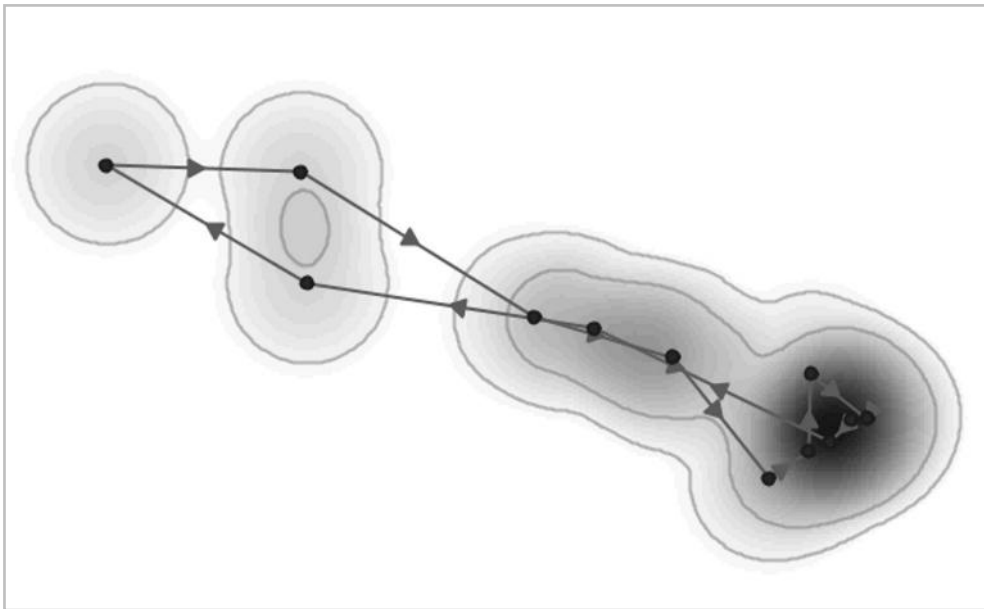# A User Manual to perform Home Range Analysis and Estimation with OpenJUMP HoRAE

By

Stefan Steiniger and Andrew J.S. Hunter

Version 16th Nov. 2011

Contact: Stefan Steiniger and Andrew J.S. Hunter are within Department of Geomatics Engineering, University of Calgary, 2500 University Drive N.W., Calgary, Alberta, Canada T2N 1N4; email: sstein@geo.uzh.ch, ahunter@ucalgary.ca

Purpose - This document describes how to use the Home Range Analysis and Estimation (HoRAE) Toolbox developed for the free GIS OpenJUMP. The toolbox was originally created for the evaluation of GPS collar data from grizzly bears (*ursus arctos*). However, as the base dataset for the home range estimation is an ordered point dataset, i.e. the GPS locations from the collar, GPS locations from other (slow moving) mammals obtained with GPS collars might be analysed too. We will describe the functions that allow to estimate home ranges, returning a geometry object for the home range, and functions for home range analysis. Currently the toolbox contains the following functions:

Home Range Estimation Functions:
- Minimum Convex Polygon (MCP),
- Kernel Density Estimation (KDE), methods: (a) href, (b) LSCV, and (c) ad-hoc (region split)
- Line-based KDE, scaled and un-scaled
- Brownian Bridges
- Line Buffer
- Local Convex Hulls (LoCoH), methods: radius r, k-1 nearest neighbours, alpha region.

Home Range Analysis Functions:
- Asymptote analysis for MCP, Point-KDE, Line-Buffer
- Calculate core areas
- Classify home range parts
- Derive the skeleton for a home range region.
- Analyze daily travel

Keywords - home range analysis, kernel density estimation, movement analysis, GIS, open source software

General Notes - As by now this manual is not aiming at describing the basics and details of home range analysis we refer the interested reader/newbie on the topic to the ABODE manual that we have made available here:

http://www.geo.uzh.ch/~sstein/ojhorae/laver_abode_user_manual_2005.pdf

Also the manual assumes that you are familiar with the use and terminology of Geographic Information Systems. If you have no previous GIS experience, then we recommend to read "*Desktop GIS: Mapping the Planet with Open Source Tools*" by Gary Sherman[1]. Also we recommend going through an OpenJUMP GIS tutorial as the HoRAE toolbox is an extension for OpenJUMP. OpenJUMP tutorials are available here: http://sourceforge.net/projects/jump-pilot/files/Documentation/[2]

---

[1] The desktop GIS book at Amazon: http://www.amazon.ca/Desktop-GIS-Mapping-Planet-Source/dp/1934356069
[2] The most actual tutorial is the OpenJUMP General Tutorial for OpenJUMP 1.4.2:
http://sourceforge.net/projects/jump-pilot/files/Documentation/OpenJUMP%201.4%20Tutorials/

Contents

1. Download and Installation

The Home Range Analysis and Estimation (HoRAE) Toolbox is an extension for the free and open source GIS OpenJUMP (www.openjump.org). Hence, for the toolbox the same conditions on supported operating systems, licenses and installation requirements hold as for OpenJUMP. We will give the details on those below.

Supported Operation Systems

OpenJUMP HoRAE should run on **Windows**, **Linux** and **MacOSX**. There is only one edition for all OS since OpenJUMP is using the JAVA[3] platform.

Free Software License and Exclusion of Warranty

OpenJUMP and also the HoRAE toolbox are distributed under the General Public License (GPL) Version 2.0 that can be obtained from here: http://www.gnu.org/licenses/gpl-2.0.txt. This license allows free distribution of the program at no charge, and allows the modification of the source code and the distribution of those modifications as long as the GPL 2.0 is retained and distributed as well. For more information about Free and Open Source GIS (FOSS4G) software and licenses we refer the interested reader to Steiniger and Hay (2009).

Please note that the paragraphs 11 and 12 of the GPL 2.0 state "THERE IS NO WARRANTY FOR THE PROGRAM", and contain the details to this statement.

The source code of the HoRAE toolbox should be contained in the java library named *movantool_XXX.jar* in OpenJUMPs "/lib/ext/" folder (with XXX as a place holder for some specifications in the file name). Renaming that file to "*.zip" should make the sources accessible. However, the sources can be also obtained from the authors, i.e. Stefan Steiniger or Andrew Hunter, by writing an email.

Download

We deliver the toolbox bundled with OpenJUMP to ease installation. To download the HoRAE toolbox go to:

http://sourceforge.net/userapps/mediawiki/mentaer/index.php?title=Movement_Analysis

From that wiki page you can download a *.zip file that contains the latest OpenJUMP HoRAE edition. The names of the file to download should be something like *openjumpXXXmovantools_MonthYear.zip*.

Before you can start OpenJUMP make sure that you have JAVA installed on your computer. OpenJUMP needs at least JAVA version 1.6; but depending on the utilized Sextante toolbox, newer JAVA version might be necessary. On Windows you can check if Java is installed by looking for a folder "C:\Program Files\Java" or "C:\Program Files (x86)\Java". If you are not having JAVA installed, then you can download it for Windows or Linux from the webpage http://www.oracle.com/technetwork/java/javase/downloads/. If you are a MacOSX user you may go to: http://support.apple.com/downloads/.

Installation and Starting

If you (i) have downloaded OpenJUMP HoRAE, and (ii) you have made sure that JAVA is installed, you can install and start the software. It`s done in 3 steps:

1. Unzip *openjumpXXXmovantools_MonthYear.zip* to a folder of your choice
2. Go to the subfolder */bin/*
3. Start OpenJUMP using the appropriate start file:

---

[3] JAVA is a programing language and a platform that functions as an abstraction layer. The JAVA platform is available for a range of operation systems. This way software programmed for/in JAVA can be run on all supported operating systems for which the JAVA platform is available.

o Under Windows use "openjump.exe" or in case it does not work use: "oj_windows.bat"
o For Linux use "oj_linux.sh"
o For MacOSX click on the application icon that shows a kangaroo.

Please note that OpenJUMP has by default assigned 64 MB of memory as minimum and 512 MB as maximum memory to work with. For larger point datasets or finer raster resolutions this amount may be insufficient. To increase the values, e.g. to allow OpenJUMP to work with 1024 MB (= 1 GB) of memory, you can modify the following files with a text editor: (i) Windows: openjump.ini when starting with openjump.exe, otherwise oj_windows.bat, (ii) Linux: oj_linux.sh, (iii) MacOSX: oj_linux.sh. Search in the file for the letter sequence `–Xmx512M` and replace it with `–Xmx1024M` . However, you cannot assign more memory than your computer has physical RAM, and also older JAVA versions have a memory assignment limit of 2GB.

Using the HoRAE Toolbox with a Newer OpenJUMP Version

It may be possible that the OpenJUMP HoRAE edition you downloaded does not come with the latest version of OpenJUMP. You can check what your OpenJUMP version is by choosing in OpenJUMP the menu entry [**Help>About...**]. In case there is a newer version of OpenJUMP and you want to use the toolbox with that version, then you would need to download OpenJUMP Plus edition that contains the Sextante toolbox. We need Sextante because all most raster functions use the Sextante framework for processing and raster analysis. The OpenJUMP Plus edition should also contain some other useful plugins for example the Print plugin and the PostGIS plugin, of which the latter can be used to connect to a PostgreSQL/PostGIS database. The latest versions of OpenJUMP GIS can be obtained from here: http://sourceforge.net/projects/jump-pilot/files/OpenJUMP/ or from the OpenJUMP Wiki http://sourceforge.net/apps/mediawiki/jump-pilot/index.php?title=Main_Page which is also the best source for learning about OpenJUMP.

2. Loading Data

With OpenJUMP HoRAE tracking/movement data as well as other types of geographic data (i.e. mapping data) can be loaded for analysis and map creation. We will describe the supported data formats and the requirements for the movement data. However, we will start with loading an example dataset.

Loading a Tracking Sample Dataset

OpenJUMP HoRAE should come with a sample dataset. It contains about 14 artificial grizzly bear GPS points. "Artificial" refers here to the coordinate of the points, not to the characteristics of the points to each other or the time information. The real location was disclosed due to data restrictions that should ensure wildlife protection. You can find the dataset in the subfolder "/testdata/" and the file with the GPS data named: "gb_tracking_sampledata.jml". The dataset is stored in OpenJUMPs own *.jml format, which is a special GML format. But, as we will outline below, it is also possible to load points stored in *.shp files.

To load the dataset go to menu entry [**File>Open...**], select the symbol for "**File**" in the left sidebar, and browse to the *testdata* subfolder. Select the file "gb_tracking_sampledata.jml" and click the [**Finish**] button. You should see now the image shown in Figure 1.
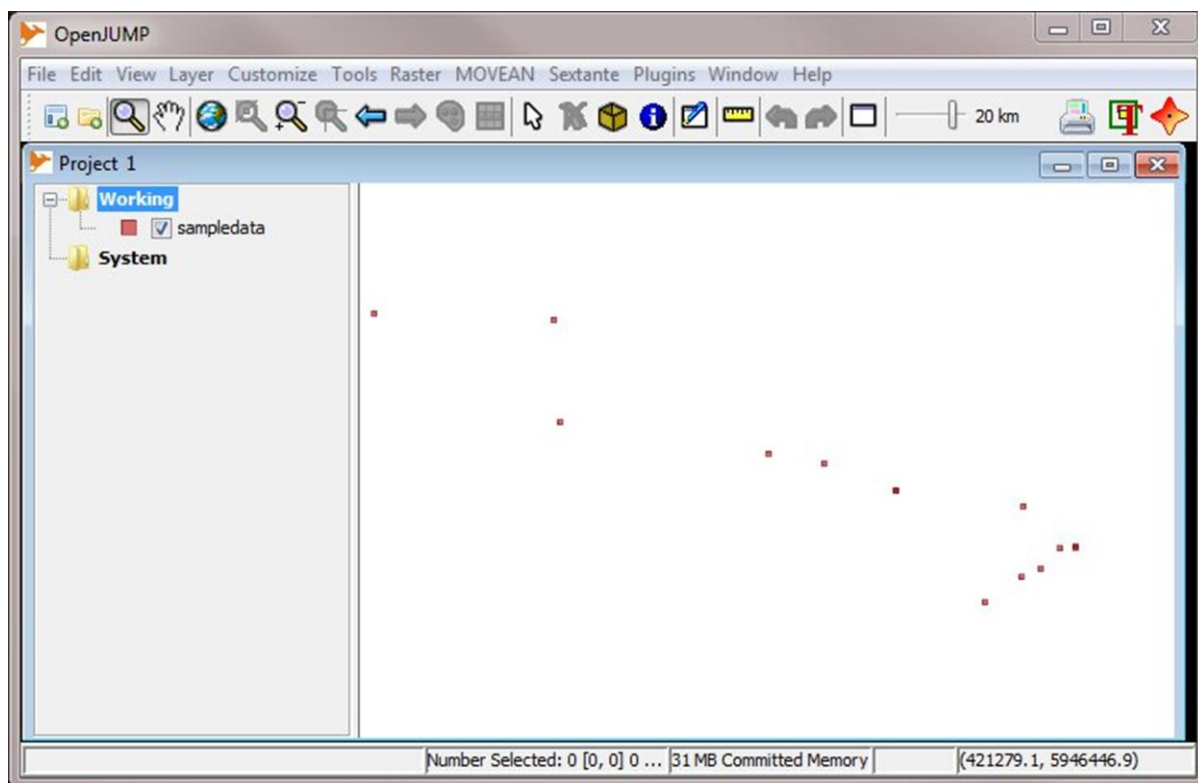
Figure 1. OpenJUMP HoRAE with the sample data loaded. The menu "MOVEAN" contains all the functions for home range estimation and analysis that are added by the toolbox to OpenJUMP.

Please note that OpenJUMP in its current version (v 1.4.2) does not care about map projections or coordinate systems. OpenJUMP will assume that all data loaded are in the same coordinate system. For more information please see the Notes section.

Supported Spatial Data Formats for Tracking Data

The tracking data to be evaluated in OpenJUMP HoRAE should be stored as point data. HoRAE can not read GPS data directly from a GPS receiver. Hence, the tracking data needs to be readout out from the receiver with another software[4] and prepared for loading in OpenJUMP. However, there exists a *GPS Extension* for OpenJUMP that can load data from a receiver, please see the following link for details: http://sourceforge.net/apps/mediawiki/jump-pilot/index.php?title=GPS_Plugin .

Besides the OpenJUMP specific *.jml format (i.e., JUMP GML) used for the sample dataset, other data formats are supported by OpenJUMP. Among those are: (i) Shapefiles (*.shp), (ii) Feature Manipulation Engine (FME) GML (*.gml), (iii) the Well Known Text (WKT) Format (*.wkt), and (iv) the Comma-Separated Value (CSV) Format (*.csv). Note that the latter format is provided by an OpenJUMP plugin for HoRAE. Similarly the reading of data in *.dxf format is possible if the specific plugin is added (see the Notes section).

Tracking Datasets Requirements

The point datasets that should be analysed need to fulfil some requirements:

- First, the point coordinates [x,y] should be in a mapping projection, such as UTM. The coordinates should not be in geographic coordinates, i.e. not in latitude and longitude in degrees. An indicator for lat & long values in degrees is when the first digits contain only values between 0 and 360. If your data is stored in such lat & lon values they should be

---

[4] A GIS software that may support reading your GPS point data from a receiver is Quantum GIS (www.qgis.org). You may also consider QGIS for creating maps of the home ranges for presentation purposes.

6

transformed into UTM values, either with OpenJUMPs Coordinate Transformation Service (CTS) Extension or for instance with Quantum GIS.

- Second, depending the type of home range estimator that should be used different requirements exists on the attributes that describes the observation time for each location. For requirements for a particular method we refer to the Section X, where the use of the function is described. In general each location (i.e. point) should be described by the following attributes:

  o (i) animal ID,

  o (ii) location ID in order of the observation sequence,

  o (iii) time of observation: the time may be needed in different granularity – i.e. year, month, day, hours, minutes, seconds – with each time unit (y,m,d,h,s) stored as a different attribute with the data type "Double".

  However, not all methods require all attributes. For instance the *Minimum Convex Polygon* (MCP) estimator only needs the animal ID attribute.

Important Note: All the functions have been developed to analyse grizzly bear movement data. Grizzly bears are hibernating and hence, we had only datasets with locations from spring to autumn of one year, and no movement tracks that spread over two years. Currently it is unclear how the analysis functions of the toolbox react if movement data are analysed that extend over two or more calendar years. However, a visual plausibility analysis should help to detect calculation errors.

Using Other Types of Geographic Data and Analysis

Since the home range toolbox is based on the GIS OpenJUMP, other types of data can be added in the background to see the context of observations and animal movements. OpenJUMP is primarily a vector GIS. Hence, it focuses on the display and analysis of vector data. So, if you have land use data in vector format, e.g. Shapefiles, then you can load that data as well. Also, the loading of areal or satellite images is possible. OpenJUMP can load TIF, PNG and JPEG images (but not all types of TIFF)[5], and under Windows MrSID and ECW images can be displayed as well. However, for image data it is important that they are in the same map projection as your vector data to ensure proper alignment of the different datasets.

Finally, when using OpenJUMPs overly analysis functions, the generated home ranges can be further analysed with available background data. For instance, the percentage of different types of land-cover such as wetlands and forest within an animal's home range could be evaluated. Or the overlap of home ranges of different individuals (e.g. two solitary grizzlies) can be calculated.

3. Visualization Functions

OpenJUMP HoRAE offers two visualization functions for tracking data. We will describe those below and then, in the next chapter, we will explain the home range estimation functions. From here on, all toolbox functions will be described with the following structure: (i) purpose of the function, (ii) working principle (if applicable), (iii) literature reference (if applicable), (iv) dataset attribute requirements, (v) input data, (vi) running the function, and (vii) expected output.

General Styling of Layer Objects in OpenJUMP

In the OpenJUMP toolbar as well in layer context menu you will find a 'palette' icon to [**Change Styles...**]. Use this function to define how you want to have points, lines and polygons rendered. The style dialog contains 5 tabs: (1) Rendering, (2) Scale, (3) Colour Theming, (4) Labels, and (5) Decorations. Most important is the *Rendering* tab dialog to change the line width, colours and point types. The *Scale* dialog allows you to define when at what scales a layer is displayed. The *Colour* theming dialog allows to colour polygons by their values – i.e. to

Style button

---
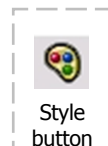
[5] Note that there are two options for loading TIF images. Using [File>Open File…] is the standard way. But if images should be analysed, then [File>Open...] and chosing "Sextante Raster Image" in sidebar on the left is needed to load image data.

generate choroplete maps. Different types of attribute classifications are available. The *Labels* dialog can be used to display attribute information as small labels in the map. For instance the number of an observation location or the names of rivers (if you have such data) can be displayed. Finally, with the functions of *Decorations* tab you can define the style of a line. For instance for the movement data it useful to change the style of line so that arrows indicating the direction of movement are shown.

Display Movement Tracks

*Purpose* - Displays the tracks of animals, by converting the observation points to lines (Figure 2).

Principle – The observation points are ordered and then connected by lines.

Dataset Attribute Requirements – Each observation point needs to have a unique location ID that should resemble the observation sequence. The unique ID is used to sort the points. For each point three time attributes for year, month, day need to be given as well. These time attributes need to be of data type 'Double' – i.e. not of data type 'Date' as used in Shapefiles.

Input – A point layer, with the following attributes for each observation point: (i) animal ID, (ii) location ID, (iii) observation year, (iv) observation month, (v) observation day.

Running the Function – use [**MOVEAN>Display Movement Tracks**]

- If you have the tracks of only one animal and therefore no animal ID attribute, you can create such attribute as follows (see also the Section "Calculation of the Home Range Area and Perimeter" below):

    1. Create a new field "animal ID" of data type "Double" using [**View/Edit Schema**] from the layer context menu (Don't forget to make the layer '*editable*').

    2. Fill the "animal ID" field with the same value for all points by use the function [**Tools>Edit Attribute>Auto Assign Attribute…**]. In the functions dialog: (i) select the target layer, (ii) select the attribute to fill, (iii) uncheck "Auto-increment", (iv) check that the assigned value (it can be one or any other value), and (v) press [ok].

- Option "one output layer per individual" – check this box if you have several animals in one dataset and the tracks for each animal should be in its own layer. This option is useful to split a dataset into several smaller ones for later home range calculations. Since the home range estimators will process the data for only one animal.

- Option "generate single lines" – there exists the possibility to have one long line that connects all points. However, this causes OpenJUMP to slow down during rendering (i.e. to become unresponsive), when zoom and panning operations are performed. Hence, it is better to return each connection from one observation point to another observation point as a separate line, i.e. to leave the box checked.
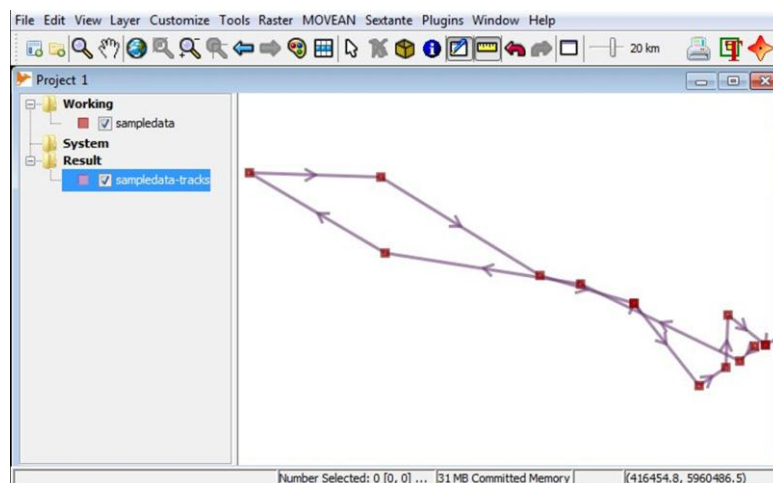
Figure 2. Using the function [MOVEAN>Display Movement Tracks] will generate lines from the point data. To see the movement direction, the "Decoration" options tab of the "Style" dialog were used (accessible via the palette button).

8

Output – A new layer of lines that connect the observed locations. The date information for start and endpoint are stored encoded in the attribute table for each line.

Display Day Travel Ranges

Purpose - Visualizes the observation points of the same day by creating a Minimum Bounding Rectangle (MBR) of the convex hull for each observation day. If there are only 2 points per day, then a circle is drawn (figure 3). Additionally the average daily travel distance is calculated. The information on the average travel distance can be used as input parameter for given

Principle – The function searches for all observations of the same day. Then it creates the so-called convex hull of those observation points. Then based on the convex hull a minimum enclosing rectangle is created to indicate the main travel direction.

Dataset Attribute Requirements – each observation point needs to have a unique location ID that should resemble the observation sequence. The unique ID is used to sort the points. For each point the observation day needs to be given. This day needs to be unique per year and should indicate the day of the year, e.g. Sun. 13th Nov. 2011 = 317th day. The information is used for grouping and calculation of the average travel distance per day. For this purpose using the day of the month as attribute would not work if the observations are spread over several months, and several days have the same "identifier" (e.g. 7th May = 7th June).

Input - A point layer, with the following attributes for each observation point: (i) location ID, (ii) observation day

Running the Function – Use [**MOVEAN>Display Day Travel Ranges**]

Output – A new layer of lines that connect the observed locations. A second layer showing rectangles and eventually circles, with each rectangle indicating the approximate area the animal traveled at a particular day. A separate graph is shown that displays the average daily travel distance. Finally, statistical data are presented in the output window on the average travel distance per day. The output window can be opened using the 'window' button of the toolbar.

Window button
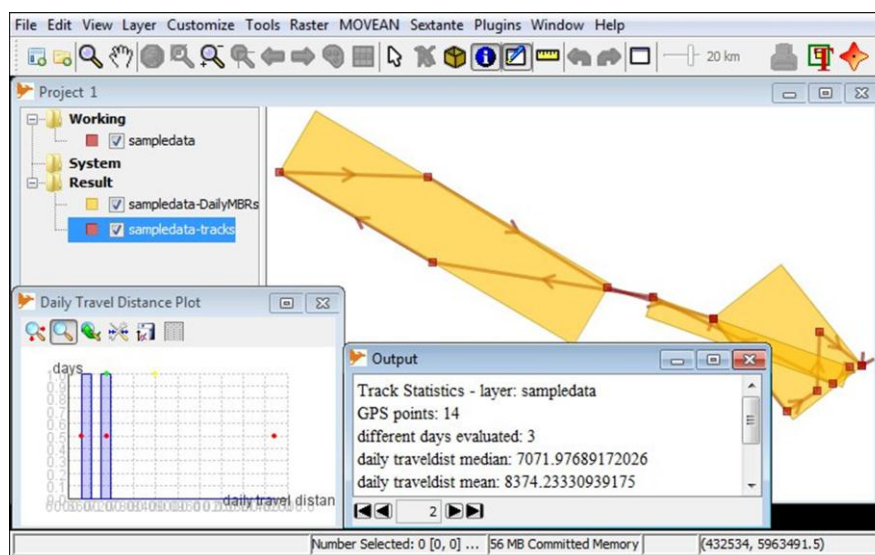
Figure 3. Rectangles created with [MOVEAN>Display Day Travel Ranges], indicating the daily travel distance of an animal.

4. Home Range Estimation Functions

With respect to the input data type we can classify home range estimators into point-based home range estimators and line- /or track-based estimators. Especially the latter are useful for GPS collar

data, as they can utilize additional information that comes with such type of data; for instance information on recording time and the possibility to calculate velocity vectors between subsequent GPS points. The toolbox contains point-based and line-based estimators that are described below. However, as for most toolbox users it will be important to calculate area and perimeter of the estimated home range polygon we will describe that procedure first. It will be the same for all estimators if a home range polygon is the result.

Calculation of the Home Range Area an Perimeter

Purpose – Calculation of area and perimeter of a polygon geometry, e.g. a home range polygon

Principle – Standard computational geometry algorithms; for instance the area calculation can be performed with the shoelace formula/Gauss' area formula[6]. The perimeter of an area is calculated by measuring the length of the outer shell plus the length of the boundary of holes.

Input – A layer with home range polygons (i.e. areal geometries)

Running the Function

1. Make the layer editable, that contains the polygons, by choosing [**Editable**] from the layer mouse context menu (i.e. right mouse click on layer name in Windows – Figure 4)
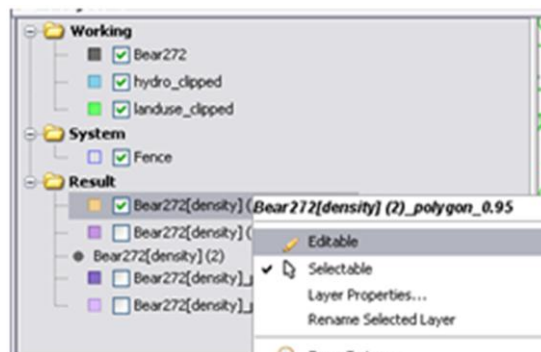
Figure 4. Making a layer editable.

2. Create new attributes 'area' and 'perimeter' by
   a. Choose [**View/Edit Schema**] from the mouse context menu
   b. Click in the last field on the left side (i.e. the *Field Name* column) and enter 'area' (Figure 5)
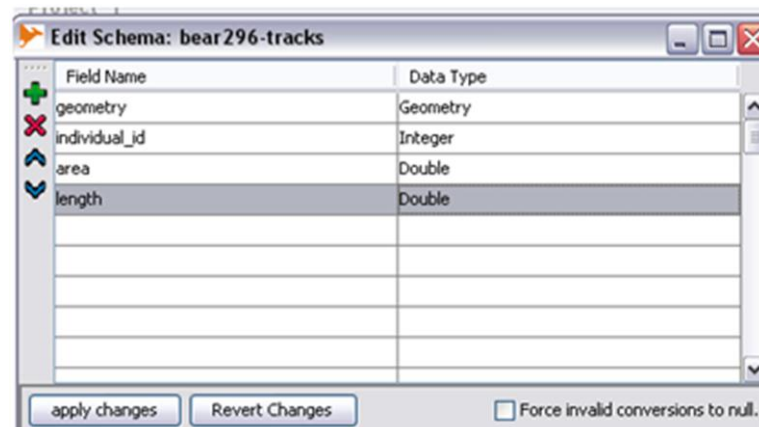
Figure 5. Adding area and length (i.e. perimeter) attributes to the layer.

---

[6] The Shoelace algroithm is used in the JTS Topology Suite Ver. 1.13, which provides the geometric models and algorithms underlying OpenJUMP horae.

  c. Click on the corresponding filed on the right side and chose 'Double' in the *Data Type* column

  d. Do the same for perimeter; choosing 'Double' as data type as well

  e. Press [**apply changes**], and close the dialog

3. Use the Matrix-Icon button from the toolbar, to display the attribute table for the currently selected layer. You should see the new columns 'area' and 'perimeter' there and their fields should be empty.

4. Go to the menu [**Tools>Analysis>Edit Attributes>Calculate Area and Lengths...**] to open the area and length calculations dialog (Figure 6)

Attribute Table button
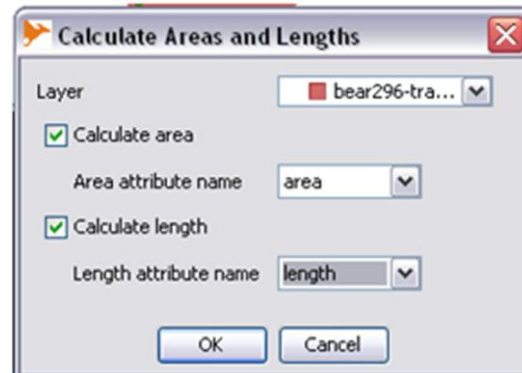
Figure 6. The dialog for area and length calculation.

5. Set the correct Layer with your data, and chose the attributes that should be filled with the area and length values; then press [ok]

6. Now, when you open the attribute table for the layer again, then you should see the area and perimeter fields filled.

7. Note, the sum for area or perimeter of all polygons (or lengths of lines) that are in a layer can be obtained using [**Tools>Statistics>Layer Statics...**]. To retrieve the sum of values for attribute data for a single layer you can use [**Tools>Statistics>Layer Attribute Statistics...**].

Minimum Convex Polygon (MCP)

Purpose – Create home range polygons.

Principle – Creates the convex hull of all input points. The convex hull is a standard computational geometry algorithm[7].

Literature References – Burgman and Fox (2003), Nielsen et al. (2007), Harris et al. (1990)

Dataset Attribute Requirements - An attribute which contains the ID of an animal, so that several animals can be processed within one file. To generate such an ID attribute see the section on "Display Movement Tracks" or use [Tools>Edit Attributes>Auto Assign Attribute…] directly.

Input – A point layer, with an animal ID attribute for each observation point.

*Running the Function* – Use [**MOVEAN>HRE>Calculate Minimum Convex Polygon**]

  Check the box for "one output layer per individual", if you have several animals in one file and want to have the MCP for each animal in a separate layer.

  Note, that we did not implement outlier removal techniques as described in Harris et al. (1990).

---

[7] A description of "Convex Hulls" on Wikipedia: http://en.wikipedia.org/wiki/Convex_hull

Output – A layer with the convex hull polygon (Figure 7).
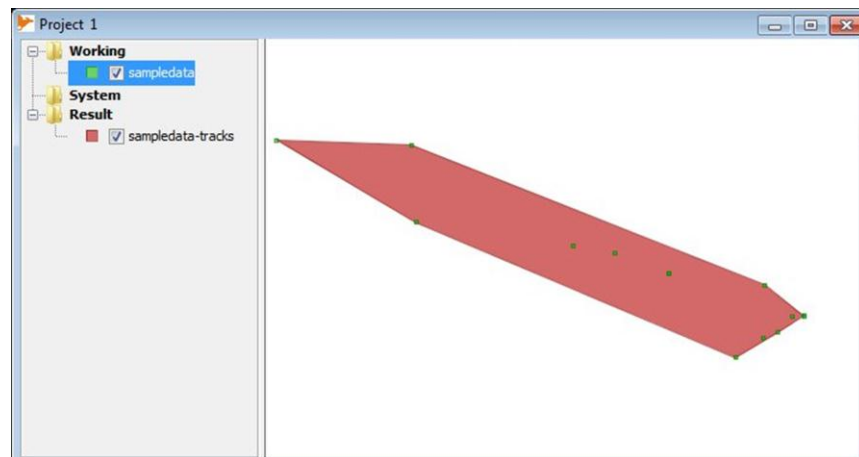
Figure 7. The convex hull of animal locations created with [MOVEAN>HRE>Calculate Minimum Convex Polygon].

Kernel Density Estimation (KDE) - with Parameter Estimators: h_{ref} , LSCV, and ad-hoc - and Creation of Probability Contours from a Density Raster.

Purpose - Create home range polygons.

Principle – A weighted kernel function is moved over a raster and point values summed. The shape and the size of the kernel function, the latter often called "bandwidth h", will influence the result. To determine the 'optimal' bandwidth h several rules of thumb have been developed that can allow to determine h in an automated or semi-automated way. Automated methods for bandwidth estimation that we have implemented are h_{ref}, Least Squares Cross Validation (LSCV), and ad-hoc.

In general the KDE method will return a raster with each grid cell containing a "density" value, and not home range polygons. To derive the home range polygons a contouring algorithm needs to be applied. Assuming that density can be related to probability it can be said that a grid cell (i.e. location) with some density value reflects a probability that the animal can be found in that cell. If the value of a cell is zero, the animal is unlikely to be found there. Hence, if we want the area that has a 95% probability of encountering the animal within, we will calculate a contour line that encloses 95% of the density of all cells (volume).

Literature References – among others we used: Silverman (1986), Powell (2000), Kie et al. (2010). For the bandwidth estimation methods: h_{ref}: Silverman (1986), LSCV: Worton (1995), Sheather (2004), and ad-hoc: Berger and Gese (2007) and Jacques et al. (2009).

Dataset Attribute Requirements – An attribute that weights each location point for the density calculation. The weight value *w* should be between $w = 0 \ldots \infty$, however, the standard value for the calculation is $w = 1$. Weighting an observation point can be used to give some points more priority (e.g., the den of a bear), or to exclude points that could be outliers. To generate such a weight attribute see the section on "Display Movement Tracks" or use [Tools>Edit Attributes>Auto Assign Attribute…] directly. But you should set the value that will be assigned to 1.

Input - A point layer, with a weight value/attribute for each observation point.

Running the Function – Use [**MOVEAN>HRE>Point KDE>Point Kernel Density**] (figures 8 and 9) to create the density raster, and then [**MOVEAN>HRE>Create Probability Contours from Raster**] (figure 10) to create contours/home range polygons from that raster.

It is important to note, that the KDE method should be used in an iterative/explorative fashion to obtain a feeling about the selection of the bandwidth and its influence on the created density raster. Only this way an "appropriate" bandwidth, and subsequently home range, for each dataset may be found.
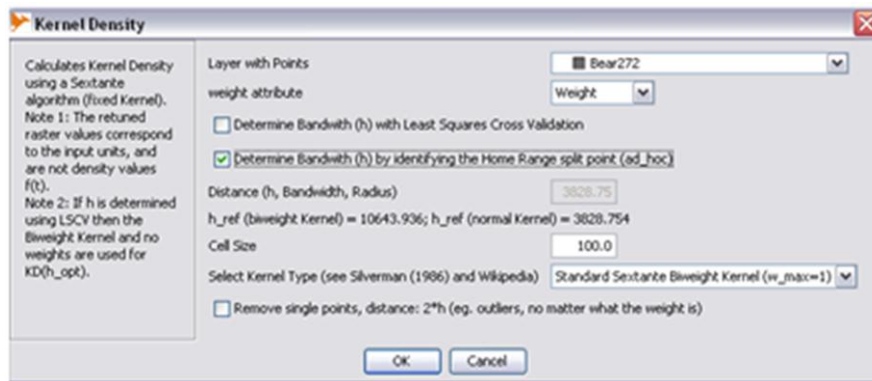
Figure 8. The function dialog for [MOVEAN>HRE>Point KDE>Point Kernel Density].

A - Using the function **Point Kernel Density**:

After setting the input layer and assigning the attribute with the weight values you need to decide how you want to set the **kernel bandwidth** h. The dialog gives you 3 options:

- To use Least Squares Cross Validation (LSCV) , check the corresponding box

- To use the ad-hoc method, check the corresponding box

- Give a bandwidth value by using the "Distance" field. To help you selecting a useful value, below the field are the values for h_{ref} for the "normal" Kernel and the "biweight" Kernel displayed for this particular dataset. (NB: The smaller value for the Gaussian/normal kernel should be used in a first step because the calculated contours will reveal more detail.)

Chose next the **Cell Size**, because the point dataset will be rasterized. The default value of 25m can be too fine, i.e. cause long processing times. The cell size should be chosen based on the extent of the point dataset. For instance if the points are distributed over 10km and the cell size is set to 25m we will retain a 400x400 cells raster with 160'000 cells. In this case choosing a cell size of 100m may be sufficient and will result in a 100x100 cell raster that can be faster processed.

Finally you need to select the **Kernel Type**. According to Silverman (1986) the kernel type has not so much influence on the results than the bandwidth selection, if the same bandwidth
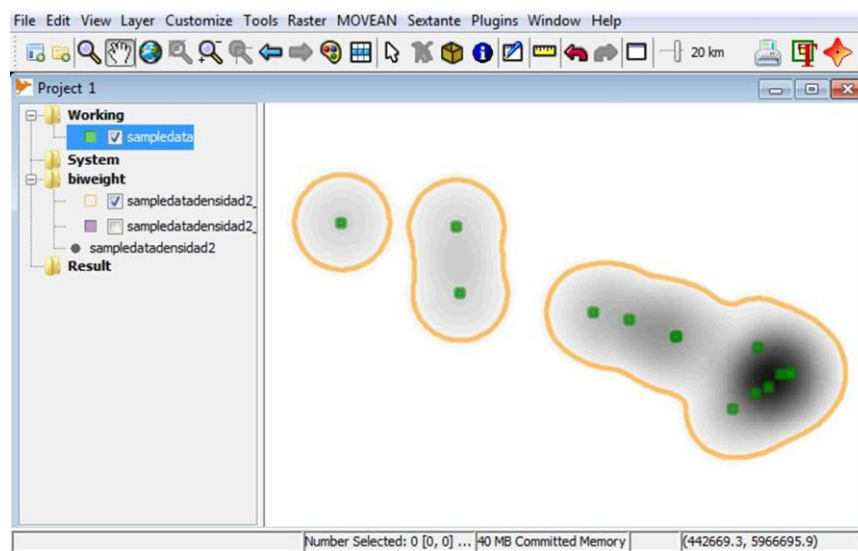
Figure 9. The density raster and the p=95% contour/home range for the sample dataset generated with the function [MOVEAN>HRE>Point KDE>Point Kernel Density], using a bi-weight Kernel, 4000m Bandwidth and 100m cell size.

is used. However, each kernel has a different maximum density value (w_max) leading to slightly different contours. If the bandwidth is determined using LSCV then the Biweight Kernel is automatically selected and no weights are used for the kernel density calculation (i.e. w=1.0).

A further option allows to **exclude outlier points** that are further than twice the bandwidth away from any other point in the dataset. Check this box if you feel comfortable, but we recommend in any case to compare the results with and without removal of outliers.

Note, if you consider to use the LSCV method to estimate the bandwidth h, then use [**MOVEAN>HRE>Point KDE>Display LSCV Function for Dataset**] to assess if the calculated bandwidth values make sense. The graph will also help to see if an optimization has taken place – i.e. a minimum value was found.

For problems that we encountered with the parameter estimation, in particular with LSCV, please see below.

B - Using the function **Create Probability Contours**:

If you have created a density raster with the Point KDE you can use this function to create density contours that can be considered as home ranges. Select the Point KDE output raster and then call the function (if no raster is selected the function cannot be used). The dialog for this function is displayed in Figure 10. The functions lets you define a probability, i.e. a value p = 0...1.0, for which the contour should be calculated. Typically the literature recommends a value of p = 0.95 (Powell, 2000).
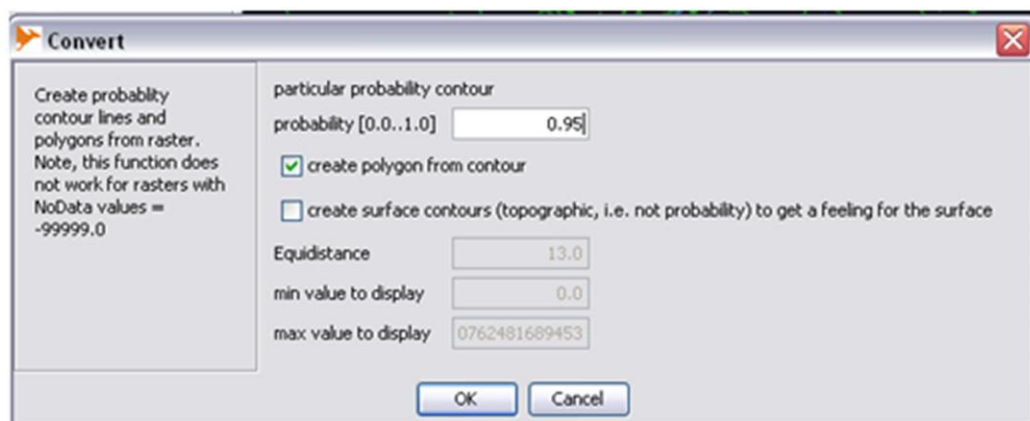
Figure 10. The function dialog for [MOVEAN>HRE>Create Probablity Contours from Raster].

Check the box next to ""create polygon from contour" if you want the wished contour (i.e. the 95% contour) returned as a (home range) "polygon" in a separate layer.

For a better visualisation of the topography of the density grid, it can be also useful to check the "create surface contours" option. Based on the generated contours it may be better possible to identify areas with a high or low probability of use by the animal. Note, that the values for those contours are not limited to the range of 0...1, but are corresponding to the calculated density values. The value for the "Equidistance" field should be chosen based on how many contours one wants to be displayed – for instance: if the calculated "min value to display" is 0 and "max value to display" is 10, and 5 contours should be returned, then (10.0-0.0) / 5 = 2.0; i.e. the equidistance value should be set to 2.0.

Output – The function "Point Kernel Density" returns a raster layer with density values. However, the retuned raster values will correspond to the input units, and are not (normalized) values of the density function $f(t)$. The function "Display LSCV Function for Dataset" presents a graph of the LSCV function, and displays calculated values for h_{ref}, h_{LSCV} etc. in the bottom information field (flashing yellow if the calculations are finished. Finally, the function "Create Probability Contours from Raster" returns a layer with a probability contour and optionally another layer with polygons, which can be considered as polygons that inscribe the home range (or the core area; see below), if the probability value was appropriately chosen.

14

Notes on Parameter Choice – Different kernel functions have been implemented in the toolbox: biweight, triweight, Epanechnikov, Gaussian, triangular, uninform and cosine functions (see Silverman 1986). As noted by Silverman (1986, pgs. 43 and 86) changing the kernel function does not change the results as much as changing the bandwidth. The implementation also assumes an isotropic distribution, i.e. the bandwidth value h is equal for both coordinate directions x and y.

Several automated methods for identification of bandwidth h are provided in the toolbox: (i) the reference method follows Silverman (1986), $h_{ref} = \sigma_{xy} \cdot n^{-1/6}$, and assumes a normal distributed point cloud. (ii) The Least-Squares Cross Validation (LSCV) method returns h_{LSCV} derived using a step-wise minimization approach over a range of bandwidths. h_{LSCV} is set to the bandwidth that minimizes a relative loss function (Worton, 1995; Sheather, 2004). (iii) As an alternative approach we implemented the ad-hoc method used in Berger and Gese (2007) and Jacques et al. (2009). h_{ad-hoc} is the minimum bandwidth that retains a single home range polygon. If h_{ad-hoc} is reduced further the home range will split into two patches. Kie et al., (2010) consider this approach repeatable and defensible if the proper biological questions are asked *before* the analysis. In our implementation the algorithm determines at what value of h the p = 99.5% contour/region splits, whereas in Berger and Gese (2007) and Jacques et al. (2009) the p = 95% contour is used. Several authors (e.g. Powell 2000, Laver 2005) note that h may also be chosen based on expert knowledge, for instance known location error, or the radius of perception of an animal. As such, the user can opt out of the data-driven methods described and apply an expert knowledge derived value for h.

Failure of LSCV and Numerical Limitations: The LSCV approach failed for all datasets that we tested with more than 500 GPS points. h_{LSCV} was estimated, but in all cases was extremely small when compared with h_{ref}. As a result the estimated home ranges consisted of small circles around each GPS point. The potential for LSCV results to vary and possibly be not usable has been also noted by Sheather (2004), Hemson et al. (2005), Huck et al. (2008) and Kie et al. (2010). The problems may in particular be bound to datasets with more than 150 points and for clumped point distributions (Hemson et al., 2005). Aside from the LSCV failures we encountered numerical problems when calculating the LSCV score CV(h) in our implementation using JAVA and a 32-Bit operating system when following Worton's (1995) formulation. In particular the calculation of the expression e^x will result in erroneous values for CV(h) if x > 709 due to 32-bit precision limits. We note that this problem also occurs during probability calculations for the Brownian Bridge approach.

Line-based Kernel Density Estimation (Line-KDE) – scaled and unscaled

Purpose - Create home range polygons.

Principle – The operating principle is the same as for the Point-based KDE. That is, a weighted kernel function is moved over a raster and point values summed. However, for the line-based KDE the function is not moved from cell to cell, but rather along the recorded GPS track segments from the start-point to the end-point of each segment. We implemented a scaled and an unscaled version. If contour lines are created, then the unscaled Line KDE results in a buffer-like shape of the contours for a single segment. In the scaled Line KDE version the contours will have a bone-like shape (see Figure 11). In theory different kernel shapes can be used as for the Point KDE, however, in the current version we have implemented only the biweight kernel.

Similar to the Point KDE this function returns a density raster. Consequently, home range polygons are to be derived by applying the contouring function to the resulting raster.

Literature References – The algorithm is described in Steiniger and Hunter (2010). However during drafting of our manuscripts an article by Benhamou and Cornélis (2010) was published that presents a similar approach for a moving kernel.

Dataset Attribute Requirements – Each location point needs to have a unique identifier that is assigned according to the order of point acquisition (similar as for the "Display Movement Tracks" function). Based on that ID the points can be connected to a track/line. In opposition to the point KDE no weight attribute is needed since the weight is set internally to $w = 1$.

Input – A point layer, with a unique ID for each point that is given according to sequence of recording.
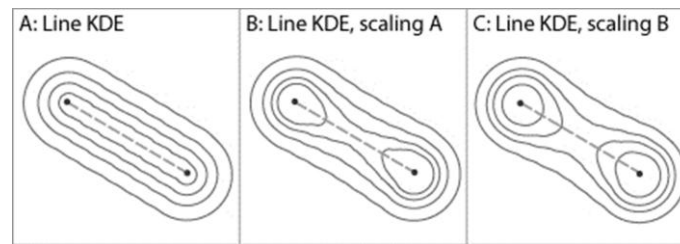
Figure 11. Contour lines for the Line KDE unscaled and scaled – with scaling variants A and B. Scaling B has a stronger scaling factor.

Running the Function – Use [**MOVEAN>HRE>Line KDE>Line-based Kernel Density for Movement Points**] (figure 12) or [**MOVEAN>HRE>Line KDE>Scaled Line-based Kernel Density for Movement Points**] to create the density raster. Then use the function [**MOVEAN>HRE>Create Probability Contours from Raster**] to create contours/home range polygons from that raster. A description of the latter function can be found in the section on the Point KDE function.
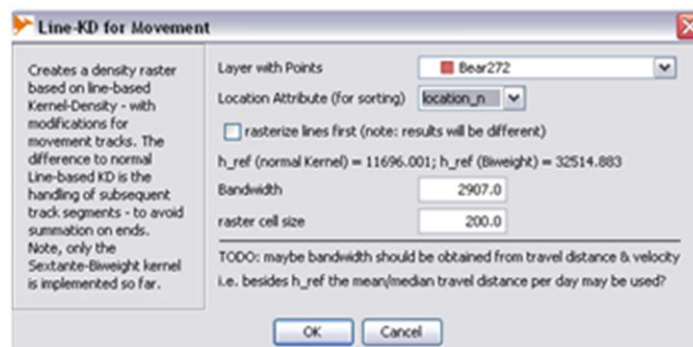
A - Using the function **Line-based Kernel Density for Movement Points**:

After calling the function set the input layer and then set the location ID attribute. Next you need to decide if lines should be rasterized first (all segments are rasterized in a first step) or segment by segment. The results for both methods will be different as described in Steiniger and Hunter (2010). We recommend leaving the box unchecked, i.e. use the segment-wise approach.

Then the bandwidth h needs to be set. The values for h_{ref} are displayed for the input dataset. The smaller value for the Gaussian/normal kernel should be used in a first step because the calculated contours will reveal more detail. However, an alternative is to use the median travel distance obtained with the function [MOVEAN>Display Day Travel Ranges].

Finally choose an appropriate cell-size value as described in the Point KDE section.

Figure 12. The dialog for the function [MOVEAN>HRE>Line KDE>Line-based Kernel Density for Movement Points].

B - Using the function **Scaled Line-based Kernel Density for Movement Points**:

This function provides almost the same options as the previous one. One difference is that the rasterization approach cannot be chosen and is internally set by default to using the "segment-wise" approach. The other difference is a new option for choosing the kernel scaling function. This function defines how strong the bone-shape will occur. By default a smoother scaling with factor values of [1.0 … 0.707] is set. Unchecking the box will use scale values [1.0 …0.5] that result in a stronger bone-like shape.

Output – The Line KDE functions both return a raster layer with density values. However, the retuned raster values will correspond to the input units, and are not (normalized) values of the density function *f(t)*. The function "Create Probability Contours from Raster" returns a layer with a probability contour and optionally another layer with polygons, which can be considered as polygons that inscribe the home range (or the core area; see below), if the probability value was appropriately chosen.

Notes on Parameter Choice – With respect to choosing the bandwidth *h* we provide the function [MOVEAN>Display Day Travel Ranges] that calculates the median of the average daily travel distance (MTD) based on the track information. A requirement for the calculation of h_{mtd} is that the recording day, e.g. the day of the year, for each GPS point is known. Alternatively, the reference bandwidth h_{ref} can be used. However, in certain instances h_{mtd} may be (much) larger than h_{ref}. If that is the case, then we recommend that users adopt h_{ref} to retain more detail in the resulting home range geometries.

The scaling function controls the (bone-)shape of the probability contours for each segment of the GPS tracks. We undertook testing using four point datasets obtained from grizzly bear GPS collars to compare the two built-in scaling functions; one with a scaling range of [0.5...1.0] and a second with [0.707...1.0] (Steiniger and Hunter, 2010). The tests did not reveal major visual differences for *p* = 95% contours. But further testing with additional datasets and data from different species is necessary before a conclusive statement can be made.

Brownian Bridges (BB)

Purpose - Create home range polygons.

Principle – The Brownian Bridge estimator is based on a probability model that describes the possible location of a particle that is constrained in space, i.e. consecutive GPS points, and time. The particle has a certain amount of time to get from location 1 to location 2, and whose movement is described by Brownian motion (Einstein 1905 in Turchin (1998)). For the home range estimation the GPS locations are considered as observed particle locations.

Similar to the KDE functions this function returns a density raster. Consequently, home range polygons are to be derived by applying the contouring function to the resulting raster.

Literature References – The method was proposed by Bullard (1991) and Horne et al. (2007) published an article on parameter selection using a "Liker" function. An implementation is also available in the adehabitat package for R (Calenge, 2006).

Dataset Attribute Requirements – Each location point needs to have a unique identifier that is assigned according to the order of point acquisition (similar as for the "Display Movement Tracks" function). Based on that ID the points can be connected to a track/line. Additionally a second "Time Attribute" is necessary that contains the recording time in seconds. At what point in time the "zero" value is set for the seconds is not important, as long as the time difference between two consecutive points can be calculated correctly in seconds. Hence, a value in seconds based on the first day of the year as "zero" point is possible if the track to be analyzed is recorded within that one year and not over two years.

Input - A point layer, with the following attributes for each observation point: (i) location ID, (ii) (relative) observation time in seconds.

Running the Function – The Brownian Bridge estimation contains three different functions. One function for the parameter estimation: [**MOVEAN>HRE>Brownian Bridge>Display Liker Function for BB**]. The two other functions to perform the BB calculations: (i) [**MOVEAN>HRE>Brownian Bridge>Brownian Bridge Home Ranges – Vector Segment Version**] for using the segment-wise rasterization approach (see our explanation above in the Line KDE section), and (ii) [**MOVEAN>HRE>Brownian Bridge>Brownian Bridge Home Ranges – Raster Version**] for using the rasterize-first approach. The estimation functions create a density raster. Consequently use afterwards the function [**MOVEAN>HRE>Create Probability Contours from Raster**] to create contours/home range polygons from the density rasters. A description of the latter function can be found in the section on the Point KDE function.

A - Using the function **Display Liker Function for BB**:

The BB estimation functions require, among others, two input parameters $sigma_1$ and $sigma_2$. Bullard (1991) and Horne et al. (2007) suggest that $sigma_2$ is set to the uncertainty, or positional error of the observation point, e.g. $sigma_2 = 30$ m for GPS collar data. Assuming that $sigma_2$ is fixed for the whole data set Horne et al. (2007) propose a likelihood function and optimization routine to estimate $sigma_1$. The implemented "Display Liker Function" implements this step.

After calling the function (figure 13) - the following parameters need to be set: the input layer with the points and then the location attribute needs to be chosen. Next the time attribute that contains the recording time in seconds needs to be set. However, in case that one wants to test with a dataset that does not have such information, then one can check the box for "equal time difference…". In this case it is assumed that the points are recorded in the same time interval. For the calculations internally is used a time difference of 10'000 seconds.

Then set $sigma_2$, which can correspond to the accuracy of your GPS data. For instance in general one can assume a GPS point accuracy of 30m. This is why we set 30.0 as default value. Now you need to define the range and number of points within that range for which the Likelihood function values are to be calculated. As the computation may need some time it is better to choose less number of grid points in the beginning – e.g. 100 or 200, and then later refine. The returned graph should contain a minimum value – which is the sought parameter $sigma_1$ that is somewhere in the middle of the graph. To achieve this the search range often needs to be iteratively refined by adjusting the start and end-values.
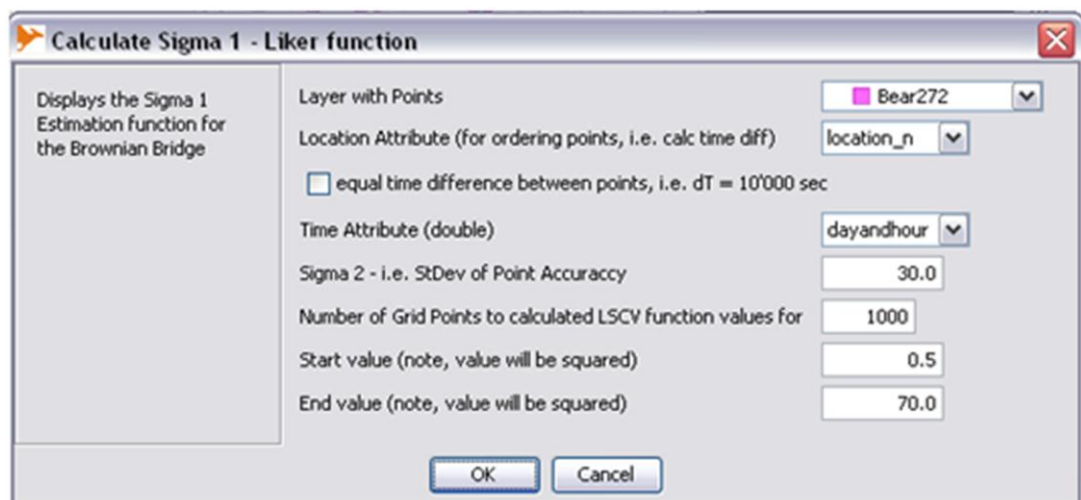
Figure 13. The dialog for the function [MOVEAN>HRE>Brownian Bridge>Display Liker Function for BB].

For the sample data delivered with the toolbox one can set the number of grid points to 200 and leave the range from 0.5 to 70.0.

After pressing [ok] the function then displays the Liker-Function graph (see also figure 15). Also in the lower left corner (blinking in yellow) the calculated minimum value for the function is given. For the sample data this will be $sigma_1 = 30.515$m with 200 grid point values, and $sigma_1 = 25.34$m with 1000 values.

We note, that our software does not determines the minimum based on an optimization routine as described by Horne et al. (2007), but by choosing the smallest value calculated from all the points (e.g. for 200 or 1000 points, depending on the what value that was set earlier for the number of grid points).

B - Using the function **Brownian Bridge Home Ranges** (segment-wise and raster version):

Both functions require the same input data and parameter settings (figure 14). Hence, set the input layer with the GPS points, then set the location attribute. Next you should uncheck the box for "equal time difference…" and set the time attribute. Use "equal time difference…" only

for testing purposes. Set sigma$_1$ (as obtained from the Liker function) and sigma$_2$ as used for the Liker function calculation. Finally, choose the raster cell size appropriately as described in the Point KDE section.
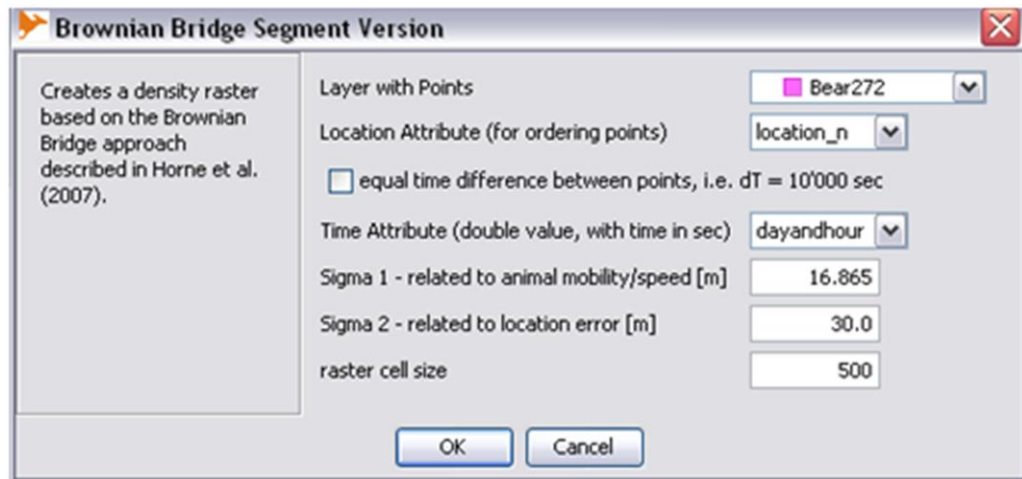

Figure 14. The dialog for the function [MOVEAN>HRE>Brownian Bridge>Brownian Bridge Home Ranges – Vector Segment Version].

Output – The "Display liker function" returns a graph with the likelihood function calculated and a minimum value that can be used for sigma$_1$. The Brownian Bridge functions both return a raster layer with probability values. The function "Create Probability Contours from Raster" returns a layer with a probability contour and optionally another layer with polygons, which can be considered as polygons that inscribe the home range (or the core area; see below), if the probability value was appropriately chosen.

Notes on Parameter Choice – The earlier mentioned numerical problems for Kernel Density calculations also occur during probability calculations for the Brownian Bridge approach, since using a normal distribution requires evaluation of e^x as well. The issue is particularly evident in the form of jumps when plotting the likelihood function (figure 15). As a result the parameter sigma$_1$ may be incorrectly identified as implemented.
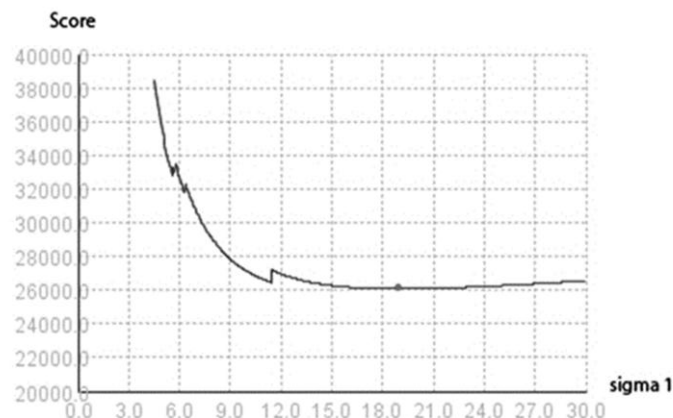
Figure 15. Reaching the numerical limits for 32-bit computers causes jumps in the graph for the Brownian Bridge score value calculations. The dot in the middle marks the minimum score value that determines parameter sigma$_1$.

Line Buffer

*Purpose* - Create home range polygons.

Principle – The method uses positive and negative buffer operations on the movement trajectories. Negative buffers are applied to remove portions of the home range stemming from exploratory walks by the animal, i.e. occasional sallies. The method can deliver a simple home range polygon, but is also able to create density surfaces based on buffer overlap counts.

Literature References – no earlier reference, but applied in Steiniger et al. (2010).

Dataset Attribute Requirements – Each location point needs to have a unique identifier that is assigned according to the order of point acquisition (similar as for the "Display Movement Tracks" function). Based on that ID the points can be connected to a track/line. For a more reasonable selection of the buffer radius time information is helpful by using the average travel distance per day. To utilize this option for each point the observation day needs to be given. Import is that this day should be unique as described in the "Display Day Travel Ranges" requirements.

Input - A point layer, with the following attributes for each observation point: (i) location ID, and if possible: (ii) observation day

Running the Function – Use [**MOVEAN>HRE>Line Buffer-based Home Ranges**] (figure 16) to obtain the home range polygons. If you select the option to create a density grid, then you can use [**MOVEAN>HRE>Create Probability Contours from Raster**] to create contours/home range polygons from the generated density raster.
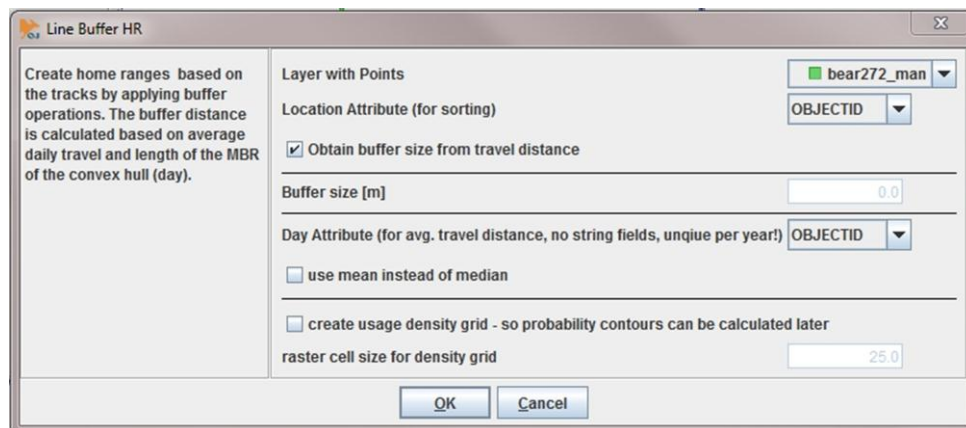
Figure 16. The dialog for the function [MOVEAN>HRE>Line Buffer-based Home Ranges].

After calling the function set the input layer and the location ID attribute. Then set the buffer size value. However, if you also have attribute data for the day of recording for each point you can check the box for "Obtain buffer size from travel distance". This will allow you set the attribute for the recording day and to check the box for "use mean instead of median". The latter option was added to see how different the results would be if the mean of for all daily travel distance is used instead of the median value. However, we recommend to leave the box unchecked as the 'median' is more robust than the 'mean' against outliers in the daily travel statistics.

If you check the box for the option "create usage density grid…". Then the function also returns a density raster and you need to provide a raster cell size. For appropriate selection of the cell size see the Point KDE section.

Output - The function returns a range of results. If no additional options are checked it will return the movement track, the final home range polygons, as well as the intermediate processing results for the positive buffering and the negative buffering (called 'erosion'). If the buffer size is calculated based on the daily travel, then the associated minimum bounding rectangles (one for each) are displayed as well and the used buffer value (the median travel distance) is given – also in the output layer name. Finally if the density grid option was checked a raster layer is returned that can be used with the contouring function.

Notes on Parameter Choice – We note that the buffer size for the erosion operation is derived by taking the input buffer size and multiplying it with -1.10. Erosion is applied to remove sallies.

Local Convex Hulls (LoCoH) – Methods: radius r, k-1 nearest neighbours, alpha region

Purpose - Create home range polygons.

Principle – The Local Convex Hull (LoCoH) method is essentially a localized version of the MCP approach with some post-processing, i.e. unioning of the hulls. Getz et al. (2007) present three different methods to define the neighbourhood of a point to construct the local hulls: (i) a so-called adaptive LoCoH uses all those points within sum distance of the center point for the local convex hull where the sum of distance between the points and the center point is smaller than a distance a, (ii) a k-nearest neighbour approach using all k nearest points from the center point, and (iii) a radius-based LoCoH using all points within a circle of radius r from the center point to build the local convex hull.

Literature References – The approach was proposed in Getz and Wilmers (2004) and refined in Getz et al. (2007)

Dataset Attribute Requirements – No attribute requirements, but in opposite to the MCP function it is assumed that the location data are only from one individual.

Input - A point layer with the observed locations.

Running the Function – Use [**MOVEAN>HRE>LoCoH Home Ranges**].

> In the function dialog (figure 17) choose the input layer with the observations. Then select one of the three methods that should be used to define the neighbourhood for each local convex hull. Provide the corresponding input parameter in the field. Finally you can select if isopleths/contours should be generated as output as well.
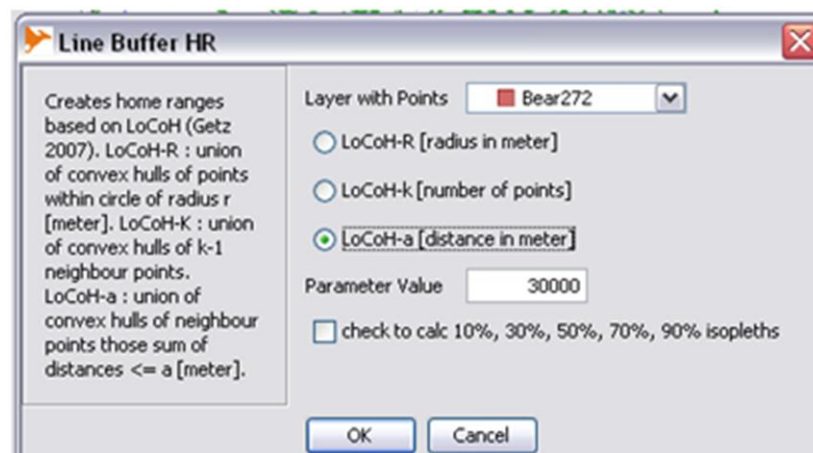
Figure 17. The dialog for the function [MOVEAN>HRE>LoCoH Home Ranges].

Output – The function should return three layers: (i) a layer that contains the home range polygons (i.e. the 100 percent isopleth), (ii) a layer that contains all generated local hulls (called single hulls): an intermediate processing result, and (iii) if the isopleth option was selected, then the 10%, 30%, 50%, 70% and 90% isopleths will be included in the result layer as well.

Notes on Parameter Choice – The toolbox doesn't include automated methods for parameter estimation, hence, the parameter must to be set manually. Generally, an iterative approach based on qualitative (e.g. number of holes in the home range) or quantitative criteria (e.g.: values for area and shape index) should be used. Getz et al. (2007) make suggestions for initial parameter values.

Home Ranges Calculated for Point KDE, Line KDE and BB for the Sample Dataset

In figure 18 we show results for different home range estimators for the sample dataset that comes with the toolbox.
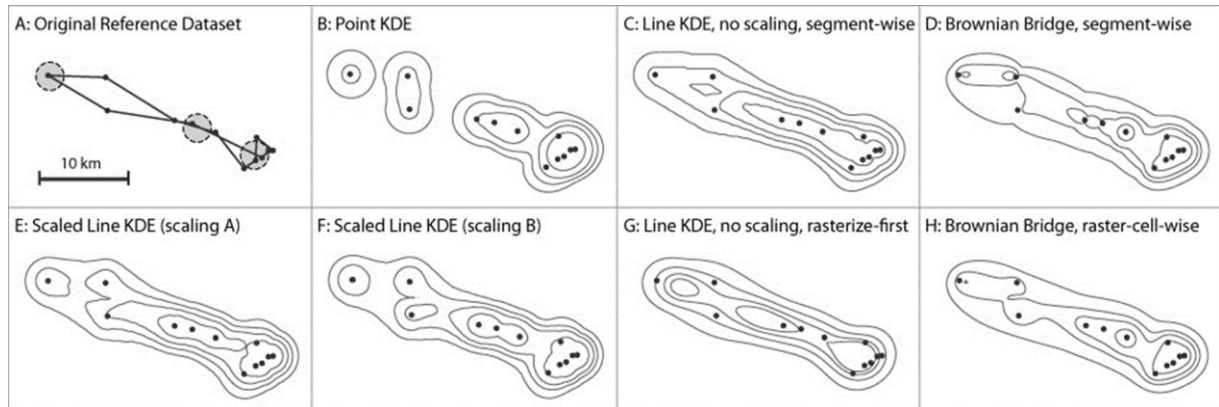
Figure 18. Reference point dataset and 25, 50, 75 and 95 percent contours for probability of utilization with point–KDE, line-KDE (rasterize-first and segment version), scaled line-KDE (scaling A and B), and Brownian Bridge. The dashed circles in image A mark important configurations, i.e. i - acute angle, ii - parallel tracks, iii - crossing tracks.

5. Home Range Analysis Functions

The toolbox does not only provide functions for estimation of home ranges. Several functions for the analysis of movement data as well as for the analysis of the created home range polygons exist. We will introduce these functions in this section. All functions are to be found in the menu [MOVEAN>HR Analysis].

However, we note that OpenJUMPs GIS functions for vector analysis and raster analysis can also be helpful in the evaluation of movement data in conjunction with other geographic datasets. For instance overlay analysis functions in [Tools>Analysis>…] and statistics tools in [Tools>Statistics>…] should be considered. Raster analysis tools are available using the Sextante toolbox [Sextante>Toolbox].

Asymptote Analysis

Purpose – Asymptote analysis allows to determine if the area of the home range exhibits asymptotic behaviour if more (GPS) points are added to the dataset. Analysing this can be helpful in two instances: (a) it shows if the number of sampling points (observations) is sufficient for home range estimation, or if more points are needed. That is, more points are needed if no asymptotic behaviour occurs. (b) An asymptote plot (figure 19) allows the user to study movement behaviour of one animal between seasons within a year. For instance, with sample data used during development we could see that during the summer months, for some animals, home ranges were substantially larger than home ranges observed during spring or autumn. The toolbox allows asymptote analysis for sequential data with three home estimators: MCP, Line-Buffer and KDE. Asymptote analysis using random points has only been implemented for MCP.

Principle – A home range estimator is used to calculate the home range for a given set of observation points. The program starts with 3 points, estimates the home range and calculates the area for the home range. Then a few points are added sequentially or randomly and the estimation and area calculation is done again, then more points are added, and so forth. Finally, a graph showing the number of points vs. the area is drawn for visual analysis (figure 19).

Literature References – Asympote analysis is discussed in Harris et al. (1990), Laver (2005)

Dataset Attribute Requirements – What type of attributes are needed for the calculations depends on what home range estimator is chosen, i.e. MCP, Line-Buffer or Point KDE. However, all methods will require a location ID attribute to order the point set according to the recording sequence. The line-buffer method needs additionall an attribute that contains the recording day to calculate the median travel distance. For the details we refer to corresponding estimator function descriptions above.

Input – A point layer, with the following attributes for each observation point: (i) location ID, and if the line-buffer estimator is used, then (ii) observation day.

Running the Function – Use [**MOVEAN>HRE Analysis>Asymptote Analysis**]

Chose the layer with your observation data and set the attribute that contains the location ID.

The check box "point data are track data" is important to decide if points are added randomly or sequentially. If you recorded the points in a sequence, e.g. as recorded with a collar, then points should be added according to the sequence. However, if you have observation data where the recording time is unknown, then uncheck the box and the points to be added will be selected randomly. In the case of non-sequential data for each point more 10 draws are taken and, consequently 10 geometrie are created. Then the median area value is used for the statistics.

Next chose one of the methods that can be used for estimation of the home range. We implemented MCP, Line-Buffer and Point KDE as these are comparably fast. Each of those functions will be executed a few dozens to hundred times, depending on the size of the dataset. Hence, preforming an asymptote analysis with computational expensive home range estimators will take days.

If you chose the Line-Buffer estimator you need to provide the attribute that contains the recording day. If you chose the Point KDE estimator, then you need to select if h_{ref} is adaptive, i.e. is calculated for each 10 points more, or is used for the whole dataset. Also the provision of the raster cell size is necessary (see the notes in the Point KDE section for chosing this value). Other parameters for the Point KDE estimator, such as kernel type and probabilty value to create the home polygon have been fixed, i.e. biweight kernel, h_{ref}, p = 95%. To have fewer calculations for the point KDE points are added in chunks of 10.

Figure 19. An asypmtote graph generated with [MOVEAN>HRE Analysis>Asymptote Analysis]. The blue graph shows the change of area if more observation points are evaluated. The red line displays the difference between two consecutive home range evaluations with n and with n+x points (with usually x=1) from the blue graph. Hence, the red line can be used to identify strong changes in the home range. Note that a right-mouse click on a line in the graphics returns the x-value/number of points.

Output – Two outputs are produced by the function: (i) a layer with all the generate home ranges that can be used as input for the Asymptote Movie function, and (ii) a graph showing how the

23

area changes if points are added to the dataset (figure 19, for an interpretation of the graph we refer to literature mentioned above).

Asymptote Movie

Purpose – To show in a simple animation how the home range (i.e. the polygon geometry) changes over time. This function is only useful if the original observation dataset was sequentially recorded.

Principle – Displays each home range polygon calculated with [MOVEAN>HRE Analysis>Asymptote Analysis], ordered by the number of points that have been used to create the home ranges.

Literature References – no references

Dataset Attribute Requirements – needs the output layer from the function [MOVEAN>HRE Analysis>Asymptote Analysis] and its attribute "number of points". However, the animation makes only sense for sequentially recorded data.

Input – A layer with home range polygons and an attribute that allows to order the polygons of the layer in time.

Running the Function – Use [**MOVEAN>HRE Analysis>Asymptote Movie**].

Select the layer generated with the function [MOVEAN>HRE Analysis>Asymptote Analysis] described above as "Layer with Home Rages". In the field "time attribute" chose "numpoints" from the drop down list.

Note that it may be useful to display topographic data in the background to better interpret movement behaviour. E.g. Grizzly bear movements in the Rocky Mountains are for instance constrained by topographic features, such as mountain ridges, passes and lakes.

Output – An animation showing how the home range changes/evolves over time. Note that the play time of the animation depends on the number of points (i.e. home ranges) of the dataset. Rewind, pause and fast forward are not implemented.

Create Core Area from Density Raster

*Purpose* – Identify areas of the home range that are used more frequently by an individual.

Principle – The determination of the core area is based on evaluation of the density of use surface generated with one of the estimators, such as Point KDE and Line KDE. The definition for "more frequent use" is based on comparing an area of the home range (i.e. a cell) to a uniform and random utilization of home range space (Powell, 2000). Obtained is a probability p_{core}, which defines the core area. The contour line formed by p_{core} will divide the home range into (i) frequently used areas and (ii) areas that may have been used by chance (uniform).

The methods we implemented are based on the evaluation of a graph that displays probability of use vs. area of use. The first method generates a graph by evaluating the density/probability of use for each cell of the raster grid as described in Seaman and Powell (1990). We will call this the "cell-based method". The second method is partly described in Harris et al. (1990) and is based on the calculation of contours for certain probability values, e.g. p = 2.5...97.5% in steps of 2.5%, and the area that the contours enclose. We will call this method the "contour-based method". The approach for choosing the core probability from these generated graphs was proposed in Seaman and Powell (1990): plot the curve of area vs. probability of use, and select the probability where this curve and a diagonal line, representing random use, from (area = 100%, probability = 0%) to (area = 0%, probability = 100%) are most distant from each other (figure 20).

Literature References – Powell (2000) can be used as a general introduction into the thematic. Samuel and co-authors (1985; 1988) first described an approach to quantitatively estimate the core area based on point data. However, we implemented two different methods as described by Seaman and Powell (1990) and Harris et al. (1990). See also Laver (2005) for more information.

Dataset Attribute Requirements – To perform the calculations a raster is needed, in which the raster values should correspond to density or probability of occurrence.

Input – A raster with density values generated with the Point KDE or the Line KDE estimator. Also the resulting raster of the Brownian Bridge estimator can be used, but see the notes below.
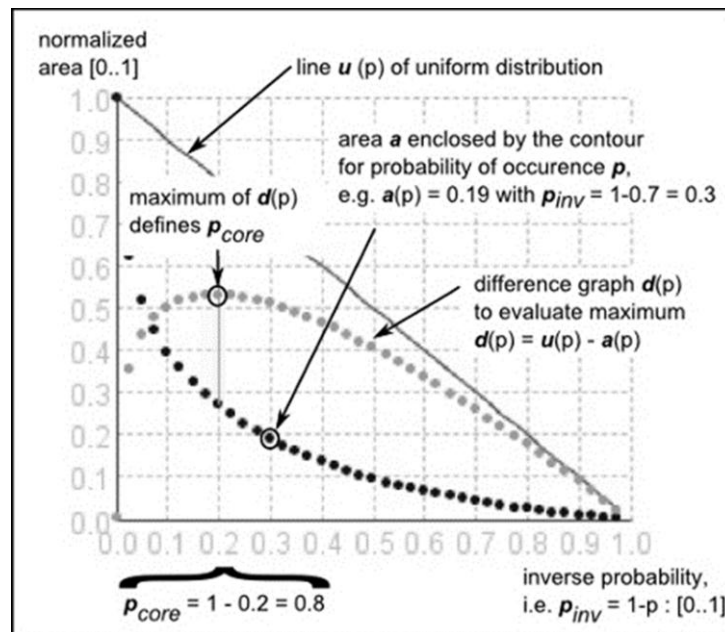
Figure 20. The graph for the core area calculation as obtained with the toolbox and with additional explanations. The graph is for the method that evaluates the area for a range of probability contours as described in Harris et al. (1990). However, core area probability is not derived from the inflection point but from the maximum distance proposed in Seaman and Powell (1990).

Running the Function – Use [**MOVEAN>HRE Analysis>Create Core Area from Density Raster**] while having selected the density raster that was returned from one of the estimator function (if you don't select the raster, the function will not be accessible).

Select if you want a polygon to be created from the derived value for p_{core} otherwise only a layer with the contour line is returned.

Output – Two graphics that display "area" vs. "probability/density of use" are generated; one for each method. In the output window the calculated values for p_{core} for both methods are returned. The value "Calculated Contour Value" gives not the probability but a value that corresponds to the values of the input raster. Additionally, a layer with the core contour is displayed. However, that core contour is based on the value derived with the cell-based method. To generate the contour or core polygon with the value obtained with the contour-based method use the function [MOVEAN>HRE>Create Probability Contours from Raster]. As the graphics may contain many points, it may take time to move those graphics around. So please be patient.

Notes – *Computation time & resources* – The method based on the evaluation of contour lines is fairly time intensive. So please be patient. Also, if the rasters are very big (remember: a 1000 x 1000 raster contains 1 million cells), then it may occur that the program will run out of memory when the raster cells are evaluated.

Different results for both methods - We observed that in our experiments both methods, i.e. the raster-cell based by Seaman and Powell (1990) and the contour-based by and Harris et al. (1990), often returned different probabilities for the core area. However, when there are large differences, e.g. 10% or more, then we suggest using the value from the contour-based method as its computation is more intuitive.

Failure of Core Area Calculation for Brownian Bridge Results – When calculating p_{core} based on the (density) surface produced with using the BB approach, we occasionally obtained values

close to or larger than p = 95%, when they should be much smaller. Both calculation methods, but in particular the cell-based method, returned high p values. The reason for this behaviour may be the shape of the generated density distribution, i.e. if a profile of the probability surface perpendicular to a single track segment is drawn (you can do this with the function [Raster>Profile Graph…]), then very long tails are visible. This may also explain why Huck et al. (2008) found the calculated home-range for the p = 90% contours nonsensical, i.e. too big, and instead used the p = 40% contours. As a consequence care needs to be taken when knowledge gained from other methods developed for KDE are applied to BB results.

Skeletonization of HR

Purpose – To create a skeleton of an existing home range polygon. A skeleton resembles the essential movement paths of a home range[8] (figure 21). They can be used to compare more easily if a home range of one individual changed significantly over time or, if the home ranges of two individuals are similar.

Principle – The function takes a home range polygon, converts that polygon to a (binary) raster and derives the skeleton of that raster by application of morphological operators. As the skeletonization result is a raster as well, the skeletons are vectorized to return a vector layer. Raster-vector conversion and skeletonization is performed with functions of the Sextante Toolbox (Olaya, 2010).

Literature References – There are no references existing on the use of skeletons for home range analysis.

Dataset Attribute Requirements – No attribute requirements.

Input – A layer with (home range) polygons.

Running the Function – Use [**MOVEAN>HRE Analysis>Skeletonization of HR**].

Chose the layer with the home range polygons and define the raster cell size. The latter is necessary since the skeleton extraction is not performed as a vector but a raster operation.

Output – A vector layer with the skeletons. If a home range polygon will not have a skeleton (because it may be too small with respect to the raster cell size), then this will be reported in the output window.
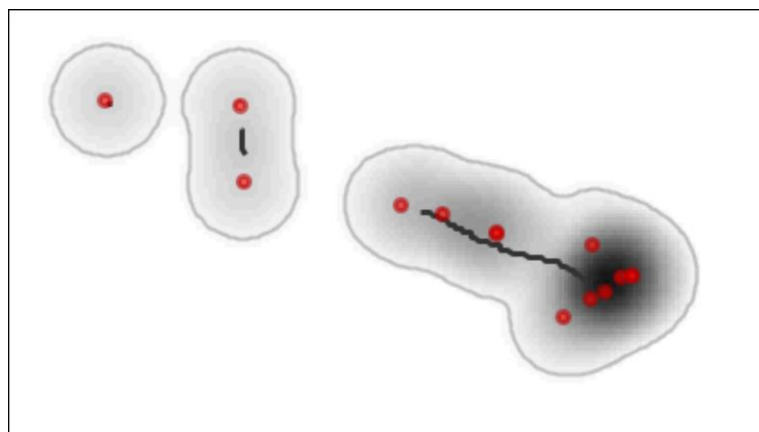
Figure 21. A skeleton for the home range polygon of the sample dataset produced with the function [MOVEAN>HRE Analysis>Skeletonization of HR]. The grey lines delineate the home range and the black lines are the skeletons.

---

[8] The skeletonization process is sometimes also referred to as medial axis transform: http://en.wikipedia.org/wiki/Topological_skeleton

26

Calculate Buffer Distance for HR Classification

Purpose – To calculate a buffer distance, i.e. the distance between the core area contour and the home range contour. The value can then be used for the classification of home ranges with the functions [MOVEAN>HRE Analysis>Extract Core, Patch, Edge] and [Extract Corridors].

Principle – Matches a given home range with its given core area (both obtained with other functions from the toolbox). Then it calculates the average distance between home range contour and core area contour for several matches.

Dataset Attribute Requirements - No attribute requirements.

Input – Two polygon layers are needed: (i) one layer that contains the home range polygons, and (ii) a layer that contains the core area polygons.

Running the Function – Use [**MOVEAN>HRE Analysis>Calculate Buffer Distance for HR Classification**].

Select the input layers as necessary. Optionally you can chose to output the distance measurements in a layer.

Output – The calculated distance between core and home range boundary are returned in the output window. If it was chosen to output the single measurements, then a layer with lines is shown (figure 22). The difference between two lines next to each other corresponds to one measure.
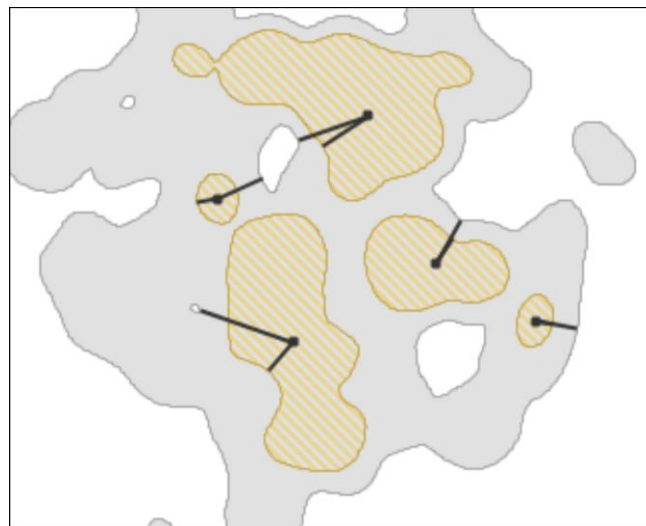
Figure 22. Measurements (in black) used to calculate the distance between home range boundary and core area boundary using [MOVEAN>HRE Analysis>Calculate Buffer Distance for HR Classification]. Brown and hatched are the core areas and grey the home range areas.

HR Classification 1: Extract Core, Patch, Edge

Purpose – Classify a home range polygon into core areas, edge areas and patches. Patches are solitary home ranges that are too small to contain a core area. Additionally distinctions are made between inner edges and perforated edges (surrounding small holes in the home range).

Principle – Knowing the distance between home range boundary and core area boundary allows the application of buffer-operations to identify the three classes of home range areas. The three classes and their calculation with raster and vector methods are discussed in Steiniger (2009).

Literature References – Upon publication there is no literature on the use of the classification in home range analysis. However, there is a publication by Vogt et al. (2009) for the analysis of animal movements that where generated with a simulator.

Dataset Attribute Requirements - No attribute requirements.

*Input* – The layer containing home range polygons.

Running the Function – Use [**MOVEAN>HRE Analysis>Extract Core Patch, Edge**].

Select the layer with the home range polygons and then set the buffer distance. A meaningful buffer distance can be obtained with the function [MOVEAN>HRE Analysis>Calculate Buffer Distance for HR Classification].

Output – A polygon layer that contains the classified home range into core and edge areas, as well as patches (figure 23). Use can use the styles dialog and the tab "colour theming" to assign the classes different colours, based on the 'ptype' attribute.
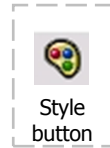
Note – If one compares the core area generated with this method and generated with the function [MOVEAN>HRE Analysis>Create Core Area from Density Raster] then the areas may be quite different. The core area generated with this method should be seen as an approximation for the core area generated with the surface based method.

Style button
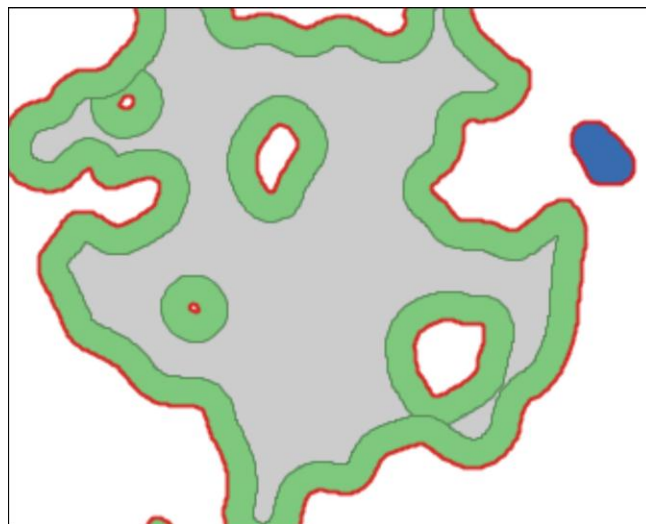
Figure 23. The classification result for a home range using [MOVEAN>HRE Analysis> Extract Core Patch, Edge]. The red line delineates the original home range. In green are edge areas, in grey a core area, and in blue is one patch. Inner edges and perforated edges, which both surround holes in the home range, are not separately displayed in this image.

HR Classification 2: Extract Corridors

Purpose – To extract from a home range polygon corridors, branches, shortcut areas, and patches. Corridors are hereby defined as narrow areas within the home range polygons. Branches may be exploratory areas when the shape of the home range is assumed to be convex. Patches are again – as for the previous function – home range areas that are smaller than a certain minimum area.

Principle – Similar to the "Extract Core, Patch, Edge" function the parts are detected by buffer operations and overlay analysis described in Steiniger (2009). Here as well knowing the distance between home range boundary and core area boundary is helpful for the selection of the buffer distance and is helpful to retrieve meaningful results.

Literature References – Vogt et al. (2009) use a similar classification for the analysis of animal movements that where generated with a simulator.

Dataset Attribute Requirements - No attribute requirements.

*Input* – The layer containing home range polygons.

Running the Function – Use [**MOVEAN>HRE Analysis>Extract Corridors**]

Select the layer with the home range polygons and then set the buffer distance. A meaningful buffer distance can be obtained with the function [MOVEAN>HRE Analysis>Calculate Buffer

Distance for HR Classification]. Optional it is possible to check the box, so that very small branches are removed by the function immediately.

Output – A polygon layer that contains the classified home range into core and edge areas, as well as patches (figure 24). Similar as for the other classification function to detect cores and edges, one can use the styles dialog and the tab "colour theming" to assign the classes different colours. Use the 'ptype' attribute for that.
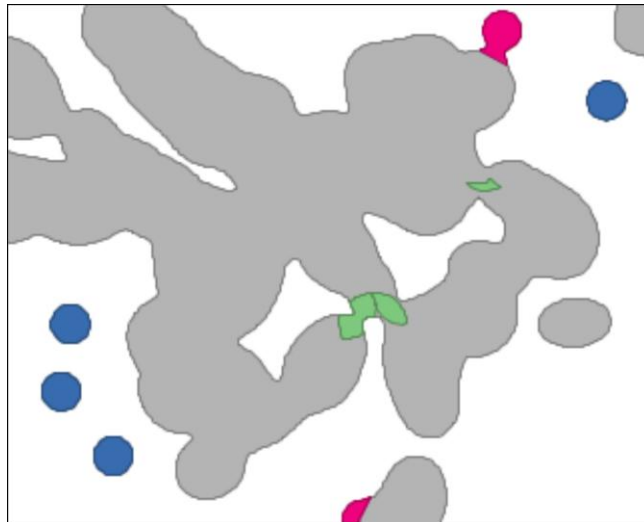
Figure 24. The classification result for a home range using [MOVEAN>HRE Analysis> Extract Corridors]. The red line delineates the original home range. In grey are the home range areas, in green are corridor areas, and blue are patches. Branches are shown in purple.

Note – The idea for the detection of corridors and branches is somewhat experimental and has not been tested with different individual and species data. We often discovered that using the distance between core area and home range boundary may not be successful to detect corridors. Hence, it seems to be advisable to rather choose the buffer distance parameter based on expert knowledge about a species.

6. Notes

Temporary Storage of Data

As mentioned earlier OpenJUMP HoRAE uses Sextante toolbox functions. These functions sometimes produce intermediate vector and raster files in the temp folder (e.g. on Windows Vista: "C:\Users\YourLoginName\AppData\Local\Temp\"). Subsequently the temp folder needs to be cleaned once in a while.

Adding other Extensions and Plugins to OpenJUMP for Enhancing its Functionality

OpenJUMP comes with a plugin system that allows adding functions to the OpenJUMP Core distribution. This is for instance used by the OpenJUMP Plus edition and OpenJUMP HoRAE to add the Sextante toolbox, printing functions and the home range analysis tools. To make the functions of a plugin or extension[9] available in OpenJUMP you need to place the *.jar files[10], which contain the additional functions, into the OpenJUMP subfolder named "/lib/ext/". In this place is also the *movean*.jar* file that contains the home range analysis and estimation functions. If you have place the library file (i.e. the *.jar file) into the mentioned subfolder, then you need restart OpenJUMP HoRAE, so it can load the additional functions. A list of most of the plugins available for OpenJUMP is given here: http://sourceforge.net/apps/mediawiki/jump-pilot/index.php?title=Plugins_for_OpenJUMP

---

[9] An OpenJUMP *Extension* consists of several *Plugins*.
[10] *.jar files are a type libraries used in JAVA programs. If you rename the ending of these files to *.zip you can see what functions are in the library.

Projections and Coordinate Systems in OpenJUMP

When loading data into OpenJUMP HoRAE information on used coordinate systems and projections is discarded, even if it is available. That means, in its current version, OpenJUMP does not care about map coordinate systems. It will assume that all data loaded are in the same coordinate system. However, for OpenJUMP exists a Coordinate Transformation Services (CTS) Extension that allows to transform vector data from one mapping projection to another one. More information on the CTS extension can be found here: http://sourceforge.net/apps/mediawiki/jump-pilot/index.php?title=CTS_Extension

7. References

Benhamou, S., & Cornélis, D. (2010). Incorporating Movement Behavior and Barriers to Improve Kernel Home Range Space Use Estimates. *The Journal of Wildlife Management*, *74*(6), 1353-1360. doi:10.1111/j.1937-2817.2010.tb01257.x

Berger, K. M., & Gese, E. M. (2007). Does interference competition with wolves limit the distribution and abundance of coyotes? *Journal of Animal Ecology*, *76*(6), 1075-1085. doi:10.1111/j.1365-2656.2007.01287.x

Bullard, F. (1991). *Estimating the home range of an animal: a Brownian bridge approach* (B.S. thesis). University of North Carolina, Chapel Hill.

Burgman, M. A., & Fox, J. C. (2003). Bias in Species Range Estimates from Minimum Convex Polygons: Implications for Conservation and Options for Improved Planning. *Animal Conservation*, *6*(01), 19-28. doi:10.1017/S1367943003003044

Calenge, C. (2006). The package "adehabitat" for the R software: A tool for the analysis of space and habitat use by animals. *Ecological Modelling*, *197*(3-4), 516-519. doi:10.1016/j.ecolmodel.2006.03.017

Getz, W. M, & Wilmers, C. C. (2004). A local nearest-neighbor convex-hull construction of home ranges and utilization distributions. *Ecography*, *27*(4), 489–505.

Getz, Wayne M., Fortmann-Roe, S., Cross, P. C., Lyons, A. J., Ryan, S. J., & Wilmers, C. C. (2007). LoCoH: Nonparameteric Kernel Methods for Constructing Home Ranges and Utilization Distributions. *PLoS ONE*, *2*(2), e207. doi:10.1371/journal.pone.0000207

Harris, S., Cresswell, W. J., Forde, P. G., Trewhella, W. J., Woollard, T., & Wray, S. (1990). Home-range analysis using radio-tracking data–a review of problems and techniques particularly as applied to the study of mammals. *Mammal Review*, *20*(2-3), 97–123.

Hemson, G., Johnson, P., South, A., Kenward, R. E., Ripley, R., & MacDonald, D. (2005). Are kernels the mustard? Data from global positioning system (GPS) collars suggests problems for kernel home-range analyses with least-squares cross-validation. *Journal of Animal Ecology*, *74*(3), 455-463. doi:10.1111/j.1365-2656.2005.00944.x

Horne, J. S., Garton, E. O., Krone, S. M., & Lewis, J. S. (2007). Analyzing animal movements using Brownian Bridges. *Ecology*, *88*(9), 2354-2363. doi:10.1890/06-0957.1

Huck, M., Davison, J., & Roper, T. J. (2008). Comparison of two sampling protocols and four home-range estimators using radio-tracking data from urban badgers Meles meles. *Wildlife Biology*, *14*, 467-477. doi:10.2981/0909-6396-14.4.467

Jacques, C. N., Jenks, J. A., & Klaver, R. W. (2009). Seasonal Movements and Home-Range Use by Female Pronghorns in Sagebrush-Steppe Communities of Western South Dakota. *Journal of Mammalogy*, *90*(2), 433-441. doi:10.1644/07-MAMM-A-395.1

Kie, J. G., Matthiopoulos, J., Fieberg, J., Powell, R. A., Cagnacci, F., Mitchell, M. S., Gaillard, J.-M., et al. (2010). The home-range concept: are traditional estimators still relevant with modern telemetry technology? *Philosophical Transactions of the Royal Society B: Biological Sciences*, *365*(1550), 2221 -2231. doi:10.1098/rstb.2010.0093

Laver, P. N. (2005). *ABODE – Kernel Home Range Estimation for ArcGIS, Using VBA and ArcObjects.* (User Manual). Department of Fisheries and Wildlife Sciences, Virginia Tech. Retrieved from http://www.geo.uzh.ch/~sstein/ojhorae/laver_abode_user_manual_2005.pdf

Nilsen, E. B., Pedersen, S., & Linnell, J. D. C. (2007). Can minimum convex polygon home ranges be used to draw biologically meaningful conclusions? *Ecological Research*, *23*, 635-639. doi:10.1007/s11284-007-0421-9

Olaya, V. (2010). SEXTANTE, a free platform for geospatial analysis. *OSGeo Journal*, *6*(1), 8.

Powell, R.A. (2000). Animal home ranges and territories and home range estimators. In L. Boitani & T. Fuller (Eds.), *Research techniques in animal ecology: controversies and consequences* (pp. 65–110). Columbia University Press.

Samuel, M. D., Pierce, D. J., & Garton, E. O. (1985). Identifying Areas of Concentrated Use within the Home Range. *Journal of Animal Ecology*, *54*(3), 711-719. doi:10.2307/4373

Samuel, M., & Green, R. (1988). A Revised Test Procedure for Identifying Core Areas Within the Home Range. *Journal of Animal Ecology*, *57*(3), 1067–1068.

Seaman, D. E., & Powell, R. A. (1990). Identifying Patterns and Intensity of Home Range Use. *Bears: Their Biology and Management*, *8*, 243-249. doi:10.2307/3872925

Sheather, S. J. (2004). Density estimation. *Statistical Science*, *19*(4), 588–597.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis* (Vol. 26). London: Chapman & Hall/CRC.

Steiniger, S. (2009). *A vector-based algorithm to detect core, edge, patch and corridor areas and comparison with its raster-based complement.* Retrieved from http://sourceforge.net/userapps/mediawiki/mentaer/index.php?title=Movement_Analysis

Steiniger, Stefan, & Hay, G. J. (2009). Free and open source geographic information tools for landscape ecology. *Ecological Informatics*, *4*(4), 183-195. doi:16/j.ecoinf.2009.07.004

Steiniger, S., & Hunter, A. J. S. (2010). *A scaled line-based kernel density estimator for the retrieval of utilization distributions and home ranges from GPS movement tracks.* Retrieved from http://sourceforge.net/userapps/mediawiki/mentaer/index.php?title=Movement_Analysis

Steiniger, Stefan, Timmins, T. L., & Hunter, A. J. S. (2010). Implementation and Comparison of Home Range Estimators for Grizzly Bears in Alberta, Canada, based on GPS data. Presented at the GIScience 2010, Zurich, Switzerland.

Turchin, P. (1998). *Quantitative analysis of movement: measuring and modeling population redistribution in animals and plants*. Sinauer Associates, Sunderland (MA).

Vogt, P., Ferrari, J. R., Lookingbill, T. R., Gardner, R. H., Riitters, K. H., & Ostapowicz, K. (2009). Mapping functional connectivity. *Ecological Indicators*, *9*(1), 64-71. doi:10.1016/j.ecolind.2008.01.011

Worton, B. J. (1995). Using Monte Carlo Simulation to Evaluate Kernel-Based Home Range Estimators. *The Journal of Wildlife Management*, *59*(4), 794-800. doi:10.2307/3801959