**Overview, Design concepts, and Details protocol**

The following text presents the Overview, Design concepts, and Details (ODD) protocol for agent-based models as described by Grimm et al. (2006) and Grimm et al. (2010). All simulations were constructed using Netlogo v6.1.1 (Wilensky 1999). Netlogo code and supporting GIS files for these simulations can be found in Supplementary Materials.

**Overview**

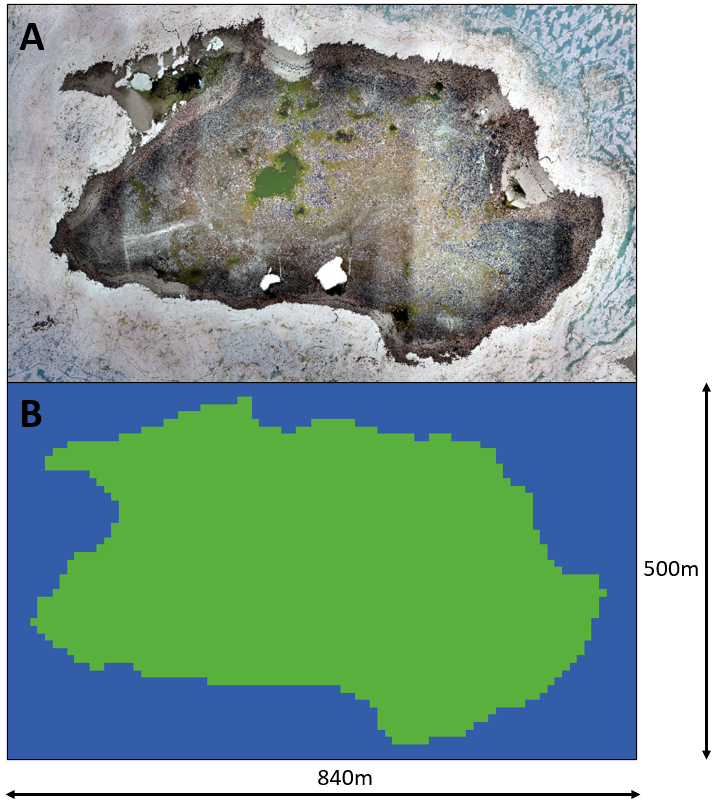
**1. Purpose**

The purpose of these agent-based simulations is to estimate the detection probability of polar bears (*Ursus maritimus*) on East Bay Island, Nunavut (64˚02’N, 81˚47’W) with a remote camera array. In summary, we simulate an array of 35 remotely operated cameras with known GPS locations and cardinal directions, as well as a single polar bear performing a correlated random walk through the island. Once the polar bear has left the island, we determine if the individual was detected by at least one camera, and use this to estimate detection probability of a bear given that it was present on the island.

**2. Entities, state variables, and scales**

Simulations are made up of bear and camera agents, while the landscape is represented by land and water patches. Camera agents are simulated based on a real array of 35 BTC-5HDPX trail cameras deployed in 2019. Camera agents have defined X and Y coordinates used to determine location, as well as manufacturer specified detection distances and angles, which is used to define a cone shaped detection zone (see section 6 Input Data). Cameras have a single state variable for bear detection, which records if a bear was detected by the camera. Bears are represented simply as an agent moving through the landscape, and do not hold any state variables. However, bears do have a literature-derived movement speed (see section 6 Input data) and a user defined turning-angle to produce a correlated random walk (see section 3 Process overview and scheduling).

The landscape is an explicit representation of East Bay Island using drone-acquired imagery and Netlogo’s *gis* extensions. For simplicity, we do not describe the technical details of the original drone imagery collection and orthomosaic creation. Briefly, still-imagery of East Bay Island was collected and used to create a 3cm RGB orthomosaic, which was used to create a simple landcover classification map of land and water cells in ArcMap v10.7.1 (ESRI, Redlands, CA). The landscape is represented as an 840×500m rectangle, composed of individual 100m2 patches. Other than representing water (blue) or land (green), patches in simulations hold no state variables and only serve as a surface for the placement of cameras and movement of bears (Figure 1). World wrapping was not used in these simulations. Simulations proceed in discrete one-second timesteps, thus polar bear movement speed is scaled to meters-second.

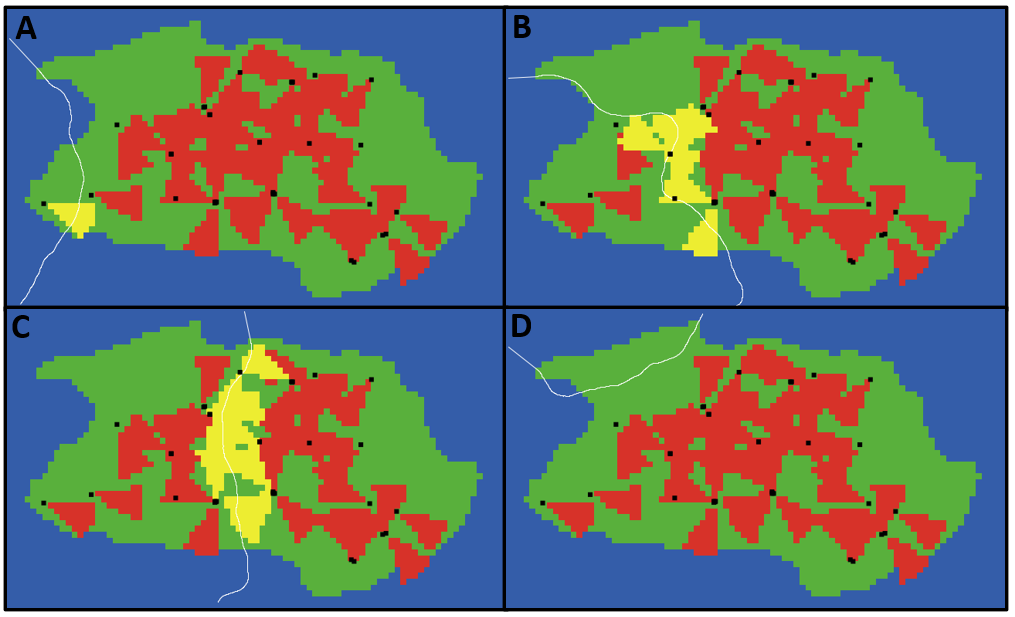


**Figure 1.** A) RGB orthomosaic of East Bay Island produced from drone imagery. B) Simulated landscape in Netlogo. Two types of patches are included, land (green) and water (blue).

**3. Process overview and scheduling**

Individual simulations begin by loading in drone imagery to create the East Bay Island landscape. Following this, the camera array is set up and detection zones are defined based on manufacturer specifications of camera detection distances and angles. These zones are represented by red patches. A single polar bear agent is produced at the edge of the world, and moves to the closest land patch to ensure the bear actually visits the island (rather than turning and leaving the world). For further details on processes that occur on the very first step of the model, see section *5. Initialization*.

The model proceeds in discrete one second timesteps. Each step, the bear first turns a direction left or right within a 15° field of view (to produce a random correlated walk), and moves forward 0.94m (based on literature derived movement speed). Bear agents have no preference for land or water patches, and movement speed does not differ between the different patch types. At each step of the simulation, cameras assess whether or not a bear is within their detection zone. If a bear is detected, the detection zone turns yellow (to assist the observer), and the camera records a positive detection. These steps are repeated until the bear reaches the edge of the world and cannot proceed any further. At this point, the individual model run is complete and detection data is output (Figure 2)



**Figure 2.** Example runs of polar bear walk paths and individual camera detections. A) The bear agent enters the world on the left and is detected by only a single camera. B and C) The bear takes a more central route through the island and is detected by several cameras. D) The bear enters the world at the top of the island and is not detected by any cameras.

**Design Concepts**

**4. Design Concepts**

*4.1 Basic Principles*

The basic principles regarding context and justification of these simulations are provided in section 1 *Purpose*.

*4.2 Emergence*

The only data output produced from these simulations is whether or not a polar bear was detected using the camera array deployed on East Bay Island.

*4.3 Adaptation*

Individual simulation runs only occur for a single bear visitation to the island, and the bear’s paths are based on a correlated random walk rather than occurrence of bird nests or other food resources. As such adaptation is not considered in these simulations.

*4.4 Objectives*

Agents do not have objectives in these simulations

*4.5 Learning*

Learning is not considered in these simulations.

*4.6 Prediction*

Prediction is not considered in these simulations.

*4.7 Sensing*

Cameras are capable of sensing the presence of a polar bear agent within their detection zone. This zone is defined as a cone of patches in front of the camera, with a 90m detection distance and a 41.9° detection angle to produce an approximate 43.7° field of view. We assume perfect detection of an individual bear, given that it is present within the detection zone. See section 6. *Input data*.

*4.8 Interaction*

Polar bear and camera agents interact when a polar bear walks within a camera’s detection zone. If a polar bear is detected, the camera asks the patches within the detection zone to turn from red to yellow, to inform observers which camera detected the bear.

*4.9 Stochasticity*

Stochasticity is only considered with polar bear agents. The initial position of the bear is randomly chosen at the edge of the world, and then the bear moves to the closest land patch. Once the bear arrives to land and begins it’s correlated random walk, turning direction is based on random number generation, whereby each step bears choose a direction to walk within a 15° field of the bear’s current travel bearing. The bear can leave the island, and simulations end once the bear reaches the edge of the world. Although we did not rely on empirical data to parameterize bear walk paths, we chose this value based on the ability to produce a relatively straight walking path.

*4.10 Collectives*

Collectives are not considered in these simulations.

*4.11 Observation*

Several observer processes are used during simulation runs. Polar bear walk paths are traced with a white line so that observers may easily see where the bear has walked, and camera detection zones are depicted as yellow or red for if a bear has been detected or not respectively. At the end of each simulation run, the model outputs if any cameras detected a bear. We then use this to calculate the proportion of simulation runs in which a bear was observed by a camera, given that the bear was on the island.

**Details**

**5. Initialization**

*Landscape Setup*

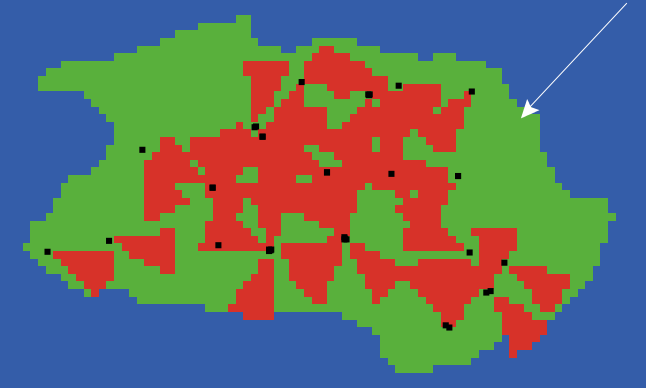
The model begins by clearing all current agents and patches from the interface, and by resizing the world to be 840×500m. Simulations use the *gis* extension to load landscape classification files, and create either land or water patches based on the landcover classification in the associated .shp file.

*Camera Placement*

Following landscape creation, 35 cameras agents are created (represented by small black squares) and the state variable *saw-bear?* is set to FALSE. Using Netlogo’s csv extension, real coordinates for the camera array are read in, and X/Y coordinates are assigned for each camera agent. Cameras move to their location, and then turn to face the cardinal direction for each real camera placement (whole circle bearing from 0-359). Following placement, cameras ask the patches in their detection zone to turn red so that observers can easily visualize the detection zone.

*Polar Bear Agents*

Following camera placements, a single bear agent is created (represented by a large white arrow) and placed randomly at the edge of the world. To ensure the bear actually walks on the island, rather than leave the system without visiting the island, the bear then faces the closest land patch and moves directly to this patch. This is the last step of the initialization process (Fig 3).



**Figure 3.** Interface view of simulations following initialization. Red patches represent detection zones for individual cameras, but notably some of these zones overlap between individual patches. The polar bear agent was originally created in the top right corner of the world, and then moved directly to the closest land cell.

**6. Input Data**

For simplicity here, we do not describe the acquisition and creation of the orthomosaic from drone imagery. For further details on this process, see Geldart et al. (in review). To produce the landscape classification files used with the *gis* extension, we manually clipped out a polygon of East Bay Island within ArcMap, and classified this as the island while areas outside of this were considered water. We exported the .prj and .shp files associated with the newly created layer, which were inputted using the *gis* extension.

Other input data used in simulations were the manufacturer specifications of the BTC-5HDPX trail cameras used in the current study. We specified a 90m detection distance and a 41.9° detection angle to produce an approximate 43.7° field of view (detection zone represented by red patches). Specifications found at <https://www.trailcampro.com/collections/trail-camera-reviews/products/browning-strike-force-hd-pro-x>.

Lastly, we defined polar bear movement speed as 3.4km-hr based on Jagielski et al. (2021), which was rescaled to m-s to match the temporal scale of each step.

**7. Submodels**

All model processes were described in full above, no complicated submodels were used in simulations.

**References**

Grimm V et al. (2006) A standard protocol for describing individual-based and agent-based models. Ecol Modell 198:115-126

Grimm V, Berger U, DeAngelis DL, Polhill JG, Giske J, Railsback SF (2010) The ODD protocol: a review and first update. Ecol Modell 221:2760-2768

Jagielski PM, Dey CJ, Gilchrist HG, Richardson ES, Semeniuk CA (2021) Polar bear foraging on common eider eggs: estimating the energetic consequences of a climate-mediated behavioural shift. Animal Behaviour 171:63-75

Wilensky U (1999) NetLogo. <http://cclnorthwesternedu/netlogo/> Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL