Ecological Informatics 15 (2013) 91–95

Contents lists available at SciVerse ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

Introducing AMV (Animal Movement Visualizer), a visualization tool for animal movement data from satellite collars and radiotelemetry

Devtulya Kavathekar, Thomas Mueller, William F. Fagan *

Department of Biology, University of Maryland, College Park, MD 20742, United States

ARTICLE INFO

Article history:
Received 11 April 2012
Received in revised form 26 November 2012
Accepted 10 December 2012
Available online 20 December 2012

Keywords:
Satellite collar data
ARGOS-GPS
Radiotelemetry
Migration
Nomadism
Range-residency

ABSTRACT

Researchers and wildlife managers often want to understand how landscape features influence an individual animal's movement. Animal movement data, whether derived from satellite collars, cellphone/hydrophone nets, or radiotelemetry studies, provide a range of information on movement including large-scale displacements and small-scale changes in orientation and velocity. To help contextualize such data and facilitate their interpretation, we developed a Java informatics tool, Animal Movement Visualizer v1.0. Built on the NASA World Wind v1.2 development kit, our free, downloadable software can display simultaneously the pathways of multiple animals moving against a backdrop of digital imagery of the Earth's surface, allowing researchers and managers to observe how multiple individuals move about with respect to one another in relative time. The program can accommodate datasets with irregularly timed relocations and relocation intervals that vary among individuals. The software displays the Earth's surface in a scalable way, facilitating visualization of specific landscape features. To illustrate possible uses for AMV, we provide a sample dataset for movement tracks of Mongolian gazelles (*Procapra gutturosa*) moving across steppe habitat.

1. Introduction

Animal movements, such as searching for food or dispersal to find a mate, are essential determinants of a species' autecology and geographic distribution. Such movements, whether they are short-term displacements within a home range or part of a long-distance annual migration, take place within a spatial context. Landscape features—ranging in size from individual trees and fence lines to whole mountain ranges—may influence those movements in many ways. Starting with radio-telemetry studies and progressing through to today's coupled ARGOS-GPS satellite collars, scientists have long relied upon the movement records of uniquely identified animals to understand aspects of species' resource use, territoriality, migration, and dispersal (Cooke et al., 2004; Moorcroft and Lewis, 2006; Morales et al., 2004; Mueller et al., 2011; Schick et al., 2008). The Movebank Consortium (Wikelski and Kays, 2012) has compiled such datasets into an online resource that includes additional informatics and visualization tools. As the spatial precision and temporal frequency of movement records have improved with technology, there are increasing opportunities for researchers and wildlife managers alike to hone their understanding of how landscape features drive and determine animal movements. For example, detailed analyses of cougar movements in southern California have helped identify the key elements necessary for successful dispersal through habitat corridors in fragmented landscapes (e.g., Beier, 1995). Similarly, Morales et al. (2004) analyzed relocation data from elk to identify environmental covariates that trigger switches between alternative movement states (e.g., feeding or relocating). In a comparative study, Mueller et al. (2011) demonstrated the key role that broad-scale variability in vegetation productivity can play in separating range residency from long distance movements such as migration and nomadism.

However, understanding how the spatial distribution of resources or the location of particular landscape features influence the movements of individual animals can be challenging. This is because researchers need to juxtapose both large scale displacements and small scale changes in orientation and velocity against the landscape backdrop. The need for such scalable visualization capacity is especially important for species, such as many ungulates, that undertake long distance seasonal movements during which local movement decisions may be influenced by resources or terrain features. Consequently, there exists a need for software tools that can readily transition from local to regional scales while still retaining the ability to simultaneously visualize complicated movement paths of multiple individuals. By facilitating visualization of movements across multiple individuals, such software could also help fill another important resource gap, namely the need to visualize *relative movement* as individuals vary their paths and positions in time and space with respect to one another.

To meet these needs we introduce AMV—Animal Movement Visualizer, a Java-based software tool that facilitates the visualization of animal movements in the context of manmade and natural landscape features. To build AMV, we combined existing technologies to create a new tool for researchers. To illustrate the potential utility of AMV, we

* Corresponding author. Tel.: +1 301 405 4672; fax: +1 301 314 9358.
E-mail address: bfagan@umd.edu (W.F. Fagan).

1574-9541/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ecoinf.2012.12.005

provide a link to an online demonstration of the tool in operation where we visualize movement data from Mongolian gazelles (*Procapra gutturosa*) equipped with ARGOS-GPS satellite collars (Mueller et al., 2011; Olson et al., 2009).

2. Materials and methods

2.1. Overview

We developed AMV—Animal Movement Visualizer using NASA World Wind version 1.3, a free, open-source Java software development kit (http://worldwind.arc.nasa.gov/java/). World Wind is used by many researchers, companies, and organizations in such diverse applications as geological mapping, search and rescue, and delineation of airspaces. Some demonstrations of applications based on NASA World Wind are available at http://goworldwind.org/demos/.

Combining NASA and USGS satellite imagery to render the Earth's surface in a scalable interface (similar to Google Earth), we developed an algorithm to display a time-lapse representation of locations and movement of uniquely identified animals on a 3D overlay. We implement multiple programming schemes that extract geospatial coordinates from a file of animal relocation data and display them in relative time, eventually removing plotted points as the animation concludes (i.e. as collar sequences conclude, animals die, etc.). We include multiple functionalities for the movement data such as speed adjustment, path trail between recaptures, first person perspective on an individual, and color changes to the visual components.

2.2. Software design overview

AMV is standalone Java application that does not require any backend server infrastructure. Total size for the software—including required libraries, picture files, and the JAR file results—is less than 10 MB, which is ideal for file compression and transfer via email and for posting on a website.

The software requires an arbitrary Excel .xls file containing animal coordinate data. Once loaded, our algorithm utilizing JavaExcelAPI (JXL) functionality parses through the .xls file to instantiate and store animal locality data and related information as virtual data structures. Eventually these structures are remembered and called upon by other algorithms coded in the Controller abstraction.

A speed bar, a box menu for user selection, showing the number of lines between relocations, a time view, and animal display functionality are coded on the left pane of the main frame. The right side displays the scalable globe view, a set of "Layers" tab checkboxes, and a bottom side bar that is called upon from the NASA WorldWind v1.2 view classes.

Right clicking an animal icon will bring up a dialog box allowing a user to change the color of the icon as well as the trail. Left clicking an animal icon will track only that individual animal's path in a first person view perspective. Right click dragging allows 3D vertical display showing actual height relative to terrain.

2.3. Input data format requirements

Input data is parsed from an Excel file with an .xls extension that must contain animal coordinate information with respect to our specifications. We require a sorted top down, time-increasing order for each individual animal without empty lines. From left to right, input data fields must contain ID # (an entry that uniquely identifies individual animals, such as a collar ID number, and which later appears as a label on the display), year, month (values are numbers 1–12), day (values are numbers 01–31), time (24 hour clock with hours, minutes, and seconds formatted as ##:##:##), latitude (decimal values ranging from −90 to 90), and longitude (decimal values ranging from −180 to 180). The first row, which is ignored by the program, should have field labels or remain blank. Data should start in cell B1 of the Excel spreadsheet. Adhering to this specification allows any coordinate locations to be traversed regardless of the type of dataset.

2.4. Software architecture

Fig. 1 provides a schematic of AMV and the interrelationships of its component algorithms.

2.4.1. NASA's Java World Wind v1.2

After exploring industry standard development kits (e.g. Google Earth, Google Earth Plugin [KML], GIS), we decided to use World Wind v1.2 developed by NASA in our AMV implementation after considering some bundles that were proprietary, closed-source software and others that were less efficient for our needs. AMV involves a relatively simple interface that is programmed efficiently (taking computer bit architecture type and memory overload into consideration) using the documented classes within the World Wind library. The open source nature of World Wind allowed us to resolve complications that arose when we needed to concurrently update the application framework and the World Wind OpenGL display to fit what we wanted the user to see. For parsing and manipulating Excel files, we opted for the Java Excel API library because it is straight forward, memory light, and well known.

2.4.2. Execution of AMV

Upon running AMV, a Java file chooser widget appears for input. A recursive prompt to the user requests a valid ".xls" file that conforms to design-specific content requirements. Instead of showing the main World Wind display at the start, we decided to force the user for input to ensure sequential ordered processes and proper utilization of World Wind. Tab delimited .csv options and a data filtering functionality are planned for future releases.

2.4.3. Clickable widgets

After encapsulating the animal data, the GUI changes, displaying an OpenGL World Wind display on the right, alongside two tabbed panes each housing pertinent functions for the user. Simplicity was the focus here, using Java Swing widgets to represent the core components of AMV. We use JSwing *JButtons* inside the main display of animals. These retain the aesthetic look of JLabels, but we created a JButton subtype to accommodate clickable functionality. Mouse hovering displays how the JButton is to be used for the first time user. Careful and extensive testing was done to ensure optimal failsafe features regarding software reliability.

2.4.4. Algorithms

We relied on existing Java sorting algorithms to keep track of sequential animal movements for the list of animals. Each animal representation was associated with a root data structure that housed all coordinate data (timestamp, position), polyline information, color, ID value, and other World Wind specific attributes. To save heap space and decrease memory allocation, static modifiers and mathematical optimizers were deployed. For example, we used numeric index assessors instead of calling performance-consuming internal methods of each data structure upon constant rendering. The SurfaceIcons and Polylines to OpenGL relied on three distinct algorithms depending on user input for visualizing animal movements (i.e., full lines linking all recapture locations sequentially, presenting only a specified number of line segments, or no lines). These algorithms update each individual's position both graphically and discretely in the data structure.

2.4.5. Time representation

A relative counter, starting from the beginning of the data set (this value was calculated from the primary algorithm upon upload of animal info) to the termination of every animal's movement, is displayed