Home Range Modelling

MATH 371 Final Project

Andrew Nisbet

April 22, 2014

Contents

List of Figures

1 Introduction

1.1 About Home Ranges

The home range of an animal is the area in which it lives and travels (Okubo & Levin, 2001, p. 238). Home ranges cover territories, nesting sites, breeding grounds, and wherever an animal spends its time. The home ranges of individual animals often overlap, and for herding animals the home range can be considered for the entire herd.

1.2 Project Aims

This project is adapted from Project 19 in de Vries et al. (2006, p.253). The goal of this project is to develop a mathematical model for animal movement behaviour that results in a realistic and stable home range. In particular, this project considers models for the home range of a herd of animals.

Models were be created for animal movement, and the resulting home ranges were evaluated. The models were be implemented numerically in the Python programming language. Relevant sections of the code are included in Appendix A.

First, this report outlines a strict definition for home ranges, followed by a framework for modelling animal movement. Four models for home ranges are then presented, and their limitations discussed.

1.3 Approach

Random walk models were selected for their ubiquity in animal movement modelling. The models are evaluated for stability, performance and biological relevance.

1.3.1 Stability

A stable home range should remain the same shape and size over time. Calculating and comparing the exact shape of the home range at each time step is prohibitively expensive. Instead, the root mean square (RMS) deviation $\hat{\sigma}$ of all the animals from the centroid C is analysed for stability.

For n animals with position $\mathbf{x}(t)$, the centroid location is given by

$$C(t) = (\bar{x}, \bar{y}) = \frac{\mathbf{x}_1 + \mathbf{x}_2 + ... + \mathbf{x}_n}{n}$$

The RMS deviation can then be calculated

$$\hat{\sigma}(t) = \sqrt{\sum_i \Big((x_i - \bar{x})^2 + (y_i - \bar{y})^2 \Big)}$$

A stable home range should exhibit

$$\lim_{t \to \infty} \frac{\mathrm{d}}{\mathrm{d}\,t} \hat{\sigma}(t) \approx 0$$

This metric has more local variability than home range size, but long term trends can still be assessed.

1.3.2 Performance

Performance is analysed with big O notation. A model that is $O\big(n^2\big)$ will take 4 times as long to run when n is doubled. Only the highest order terms are considered, so $O\big(n^2{+}2^n\big) = O\big(2^n\big)$. Algorithms with lower big O will scale better to larger problems.

This use of big O notation is consistent with computer science, rather than the mathematical usage of defining the error of functional approximations.

Model runtime may be proportional to the number of timesteps, environment size, or number of animals modelled.

1.3.3 Biological relevance

In the absence of fitting data, biological relevance is estimated quantitatively and justified in the report. Empirical and experimental research will be drawn on where possible.

1.4 Results Overview

The final model, consisting of a combination of environmental effects and olfactory diffusion, meets the project aims. It is stable and biologically valid, at the expense of performance.

2 Home Range Definition

For home range models to be evaluated and compared, a mathematical definition for a home range is needed.

Animal data is typically collected in the form of distinct location coordinates at given times. For field observations, this usually consists of GPS coordinates. The models developed for this project produce the same type of data: animal locations in discrete time and space.

Time information is typically discarded when calculating home range size, by considering all the coordinates in a given time interval together. In defining home ranges for this project, data is taken as a set of spatial points, although time information is retained for analysing stability and plotting walk paths.

Two main methods exist for relating this spatial data to a well-defined home range shape: minimum convex polygon fitting, and kernel density estimation.

2.1 Minimum Convex Polygon

The simplest method is to construct the smallest possible convex polygon around the coordinates. This method has several benefits. For a start, the minimum convex polygon is strictly defined for a set of points, so needs no parameterisation and is easy to compare across home ranges. It is also simple and fast to compute. It works reasonably well for round home ranges like the one in Figure 1.

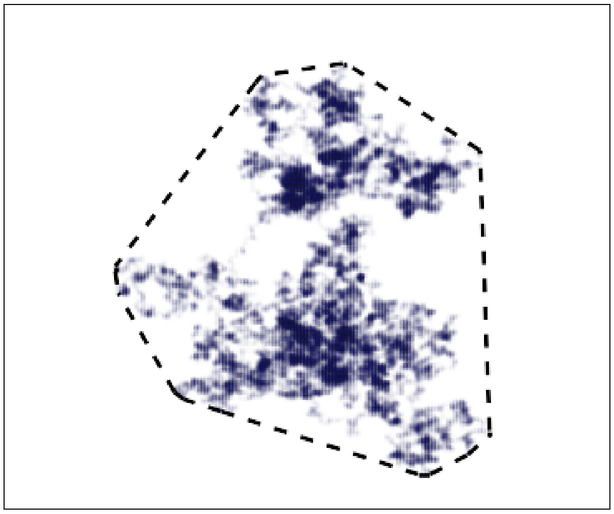

Figure 1: Convex polygon (dashed line) around scatter plot of a single random walk. Darker areas represent an overlapping of the path.

However, because the polygon contains all location coordinates, this method tends to overestimate the size of the home range. This effect is particularly pronounced if the animal's walk is not uniform, or if the animals wander before forming a stable home range, like in Figure 2.

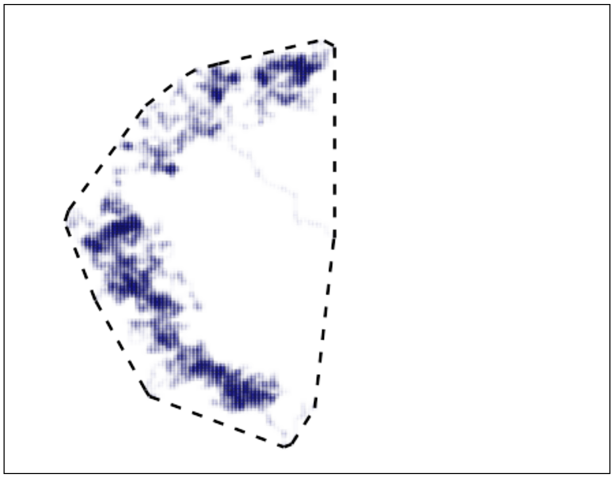

Figure 2: Convex polygon (dashed line) around scatter plot of random walk. The animal began its walk at the centre of the figure, before moving to a stable home range area on the left.

Wandering can be excluded by ignoring initial points, but this requires additional complexity and parameters, defeating the advantages of the method. Because this simulation project looks at animals converging to a stable home range, minimum convex polygon is not suitable, as it cannot handle the initial settling to a home range.

2.2 Gaussian Kernel Density Estimation

It seems reasonable that an animal may occasionally move outside of its home range. If this is the case, a better approach to home range definition would be to consider where an animal spends most of its time, rather than all of it. Alternatively, this can be interpreted as considering where an animal is most likely to found.

This intuition leads to a more robust method of defining home ranges: Kernel Density Estimation (KDE). Given a set of points, KDE produces a probability density function for the entire domain. Sampling the density function results in a set of coordinates similar to the distribution of points used in the estimation.

In particular, a multivariate Gaussian kernel density estimation is used for this project.

Figure 3 shows KDE on the same walk as in Figure 2, which the convex polygon method failed to accurately define. The kernel is an array of the same size as the environment, with a value in each cell that represents the probability of an animal occurring there. This is indicated by the darker areas in Figure 3. It can be seen that the initial wandering movement is no longer considered.

To produce a distinct home range area, a probability threshold is defined. Figure 4 shows the home range containing all areas with an occurrence probability within 90% of the maximum. The exact threshold value is relatively arbitrary, but allows for consistent comparison between home ranges. An example of a KDE threshold is shown in Figure 4. Low density wandering areas are excluded from the home range.

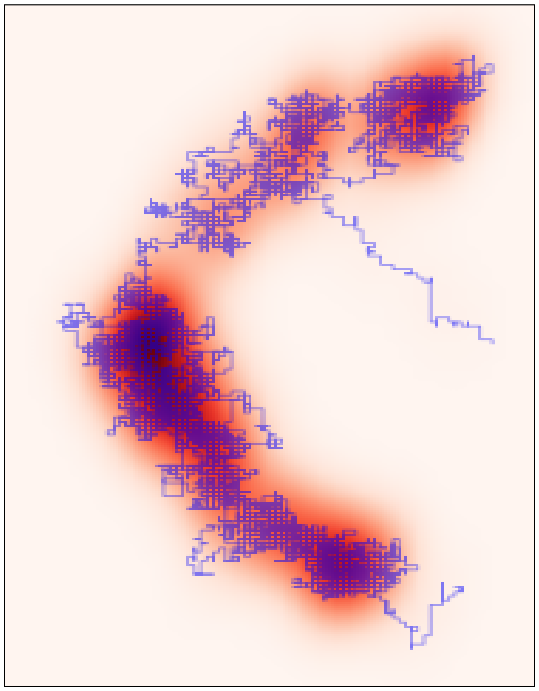

Figure 3: KDE of single animal walk. Darker areas represent higher likelihood of occurrence.

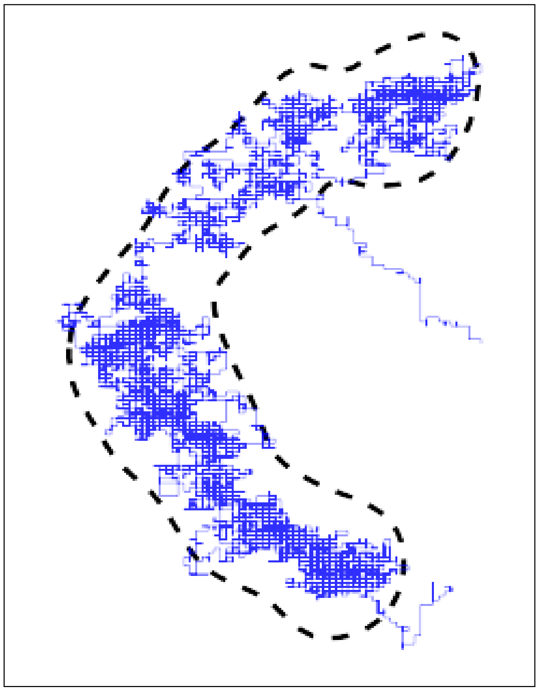

Figure 4: KDE of single animal walk. Dashed line represents 90% home range threshold.

The main disadvantage of the KDE method is its performance. Not only is it more complicated to implement than the convex polygon method, but it also performs worse. KDE typically has $O(n^2)$ runtime, whereas convex polygon calculation is roughly $O(n \log n)$.

To improve performance of KDE, the calculation is performed on a uniform (over time) subset of animal coordinates. The function runs much faster, with little difference from the full calculation. The KDE method is used for all home range plots in the remainder of the report.

3 Random Walk Modelling

Because home ranges are a product of animal locations, home range models ought to be based on animal movement. Random walks have long been used for animal movement modelling, from simple microbial taxis to complex mammal behaviour. Random walk models generally incorporate various biases to capture the biological intent and motivations for complex organisms to move preferentially though their environment.

For this project, 2-dimensional random walks were used to produce a series of location points, to which home ranges were fitted.

3.1 Environment Grid

The environment was discretised into a square grid. Because of their non-adjacent diagonal cells, square grids are sometimes considered inferior to hexagonal and rectagonal square grids (Birch et al., 2007). However, the advantage of square grids is that they are simple to implement, and when represented by a square array can rely on optimised procedures in computational mathematics packages. The increased performance allows for consideration of smaller cells, reducing inaccuracies.

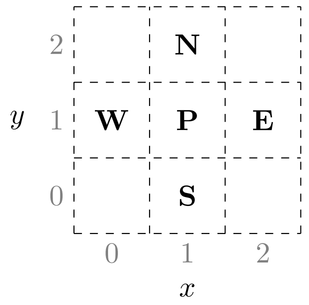

Figure 5: Grid notation for an animal located at **P**.

An animal located at point $P = (x, y)$ has a choice of moving in any one of four directions, which are represented by the four compass directions N, S, E, and W as in Figure 5. Additionally, the animal may remain at point P. The random variable $\mathbf{X}(t) = \big(X(t), Y(t)\big)$ is defined to represent an animal's location.

3.2 Quality Driven Movement

Each location in the environment is assigned a quality $q(x, y)$. This attribute represents the desirability of a certain position as perceived by the animal, and may incorporate factors such as food availability, predator territory, and climate. These qualities are combined into an environment matrix Q, with $Q_{xy} = q(x, y)$.

The animal movement was modelled as a discrete time Markov chain random walk, meaning movement to the next location is dependent only on the current position. Limiting movement to adjacent cells imposes a maximum velocity $\frac{\Delta x}{\Delta t}$ on animal movement, which is good for biological relevance. Also, by allowing the animal to remain at its current position and considering a small Δx, speeds of less than this maximum are observed. Finally, animal movement over several timesteps will contain backtracking, again resulting in speeds less than this maximum.

The bias of the random walk is based only on the quality of the adjacent locations. This can thought of as animals only observing the environment quality in those locations close to them. This is particularly valid for olfactory perception.

To convert the environment qualities to transition probabilities, they are normalised so the of qualities at adjacent cells to P sum to one. The random walk is given by

$$Pr\Big(\Delta X(t) = i,\, \Delta Y(t) = j \,\Big|\, X(t), Y(t)\Big) = \begin{cases} \dfrac{q_N}{Q_{tot}}\,, & (i,j) = (\ \ 0, +1) \\ \dfrac{q_S}{Q_{tot}}\,, & (i,j) = (\ \ 0, -1) \\ \dfrac{q_E}{Q_{tot}}\,, & (i,j) = (+1,\ \ 0) \\ \dfrac{q_W}{Q_{tot}}\,, & (i,j) = (-1,\ \ 0) \\ \dfrac{q_P}{Q_{tot}}\,, & (i,j) = (\ \ 0,\ \ 0) \end{cases}$$

with

$$q_N = q(X,\, Y{+}1) \quad \text{and} \quad Q_{tot} = q_N + q_S + q_E + q_W + q_P$$

When the environment quality is known at the start of the simulation, these equations can be precompiled into a sparse transition matrix P, with $P_{(x,y)(x^*,y^*)}$ representing the probability of an animal at (x, y) transitioning to (x^*, y^*). However, because much if the environment remains unexplored, and later models deal with dynamic qualities, this step was not performed, and the quality-to-probability conversions were performed on the fly.

To alter the strength of quality as a bias, the quality matrix can be scaled with an exponent weight: Q^m. Higher m means relatively more attraction towards high quality; lower m means lower attraction.

Edge cases were not specified. Because the simulations are approximating an infinite environment, the simulation stopped (crashed) if an animal reached the edge, and was manually restarted with a larger environment size.

This random walk approach was used to produce the data used for fitting home ranges in Figures 1-4.

All simulations were run on a square environment, with the animals beginning in the centre. A timestep of $\Delta t = 0.001$ was used. This value is arbitrary, but allows conversion of time convergence plots to number of iterations.

4 Random Walk Home Range Models

With a framework for animal movement and a method to define home range size, the modelling can begin. Four different models were developed and evaluated.

4.1 Environment Quality Model

Many of the factors that influence quality are inherent to the environment, such as food availability and climate. This inherent environment quality can be produced randomly. A purely random array like the one in Figure 6 does not resemble an actual environment. Applying a Gaussian filter to the noise results in more realistic variation, as seen in Figure 7.

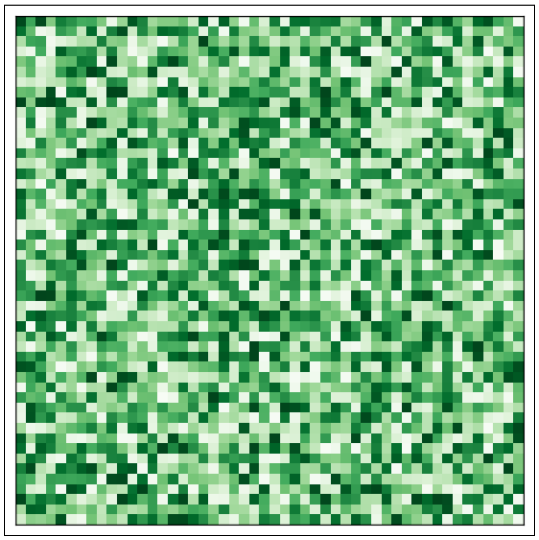

Figure 6: Random quality noise.

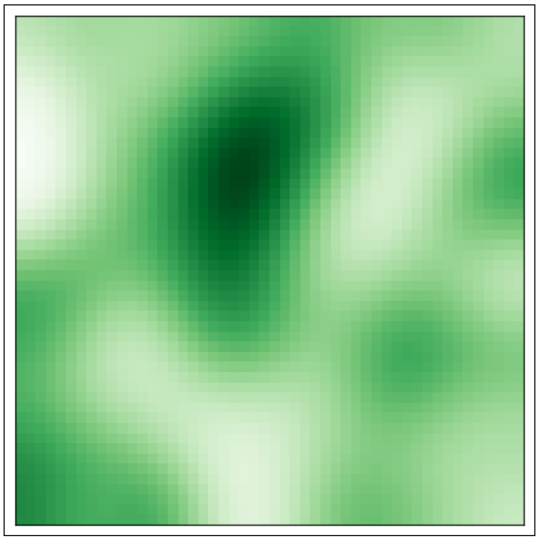

Figure 7: Smoothed environment quality. Darker areas represent higher quality.

Gaussian filters are widely used in image processing to reduce noise and introduce blur effects. They work by assigning each cell with a weighted average of the quality of its neighbours. The weighting decreases with distance, and this weighting attenuation is the only parameter. Setting this parameter as a function of the environment dimensions results in environment with similar feature scales for all Δx environment sizes.

This model uses a random environment to bias a random walk. After plotting, the environment is scaled by the exponent m. Quality is assumed to be constant over time, and unaffected by the presence of the animals being modelled.

The algorithmic complexity of the simulation is $O(n)$, where the total number of iterations n is equal to the product of the number of animals and the number of iterations for the simulation. As a result, the simulation runs extremely quickly.

When placed in the environment, an herd of animals animal travels together to a local maximum, as shown in Figure 8. The model is a discretised, stochastic form of gradient descent, and for $m \to \infty$ is identical to gradient descent. Provided the local maximum is sufficiently large and steep, stable behaviour is exhibited. Figure 9 shows the change in RMS animal separation size over time, which flattens out when the animals reach the high quality area at $t = 2$.

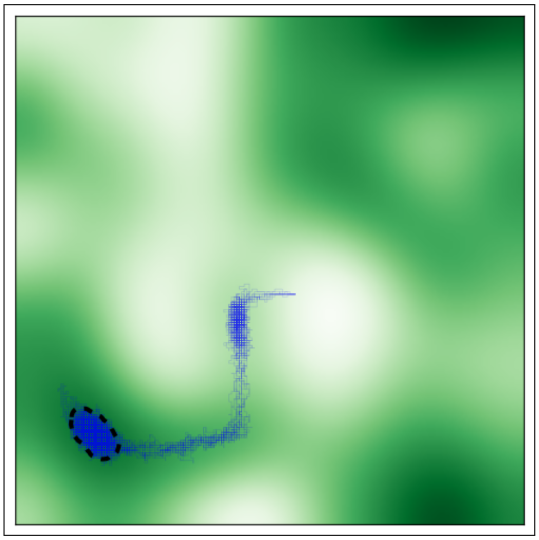

Figure 8: Biased random walk of 10 animals in a randomly generated environment, with dashed KDE home range.

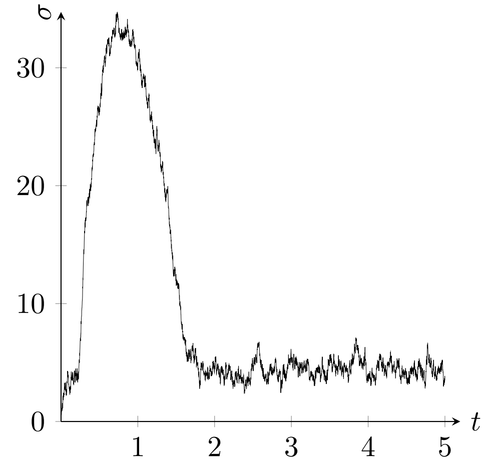

Figure 9: RMS deviation of 10 animals from centroid over 5000 iterations.

However, this nice convergence is not always seen. Multiple animal convergence depends on the steepness, size, and uniqueness of the local maxima. Figure 10 shows the result of the model running on a different environment. The herd splits, and animals end up in different areas. This is because of the two local maxima found on either side of the starting point. Additionally, the steepest quality gradient is concentrated near the edges of the map. When $\frac{\mathrm{d}}{\mathrm{d}x}q$ is small, behaviour tends towards that of an unbiased walk, making it even more likely the animals will travel in different directions.

This diverging behaviour is not unheard of in biology. Similar separations can result in speciation. However, it would be unusual to see this occur in a near-uniform environment for no real reason other than half of the herd wandering away.

This problem highlights the issue with environment-quality driven movement. The model suggests that animals only seem like they are herding because they are heading to the same place. There is no consideration for interactions between animals, which ought to be a significant driver of herding behaviour.

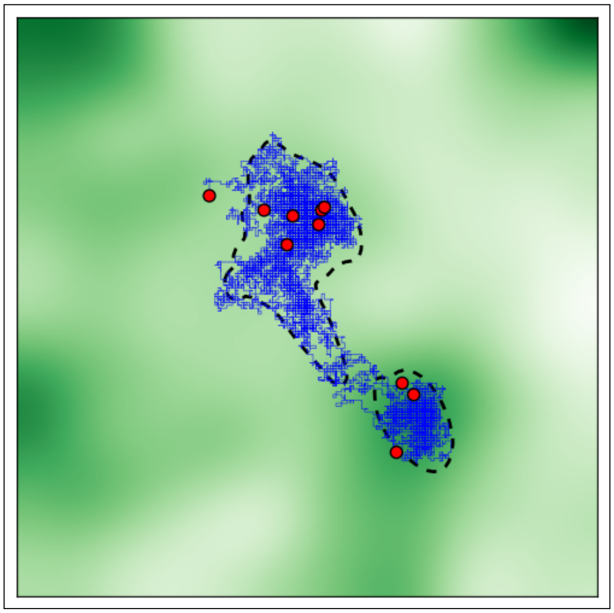

Figure 10: Diverging random walk of 10 animals, with their final locations indicated by dots.

4.2 Centre Bias Model

By incorporating a biological mechanism behind animal interaction, a more biologically valid model of animal interaction can be developed. For this model I used scent, as used by several other authors (e.g., Benhamou, 1989). Scent is used by many animals for orientation and interaction, but similar models could be developed for visual or auditory orientation.

The idea is that animals leave scent as they move through their environment. This may be discrete events, such as territory marking, as well as continuous processes, like constant pheromone production. However, in the interest of simplicity, scent was considered as a constant uniform process: the animal releases a constant amount of scent at every timestep. For many animals over many timesteps, this is an acceptable approximation.

Consider scent concentration in a home range. Because animals spend more time in areas close to the centre of their home range, it can be assumed that scent is more concentrated there, and that the scent diffuses exponentially from the centre.

$$\text{stimulus} \propto \exp(\text{distance})$$

However, the animal does not perceive the environment quality directly. Just as the quality matrix can be scaled by the exponent m to represent how strong quality biases movement, the perceived scale of the scent with decreasing intensity must be accounted for.

This intuition comes from the Fechner's law, which states that the change in perceived intensity p of a stimulus S is given by

$$\begin{aligned} dp &= k\frac{dS}{S} \\ p &= k \ln S + C \\ \text{perception} &\propto \ln(\text{stimulus}) \end{aligned}$$

with constants k and C (Masin et al., 2009). The result of these two proportionalities is a linear perceived gradient for an exponentially diffusing stimulus gradient.

$$\text{perception} \propto (\text{distance})$$

Fencher's law is extremely robust, even among animals. Supporting studies include primates (Nieder & Miller, 2003), rats, and pigeons (Gibbon, 1997).

The final linear relationship is still an approximation. Non exponential models exist for diffusion, and variations of Fencher's law exist with different logarithmic bases and scaling factors. The key intuition is that animals are able to covert to decaying stimuli to a linear scale.

A linear radial gradient environment is easy to create, and then the simulation proceeds as in the environment quality model. Placing animals in the centre of the environment results in an eventually stable home range, as shown in Figure 11. The animals spread apart until their RMS separation becomes stable. The stability conversion is shown in Figure 12.

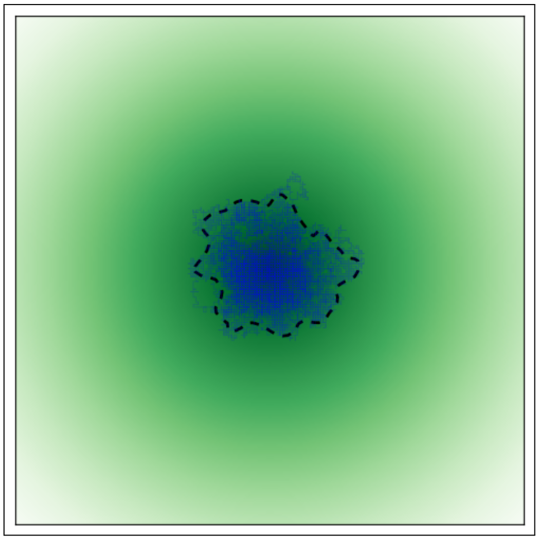

Figure 11: Stable home range in decreasing radial environment quality.

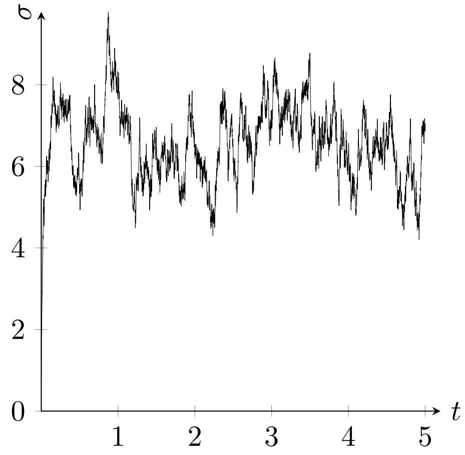

Figure 12: Stability of 10 animals over 5000 iterations.

This model has the same $O(n)$ complexity as the environment quality model.

The drawback of the central bias model is that it requires stipulating a centre for the home range. This doesn't account for stochastic fluctuations in home range location, or for biased movement of home range location.

4.3 Olfactory Model

A more accurate model would be to model the scent production directly, and allow it to diffuse over the environment. This can be set as a dynamic environment quality array that changes each iteration.

The scent quality q is diffused using the 2D diffusion equation

$$\frac{\partial q}{\partial t} = D\nabla \cdot [\nabla q] + S(q)$$

where D is a constant diffusion coefficient, and $S(q)$ is a source term. Here, the source term represents the addition of new scent, the chemical decay of the scent, and diffusion into the z direction which is not explicitly modelled.

The diffusion equation can be discretised with the finite volume method. The notation used is shown in Figure 13. In addition to the animal location P and adjacent positions $NSEW$, points $nsew$ are defined $\frac{\Delta x}{2}$ away from P.

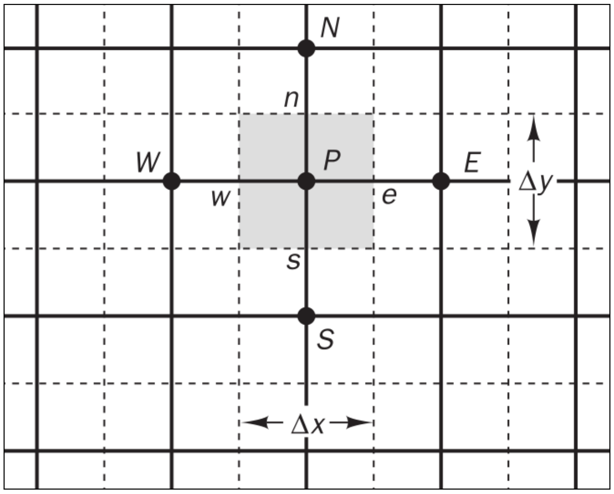

Figure 13: Notation for an animal located at P. From Versteeg and Malalasekera (2007, p. 129).

Consider the square control volume with edges passing though $nesw$ around the position P. The control volume has unit depth. Integration over the control volume and over a time interval from t to $t + \Delta t$ gives

$$\int_{t}^{t+\Delta t}\int_{\Delta V}\frac{\partial q}{\partial t}\,\mathrm{d}V\mathrm{d}t = D\int_{t}^{t+\Delta t}\int_{\Delta V}\frac{\partial}{\partial x}\left(\frac{\partial q}{\partial x}\right)\,\mathrm{d}V,\mathrm{d}t$$

$$+ D\int_{t}^{t+\Delta t}\int_{\Delta V}\frac{\partial}{\partial y}\left(\frac{\partial q}{\partial y}\right)\,\mathrm{d}V\mathrm{d}t + \int_{t}^{t+\Delta t}\int_{\Delta V} S\,\mathrm{d}V\mathrm{d}t$$

Assuming q is constant over a whole cell, the left hand side can be written as

$$\int_{t}^{t+\Delta t}\int_{\Delta V}\frac{\partial q}{\partial t}\,\mathrm{d}V\mathrm{d}t = \left(q_{_P} - q_{_P}^{o}\right)\Delta V$$

The superscript ^{o} refers to $q(t)$, while $q = q(t+\Delta t)$ is not superscripted. The subscript $_{P}$ refers to the quality at point P.

$$\left(q_{_P} - q_{_P}^{o}\right)\Delta V = D\,\Delta y_{ns}\int_{t}^{t+\Delta t}\left[\left(\frac{\partial q}{\partial x}\right)_{e} - \left(\frac{\partial q}{\partial x}\right)_{w}\right]\,\mathrm{d}t$$

$$+ D\,\Delta x_{ew}\int_{t}^{t+\Delta t}\left[\left(\frac{\partial q}{\partial y}\right)_{n} - \left(\frac{\partial q}{\partial y}\right)_{s}\right]\,\mathrm{d}t + \int_{t}^{t+\Delta t}\bar{S}\Delta V\,\mathrm{d}t$$

Considering the flux of q across each face results in

$$\left(q_{_P} - q_{_P}^{o}\right)\Delta V = D\,\Delta y_{ns}\int_{t}^{t+\Delta t}\left[\left(\frac{q_{_E} - q_{_P}}{\Delta x_{_{PE}}}\right) - \left(\frac{q_{_P} - q_{_W}}{\Delta x_{_{WP}}}\right)\right]\,\mathrm{d}t$$

$$+ D\,\Delta x_{ew}\int_{t}^{t+\Delta t}\left[\left(\frac{q_{_N} - q_{_P}}{\Delta y_{_{PN}}}\right) - \left(\frac{q_{_P} - q_{_S}}{\Delta y_{_{SP}}}\right)\right]\,\mathrm{d}t + \int_{t}^{t+\Delta t}\bar{S}\Delta V\,\mathrm{d}t$$

An explicit time differencing scheme is applied

$$\int_{t}^{t+\Delta t} q_{_P}\,\mathrm{d}t = q_{_P}^{o}\,\Delta t$$

$$\left(q_{_P} - q_{_P}^{o}\right)\Delta V = D\,\Delta y_{ns}\Delta t\left[\left(\frac{q_{_E}^{o} - q_{_P}^{o}}{\Delta x_{_{PE}}}\right) - \left(\frac{q_{_P}^{o} - q_{_W}^{o}}{\Delta x_{_{WP}}}\right)\right]$$

$$+ D\,\Delta x_{ew}\Delta t\left[\left(\frac{q_{_N}^{o} - q_{_P}^{o}}{\Delta y_{_{PN}}}\right) - \left(\frac{q_{_P}^{o} - q_{_S}^{o}}{\Delta y_{_{SP}}}\right)\right] + \bar{S}\Delta V\Delta t$$

Finally, several simplifications can be made due to the square grid

$$\Delta x_{PE} = \Delta x_{WP} = \Delta y_{PN} = \Delta y_{SP} = \Delta y_{ns} = \Delta x_{ew} = \Delta x = \Delta y$$

$$\Delta V = \Delta x \Delta y = (\Delta x)^2$$

and the source term can be split into a proportional and constant term

$$\bar{S} = (q_P S_P + S_u)$$

resulting in

$$\left(q_{_P} - q_{_P}^o\right)(\Delta x)^2 = D\,\Delta t\,[q_N^o + q_S^o + q_E^o + q_W^o - 4q_P^o] + (q_P S_P + S_u)(\Delta x)^2 \Delta t$$

The final form of the discretised diffusion equation that is used in the simulation is

$$a_{_P} q_{_P} = a_{_N} q_{_N}^o + a_{_S} q_{_S}^o + a_{_E} a_{_E}^o + a_{_W} q_{_W}^o + \left[a_{_P}^o - (a_{_N} + a_{_S} + a_{_E} + a_{_W} - S_{_P})\right] q_{_P}^o + S_u$$

where

$$\begin{aligned} a_N &= a_S = a_E = a_W = D \\ a_P &= a_P^o \\ a_P^o &= \frac{(\Delta x)^2}{\Delta t} \end{aligned}$$

What this equation provides is, given the scent quality of the environment at time t, the scent quality at time $t + \Delta t$ can be calculated.

An explicit discretisation was used: the left hand side contains only the unknown quality at the next timestep, whilst the right hand side contains only known values of quality at the current timestep. This means it can be solved iteratively for each cell, in $O\big(x^2\big)$ for each timestep.

However it does have stability issues. The explicit finite volume discretisation is conditionally stable, requiring $\Delta t < \frac{\Delta x}{4D}$. Fortunately, achieving this inequality was not an issue for the simulation.

An unconditionally stable implicit scheme is produced by differencing

$$\int\limits_t^{t+\Delta t} q\,\mathrm{d}t = q\,\Delta t$$

instead of

$$\int\limits_t^{t+\Delta t} q\,\mathrm{d}t = q^o\,\Delta t$$

Because both sides of the resulting equation would contain terms from the unknown timestep, all equations would have to be solved at the same time. This is generally implemented as a solution to the linear equation $Az = b$, where A is an x^2 by y^2 matrix. Solving such an equation is $O\left(x^4\right)$, which does not scale very well, and slows the simulation too much.

The explicit numerical simulation works as follows. An environment is created with a dynamic array to store scent quality. Each update cycle of the environment involves first calculating the new animal positions, using scent quality to bias the random walk. Scent is then added to the environment scent array at the location of the animals. Finally the scent is allowed to diffuse. Pseudocode for this routine is shown in Figure 14.

```
new array scent_quality

function update {
    move_animals(scent_quality)
    scent_quality += animal_scent
    scent_quality = diffuse(scent_quality)
}
```

Figure 14: Pseudocode for the modelling of scent diffusion.

The result is shown in Figure 15. It can be seen the animals stay together, and the stability is plotted in Figure 16. The final result is similar to that of the central bias, although with much better biological validity.

The model has $O\left(nx^2\right)$ complexity. It takes several orders of magnitude longer to run than the previous models. The simulation shown in Figures 15 and 16 had to be performed with 3 animals and 2000 iterations, as well as a smaller environment, to ensure practical running times.

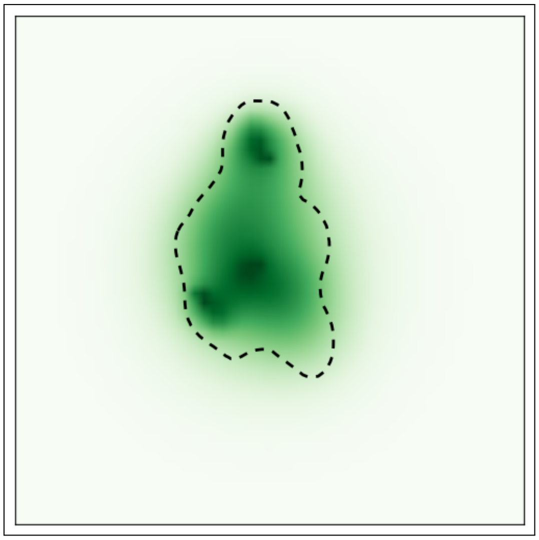

Figure 15: Home range for 3 animals. Background darkness shows scent quality.

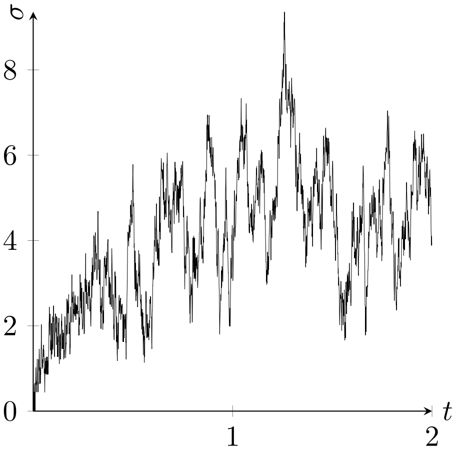

Figure 16: Stability of 3 animals over 2000 iterations.

4.4 Combined Olfactory and Quality Model

The next logical step is to combine the environment quality model with scent attraction. Consider two quality matrices: $Q_{\text{environment}}$, which is the static inherent quality of the environment; and Q_{scent}, which is the dynamic scent quality due to the other animals. They can be combined to a single quality matrix Q as follows

$$Q = (Q_{\text{environment}})^{m_1} + (Q_{\text{scent}})^{m_2}$$

The exponential weights m_1 and m_2 allow multiple qualities to be combined, as in Avgar et al. (2013).

For this model, $Q_{\text{environment}}$ is a random quality environment as in the first model, and Q_{scent} is the diffusing scent quality, as in the previous model. The combined quality matrix is then used to bias a random walk.

The results of the simulation are shown in Figure 17. The environment used is the same as in Figure 10, where the random environment model resulting in a splitting of the herd. With the combined model, the animals stick together initially despite the low environment quality gradient, due to the attraction of the scent. This is shown in the convergence plot in Figure 18, which lacks the initial peak seen in the environment quality model in Figure 9.

Of all the models, this seems to represent the most accurate model for home ranges. It is stable, results in the animals remaining in a herd, the animals respond to the environment, and it does not require a home range centre to be specified.

The inclusion of diffusion does come at a cost. The model's complexity is $O\left(nx^2\right)$.

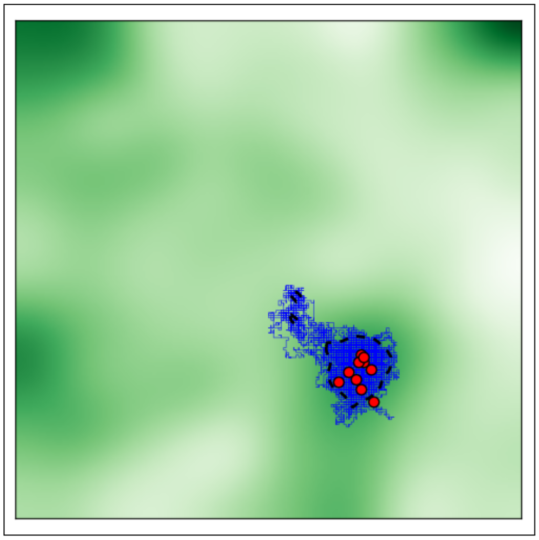

Figure 17: Home range settling for 8 animals, whose final positions are indicated with dots. Background darkness shows environment quality.

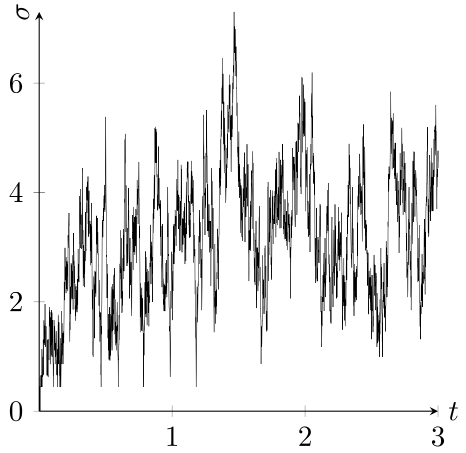

Figure 18: Stability of 8 animals over 3000 iterations.

5 Discussion

5.1 Summary

The goal of this project was to develop a mathematical model for animal movement behaviour that results in a realistic and stable home range. Four random walk models were analysed, and the best model involved combining environmental quality with scent diffusion.

5.2 Conclusions

All the models selected were stable, so biological relevance was the ultimate decider. The combined model captured the interaction between an animal and both other animals and the environment. The success of this model compared to the simple environment quality model suggests inter-animal interaction is critical for home range formation.

Herding, swarming, and flocking behaviour is well researched. These models are generally driven by interaction between animals, such as distance separation between schooling fish

(Hunter, 1966). Environmental factors are also considered, such as the effect of sunlight on plankton swarming (Buskey et al., 1996). From this, the modelling of a home range based on the environment and animal interactions seems entirely valid.

5.3 Limitations and Improvements

There are several limitations with these random walk models.

5.3.1 Parameterisation

The final combined model is parameterised by the quality weights m_1 and m_2, the diffusion coefficient D, and the source terms S_u and S_p. Additionally, parameters such as step sizes and environment dimensions are related to animal and environment properties. The kernel density estimation home range calculation is parameterised by a single parameter.

This seems to be a reasonable number, considering the complexity of the model.

The parameters for this project were chosen qualitatively, based on which numbers produced plots looking like home ranges. Fitting the model to real data would allow for better justification of the parameters chosen, as well as validation of the model itself against quantitative biological relevance.

5.3.2 Diffusion performance

The performance of the diffusion routine scales quadratically with environment size. As a result, the combined model which incorporated diffusion was extremely slow.

Python is an interpreted, single-threaded language design for prototyping over performance. If the combined model were to be used for large datasets, the diffusion code could be simplified, or rewritten in a better performing language. Alternatively, an external library could be used, as was done for the kernel density estimation.

5.3.3 Perception limit

In the final combined model, the animals are only able to perceive the quality of environment in adjacent cells. In reality, animals will be able to see and hear further than this. Accounting for this effect with the random walk model is difficult.

An alternative approach to animal movement known as state-space modelling (Patterson et al., 2008). A state-space model defines a redistribution kernel, with cells representing the likelihood of the animal moving there. These models enable the animal to evaluate a wider range of its environment when deciding where to move. This enables incorporation of more animal processes, such as memory and behavioural states, in addition to environmental drivers and animal interaction. These models are complex, but can be successfully parameterised (Avgar et al., 2013), and are likely to play a significant role in the future of animal modelling. I am unaware of any state-space model that incorporates diffusing interaction quality, which would be an interesting area for future work.

References

Avgar, T., Deardon, R., & Fryxell, J. M. (2013). An empirically parameterized individual based model of animal movement, perception, and memory. *Ecological Modelling, 251*, 158-172.

Buskey, E. J., & Peterson, J. O. (1996). The swarming behavior of the copepod Dioithona oculata: In situ and laboratory studies. *Limnology and Oceanography, 41*(3), 513-521.

de Vries, G., Hillen, T., Lewis, M., Mller, J., & Schnfisch, B. (2006). *A course in mathematical biology: Quantitative modeling with mathematical and computational methods.* Siam.

Benhamou, S. (1989). An olfactory orientation model for mammals' movements in their home ranges. *Journal of Theoretical Biology, 139*(3), 379-388.

Birch, C. P., Oom, S. P., & Beecham, J. A. (2007). Rectangular and hexagonal grids used for observation, experiment and simulation in ecology. *Ecological Modelling, 206*(3), 347-359.

Gibbon, J. (1977). Scalar expectancy theory and Weber's law in animal timing. Psychological Review, 84(3), 279.

Hunter, J. R. (1966). Procedure for analysis of schooling behavior. *Journal of the Fisheries Board of Canada, 23*(4), 547-562.

Okubo, A., & Levin, S. A. (Eds.). (2001). Diffusion and ecological problems: modern perspectives (Vol. 14). Springer.

Patterson, T. A., Thomas, L., Wilcox, C., Ovaskainen, O., & Matthiopoulos, J. (2008). Statespace models of individual animal movement. *Trends in ecology & evolution, 23*(2), 87-94.

Versteeg, H. K., & Malalasekera, W. (2007). *An introduction to computational fluid dynamics: The finite volume method.* Pearson Education.

Appendices

Appendix A: Selected Code

Complete code will be emailed to lecturer.

Software used:

- cPython 2.7.5 on Ubuntu 14.04 x64
- numpy 1.7.1 (for MATLAB-like linear algebra)
- matplotlib 1.2.1 (for MATLAB-like plotting)
- scipy 0.12.0 (for convex polygon, Gaussian filter, and kernel density estimation)

```
## Selected functions from code
# 1. Plotting mimimum convex polygon
# 2. Plotting kernel density estimation
# 3. Diffusing scent
# 4. Adding gaussian image environment quality



## imports
from __future__ import division
import matplotlib.pyplot as plt
import numpy as np
from scipy import ndimage #requried for gaussian filter
from scipy import stats #required for kde
from scipy.spatial import ConvexHull


class World(object):
    """a world consisting of an environment and some animals"""

    def plot_convex_ploygon(self, fig):
        '''Plot minimum convex polygon'''
        points = np.vstack([a.position_history for a in self.animals])
        hull = ConvexHull(points)
        for simplex in hull.simplices:
            plt.plot(points[simplex,0], points[simplex,1])
    def plot_kde(self, fig):
        '''Plot DKE contour'''
        history = np.vstack([a.position_history for a in self.animals])
        m1 = history[:, 0]
        m2 = history[:, 1]
        xmin = m1.min()
        xmax = m1.max()
        ymin = m2.min()
        ymax = m2.max()

        X, Y = np.mgrid[xmin:xmax:100j, ymin:ymax:100j]
        positions = np.vstack([X.ravel(), Y.ravel()])
        values = np.vstack([m1, m2])
        kernel = stats.gaussian_kde(values)
        P = kernel(positions)
        Z = np.reshape(P.T, X.shape)

        cutoff = (max(P) - min(P)) * 0.1 + min(P)
        levels = [cutoff]

        plt.imshow(np.rot90(Z))
        plt.contour(X, Y, Z, levels=levels)

class Environment(object):
    '''2D area explored by animal, containing 2D features'''

    def update_scent(self, n_iter=1):
        '''Explicit finite volume method'''
        dx = 0.03
        dt = 0.1
        SP = 0.1
        D = 0.0005
        aP0 = dx / dt
        aP = aP0
        aN, aS, aW, aE = D, D, D, D

        new_quality_scent = np.zeros(shape=self.size)

        for x in xrange(1, self.x_max):
            for y in xrange(1, self.y_max):
                new_quality_scent[x, y] = aN*self.quality_scent[x, y+1] \\
                    + aS*self.quality_scent[x, y-1] \\
                    + aE*self.quality_scent[x+1, y] \\
                    + aW*self.quality_scent[x-1, y] \\
                    + (aP0 - (aN+aS+aW+aS-SP)) * self.quality_scent[x, y]

        self.quality_scent = new_quality_scent / aP
        def add_gaussian_quality(self):
            '''Random gausian landscape'''
            gaussian_coefficient = 0.1 * np.average(self.size)
            noise = np.random.random(self.size)
            landscape = ndimage.gaussian_filter(noise, gaussian_coefficient)
            self.quality_basic = landscape
```