will congregate on the northern reaches of Denali to make their dens and raise cubs; thus, pathing decisions less avoidant of elevation more closely match real-world data.

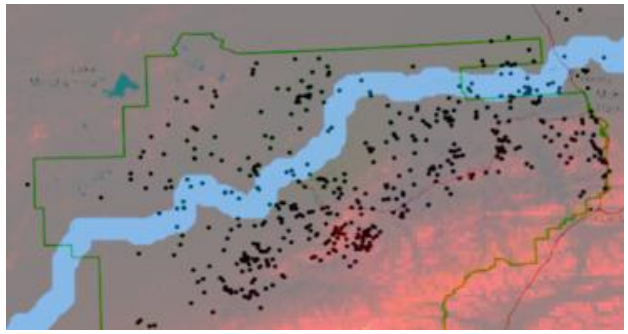


Figure 14 Wolf Dens within Denali National Park

The home range of the wolves is estimated using a kernel density estimation method as described in previous works [6] [11]. To validate the current implementation, the wolf distributions were compared to that in [11], seen in Figure 15.

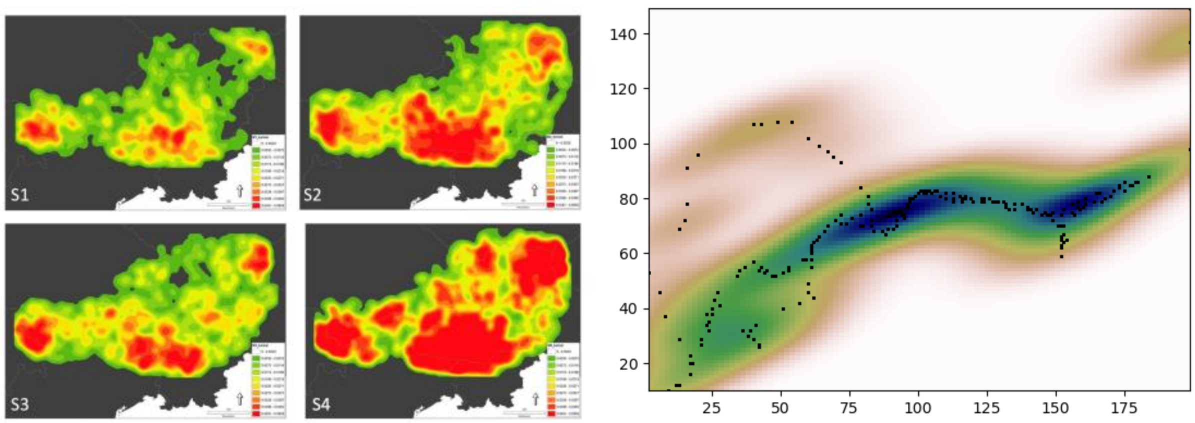


Figure 15: Kernel Density Estimation of Wolfpack Movement from [11] and from WolfSim

Project Results

Agent Behavior

Wolf Behavior

The wolf agents move throughout the environment as detailed in the previous section and can be viewed in the following figures. We keep track of the agent position at each time step, representing a single day.

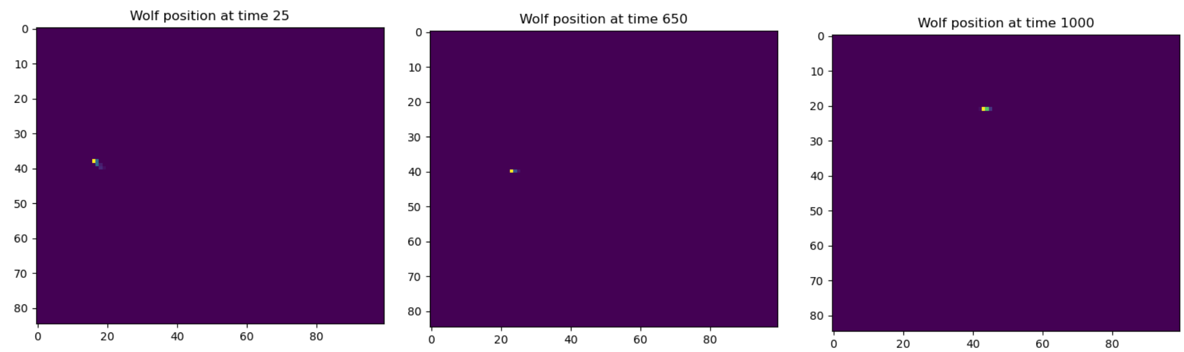


Figure 16: Wolf Movement through Environment Tracking Feeding Locations

The individual wolf movement is based on a set of likelihood for three different move actions: random, group, target. In the random action, the agent moves to a random neighbor cell. In the group action, the agent moves towards the average position of the living wolf pack. In the target action, the agent moves towards the feeding zone target. The optimal path is recalculated whenever the agent completes the destination to a feeding site.

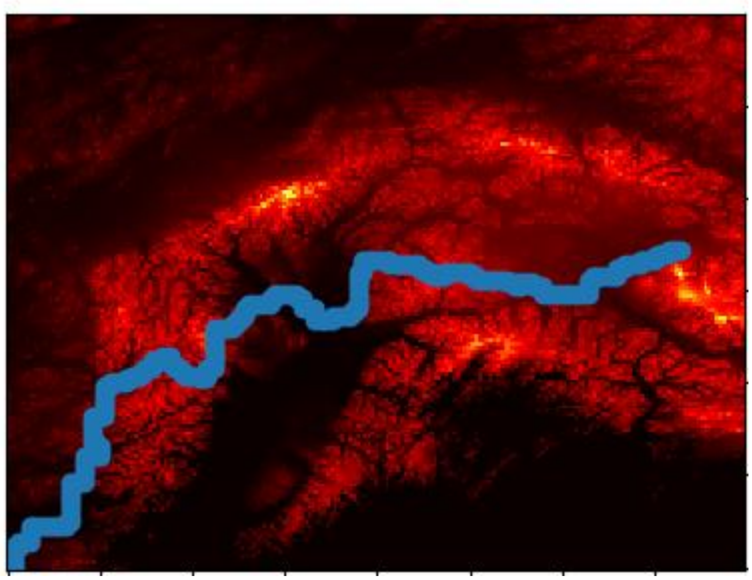


Figure 17: Optimal Path through Elevation and Vegetation for Wolfpack

Wolves will die from the environment with a small probability. In addition, after the wolf has reached a mature adult age, it will begin to randomly attack nearby wolves and cause death with some probability.

Detection Behavior

The visualization can be used to demonstrate how the detection methods work. In the following figures, the wolf agents are the small block circles moving through the environment where green circles are feeding zones, yellow regions are detection agents, and dark black cells are inaccessible (water). In the images on the right in the following figures, the detection method is demonstrated by flashing a yellow region that represents the area that wolves can be detected and then the wolf agents will change to the color red.

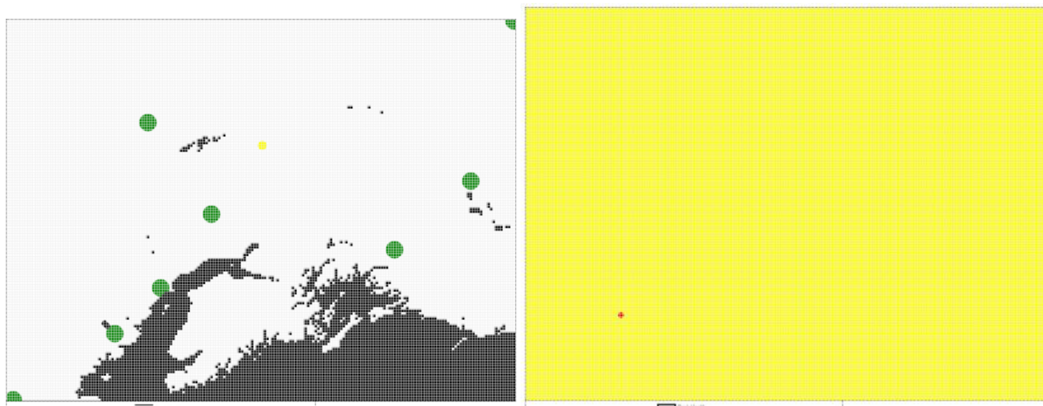


Figure 18: Wolf Detection from satellite

For satellite tracking, the entire map is pinged, with all collared, living wolves being potentially detected at a low probability (in this instance, 25%).

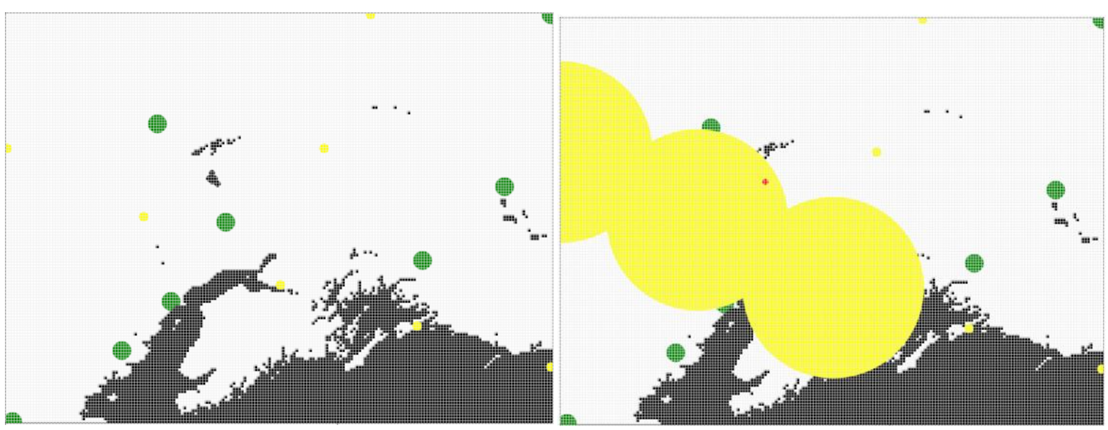


Figure 19: Wolf Detection from planes (Route #1)

For the second detection method, three plane paths are used, specifically flights from Anchorage to the destinations Juneau, Nome, and Fairbanks, with three detection pings occurring during each individual flight, and flight destinations cycling through the three options. The probability of a successful tracking ping is then calculated based on the position of the wolf in relation to the tracking point. Note: although the visualization shows yellow tracking bubbles for each ping, it is possible for wolves to be detected outside this notional range, albeit with a much lower probability.

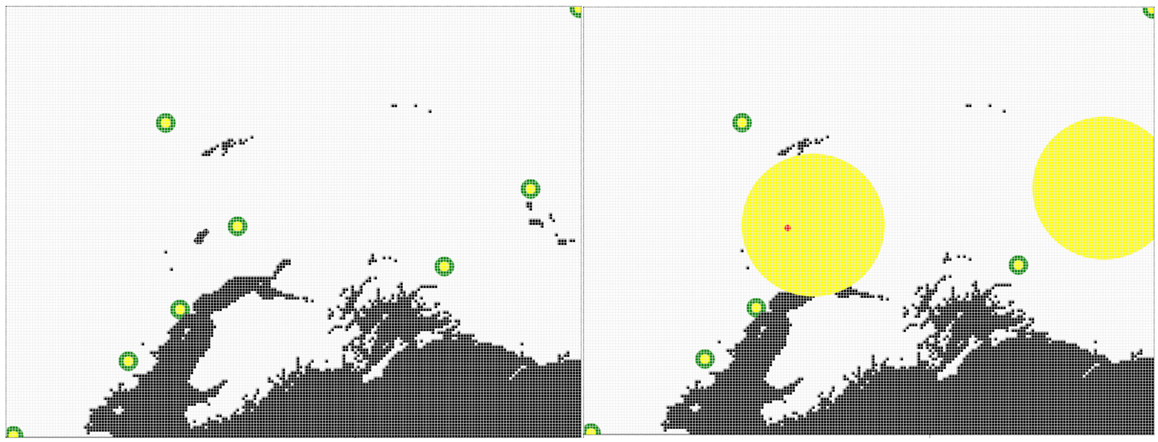


Figure 20: Wolf Detection from helicopters (Zones 5+6)

For the helicopter-based simulation, four pairs of tracking flights are designated; thus, in a single tracking ping, 2 feeding sites are chosen to be visited by the research helicopters, and nearby wolves may be detected.

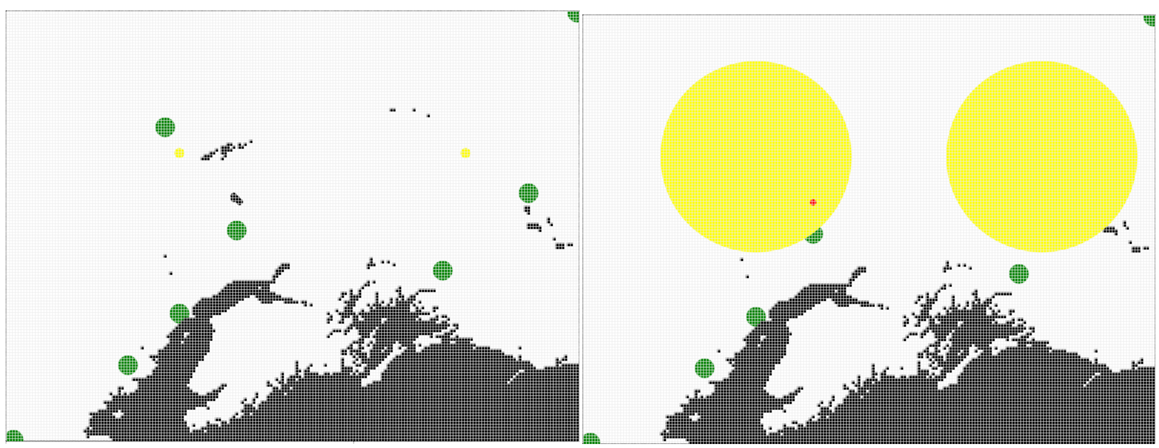


Figure 21: Wolf detection from radio tower stations

For the tracking station simulation run, centrally located stationary beacons are placed at high elevation, and provide coverage within the central range.

Data

The elevation and vegetation data can be visualized shown here. The elevation and vegetation were used to define the feeding zones, solve for the optimal paths, and for placing the detection zones.

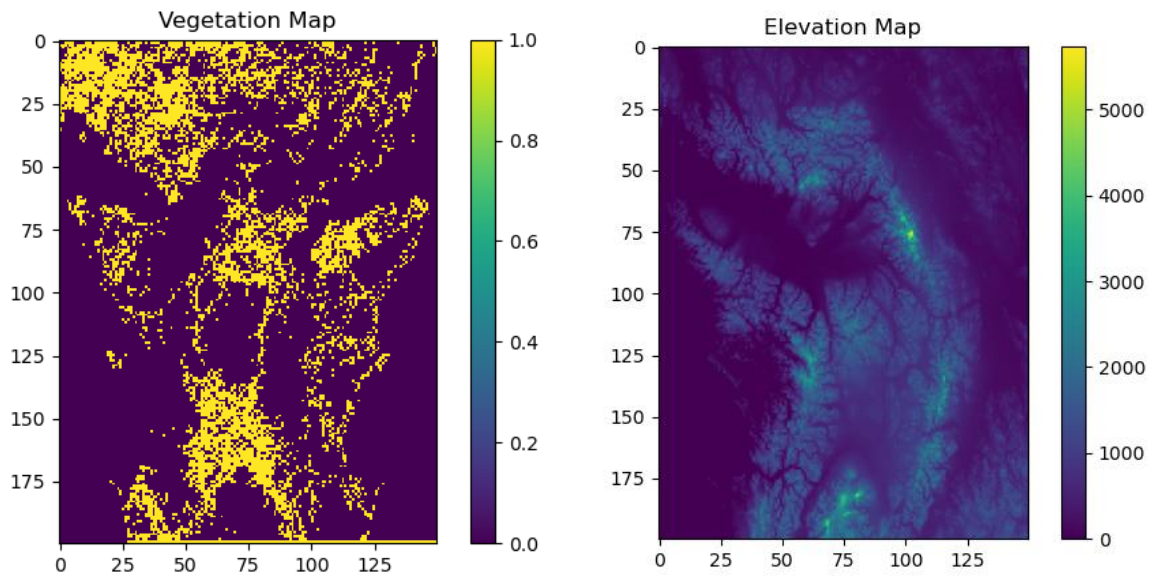


Figure 22: Feeding site location (Left) and Elevation data (Right)

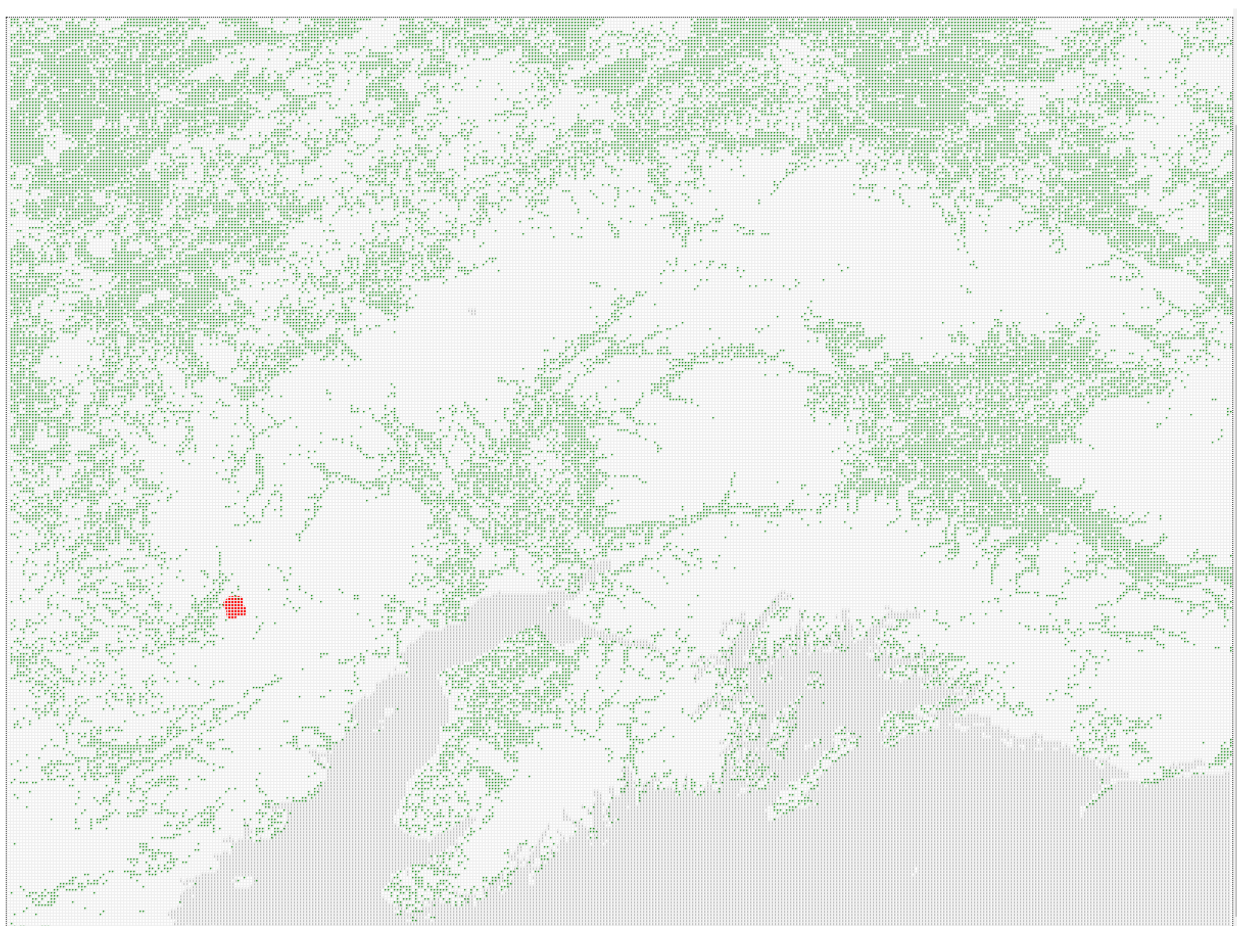
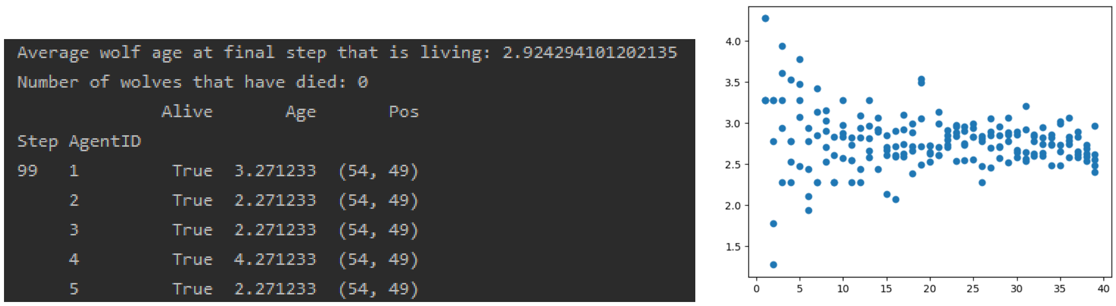


Figure 23: High resolution (400x300) view of Region with Vegetation and Water Boundaries

Analysis

Wolf Pack Health

For a single run of analysis, we can analyze the number of wolves that have died, age of wolves that are still living, and final position of living wolves. For batch runs we can collect statistics on the "pack health" (average age of living wolf). For example, the results can be seen for 200 total runs, by varying pack size from 1 – 40, and running 5 iterations of each.



```
Average wolf age at final step that is living: 2.924294101202135
Number of wolves that have died: 0
              Alive       Age       Pos
Step AgentID
99   1         True  3.271233  (54, 49)
     2         True  2.271233  (54, 49)
     3         True  2.271233  (54, 49)
     4         True  4.271233  (54, 49)
     5         True  2.271233  (54, 49)
```


Figure 24: Printed Results from a single run (Left) and Variation of Wolf Average Age with Changing Pack Size from batch run (Right)

Wolf Pack Home Range

As detailed in [6], the home range of a wolf pack can be estimated by using a kernel density distribution formed from the tracked positions of the wolves. Since we have multiple tracking method options, we can run analysis using each one to compare the success in predicting the home range of the wolf pack over 1 year and 5 years. The work here focuses on the qualitative comparison between the two using visual information. Future work will involve further quantitative analysis to compare the distributions and wolf conservation efforts.

The following images are plotted over the range of wolf movement for 1 and 5 years, respectively. The truth positions and the detected positions are shown side by side, with the colored heat maps indicating the probability value at those points on the map. The block dots indicate the data used to create the kernel density estimator, the left features all the points over each time step and the right features the tracked points which occur whenever the respective tracking method is successful at detecting an agent.

Satellite

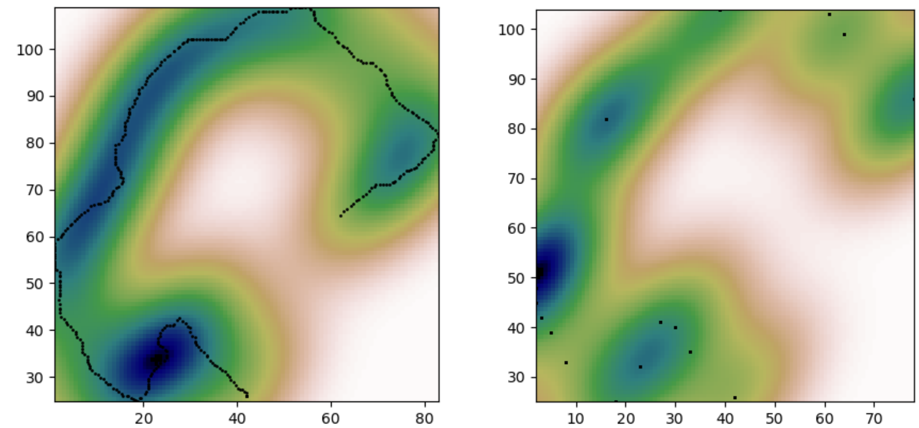


Figure 25: Home Range KDE for Truth Data (Left) and Satellite Detections (Right) for 1 year

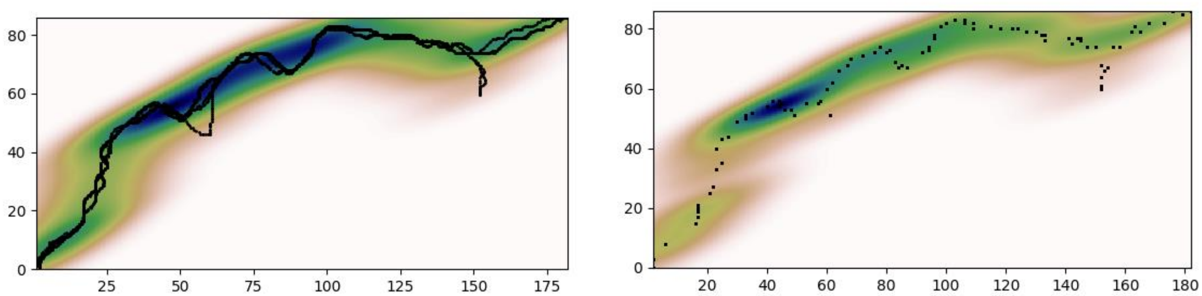


Figure 26: Home Range KDE for Truth Data (Left) and Satellite Detections (Right) for 5 years

Planes

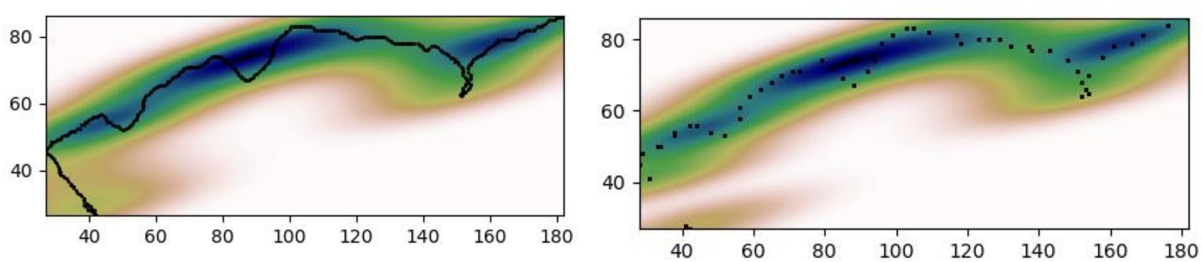


Figure 27: Home Range KDE for Truth Data (Left) and Planes Detections (Right) for 1 year

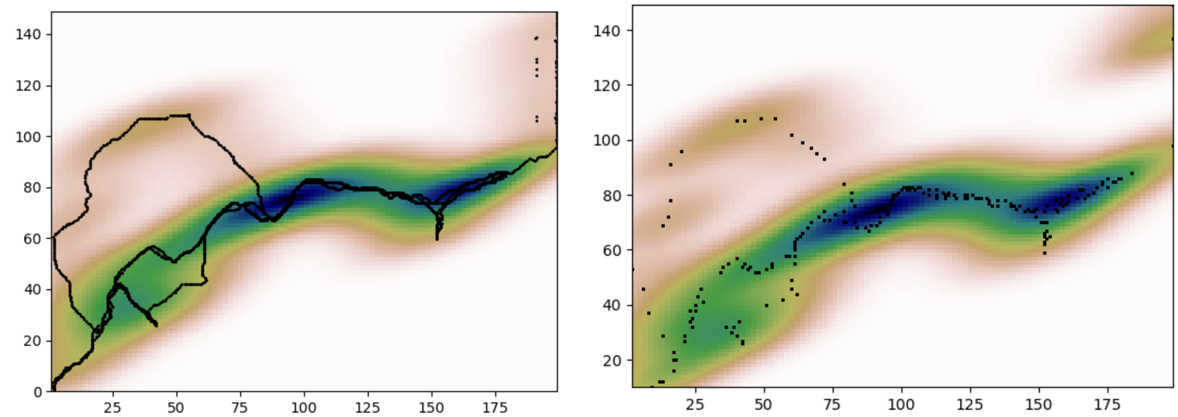


Figure 28: Home Range KDE for Truth Data (Left) and Planes Detections (Right) for 5 years

Helicopters

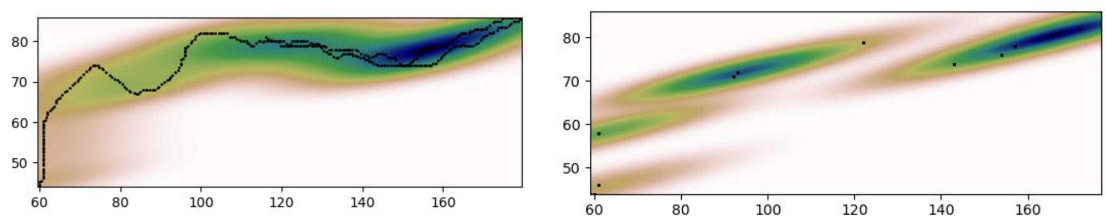


Figure 29: Home Range KDE for Truth Data (Left) and Helicopters Detections (Right) for 1 year

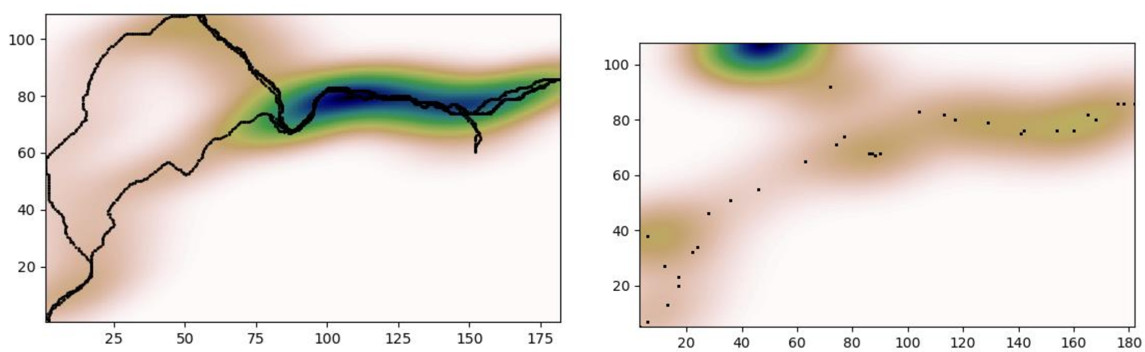


Figure 30: Home Range KDE for Truth Data (Left) and Helicopters Detections (Right) for 5 years

Stations

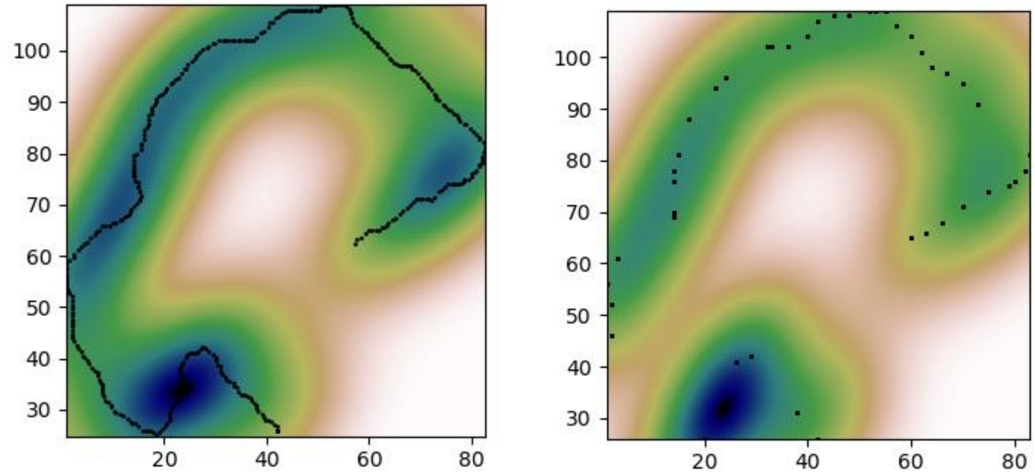


Figure 31: Home Range KDE for Truth Data (Left) and Stations Detections (Right) for 1 year

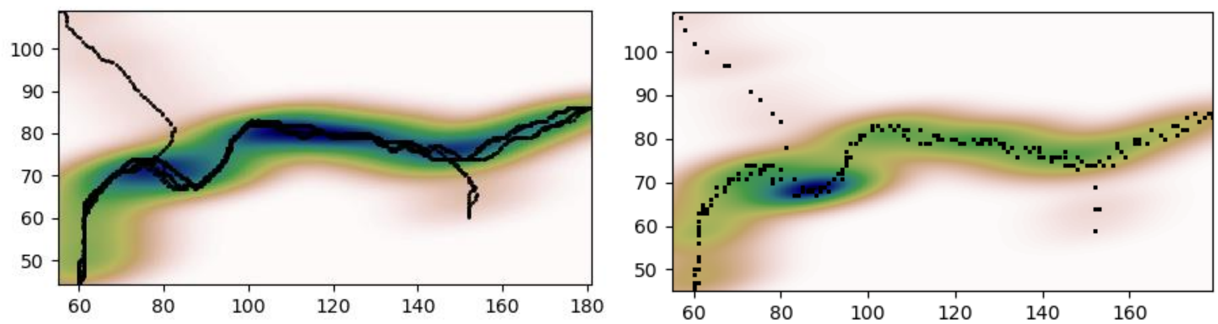
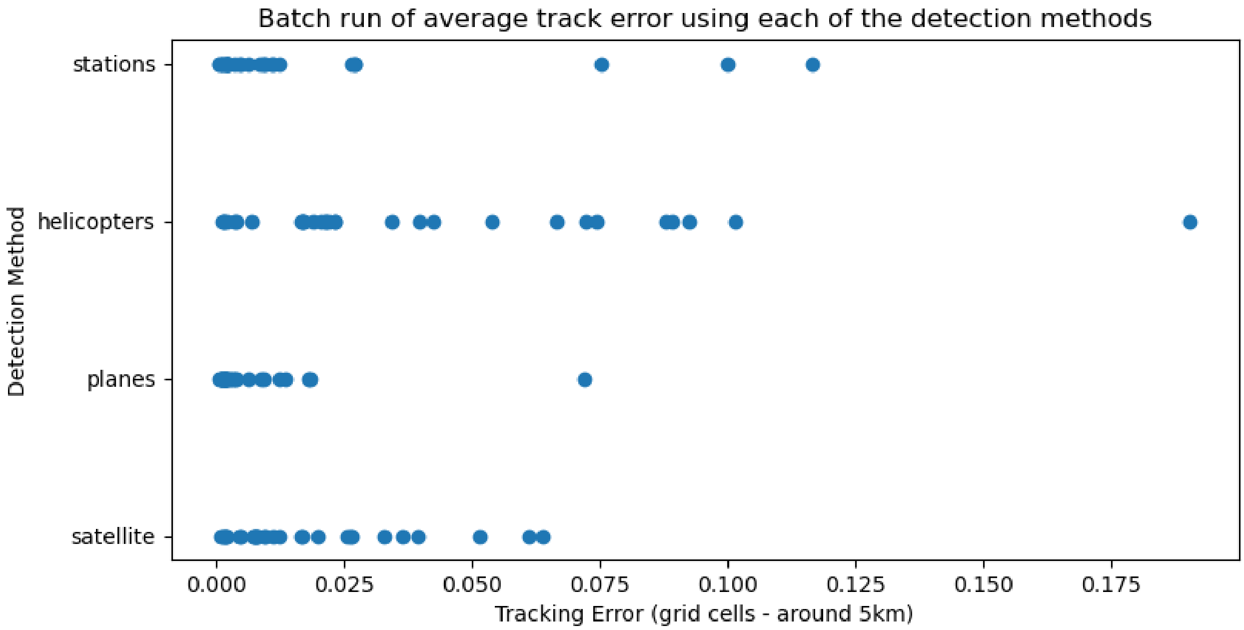


Figure 32: Home Range KDE for Truth Data (Left) and Stations Detections (Right) for 5 years

Tracking Error

We can examine the tracking error in more detail by running batches of cases and examine the average tracking error over 1 year for each of the detection methods. Each one is run 30 times, totaling 120 runs. This is done in 4 minutes and 22 seconds.



Animations + Visualization

A series of animations of wolf pack and detection behavior is available in the github repository. Several links are provided here.

Animated movement seen in Git repository:
https://github.gatech.edu/charris92/Wolfsim/blob/master/bin/wolf_animation.gif

Movement and live plotting:
https://github.gatech.edu/charris92/Wolfsim/blob/master/bin/viz_with_charts.gif

Movement and live plotting focused on satellite tracking:
https://github.gatech.edu/charris92/Wolfsim/blob/master/bin/sat_track_viz.gif

Movement and detection for planes tracking:
https://github.gatech.edu/charris92/Wolfsim/blob/master/bin/final_viz.gif

Project Tutorial

Development Platform

We are using the Mesa[1] agent-based simulation Python package to construct the foundation to our tool. The Python packages and development environment are setup in a PyCharm IDE and the files are linked the GitHub to provide for seamless collaboration and testing. The core Python packages are listed below, as well as in the setup.py file. Python 3.7 was used in development, but any Python 3.x is expected to work correctly. The specific packages are needed to create the agent-based model, the environment, and the agent behaviors.

- *Mesa*
    - Agent-based models, web-based visualizations, pandas dataframe storage
- *Numpy*
    - Efficient data storage and analysis on arrays.
- *Matplotlib*
    - Plotting charts, graphs, and images.
- *Scipy*
    - Post-processing of the data. For example, the kernel density estimation using the positions of the wolves.
- *Networkx*
    - Defining a graph structure to plan the optimal paths between feeding grounds. The Dijkstra's algorithm function is used to solve for the path.

There are four key modules of the code detailed here.

- **WolfsimModel**
    - The framework for the simulation is defined here. The mesa model is used to define the environment in the form of a grid, the scheduling of agents in the environment, the initial placement of those agents, the events at each time step of the simulation, as well as a few other supplimentary functions.
- **WolfsimAgents**
    - The definition and behavior of the wolf agent with movement, age parameters, and health indicators. The mesa agent class is used to build the wolf agent. The movement of the wolf used individual behaviors to model the swarm-like movement of the pack.
- **RunWolfsim**
    - The functions for running single cases, or batch cases of the simulation. Functions are also available for storing data in the simulation using the mesa datacollector class.
- **utils**
    - Additional functions needed throughout the other three modules.

Scenario

Our team has been tasked by the Alaska Department of Fish and Game, working in conjunction with the Alaska Department of Transportation and the Anchorage Port

[1] https://mesa.readthedocs.io/en/master/

Authority, to assess various tracking options for wolf populations inside an area of interest surrounding Anchorage, including the majority of the Alaska Range and segments of the Wrangell and Chugach Mountains. More specifically, this area spans a rectangular area 200 miles north, east, and west of the city of Anchorage, as well as 100 miles to the South. The main objective is to assess wolf-tracking methods under varying environmental factors and communications constraints to better assist in these conservation efforts. A simulation environment is created in order to evaluate the current tracking methods that use radio collars, satellite collars, or aerial spotting. Each method has its advantages and disadvantages that should be taken into consideration such as range, error, and consistency.

Example

We are interested in the region around Anchorage, Alaska, where the data was accessed in the report. We specify these regions by selecting the appropriate vegetation and elevation file.

```
vegetation_file = 'data/200x150Tree.asc'
elevation_file = 'data/200x150Elev.asc'
```

We use a custom function to read these data files and organize into numpy arrays.

```
veg_cells, width, height = readASCII(vegetation_file, version=version)
elev_cells, width, height = readASCII(elevation_file, version=version)
```

We can do three different run types:

1. Run a single case or multiple repeated cases - *sim_run()*
2. Run a batch of cases with variable settings – *batch_run()*
3. Run the internal simulation components with web-based visualization – *viz_run()*

We will go through each of these run types to demonstrate the capability of the ***WolfSim***.

Single Case Runs

We now demonstrate running cases with a single set of parameters. Therefore, we define the number of wolves in the pack, the type of detection system, and the total amount of time to simulate. We can also select the number of iterations if we wish to repeat the cases multiple times.

```
iterations = 10
tf = 365*1
wolfpack_size = 25
for j in range(iterations):
    model = WolfModel(wolfpack_size, width, height, elev_cells, veg_cells,
                      tracking_type=tracking_type, plot_movement=False)
    for i in range(tf):
        model.step()
```

For a given set of inputs, as shown above, we can examine the results through three key charts. The histogram of the wolf ages, the final position of the group of wolves, and the home range distribution prediction.

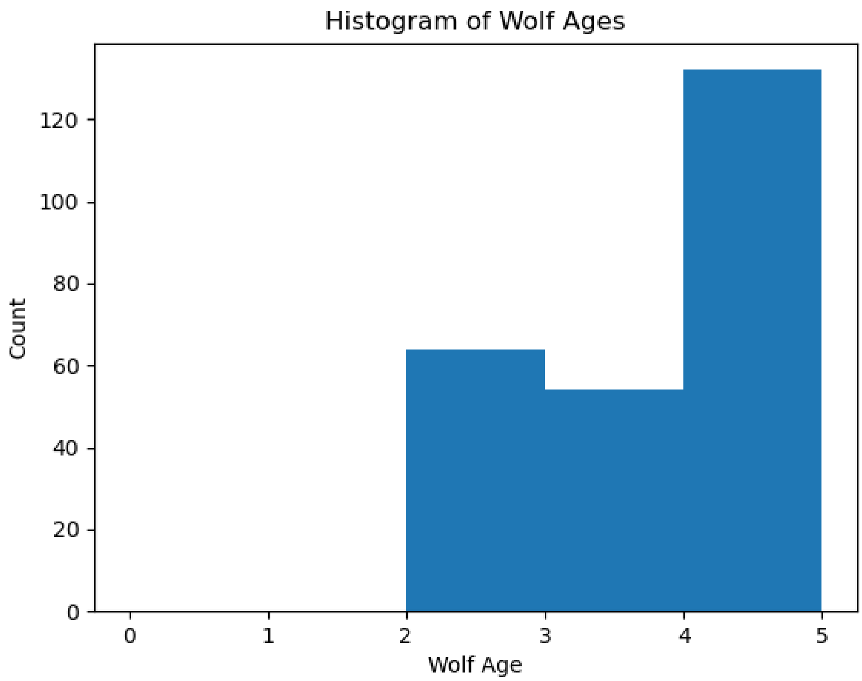


Figure 33: Wolf Ages Histogram

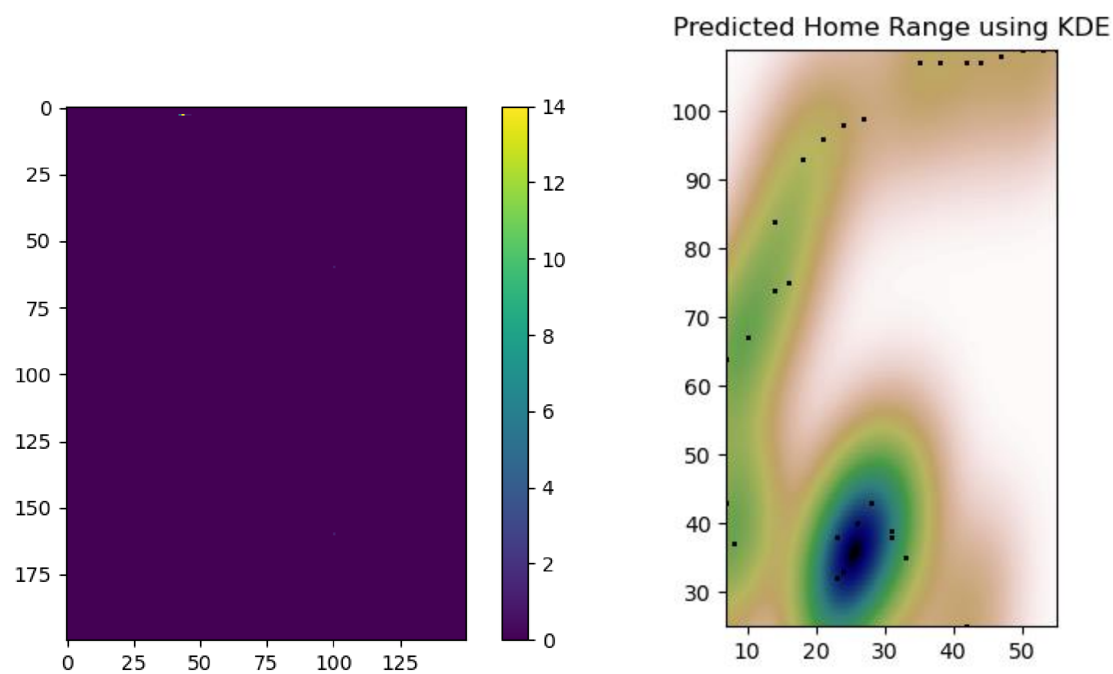


Figure 34: Final position and predicted home range for test case

Key information is also printed out at the end.

Average wolf age at final step that is living: ***3.497 years***

Number of wolves that have died: ***0***

Visualization Runs

The visualization runs are meant to be used to verify the simulation is doing as expected. The user interface opens in an html web socket and allows for visualization and controlling the simulation. This interface is seen in the following figure.

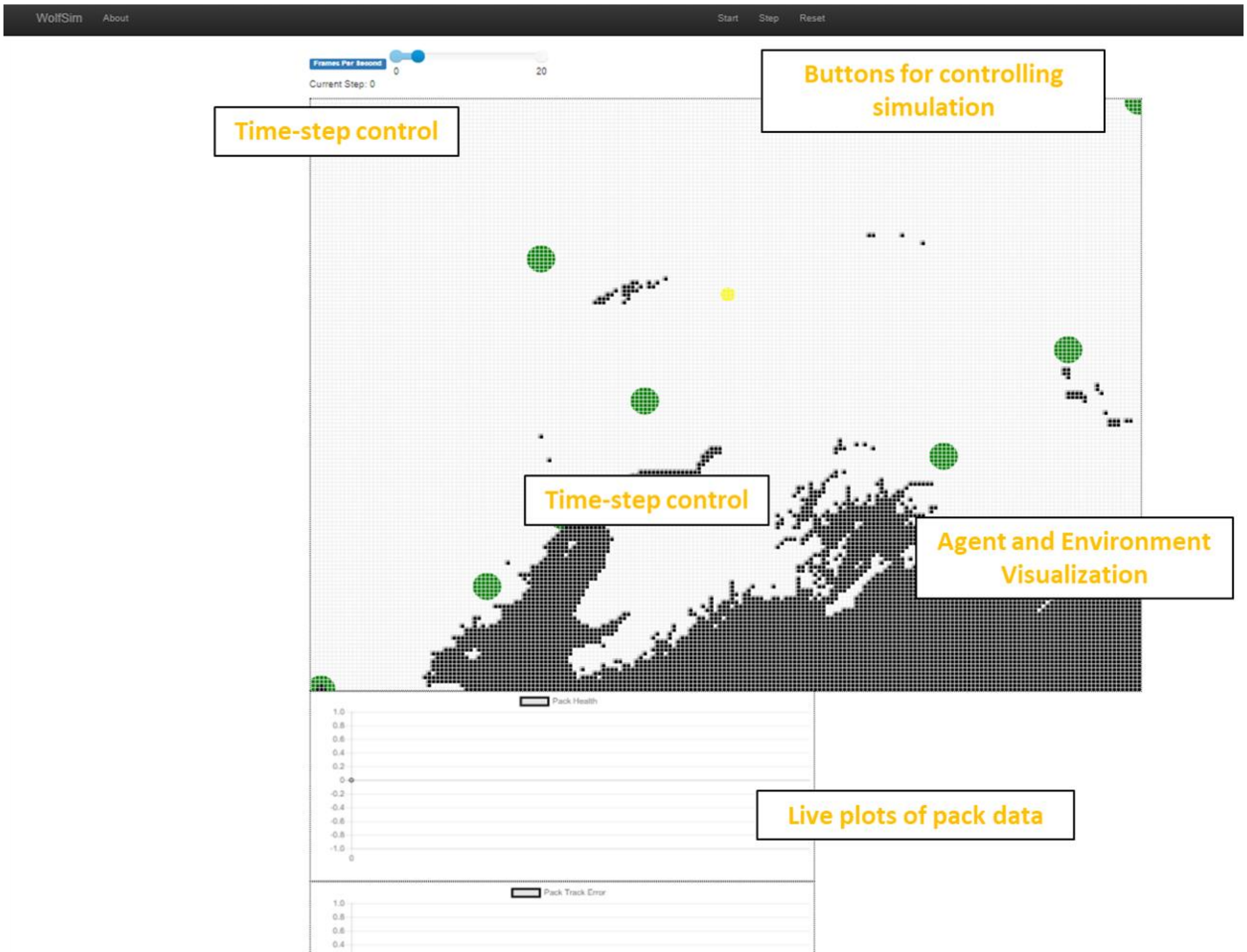


Figure 35: WolfSim Visualization and User Interface

As time moves forward in the simulation we can keep track of the agents, and view the current pack health (fraction of pack alive) and the average tracking error.

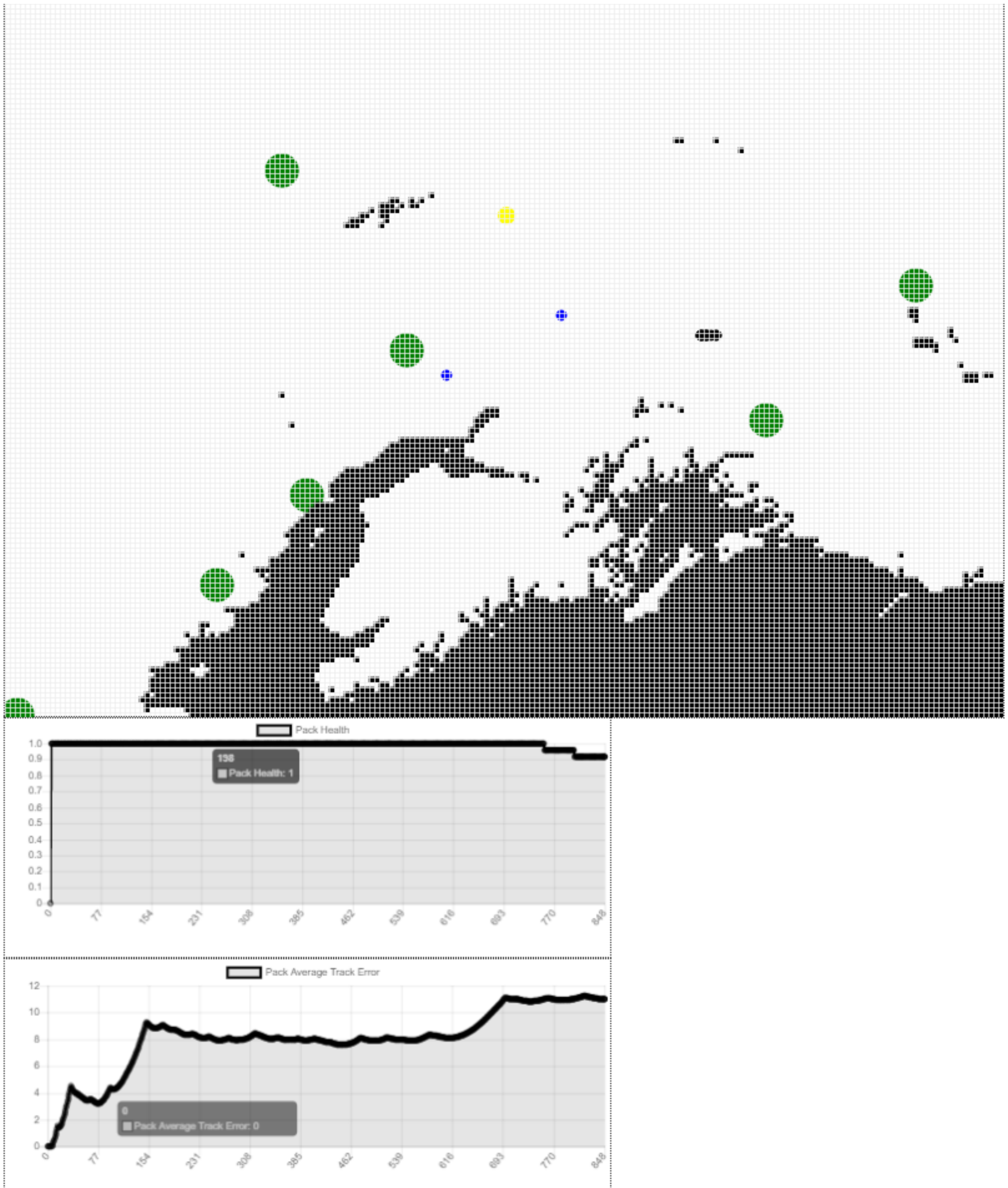


Figure 36: Visualization with Chart Graphics

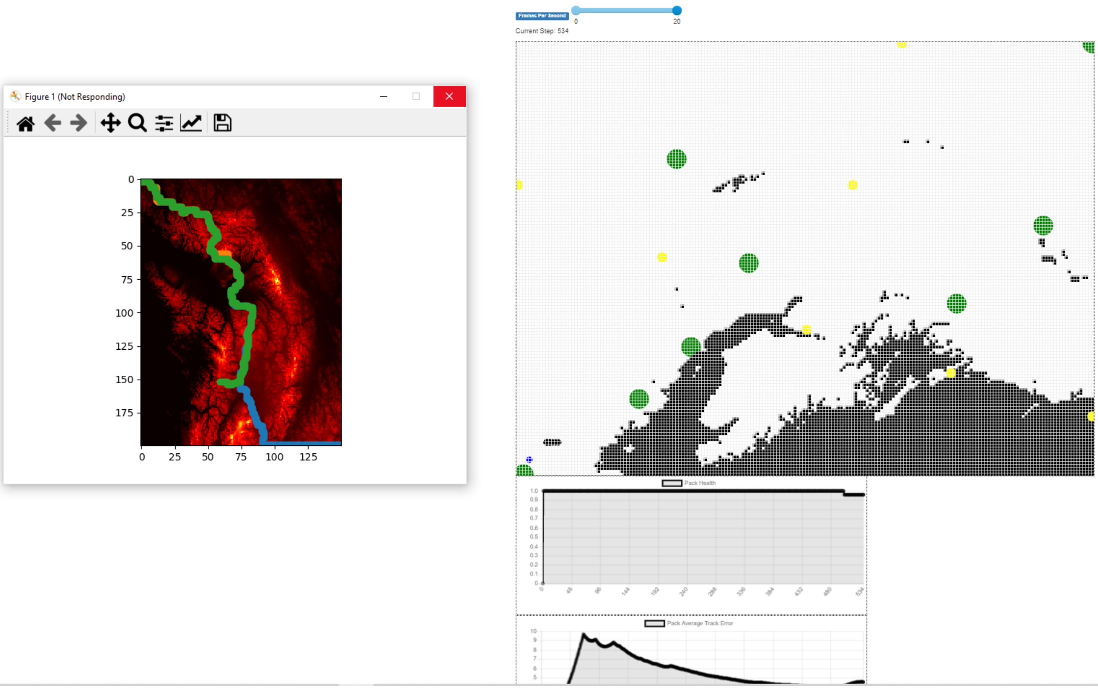


Figure 37: Visualization with Path Plots

Batch Runs

Lastly, we can run batches of cases to analyze how the change in wolfpack size or in detection method changes the success of monitoring the wolves. This is used to vary the tracking method and the number of wolves in the pack to see how this affects the tracking.

```
fixed_params = {"width": width, "height": height,
"elev": elev_cells, "veg": veg_cells,
                "N": 25}
variable_params = {"tracking_type":
['satellite', 'planes', 'helicopters', 'stations']}
batch_run = BatchRunner(WolfModel,
                   variable_params,
                   fixed_params,
                   iterations=30,
                   max_steps=365,
                   model_reporters={"pack_health": compute_pack_health,
                            "track_error": compute_track_error_average})
batch_run.run_all()
run_data = batch_run.get_model_vars_dataframe()
run_data.head()
plt.scatter(run_data.track_error, run_data.tracking_type)
```

Additional analysis using this run method is detailed in the Analysis section.

Detection Comparison

If we examine the differences between running each of the different tracking methods (satellite, planes, helicopters, stations) then we can see that, with the current implementation and parameters, the planes is the best at estimating the home range and limiting the tracking error. This is clear from the plots in the Wolf Pack Home Range and Tracking Error sections. Therefore, it can be suggested to the Alaska

Department of Fish and Game, to utilize the planes to best track the wolfpack population in the region around Anchorage.

Conclusion

The tutorial demonstrated the tools and techniques within the ***WolfSim*** environment and allowed suggestions to be made for the best detection and tracking technologies. The best technique can be chosen depending on the type of wolfpack and the region of interest. Future work could include modifying the detection agents to optimize the tracking and home range distribution estimation.

Our analysis only used 8 feeding sites, primarily out of a push for simplicity, runtime, and ease of explanation. However, increasing the simulation time and number of feeding sites allows the ***WolfSim*** environment to really shine. As shown in Figure 38, more time and feeding locations drastically improves the natural look and feel of the pathing, and results in a KDE spectrum matching that seen in arboreal predators.

Areas of potential improvement include loitering behavior, as well as pack dynamics and interactions with members of the public and human development. Roads, inhabited structures, and city limits could all be included in an additional map array and used to further describe wolf behavior through a penalty function in the pathing algorithm, which would induce an avoidance behavior that would be present in the resulting home range.

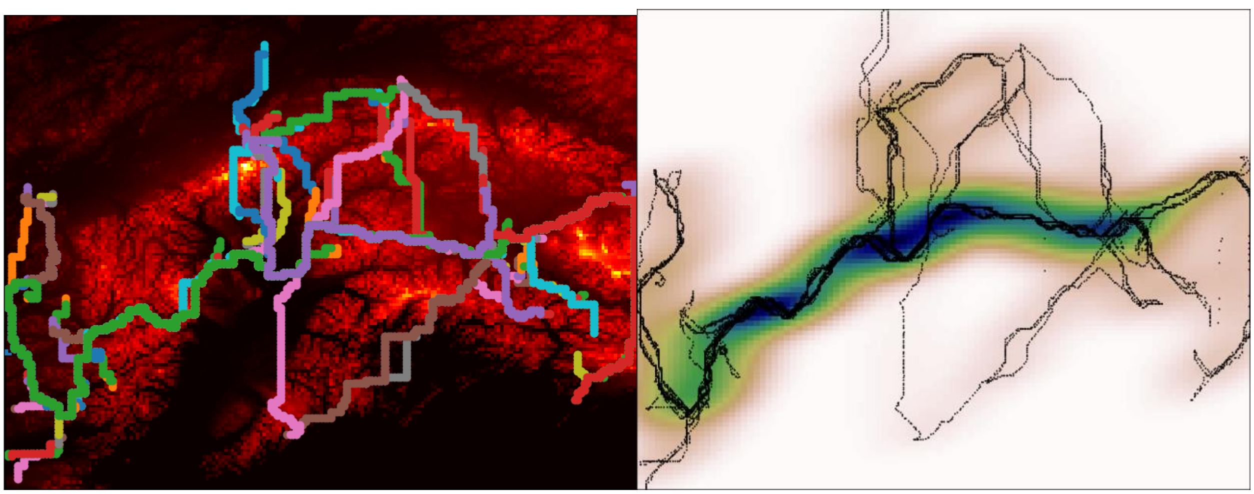


Figure 38 Path Plots and KDE for 100 feeding sites over 40 seasons

Division of Labor

Caleb Harris

The developer for the simulation model framework in Mesa, as well as the behavior of the agents

- Agent behavior
- Agent planning
- File reading and data management
- Agent visualization components
- Kernel Density estimation for home range estimation

Johnie Sublett

Implementation of the tracking devices as agents in the simulation.

- Dijkstra path integration
- Elevation and vegetation data creation
- Agent visualization setup

References

[1] https://wolf.org/programs/educator-resources-wolf-link/track-wild-wolves/

[2] https://wolf.org/wp-content/uploads/2013/09/tww_telemetrydata.pdf

[3] C. Muro, R. Escobedo, L. Spector, R.P. Coppinger, Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations, Behavioural Processes, Volume 88, Issue 3, 2011, Pages 92-197, ISSN 0376-6357, https://doi.org/10.1016/j.beproc.2011.09.006.

[4] Wlodzimierz Jedrzejewski, Krzysztof Schmidt, Jörn Theuerkauf, Bogumila Jedrzejewska, and Henryk Okarma. Daily movements and territory use by radio-collared wolves (Canis lupus) in Bialowieza Primeval Forest in Poland. Canadian Journal of Zoology. 79(11): 1993-2004. https://doi.org/10.1139/z01-147

[5] https://wolf.org/programs/educator-resources-wolf-link/track-wild-wolves/case-study-wolf-381/

[6] Downs, Joni & Horner, Mark. (2012). Analysing infrequently sampled animal tracking data by incorporating generalized movement trajectories with kernel density estimation. Computers, Environment and Urban Systems. 36. 302–310. 10.1016/j.compenvurbsys.2011.12.001.

[7] Mech, L. David; Boitani, Luigi, eds. (2003). Wolves: Behaviour, Ecology and Conservation. University of Chicago Press. ISBN 978-0-226-51696-7.

[8] Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numerische mathematik, 1(1), pp.269–271.

[9] https://wolfwatcher.org/wp-content/uploads/2015/09/PRESENTATION-BY-OLIVER-STARR-AT-SPEAK-FOR-WOLVES-EVENT.pdf, Wolf dens, 1992-1994 (NPS 2010, DENA 2008)

[10] https://apps.nationalmap.gov/downloader/#/

[11] Crook, Stephen & Paulus, Gernot. (2016). An Agent-Based Model for Exploring Wolf Recolonization in Austria. GI_Forum. 2. 3-19. 10.1553/giscience2016_02_s3.