Upslope, downslope: the influence of the third dimension on step length of roe deer

Abstract: Ecological questions regarding animal behavior, response to external factors and intrinsic motivators can all be addressed by analyzing animal movement data, often obtained by (GPS) tracking devices. The so obtained locations can be most accurately described by three coordinates; latitude, longitude and elevation. In the analysis of terrestrial movement data, the elevation is nevertheless often ignored. In this study an attempt has been made to identify under what conditions it is important to consider one of the most basic movement metrics, step length, in 3D.

The difference between 3D and 2D step lengths of 0 – 1000m has been calculated for constant, concave and rugged slopes ranging from 0 – 60°. Since the 2D definition did not hold for 3D step lengths on rugged slopes, it has been redefined to the Euclidean surface distance. The difference is subsequently calculated for the tracking steps of two contrasting roe deer datasets, one in low relief terrain and one in high relief terrain. The elevation at the start and endpoint of each tracking step as well as at every 10m along the tracking step was extracted from a Digital Elevation Model (DEM) using bi-linear interpolation.

The theoretical difference increased with increasing slope and with increasing rugosity, although a large variation in the effect size of rugosity was found. The theoretical difference for concave slopes was slightly higher than constant slopes. When applied to the roe deer movement data, the median bias was estimated at 7.3% in the high relief area and 0.1% in low relief area, while the highest median bias for one individual was found to be 18.1% in the high relief area and only 0.2% in the low relief area. The minimum underestimation could be accurately estimated by the theoretical difference on constant slopes. While rugosity leads to an increase in bias, there was found variation in the magnitude. It is therefore recommended that the step length is measured in 3D when the studied species is expected to make use of steep and rugged terrain.

Author: Marrit Leenstra
Supervisors: dr. ir. E.E. (Emiel) van Loon, J. (Johannes) De Groeve MSc, dr. F (Francesca) Cagnacci
BSc thesis Future Planet Studies
University of Amsterdam – 24/08/2018

Content

List of Figures

List of tables

Introduction

Animal movement is a key concept in ecology, linking individuals and populations with spatial processes. Understanding the causes and consequences of movement is therefore the basis of both fundamental (e.g. why is a specific habitat chosen) and applied (e.g. how does climate change affect dispersal) questions in ecology (Cagnacci, Boitani, Powell, & Boyce, 2010; Turchin, 1998). *Movement* in this context is defined as a change in the spatial location of the whole individual in time (Nathan et al. (2008), if *italic* see Glossary in Appendix I) and the simplest movement can thus be represented by distance and direction between spatial locations at two consecutive time instants. Since animals move through a 3-dimensional space, their spatial locations can be most accurately represented by the planar coordinates (usually latitude and longitude) combined with the z-coordinate, representing depth, altitude or elevation. While diving behaviour has been thoroughly assessed in marine systems (e.g. Mitani et al., 2010; Papastamatiou et al., 2018) and flight behaviour in avian species (e.g. Weimerskirch, Le Corre, Ropert-Coudert, Kato, & Marsac, 2005), little attention has been given to how movement is shaped by topographic features in terrestrial species. However, technological advancement in tracking technology and the recent emergence of high-resolution *Digital Elevation Models (DEMs*) allows for progress in this direction.

In the past, movement ecology involved direct observation in the field, sampling animal tracks in snow or sand and *mark-recapture methods* (Turchin, 1998). These labour-intensive methods are now for a major part replaced by remote *tracking devices*, enabling collection of large amounts of standardized location data. A good description of the so obtained temporal sequence of locations, termed *trajectories*, satisfies the following criteria; 1) use of a limited set of relatively easily measured metrics, 2) the relationship between these metrics is precisely defined and 3) the metrics and the relationship between them can be used to reconstruct characteristic tracks (Marsh & Jones, 1988). A combination of metrics describing direction, distance and time lag between each step is thus sufficient and can be used as input for models describing the movement pattern (e.g. *correlated random walk*) (Calenge, Dray, & Royer-Carenzi, 2009).

Overviews of such *movement metrics* are provided by Edelhoff, Signer and Balkenhol (2016) and Dodge, Weibel and Lautenschütz (2008) (Appendix II). *Step length*, defined as the Euclidean distance between two consecutive relocations is one of the most basic movement metrics. To my knowledge, *three dimensional analyses* of step lengths have not yet been conducted. However, evidence of three-dimensional analysis of *First Passage Time*, a movement metric used as indicator for area-restricted search behaviour, suggests that neglecting the third dimension in the marine environment can lead to erroneous conclusions (Bailleul, Lesage, & Hammill, 2010). Similar research for terrestrial animals has however not been found.

Remarkably, more attention has been paid to the consequences of the third dimension on the more complex *space use patterns*, there first attempts to describe home range in 3D have been made already in 1977 (Koeppl, Slade, Harris, & Hoffmann, 1977). Space use patterns describe the organization of animals in space and time and one of the core methods is estimation of home ranges (Millspaugh, 2001). *Home range* is defined as "that area traversed by the individual in its normal activities of food gathering, mating, and caring for young" (Burt, 1943). More recently, Simpfendorfer, Olsen, Heupel, and Moland (2012) concluded that the home range overlap for eel was overestimated in 2D analysis, implying that competition is smaller than previously thought. However, 3D analysis of home range overlap in juvenile salamanders showed no significant difference with 2D (Ousterhout & Burkhart, 2017). The differences in these results seem to originate in differences in scale of movement in the vertical dimension. Similar conclusions can be drawn regarding terrestrial animals; it has been found that 2D analysis underestimates home range size between 0,01%-3,37% for feral cats in New Zealand (Recio, Mathieu, Maloney, & Seddon, 2010), 12,09%-16,82% for giant pandas in China (Tracey et al., 2014) and 10%-27% for stoat in

New Zealand (Smith, Wilson, Moller, Murphy, & van Heezik, 2007). The scale of movement in the third dimension can account again for the differences in the results. In fact, Recio et al. (2010) use the difference in 3D and 2D home range size as an indirect measure of utilisation of mountain slopes by feral cats.

Based on the differences between 3D and 2D space use, it is expected that comparing step lengths with a high vertical component to step lengths with a low vertical component will lead to a bias in results. One such example could be the overestimation of step length and derivatives (e.g. speed) on valley bottoms compared to adjacent slopes.

The aim of this research is to identify under what conditions it is important to consider step length in 3D. First a theoretical exploration of the difference considering different topographic conditions is carried out, after which this is applied to movement data of European Roe deer (*Capreolus capreolus*). *European Roe deer* occupy a wide variety of habitats, under which habitats with differ reliefs (Andersen, Duncan, & Linnell, 1998) and form therefore a good model species to estimate the difference of step lengths at different topographic conditions.

The main research question is split into the following sub-questions:

1. How should step length be defined and calculated in 3D?
2. Which topographic variables influence 3D step length?
3. How large is the theoretical difference between 3D and 2D step length when calculating for different topographic variables?
4. How large is the difference between 3D and 2D step length in roe deer movement data?
5. How does the theoretical differences relate to the differences in roe deer movement data?

Methods

Definition and calculation of 3D step length

In a *2D analysis*, step length is defined as the Euclidean distance between two successive relocations. When a tracking step $(t, t+1)$ starts at point (x_t, y_t, z_t) and ends at point $(x_{t+1}, y_{t+1}, z_{t+1})$, the *2D step length* ($s_{2D}$) is calculated with

$$s_{2D} = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2} \tag{1}$$

In a *3D analysis*, I propose to define step length as the Euclidean surface distance between two successive location fixes. When considering *constant slopes*, the *3D step length* (s_{3D}) can be calculated with

$$s_{3D} = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2 + (z_{t+1} - z_t)^2} \tag{2}$$

Consequently, when $z_{t+1} \neq z_t$ neglecting the third dimension leads to underestimation of the Euclidean distance between two relocations. If slope angle (α) is known s_{3D} can be calculated with

$$s_{3D} = s_{2D}/cos(\alpha) \tag{3}$$

In Figure 1a equations 1 to 3 are illustrated.

By using the Euclidean distance between two successive relocations at (*t* and *t+1*), all the available information with respect to x and y-coordinates is used (if we disregard the use of derivatives), but for the z-coordinate that is not automatically the case. Depending on the resolution of elevation data and the distance between successive relocations, ancillary elevation information may be available (this situation is illustrated in Figure 1b). This information can be taken into account by e.g. assuming a straight path in the (*x,y*)-plane, and piecewise linear segments between the different elevation observations. To differentiate the 3D step length on constant slopes as visualized in figure 1a from the method that takes ancillary elevation information into account, it is proposed to term the latter the *surface step length* (s_{surf}). 3D step length and surface step length will together be addressed as three-dimensional step lengths.

If the shape of the *non-constant slope* can be described by a formula $f(s_{2D})$, the surface step length can be calculated with the arc length formula, being

$$s_{surf} = \int_{t}^{t+1} \sqrt{1 + (f'(s_{2D}))^2}\, \Delta s_{2D} \tag{4}$$

Non-constant slopes for which the describing formula is unknown, can be approximated by connected linear line segments (see illustration in Figure 1b). Hence, surface step length on such slopes can be calculated by a sum of the individual segment lengths. The formula for s_{surf} when using *n* segments is

$$s_{surf} \approx \sum_{i=0}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \tag{5}$$

Where x_i, y_i and z_i refer to the starting coordinates of the i^{th} segment within a tracking step $(t, t+1)$; point (x_0, y_0, z_o) coincides with (x_t, y_t, z_t) and point (x_n, y_n, z_n) coincides with $(x_{t+1}, y_{t+1}, z_{t+1})$. Increasing the numbers of segments will result in better approximation of the surface step length, until the resolution of the elevation data is reached.

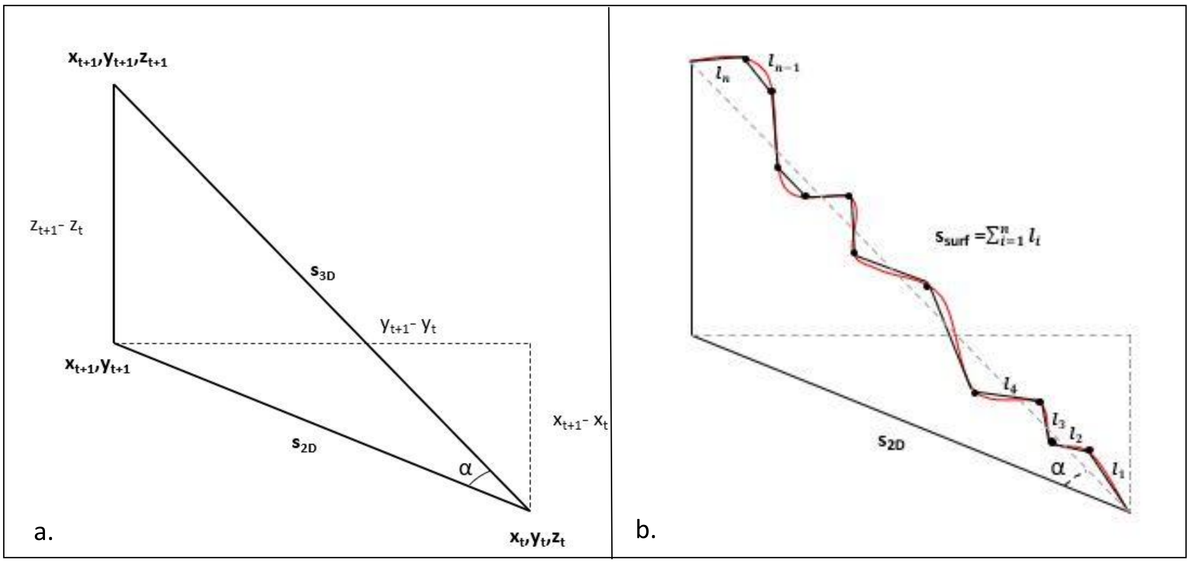

Figure 1. Step length calculations. a) On constant slopes 2D step length (s_{2D}) and 3D step length (s_{3D}) for a tracking step $(t, t+1)$ are calculated with respectively equation 1 and equation 2. z_{t+1} - z_t depends on the slope angle (α). When α is known s_{3D} can be calculated with equation 3. b) s_{surf} on non-constant slopes (red) with additional elevation data available (black dots) can be calculated by summing the lengths of *n* segments. Each segment length is in turn calculated with $l_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}$, where x_i, y_i and z_i refer to the starting coordinates of the i^{th} segment within a tracking step $(t, t+1)$.

Topographic variables

An overview of *topographic variables* is provided by Lecours, Devillers, Simms, Lucieer, and Brown (2017) (Appendix III). From their list, three variables, slope angle, profile curvature and rugosity (Figure 2), have been selected for further investigation as variables that might influence the step length.

Slope angle describes inclination of the *slope* to the horizontal plane (in degrees). Constant slopes are slopes with a constant slope angle (α) over the entire length. Slope angles can both be negative (representing a descending slope) and positive (representing an ascending slope), however, the effect on the difference between three-dimensional and two-dimensional analysis of step length is equal. Therefore, the absolute slope angle (|α|) is used (Figure 2a).

Profile curvature describes the shape of the slope, indicating if the slope is concave or convex (Blaga, 2012). A slope with zero curvature can be seen as a special case and describes a constant slope. Since concave and convex slopes have an equal effect on the difference between 3D and 2D step lengths, only concave slopes will be considered in this thesis (Figure 2b). Concave slopes can be described with a second order polynomial

$$z = as_{2D}{}^2 + bs_{2D} + c \tag{6}$$

Where z = the elevation and $s_{2D}$the 2D step length.

Slope and profile curvature will be supplemented by a *rugosity* index, quantifying the variability in slope gradient (Figure 2c). Several rugosity indices are available (see Appendix III, group 2 for an example of the terrain variables overview by Lecours et al. (2017)). For this thesis the Root-Mean-Square Deviation of the Surface (R_q) will be used, because it is a simple and easy to understand measure of surface rugosity. R_q can be understood as the equivalent of the standard deviation in statistics (Hoechstetter, Walz, Dang, & Thinh, 2008 and Equation 7) and thus as a measure of how far the elevation along a slope is deviating from a slope with a constant angle.

$$R_q = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(z_i - \bar{z})^2} \tag{7}$$

Where R_q is the rugosity, n is the number of points for which the elevation is known and z_i is the elevation at point i.

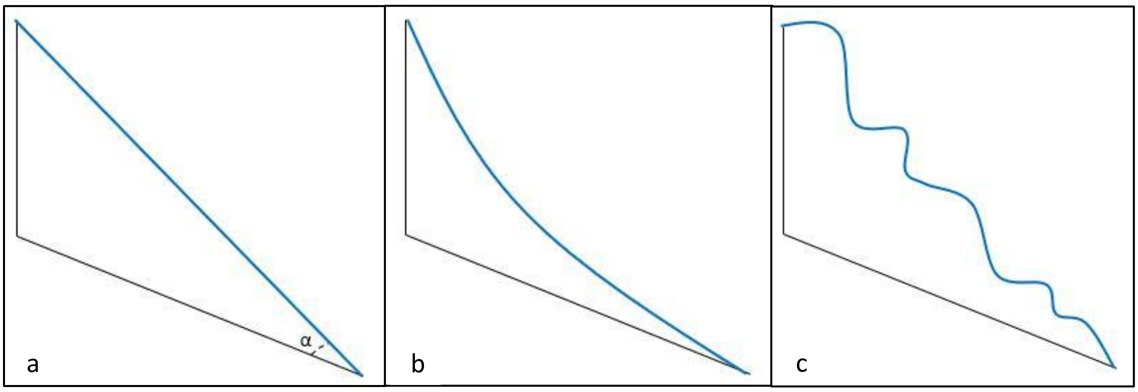

Figure 2. Topographic variables. Three topographic variables that possibly influence three-dimensional step lengths have been selected. a) slope angle.(α): describes the inclination of the slope to the horizontal plane. b) profile curvature: describes the shape of the slope, here an upward concave slope is visualised. c) rugosity: variability in slope gradient.

Theoretical data

The theoretical difference between 2D and three-dimensional step lengths is calculated for *constant*, *concave* and *rugged slopes* with slope angles (α) ranging from 0° to 60°, with an increment of 0.1° and 2D step lengths (s_{2D}) ranging from 0m to 1000m, with a step interval of 100m. The profile of the concave slope is for every step length and slope angle combination determined by varying the quadratic coefficient of equation 6 (i.e. a) according to

$$a = \frac{\tan(\alpha)}{s_{2D}} \tag{8}$$

the linear coefficient (b) and constant coefficient (c) of equation 6 are set to zero. The resulting *concave slopes* for a slope angle of 45° are visualised in figure 3.

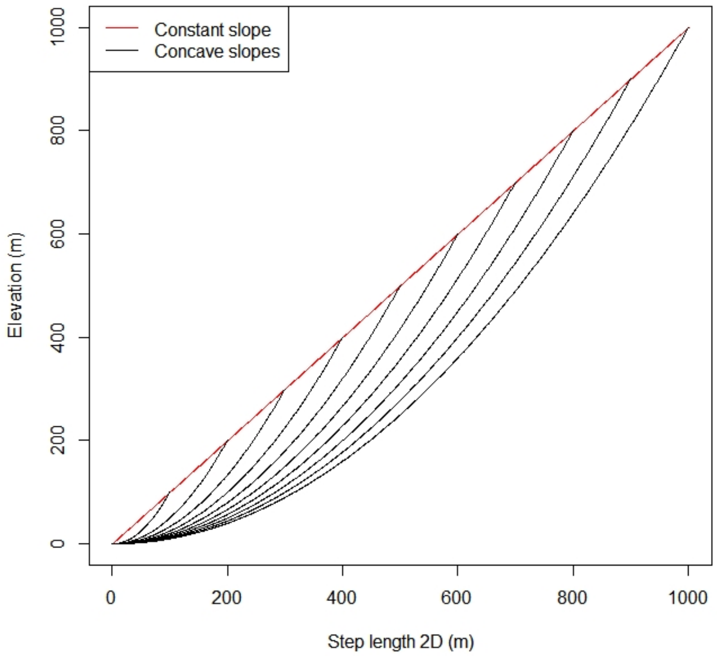

Figure 3. Concave slopes. **An example of the generated concave slopes for α=45° and 2D step lengths (s_{2D}) 100m to 1000m, with intermediate steps of 100m. The red slope represents the constant slope of 45°. The concave slopes in this example are calculated with $z = (\frac{\cos(45°)}{s_{2D}}){s_{2D}}^2 = cos(45°)s_{2D}$**

The *rugged slopes* were simulated following the steps in appendix IV. For each simulated slope of 1000 m, the slope angle and rugosity were then determined for segments of different s_{2D}-lengths (10 in total, covering the distance from point 0 to points k at 100, 200, ... 1000m). This was done by fitting a linear regression equation on each segment (from point 0 to k). The slope was subsequently calculated as the absolute tangent of the linear coefficient from the regression equation and the rugosity was calculated as the standard deviation of the residuals (Equation 7).

Three dimensional step lengths could then be calculated following respectively equation 3 for constant slopes, equation 4 for concave slopes (making use of the R package 'pracma' (Borchers, 2018)) and equation 5 for rugged slopes. Finally, the difference between 2D and three-dimensional step lengths was calculated by subtracting the 2D step length from the three-dimensional step lengths. To make the difference more general applicable the difference was divided by the 2D step length to obtain the *relative difference* between 2D and three-dimensional step lengths in percentage (Equations 9a and 9b).

$$relative\ difference = \frac{s_{3D}-s_{2D}}{s_{2D}} * 100 \tag{9a}$$

$$relative\ difference = \frac{s_{surf}-s_{2D}}{s_{2D}} * 100 \tag{9b}$$

All calculations and simulations are performed in R version 3.4.0 (R Core Team, 2018). Scripts are presented in appendix IV.

Empirical data

Study areas

Two contrasting roe deer datasets from the *EURODEER project* (http://www.eurodeer.org) were selected to examine the difference between 2D and surface step length in real movement data. The first dataset contains location fixes of 71 individuals in the northwestern Swiss Alps (center coordinates 46°54’87.50”N, 7°54’75.01”E). The mean elevation of the study area is 1470m, with a range of 557m to 2859m and the mean slope angle is 39.5° (Figure 4). The second dataset contains location fixes of 35 individuals in a study area in South West Germany (center coordinates 48°64’74.10N, 8°00’89.61E) with a mean elevation of 133m with a range from 119m to 154m and a mean slope angle of 2.4° (Figure 5).

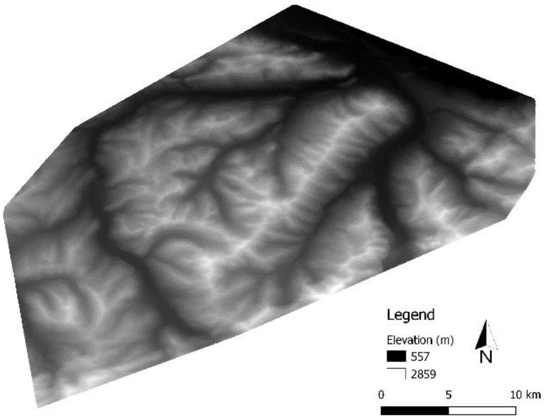

Figure 4. Digital elevation model study area Switzerland.

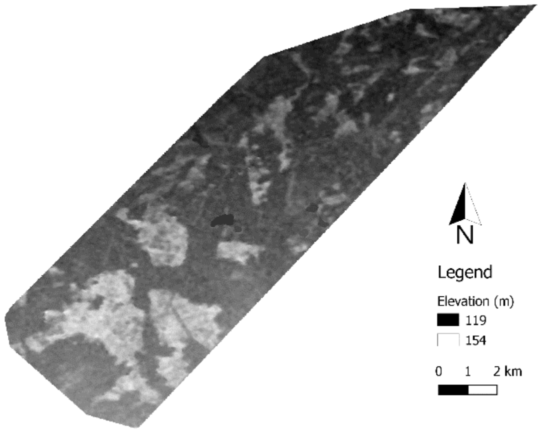

Figure 5. Digital elevation model study area Germany.

Data preparation

Each dataset was subsampled to 30 minutes resolution and for each individual the year with the highest proportion successful location fixes was selected. Subsequently, 10 individuals with a percentage successful fixes greater than 70 for Germany and greater than 90 for Switzerland were randomly selected for each study area. Measurement error was estimated at 15m, thus tracking steps shorter than 15m were removed, assuring that the animal actually moved within a given step (Table 1 and 2, next page).

Data processing

It is assumed that the animal moved in a straight line between two consecutive relocation. To obtain the slope profile for every tracking step, elevation was extracted from the EU-DEM v1.0 (European Union, 2018) at the start and endpoint of a tracking step as well as at every 10m along the tracking step. This is done by using bi-linear interpolation to interpolate from the grid cells on the DEM to points on the track, using R package ‘amoter’ (van Loon, 2018). A linear regression was fitted on the so obtained profile to determine the slope angle and the rugosity, by taking respectively the absolute tangent of the linear coefficient and the standard deviation of the residuals (Equation 7). 2D step length and surface step length were then calculated using equation 1 and equation 5, again with the R package ‘amoter’. Finally, the relative difference between the surface step lengths and 2D step lengths was calculated using equation 9b and the difference between theoretical and empirical data was visualised.

Table 4. Trajectories in Switzerland. **For the selected 10 animals the start- and end date of the trajectory are listed, together with the successful location fixes in percentage and absolute numbers. The relocations longer than 15m are used for the analysis.**

Animal Id	*Start data (yyyy-mm-dd hh:mm:ss)*	*End date (yyyy-mm-dd hh:mm:ss)*	*Successful location fixes (%)*	*Successful location fixes (number)*	*Relocations longer than 15m (number)*
2239	2012-03-01 01:00:56	2012-09-01 01:30:50	98.7	8724	5464
2240	2012-03-01 01:00:47	2012-09-01 01:30:44	98.4	8688	5419
2241	2012-03-01 14:00:11	2012-09-01 01:30:12	98.6	8710	5197
2247	2012-03-01 01:00:44	2012-09-01 01:30:56	99.1	8754	5328
2254	2013-03-01 17:01:15	2013-09-01 01:30:20	92.7	8185	4952
2259	2014-03-01 01:00:28	2014-09-01 01:30:11	94.8	8376	5153
2266	2013-03-01 01:00:26	2013-09-01 01:30:14	93.4	8252	5534
2292	2014-03-01 01:00:21	2014-09-01 01:30:22	95.8	8460	5802
2296	2012-03-01 01:00:56	2012-09-01 01:30:22	98.5	8701	5785
2297	2014-03-01 01:00:21	2014-09-01 01:30:36	96.0	8484	5832

Table 5. Trajectories in Germany. **For the selected 10 animals the start- and end date of the trajectory are listed, together with the successful location fixes in percentage and absolute numbers. The relocations longer than 15m are used for the analysis.**

Animal Id	*Start data (yyyy-mm-dd hh:mm:ss)*	*End date (yyyy-mm-dd hh:mm:ss)*	*Successful location fixes (%)*	*Successful location fixes (number)*	*Relocations longer than 15m (number)*
1453	2010-03-01 01:00:10	2010-09-01 01:30:25	99.0	8747	5447
1454	2010-03-01 01:00:50	2010-09-01 01:30:24	97.0	8566	5251
1455	2010-03-01 01:00:27	2010-09-01 01:30:53	94.6	8358	6264
2041	2011-03-01 01:00:13	2011-09-01 01:30:46	74.8	6604	4050
2043	2011-03-01 01:00:18	2011-09-01 01:00:46	71.2	6285	3509
2044	2011-03-01 01:00:38	2011-09-01 00:30:50	73.0	6442	3483
2048	2011-03-01 01:01:20	2011-09-01 01:30:21	72.3	6387	4064
2065	2012-03-01 01:00:43	2012-09-01 01:30:26	73.3	6472	3595
2076	2013-03-01 01:00:11	2013-09-01 01:30:49	72.1	6365	3887
2078	2013-03-01 01:01:01	2013-09-01 00:30:17	75.1	6635	3467

Results

Theoretical data

The theoretical relative difference between 3D and 2D step lengths for constant slopes angles of 0° to 60° ranges from 0 to 100%. The difference increases with increasing slope and the increase per degree is larger for every increment (Figure 6).

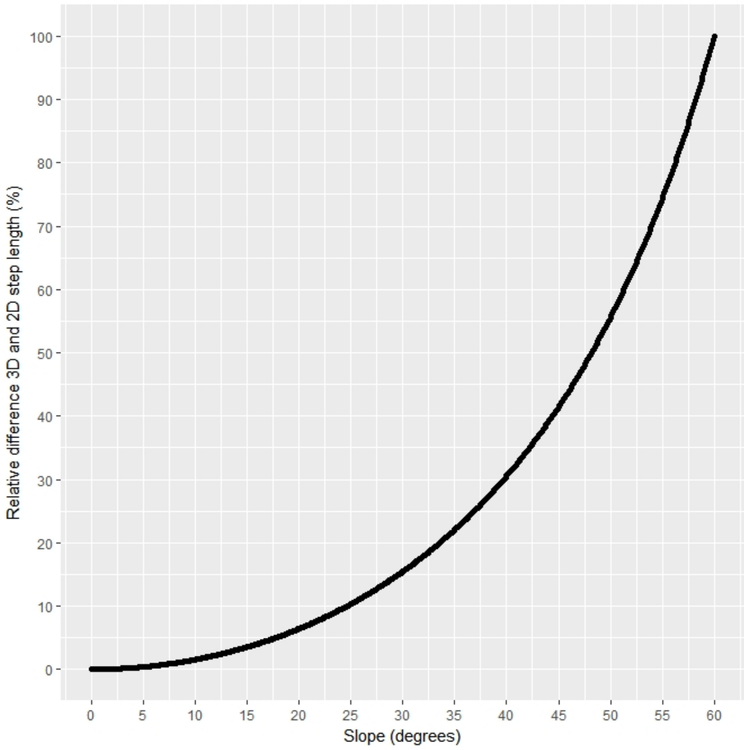

Figure 6. Theoretical relative difference constant slopes. **3D step length was calculated using equation 3 and 2D step lengths from 0m to 1000m. The relative difference was then calculated with equation 9a.**

Clearly, this is a geometrical relation, which follows from substituting equation 3 in 9a, leading to:

$$relative\ difference = \frac{s_{3D} - s_{2D}}{s_{2D}} = \frac{\frac{s_{2D}}{\cos(\alpha)} - s_{2D}}{s_{2D}} = \frac{1}{\cos(\alpha)} - 1$$

The theoretical relative difference between surface step length and 2D step lengths for concave slopes of 0° to 60° is always slightly larger than the theoretical relative difference on constant slopes and ranges from 0 to 108.5%. The difference increases with increasing slope and the increase per degree is larger for every increment (Figure 7).

The theoretical relative difference between surface step lengths and 2D step lengths on rugged slopes of 0° to 59° ranges from 0 to 92%. The difference increases with increasing slope and the increase per degree is larger for every increment. The general trend shows that the difference increases with increasing rugosity (Rq), however, there is some variance in the results. Interestingly, the results indicate a smaller difference for some rugged slopes compared to constant slopes (Figure 8).

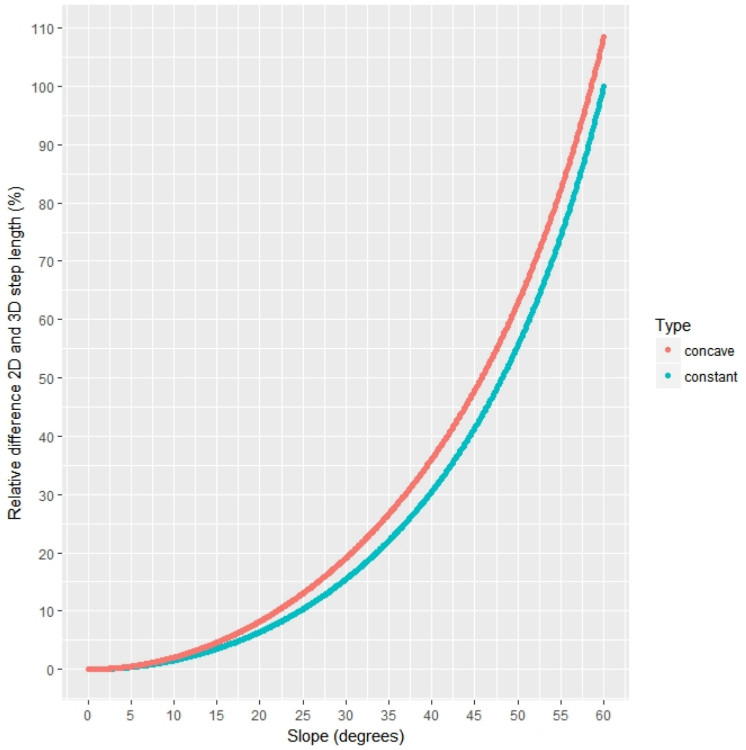

Figure 7. Theoretical relative difference concave slopes. **Surface step length for the concave slope was calculated using equation 4 and 2D step lengths from 0m to 1000m. 3D step length for the constant slope was calculated using equation 3 and 2D step lengths from 0m to 1000m. The relative difference was then calculated with equation 9b.**

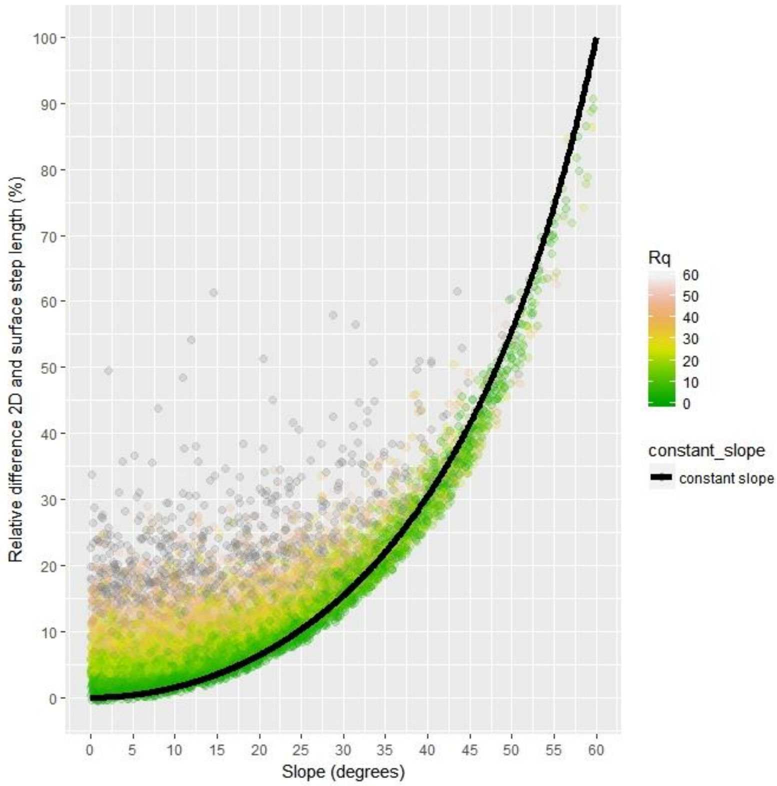

Figure 8. Theoretical relative difference rugged slopes. **Surface step length for the rugged slopes (Rq) was calculated using equation 5 and 2D step lengths from 0m to 1000m. 3D step length for the constant slope was calculated using equation 3 and 2D step lengths from 0m to 1000m. The relative difference was then calculated with equation 9b.**

Empirical data

The selected and cleaned trajectories are visualised in figure 9 (Switzerland) and figure 10 (Germany) and summarized per animal in table 3 (Switzerland) and table 4 (Germany). The median slope of the trajectories in the Swiss study area is 9.44° (Q1 - Q3 is 2.39° - 19.90°), with a median rugosity of 2.13 (Q1 - Q3 is 0.57 - 3.46). The median 2D step length is 44.52m (Q1 - Q3 is 25.66m - 91.02m) and the median 3D step length is 49.53 (Q1 - Q3 is 28.74m - 100.22m). The median difference between 2D and surface step length is then 3.58 (Q1 - Q3 is 0.27m - 9.89m), which corresponds to a median relative difference of 7.3% (Q1 - Q3 is 0.5% -17.2%). For the trajectories in the German study area the median slope is 0.61° (Q1 - Q3 is 0.00°-1.82°), with a median rugosity of 0.28 (Q1 - Q3 is 0.00 - 0.42). The mean 2D step length is 46.92m (Q1 - Q3 is 27.21m - 93.26m) and the median 3D step length is 47.02m (Q1-Q3 is 27.26m - 93.51m). The median difference between 2D and surface step length is then 0.140m, and the median relative difference is 0.05% (Q1 - Q3 is 0.00 - 0.15%).

The max slope of the trajectories in the Swiss study area is 64°, whereas the max slope for the trajectories in the German study area was much lower with 16°. This large difference between study areas reflects also on the maximal median underestimation of step length for an individual's trajectory, that was found to be 18.1% for animal 2240 in the Switzerland dataset and only 0.2% in the German dataset (for 4 different animals). The relative difference for all animals ranged in Switzerland from 0 to 135% and in Germany from 0 to 11.5%. The relative difference stayed in almost all of the steps on or above the theoretical relative difference for constant slopes (Figure 11). The pattern in the rugosity is not very strong, trajectory steps with low rugosity (< 5) however seems to stay closer to the theoretical relative difference than trajectory steps with higher rugosity (Figure 12). The maximum total difference between 2D and surface step lengths for the whole trajectory was 90554m. The rugosity in the empirical data seems to be much lower than the rugosity in the theoretical data.

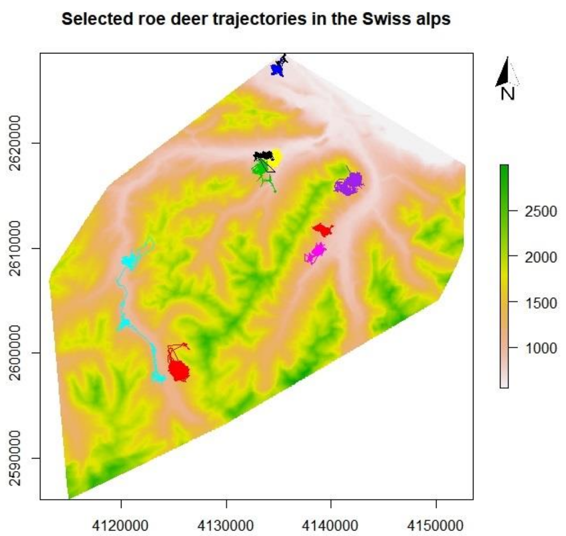

Figure 9. Map selected roe deer trajectories in Switzerland. **The trajectories are recorder over 1 summer (1-Mar to 31-Aug) and regularized to a temporal resolution of 30 minutes. Steps shorter than 15 minutes were removed.**

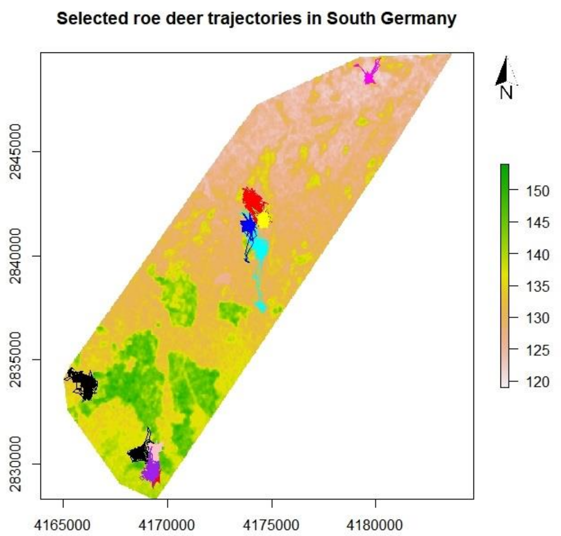

Figure 10. Map selected roe deer trajectories in Germany. **The trajectories are recorder over 1 summer (1-Mar to 31-Aug) and regularized to a temporal resolution of 30 minutes. Steps shorter than 15 minutes were removed.**

Table 6. Summary analyses trajectories Switzerland. **A summary of the trajectories of 10 animals in the high-relief study area in Switzerland. The 2D step length (S_{2D}) is calculated using equation 1, whereas the surface step length (S_{surf}) is calculated using bi-linear interpolation to extract elevation from a DEM at every 10m moving from the start point to the endpoint of the step, using equation 5. The slope and rugosity are calculated using the linear regression, on the so obtained profile. The total difference for the whole trajectory is furthermore calculated by summing the difference of all steps of one individual. The relative difference with respect to the 2D step length is calculated with equation 9b.**

Animal Id	*Slope (°), median (Q1-Q3)*	*Rugosity, median (Q1-Q3)*	*s_{2D} (m), median (Q1-Q3)*	*s_{surf} (m), median (Q1-Q3)*	*Difference (m), median (Q1-Q3)*	*Difference total trajectory (m)*	*Relative difference (%), median (Q1-Q3)*
2239	1.53 (0.16 -2.70)	0.31 (0.24-0.55)	38.17 (23.16 - 73.73)	38.36 (23.23 – 74.03)	0.10 (0.05 – 0.25)	1117.38	0.2 (0.1-0.4)
2240	19.38 (9.94 - 27.57)	3.37 (2.14-4.37)	40.03 (24.68-73.77)	47.89 (30.14-87.57)	12.03 (3.50-15.08)	65165.73	18.1 (9.2-27.7)
2241	8.98 (3.38-19.62)	2.21 (0.79-3.34)	40.71 (24.48-83.17)	45.64 (27.70 – 91.47)	2.84 (0.49-8.47)	33630.21	5.0 (1.1-6.5)
2247	1.44 (0.18-2.54)	0.31 (0.24-0.54)	45.01 (25.12-87.91)	45.18 (25.16-88.2)	0.10 (0.05-0.25)	1199.13	0.2 (0.1-0.4)
2254	8.12 (3.73-16.29)	1.61 (0.66-2.69)	41.32 (24.3-81.89)	44.67 (26.56-87.85)	2.40 (0.44-5.66)	24161.75	4.2 (1.0-11.7)
2259	15.95 (7.35-24.77)	2.88 (1.72-3.87)	38.26 (23.60-71.00)	44.78 (27.81-82.07)	6.40 (2.33 - 11.43)	46501.7	13.9 (6.3-22.5)
2266	10.98 (4.33-20.34)	2.72 (1.36-3.67)	52.30 (28.55-107.64)	58.44 (32.08-122.07)	5.06 (1.66-11.47)	49085.63	9.0 (2.7-17.9)
2292	17.50 (9.00-25.44)	3.44 (2.22-4.42)	52.05 (27.80-116.12)	61.05 (33.32-133.36)	8.92 (4.14-18.46)	88552.04	15.1 (08.2-24.0)
2296	13.72 (5.67-20.39)	2.45 (1.40-3.15)	41.44 (25.76-74.71)	46.19 (28.99-82.98)	4.19 (1.66-8.36)	37748.02	9.7 (3.4-16.6)
2297	13.07 (6.64-20.59)	2.90 (1.91-4.11)	69.85 (31.98-171.25)	80.17 (37.05-193.23)	8.98 (3.59-19.95)	90554.44	11.0 (6.0-18.5)
All	9.44 (2.39-19.90)	2.13 (0.57-3.46)	44.52 (25.66-91.02)	49.53 (28.74-100.22)	3.58 (0.27-9.89)	-	7.3 (0.5-17.2)

Table 4. Summary analyses trajectories Germany. **A summary of the trajectories of 10 animals in the low-relief study area in Germany. The 2D step length (S_{2D}) is calculated using equation 1, whereas the surface step length (S_{surf}) is calculated using bi-linear interpolation to extract elevation from a DEM at every 10m moving from the start point to the endpoint of the step, using equation 5. The slope and rugosity are calculated using the linear regression, on the so obtained profile. The total difference for the whole trajectory is furthermore calculated by summing the difference of all steps of one individual. The relative difference with respect to the 2D step length is calculated with equation 9b.**

Animal Id	*Slope (°), median (IQR)*	*Rugosity, median (IQR)*	*s_{2D} (m), median (IQR)*	*s_{surf} (m), median (IQR)*	*Difference (m), median (IQR)*	*Difference total trajectory (m), median (IQR)*	*Relative difference (%), median (IQR)*
1453	0.57 (0.00-1.88)	0.26 (0.00-0.34)	39.93 (24.75 - 72.76)	39.95 (24.77-72.88)	0.05 (0.00-0.10)	567.33	0.1 (0.0-0.2)
1454	0.65 (0.00-0.98)	0.26 (0.00-0.32)	53.57 (28.36-117.67)	53.61 (28.36-117.73)	0.05 (0.00-0.10)	337.08	0.1 (0.0-0.1)
1455	0.47 (0.00-1.55)	0.25 (0.00-0.31)	37.63 (24.37-62.39)	37.7 (24.39-62.51)	0.05 (0.00-0.06)	416.97	0.1 (0.0-0.2)
2041	1.38 (0.11-2.78)	0.32 (0.23-0.62)	51.48 (29.34-96.57)	51.83 (29.39-96.75)	0.01 (0.05-0.30)	1145.50	0.2 (0.0-0.5)
2043	0.82 (0.00-1.05)	0.26 (0.00-0.32)	52.57 (29.56-114.22)	52.61 (29.61-114.31)	0.14 (0.00-0.10)	496.06	0.0 (0.0-0.1)
2044	1.56 (0.16-2.41)	0.29 (0.23-0.39)	36.57 (23.52-64.22)	36.63 (23.55-64.33)	0.11 (0.05-0.15)	374.75	0.2 (0.1-0.3)
2048	1.02 (0.20-2.38)	0.46 (0.27-0.74)	73.18 (37.18-158.85)	73.83 (37.31-159.21)	0.20 (0.05-0.45)	1340.34	0.2 (0.1 – 0.4)
2065	0.42 (0.00-1.23)	0.29 (0.00-0.42)	65.26 (32.79-136.19)	65.40 (32.86-136.40)	0.05 (0.00-0.15)	484.33	0.1 (0.0-0.2)
2076	1.40 (0.08-2.40)	0.31 (0.23-0.52)	44.52 (27.00-81.42)	44.60 (27.06-81.66)	0.10 (0.05-0.25)	654.66	0.2 (0.1-0.3)
2078	0.32 (0.00-1.08)	0.26 (0.00-0.31)	46.82 (26.55-92.12)	46.86 (26.59-92.16)	0.05 (0.00-0.09)	201.06	0.1 (0.0-0.1)
All	0.61 (0.00-1.82)	0.28 (0.00-0.42)	46.92 (27.21-93.26)	47.02 (27.26-93.51)	0.05 (0.00-0.15)	-	0.1 (0.0-0.2)

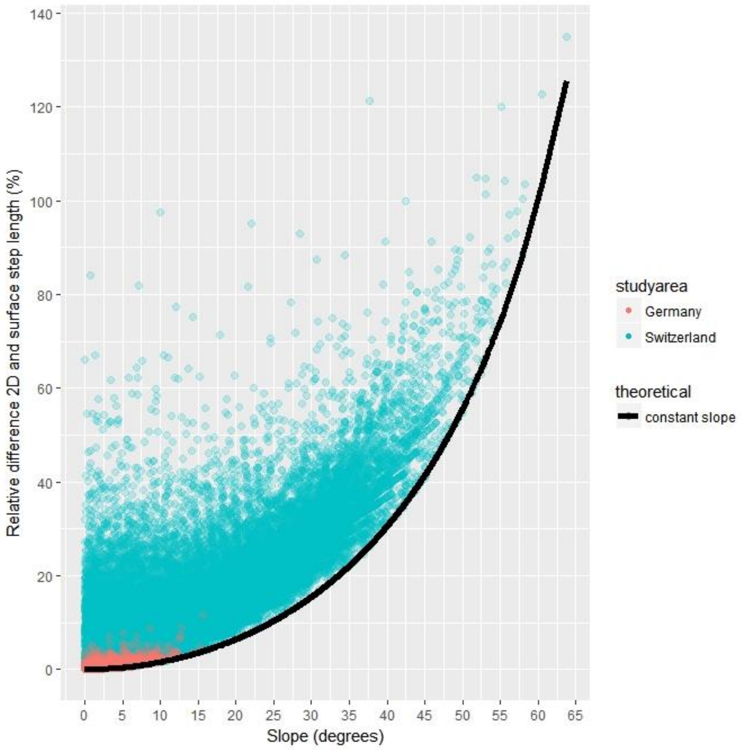

Figure 11. Relative difference trajectory steps Germany and Switzerland. **The relative difference ranges in Switzerland from 0 to 135% and in Germany from 0 to 11.5%. The relative difference stayed in almost all of the steps on or above the theoretical relative difference for constant slopes.**

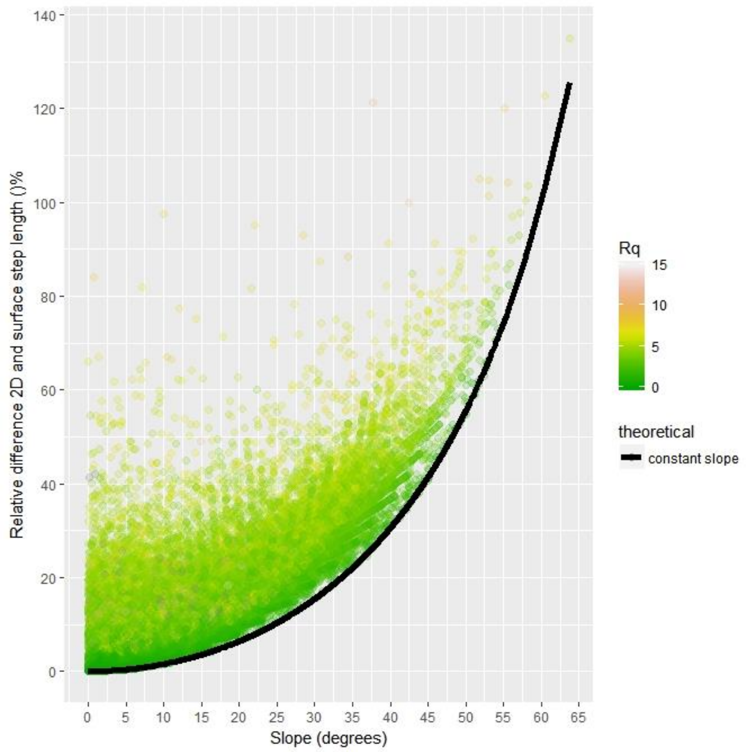

Figure 12. Relative difference trajectory steps rugged slopes. **The pattern in the rugosity is not very strong, trajectory steps with low rugosity (< 5) seems to be more often close to the theoretical relative difference for constant slopes than trajectory steps with higher rugosity.**

Discussion

Step length is one of the basic movement metrics and is used as input for models describing movement patterns (e.g. *correlated random walks* (Turchin, 1998)) and estimating space use (e.g. with *Brownian bridge analysis* (Horne, Garton, Krone, & Lewis, 2007). By defining step length as the Euclidean distance between consecutive relocations, it is obvious and widely acknowledged that the actual distance travelled is underestimated due to unobserved tortuosity of the movement path (e.g. Rowcliffe, Carbone, Kays, Kranstauber, & Jansen, 2012). However, the effect of ignoring the vertical displacement has been given less attention.

By redefining step length in 3-dimensions as the Euclidean surface distance, it was possible to estimate that neglecting the third dimension in roe deer movement data leads to a median underestimation of step length of 7.3% in a high relief area and 0.1% in a low relief area. The minimum difference (relative to 2D step length) can be calculated deterministically with the theoretical difference on constant slopes (Equation 11). While in general an increase in rugosity leads to an increase in difference between surface step length and 2D step length, the magnitude is varying. Given that roe deer are known to have relatively small home ranges, in which topographic differences could be expected to be minimal, it was expected that the difference between 2D and surface step length would to be rather small. Nevertheless, some animals and individual steps showed large underestimations for step lengths.

These findings suggest that previous estimates of step length, based on only longitude and latitude are biased. It is expected that bias for species preferring steep, rugged slopes is larger (e.g. ibex and desert bighorn sheep (Namgail, 2006; Sappington, Longshore, & Thompson, 2007)), whereas, there might be less bias for species using flatter and less rugged terrain, being on step level (wolfs and cougars using linear features in the landscape (Dickie, Serrouya, McNay, & Boutin, 2017; Dickson, Jennes, & Beier, 2005) or on trajectory level in low relief habitats.

While few studies in animal ecology addressed this question, this viewpoint is supported by evidence from other disciplines. Hoechstetter et al. (2008) showed that patch perimeter in landscape ecology is strongly connected to the variability of the terrain and concluded that ‘the effect of relief on the “true” surface distance should not be neglected in rough terrain’. The effect of the third dimension on travelled distances has furthermore been long recognized for navigational purposes; Gaw and Meystel proposed already in 1986 a method to calculate the shortest path on a grid by adding additional costs to slopes and Tavares, Zsigraiova, Semiao and Carvalho (2009) distinguished between “2D distance” and “3D distance” in their effort of optimizing municipal solid waste collection in Praia, the capital of Cape Verde.

The findings of this research seem furthermore to correspond with the body of literature on the difference between 3D and 2D surface metrics, which point out that the underestimation is dependent on the scale (of movement) in the vertical direction (Hoechstetter et al., 2008; Rogers, Cooper, McKenzie, & McCann, 2012; Simpfendorfer et al., 2012; Tracey et al., 2014). In this light it has to be pointed out that while calculating the surface distance through interpolated points between the start point and endpoint of a tracking step is already an improvement on using only the start point and endpoint, it is nevertheless an approximation and largely effected by the resolution of the DEM (i.e. Deng, Wilson, & Bauer, 2007). The use of high-resolution DEMs would allow for an increase in the number of points that can be interpolated between the start and endpoint, which results in a better estimation of the profile for the step. Lidar-derived DEMs seem a promising way to create such high-resolution DEMs. The resolution of the EU-DEM that was used in this study was in fact rather low (1 arc-second, approximately 30m x 30m (Tøttrup, 2014)) compared to the 30 minutes step length of roe deer. Artificially steep slopes could be generated when for a short step the start and endpoints fall in different cells. That seemed to result in overestimations.

When extending this methodology to longer step lengths, the linear regression as estimator of the slope and rugosity must be reconsidered. A linear regression might not fit the profile when an individual moved in succession up and down slope in a single tracking step. It might in fact not be possible to estimate the slope in one value for such a profile. However, it might be possible to estimate the rugosity by fitting higher order regressions. The 3D step length calculation is nevertheless valid, because it the overall slope and rugosity are not needed in the calculations. The 3D step length on a rugged slope, furthermore, was in the theoretical analysis in some cases lower than the 3D step length on the constant slope. This seems a result of the choice of simulation methodology; the endpoint of the constant slope as calculated by the linear regression was not forced to be the same as the endpoint of the rugged slope and the rugged slope could thus be shorter than the constant slope, while this can never be the case if the constant slope and rugged slope have the same start and endpoint.

While this study focuses on the underestimation of the real distance travelled by the animal with 2D rather than 3D calculations, it must be acknowledged that also the observation frequency, interpolation method between fixes (in this case bi-linear interpolation is used) and GPS error do have a potential large impact. Hence, the effect size of the 2D calculations should ideally be considered in the context of these other factors. While a temporal resolution of 30 minutes seems an often applied sampling frequency, Rowcliffe et al., (2012) argued that many fixes per minute are necessary to obtain a reasonable estimation of the real travelled path. Although at the moment this is often impossible due to short battery live and limited memory of the GPS-devices, technological advancement might hold the solution in the future (Tomkiewicz, Fuller, Kie, & Bates, 2010). Another possible approach to obtain detailed trajectories, could be *dead-reckoning* in combination with GPS fixes. Dead-reckoning calculates the travel vector for an animal using information on compass heading, speed and body-orientation and adds this to the previous location, hence the spatial error accumulates in time which in turn can be corrected by periodic GPS fixes (Wensveen, Thomas, & Miller, 2015; Wilson et al., 2007). While this methodology has been mostly used in marine environments, it is also applicable to terrestrial animals (e.g. Bidder et al., 2015).

The bias in step length suggests that other movement metrics might be susceptible to bias as well. In fact, Bailleul et al., (2010) found bias in the First Passage Time (FPT) of 79% of simulated shallow dives (<80m, n=42) and 93% of the simulated deep dives (>80m, n = 558) for the model species beluga whale (*Delphinapterus leucas*) by using a spherical FPT. While this seems a good method for aquatic and avian species which use true volumetric space, terrestrial animals are bound to the surface. The FPT circle is therefore better represented with a surface circle in which the same methods as the three-dimensional step lengths calculations is used to calculate the radius to all sides. In an initial attempt to calculate the difference of the *3D FPT* and the 2D FPT, it has been estimated that the mean theoretical difference for slopes ranging from 0° to 60° and a radius of 5 to 700m ranges from 0 to 13.6h (Figure 13 and methods in appendix V). These results indicate that more research is needed and that neglecting the vertical displacement in tracking data leads possibly to error in more movement metrics.

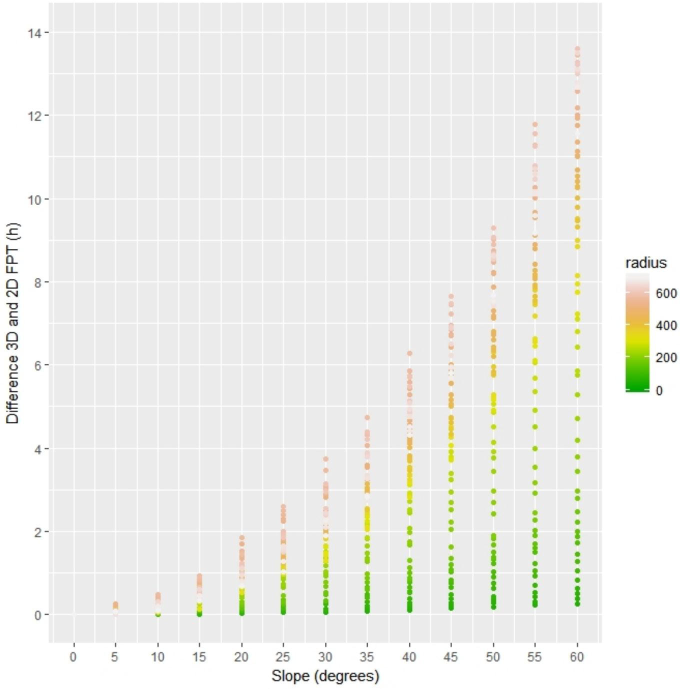

Figure 13. Theoretical mean difference FPT on constant slopes. **The mean difference between 3D and 2D FPT increases with increasing slope and increasing radius (indicated by color). However, the mean difference at a radius of 700 (white) is smaller than the 600m radius, indicating that more simulated trajectories did not exit the FPT circle.**

Conclusion

By calculating 2D and 3D step length at different slope angles and different slope types, it was possible to calculate the theoretical difference caused by ignoring the third dimension. This difference increases with increasing slope and with increasing rugosity, although a large variation in the effect size of rugosity was found. When applied to two roe deer movement datasets in contrasting study areas, the median bias was estimated at 7.3% in the high relief area and 0.1% in low relief area, while the highest median bias for one individual's trajectory was found to be 18.1% in the high relief area and 0.2% in the low relief area.

The minimum underestimation could be deterministically calculated by the theoretical difference on constant slopes, while rugosity added uncertainty to the magnitude. It is thus expected that when extending this methodology to other species and study areas the bias will be larger when steeper and more rugged slopes are used, whereas the bias will be smaller when flatter and less rugged slopes are used. It is therefore recommended to calculated surface step lengths when steep and rugged terrain is used. However, to make accurate estimations of surface step length, high-resolution elevation data is essential. The availability of such data is therefore a prerequisite for this three-dimensional analysis.

In a first exploration, it has been shown that it is likely that other movement metrics are prone to bias as well. While this problem has been acknowledged in other disciplines, it remained a blind spot in terrestrial movement ecology. It is therefore recommended to extent the research of 3D step lengths to other movement metrics and the effects of the bias on applied methodology, like correlated random walks or Brownian bridge models.

Acknowledgments

I would like to thank my supervisors Emiel van Loon, Johannes De Groeve and Francesca Cagnacci for their feedback and suggestions that improved my research, as well as the other students and staff at the movement ecology lab of the Edmund Mach Foundation that gave me advice when needed. I would furthermore, like to thank the Edmund Mach Foundation that hosted me during this research and Gehr Benedikt and Max Kröschel for the use of their datasets through the EURODEER network.

References

Andersen, R., Duncan, P., & Linnell, J. (1998). *The European roe deer: the biology of success*. Oslo: Scandinavian University Press.

Bailleul, F., Lesage, V., & Hammill, M. O. (2010). Spherical first passage time: A tool to investigate area-restricted search in three-dimensional movements. *Ecological Modelling*, *221*(13–14), 1665–1673. http://doi.org/10.1016/j.ecolmodel.2010.04.001

Bidder, O. R., Walker, J. S., Jones, M. W., Holton, M. D., Urge, P., Scantlebury, D. M., … Wilson, R. P. (2015). Step by step: Reconstruction of terrestrial animal movement paths by dead-reckoning. *Movement Ecology*, *3*(1), 1–16. http://doi.org/10.1186/s40462-015-0055-4

Blaga, L. (2012). Aspects regarding the significance of the curvature types and values in the studies of geomorphometry assisted by GIS. *Analele Universitatii Din Orada - Seria Geografie*, *22*(2), 327–337.

Borchers, H. W. (2018). pracma: Practical Numerical Math Functions. R package version 2.1.4. Retrieved from https://cran.r-project.org/package=pracma

Burt, W. H. (1943). Territoriality and Home Range Concepts as Applied to Mammals. *American Society of Mammalogists*, *24*(3), 346–352. Retrieved from https://www.jstor.org/stable/1374834

Cagnacci, F., Boitani, L., Powell, R. A., & Boyce, M. S. (2010). Animal ecology meets GPS-based radiotelemetry: A perfect storm of opportunities and challenges. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *365*(1550), 2157–2162. http://doi.org/10.1098/rstb.2010.0107

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*, 516–510.

Calenge, C., Dray, S., & Royer-Carenzi, M. (2009). The concept of animals' trajectories from a data analysis perspective. *Ecological Informatics*, *4*(1), 34–41. http://doi.org/10.1016/J.ECOINF.2008.10.002

Deng, Y., Wilson, J. P., & Bauer, B. O. (2007). DEM resolution dependencies of terrain attributes across a landscape. *International Journal of Geographical Information Science*, *21*(2), 187–213. http://doi.org/10.1080/13658810600894364

Dickie, M., Serrouya, R., McNay, R. S., & Boutin, S. (2017). Faster and farther: wolf movement on linear features and implications for hunting behaviour. *Journal of Applied Ecology*, *54*(1), 253–263. http://doi.org/10.1111/1365-2664.12732

Dickson, B. G., Jennes, J. S., & Beier, P. (2005). Influence of Vegetation, Topography, and Roads on Cougar Movement in Southern California. *Journal of Wildlife Management*, *69*(1), 264–276. http://doi.org/10.2193/0022-541X(2005)069<0264:IOVTAR>2.0.CO;2

Dodge, S., Weibel, R., & Lautenschütz, A.-K. (2008). Towards a Taxonomy of Movement Patterns. *Information Visualization*, *7*, 240–252. http://doi.org/10.1057/palgrave.ivs.9500182

Edelhoff, H., Signer, J., & Balkenhol, N. (2016). Path segmentation for beginners: An overview of current methods for detecting changes in animal movement patterns. *Movement Ecology*, *4*(1). http://doi.org/10.1186/s40462-016-0086-5

European Union. (2018). EU-DEM v1.0. Retrieved from https://land.copernicus.eu/pan-european/satellite-derived-products/eu-dem/eu-dem-v1-0-and-derived-products/eu-dem-v1.0?tab=mapview

Fauchald, P., & Tveraa, T. (2003). Using First-Passage Time in the Analysis of Area-Restricted Search and Habitat Selection. *Ecology*, *84*(2), 282–288.

Gaw, D., & Meystel, A. (1986). Minimum-time navigation of an unmanned mobile robot in a 2-1/2D world with obstacles. *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, *3*, 1670–1677. http://doi.org/10.1109/ROBOT.1986.1087445

Hoechstetter, S., Walz, U., Dang, L. H., & Thinh, N. X. (2008). Effects of topography and surface roughness in analyses of landscape structure - A proposal to modify the existing set of landscape metrics. *Landscape Online*, *3*(1), 1–14. http://doi.org/10.3097/LO.200803

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

Koeppl, J. W., Slade, N. A., Harris, K. S., & Hoffmann, R. S. (1977). A Three-Dimensional Home Range Model. *Journal of Mammalogy*, *58*(2), 213–220. http://doi.org/10.2307/1379578

Lecours, V., Devillers, R., Simms, A. E., Lucieer, V. L., & Brown, C. J. (2017). Towards a framework for terrain attribute selection in environmental studies. *Environmental Modelling and Software*, *89*, 19–30. http://doi.org/10.1016/j.envsoft.2016.11.027

Marsh, L. M., & Jones, R. E. (1988). The form and consequences of random walk movement models. *Journal of Theoretical Biology*, *133*(1), 113–131. http://doi.org/10.1016/S0022-5193(88)80028-6

Millspaugh, J. J. (2001). *Radio Tracking and Animal Populations*. (J. M. Marzluff, Ed.). Academic Press. http://doi.org/https://doi.org/10.1016/B978-0-12-497781-5.X5000-2

Mitani, Y., Andrews, R. D., Sato, K., Kato, A., Naito, Y., & Costa, D. P. (2010). Three-dimensional resting behaviour of northern elephant seals: Drifting like a falling leaf. *Biology Letters*, *6*(2), 163–166. http://doi.org/10.1098/rsbl.2009.0719

Namgail, T. (2006). Winter Habitat Partitioning Between Asiatic Ibex and Blue Sheep in Ladakh , Northern India. *Journal of Mountain Ecology*, *8*(June), 7–13. Retrieved from http://mountainecology.org/index.php/me/article/view/18

Nathan, R., Getz, W. M., Revilla, E., Holyoak, M., Kadmon, R., Saltz, D., & Smouse, P. E. (2008). A movement ecology paradigm for unifying organismal movement research. *Proceedings of the National Academy of Sciences*, pnas.0800375105. http://doi.org/10.1073/PNAS.0800375105

Ousterhout, B. H., & Burkhart, J. J. (2017). Moving beyond the plane: measuring 3D home ranges of juvenile salamanders with passive integrated transponder (PIT) tags. *Behavioral Ecology and Sociobiology*, *71*(4). http://doi.org/10.1007/s00265-017-2284-6

Papastamatiou, Y. P., Iosilevskii, G., Leos-Barajas, V., Brooks, E. J., Howey, L. A., Chapman, D. D., & Watanabe, Y. Y. (2018). Optimal swimming strategies and behavioral plasticity of oceanic whitetip sharks. *Scientific Reports*, *8*(1), 1–12. http://doi.org/10.1038/s41598-017-18608-z

R Core Team. (2018). A Language and Environment for Statistical. Vienna, Austria: R Foundation for Statistical Computing. http://doi.org/citeulike-article-id:2400517

Recio, M. R., Mathieu, R., Maloney, R., & Seddon, P. J. (2010). First results of feral cats (Felis catus) monitored with GPS collars in New Zealand. *New Zealand Journal of Ecology*, *34*(3), 288–296.

Rogers, D., Cooper, A., McKenzie, P., & McCann, T. (2012). Assessing regional scale habitat area with a three dimensional measure. *Ecological Informatics*, *7*(1), 1–6. http://doi.org/10.1016/j.ecoinf.2011.10.003

Rowcliffe, J. M., Carbone, C., Kays, R., Kranstauber, B., & Jansen, P. A. (2012). Bias in estimating animal travel distance: The effect of sampling frequency. *Methods in Ecology and Evolution*, *3*(4), 653–662. http://doi.org/10.1111/j.2041-210X.2012.00197.x

Sappington, J. M., Longshore, K. M., & Thompson, D. B. (2007). Quantifying Landscape Ruggedness for Animal Habitat Analysis: A Case Study Using Bighorn Sheep in the Mojave Desert. *Journal of Wildlife Management*, *71*(5), 1419–1426. http://doi.org/10.2193/2005-723

Simpfendorfer, C. A., Olsen, E. M., Heupel, M. R., & Moland, E. (2012). Three-dimensional kernel utilization distributions improve estimates of space use in aquatic animals. *Canadian Journal of Fisheries and*

Aquatic Sciences, *69*(3), 565–572. http://doi.org/10.1139/f2011-179

Smith, D. H. V, Wilson, D. J., Moller, H., Murphy, E. C., & van Heezik, Y. (2007). Selection of alpine grasslands over beech forest by stoats (Mustela erminea) in montane southern New Zealand. *New Zealand Journal of Ecology*, *31*(1), 88–97.

Tavares, G., Zsigraiova, Z., Semiao, V., & Carvalho, M. G. (2009). Optimisation of MSW collection routes for minimum fuel consumption using 3D GIS modelling. *Waste Management*, *29*(3), 1176–1185. http://doi.org/10.1016/j.wasman.2008.07.013

Tomkiewicz, S. M., Fuller, M. R., Kie, J. G., & Bates, K. K. (2010). Global positioning system and associated technologies in animal behaviour and ecological research. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *365*(1550), 2163–2176. http://doi.org/10.1098/rstb.2010.0090

Tøttrup, C. (2014). EU-DEM Statistical Validation Report. Retrieved from https://land.copernicus.eu/user-corner/technical-library/eu-dem-2013-report-on-the-results-of-the-statistical-validation

Tracey, J. A., Sheppard, J., Zhu, J., Wei, F., Swaisgood, R. R., & Fisher, R. N. (2014). Movement-Based estimation and visualization of space use in 3D for wildlife ecology and conservation. *PLoS ONE*, *9*(7). http://doi.org/10.1371/journal.pone.0101205

Turchin, P. (1998). *Quantitative Analysis of Movement*. Beresta Books.

van Loon, E. (2018). amoter: Analyse movement over terrain. Retrieved from https://github.com/emielvanloon/amoter

Weimerskirch, H., Le Corre, M., Ropert-Coudert, Y., Kato, A., & Marsac, F. (2005). The three-dimensional flight of red-footed boobies: Adaptations to foraging in a tropical environment? *Proceedings of the Royal Society B: Biological Sciences*, *272*(1558), 53–61. http://doi.org/10.1098/rspb.2004.2918

Wensveen, P. J., Thomas, L., & Miller, P. J. O. (2015). A path reconstruction method integrating dead-reckoning and position fixes applied to humpback whales. *Movement Ecology*, *3*(1), 1–16. http://doi.org/10.1186/s40462-015-0061-6

Wilson, R. P., Liebsch, N., Davies, I. M., Quintana, F., Weimerskirch, H., Storch, S., … McMahon, C. R. (2007). All at sea with animal tracks; methodological and analytical solutions for the resolution of movement. *Deep-Sea Research Part II: Topical Studies in Oceanography*, *54*(3–4), 193–210. http://doi.org/10.1016/j.dsr2.2006.11.017

Zeileis, A., & Grothendieck, G. (2005). zoo: S3 Infrastructure for Regular and Irregular Time Series. *Journal of Statistical Software*, *14*(6), 1–27. http://doi.org/10.18637/jss.v014.i06

Geen gegevens voor lijst met afbeeldingen gevonden.

Appendix

I. Glossary

2D analysis: two-dimensional analysis using only *x* and *y* coordinates (usually latitude and longitude) of location fixes.

3D analysis: three-dimensional analysis using besides *x* and *y* coordinates also *z* coordinates of location fixes, representing altitude, depth or elevation.

Brownian bridge analysis: analysis estimating trajectories and home ranges based on temporal correlated location fixes, including the variance in step length and time lag into the predication (Horne, Garton, Krone, & Lewis, 2007).

Concave slopes: non-constant slopes with a profile-curvature following the equation 6 (see also: *profile curvature)*

Constant slopes: slope with a constant rate of change in elevation (see also: *slope* and *non-constant slope*)

Correlated random walk: a movement pattern in which the turning angle of a step is correlated to the previous steps.

Dead-reckoning: method to estimate a trajectory by calculating the travel vector for an animal using information on compass heading, speed and body-orientation and adding this to the previous location.

DEM: Digital Elevation Model. In this thesis a raster DEM is used, in which every cell contains the elevation data.

EURODEER project: a collaborative science project on roe deer biology within the environmental and evolutionary context joined by 39 research groups throughout Europe, based on a shared spatial database for movement and ancillary information on roe deer (http://www.eurodeer.org).

European roe deer (Capreolus capreolus): Widespread and common small deer species (18-49kg), with a red-brown summer pelage and a grey winter pelage. Roe deer occupying a wide variety of habitats. Only high alpine areas and open grasslands are rarely occupied (Andersen, Duncan, & Linnell, 1998).

First Passage Time (FPT): The time required for an animal to cross a circle with a given radius (Johnson et al. 1992b as cited in Fauchald & Tveraa, 2003).

First Passage Time 3D (FPT_{3D}): The time required for an animal to cross a circle with a given radius (Johnson et al. 1992b as cited in Fauchald & Tveraa, 2003) applied to three dimensional constant slopes. The radius of the circle meets in all directions the Euclidean distance calculated using three coordinates (x, y, z).

Home range: "that area traversed by the individual in its normal activities of food gathering, mating, and caring for young" (Burt, 1943).

Location fix: The location of an animal as estimated by a tracking device at time (t) . In 2D analysis represented by the notation (x_t, y_t) and in 3D analysis represented by the notation (x_t, y_t, z_t).

Mark-recapture methods: method to estimate movement metrics by capturing, marking and releasing individuals. The number of individuals recaptured in a second capturing attempt gives estimates of for example the overall displacement or movement rate (besides population size).

Movement: A change in the spatial location of the whole individual in time (Nathan et al., 2008).

Movement metric: Variable describing movement. See Edelhoff et al. (2016) and appendix II for a comprehensive list.

Non-constant slope: slope with varying rate of change in elevation. (See also: *slope*, *constant-slope* and *rugosity*)

Profile curvature: the shape of the slope measured parallel to the direction of the maximum slope. In this thesis the upward concave slope is used, as visualized in figure 1 and 2.

Random walk: a movement pattern in which the successive steps are not correlated and thus random (see also: *correlated random walks*).

Relative difference: proportional change, for step length calculated with equations 9a and 9b.

Rugged slopes: Non-constant slopes with rugosity (see also: *rugosity*)

Rugosity: the variability in slope gradient. In this thesis calculated with the measure Rq and used interchangeable with R*q* (see also: R_q and *rugged slopes*).

R_q: the root-mean-square deviation of the surface as calculated with equation 7. As measure of rugosity used in this thesis interchangeable with rugosity (see also: *rugosity*).

Slope: the rate of change in elevation, in this thesis described by slope angle (α) in degrees.

Space use patterns: patterns that describe the organization of animals in space and time (Millspaugh, 2001)

Step length 2D (s_{2D}): Euclidian distance between successive location fixes using only the planar coordinates (x, y).

Step length 3D (s_{3D}): Euclidean distance between two successive location fixes (x_t, y_t, z_t) and $(x_{t+1}, y_{t+1}, z_{t+1})$ for tracking step $(t, t+1)$. Calculated with equation 2 and 3.

Surface step length (s_{surf}): Euclidean surface distance between two successive relocations, using an interpolation technique to obtain information about the elevation in between the start point and the endpoint of a tracking step. Calculated with equation 5.

Topographic variable: variable describing the topography of an area. See Lecours, Devillers, Simms, Lucieer, and Brown (2017) and appendix III for a comprehensive list..

Tracking devices: animal-borne instruments that measure and memorize the animal's location at specific time-intervals. The roe-deer datasets in this thesis were tracked with GPS-collars.

Trajectory: temporal sequence of locations.

II. Movement metrics (Edelhoff et al., 2016)

Characteristic	Description	Type	Calculation
Displacement	Increment of the X and Y values between two consecutive relocations, change in absolute spatial position	primary	stepwise
Time lag	Duration / increment in time between consecutive relocations (usually determined by sampling regime)	primary	stepwise
Turning angles / heading	Relative and absolute turning angles between consecutive relocations, change in direction	primary	stepwise
Step length	Euclidean distance between two consecutive relocations	primary	stepwise
Velocity / speed	Distance traveled in a given time interval between two relocations; less sensitive to missing data than step length	primary	stepwise
Persistence / turning velocity	Transformations of speed and turning angle: persistence velocity represents the tendency and degree of a movement to persist in a certain direction. Turning velocity shows the tendency of a movement to turn in a perpendicular/opposite direction	secondary	stepwise
Net / mean squared displacement	Squared displacement between the first and current relocation of the trajectory; applied to characterize diffusion behavior or migration patterns	secondary	stepwise
First passage time	Time required for crossing a predefined endpoint based on a circle (radius) around a starting relocation. Sums the times of all forward and backwards relocations within the radius; index of area-restricted search behavior	secondary	stepwise
Residence time	Extension of the first passage time accounting for returns of the animal in a given area. Sums the times of all relocations (backwards and forwards) of a trajectory within a given vicinity around a relocation.	secondary	stepwise
Pseudo-Azimuth	Recalculates the basic azimuth value at the midpoint between two consecutive steps to range within 0 and 360. Can be used as indicators for movements with same or parallel directions.	primary	stepwise
Straightness index	Ratio of Euclidean distance between the beginning and end of a trajectory and the total path length (sum of all step lengths)	secondary	across multiple steps
Sinuosity / Tortuosity	Adaptions of the straightness index analyzing the probabilistic distributions of the changes in the turning angles and the beeline distance between the start and end points of the trajectory; index of path orientation	secondary	across multiple steps

Fractal dimension	Measure of path tortuosity; non-Euclidean dimension of the trajectory varying between one (completely straight) and two (tortuous, completely spanning two-dimensional space); different implementations exist	secondary	across multiple steps
Multi-scale straightness index	Repeated calculation of the straightness index of a trajectory over a range of different temporal scales	secondary	across multiple steps
Area interest index	Repeated calculation of the straightness index for a limited size of a sliding window along the trajectory. With each repetition, the number of relocations within the trajectory is reduced	secondary	across multiple steps

III. Topographic Metrics (Lecours et al., 2017)

a)

		Group 1	Group 2	Group 3A	Group 3B	Group 4	Group 5A	Group 5B
MAJOR GROUPS	Category	Curvatures and Relative Position	Rugosity Indices	Orientation (Aspect)		Local Statistical Attributes	Slope	
	Terrain Attributes	Topographic Position Index, General Curvature, Deviation from Mean Value, Plan Curvature, Convergence Index, Minimum Curvature, Maximum Curvature, Mean Curvature, Morphometric Position Index	Terrain Ruggedness Index, Surface Ratio, Standard Deviation, Melton Ruggedness Index, Roughness, Range	Easterness	Northerness	Mean, Median, Minimum, Maximum	Slopes (Horn's Method)	Slopes (4-Cell Method, Zevenbergen & Thorne Method)

		Group 5C	Group 5D	Group 5E	Group 6A	Group 6B	Group 7	Group 8
MINOR GROUPS	Category	Slope			Vector Ruggedness Measures		Rugosity Indices	
	Terrain Attributes	Slopes (Maximum Slope Algorithm and Maximum Triangle Slope Algorithm)	Slopes (Quadratic Surfaces, Least-Square Fit Algorithm)	Other Slope Algorithms	Vector Ruggedness Measures (Sappington's Method)	Vector Ruggedness Measures (SAGA GIS Algorithms)	Planimetric-to-Surface Ratio, Ruggedness Index	Representativeness, Residual at Centre

		Group 9A	Group 9B	Group 9C	Group 9D	Group 9E	Group 9F	Group 9G
MINOR GROUPS	Category	Curvatures						
	Terrain Attributes	Profile Curvatures (TNT mips Algorithms)	Plan Curvatures (TNT mips Algorithms)	Profile & Plan Curvatures (SAGA GIS Algorithms)	Curvatures (DEM Surface Tools' Algorithms)	Curvatures (WhiteBox GIS Algorithms)	Curvatures (Haralick's Algorithm)	Curvatures (uDig Algorithms)

IV. Rugged slope simulations

1000 rugged slope profiles of 1000m 2D length are simulated at 1m resolution. De elevation e_i at every point i along the profile is generated with the following rules;

1) e_0 set to zero
2) e_1 is random generated for a normal distribution with mean e_0 and standard deviation 20.
3) e_2 is random generated for a normal distribution with mean $(e_0 + e_1)$ and standard deviation 20.
4) For i = 3 to 1000; e_i is random generated for a normal distribution with mean $(e_{i-1} + e_{i-2} + e_{i-3})$ and standard deviation 20.
5) The 300 points moving average is calculated for the whole profile, using the R package 'zoo' (Zeileis & Grothendieck, 2005).

V. Methods theoretical difference FPT on constant slopes

To calculate the difference between 3D FPT (FPT_{3D}) and 2D FPT (FPT_{2D}), 250 simulated random walks of 100 steps are created using the 'adehabitatLT' package in R (Calenge, 2006). The time interval between each location fix had been set to 30 minutes to match the roe deer datasets. The so simulated random walks have been assumed to be located on constant slopes, of subsequently 0° to 60°, with intermediate steps of 5°. The time needed to cross a circle of a given radius could thus be calculated by finding the first intersect of the trajectory and the circle, which is FPT_{3D}. However, this circle is distorted in 2D and forms an ellipse (Figure 14). The second radius of the ellipse (r_{3D}) can be calculated with

$$r_{2D} = r_{3D} * cos(\alpha) \tag{11}$$

Where r_{2D} is the radius of the circle and α is the slope angle. The first intersect of the trajectory and the ellipse is the FPT_{2D}. Since the FPT in 3D is longer than the FPT in 2D, the difference is calculated by subtracting FPT_{2D} from FPT_{3D} for slopes ranging from 0° - 60° and r_{3D} ranging from 5m – 700m. Subsequently, the mean of the difference for every combination of slope and radius has been calculated.

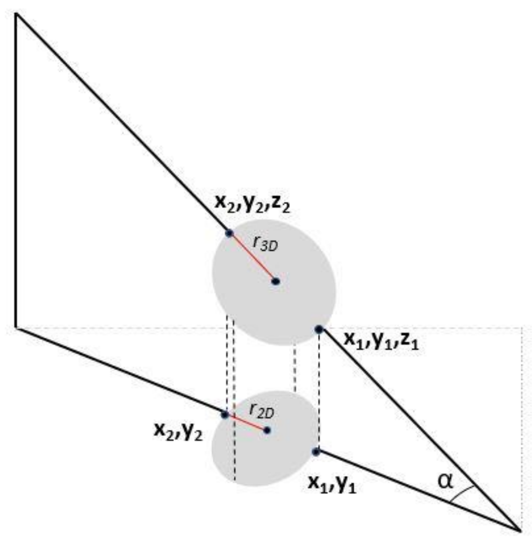

Figure 14. Circles FPT in 3D and 2D. **The second radius (r_{2D}) of the FPT_{2D} can be calculated with equation 11.**

VI. Scripts

a. Theoretical difference 3D and 2D step length at constant slopes

In this script the difference between 3D and 2D step length on a constant slope is calculated. This is done for 2D step lengths ranging from 0 to 1000m and slopes ranging from 0 to 60 degrees. However, this can be changed according to other needs) The difference is calculated by subtracting the 2D step length from the 3D step length. Dividing by the 2D step length results in the relative difference (proportional change in step length).

```
library(ggplot2)
```

Change these variables if other ranges are needed.

```
# range of step lengths in 2D
min_steplength2D <-  0
max_steplength2D <- 1000
increment_steplength2D <- 100
# range of slopes
min_slope <- 0
max_slope <- 60
increment_slope <- 0.1
# sequences of step length and slope to calculate
steplength_2D <-  seq(min_steplength2D, max_steplength2D,
increment_steplength2D)
slope <- seq(min_slope, max_slope, increment_slope)
# make sure lists are empty
steplength_3D <- NULL
diff_3D_2D <- NULL
x <- NULL
y <-  NULL
```

In the following block of code the actual calculations are conducted.

```
# calculate steplength in 3D and the difference between 3D-2D for every
combination
for (step in steplength_2D) {
  for (angle in slope) {
    D3 <- step/cos(angle*pi/180)
    steplength_3D <- append(steplength_3D, D3)
    diff <- D3 - step
    diff_3D_2D <- append(diff_3D_2D, diff)

    # append step and angle to lists for plotting
    x <- append(x, step)
    y <- append(y, angle)
  }
}
```

Plot the relative difference.

```
# create dataframe
th_slope <- data.frame("slope"=y, "di2D"=x, "di3D"=steplength_3D,
"didiff"=diff_3D_2D)
th_slope$reldiff=th_slope$didiff/th_slope$di2D
# plot
pl <- ggplot() + geom_point(data = th_slope, aes(x=slope, y = reldiff)) +
labs(x= 'Slope (degrees)', y="Relative difference 3D and 2D step length") +
scale_x_continuous(breaks = scales::pretty_breaks(n = 13)) +
scale_y_continuous(breaks = scales::pretty_breaks(n=10))
```

```
pl
```

b. Theoretical difference 3D and 2D step length at concave slopes

In this script the difference between 3D and 2D step length on concave slopes, ranging from 0 - 60 degrees (default, but this can be changed according to other needs) is calculated as the 3D step length minus the 2D step length. Dividing the difference by the 2D step length returns the relative difference, which is the output of this script.

```
library(pracma)
library(tidyr)
library(ggplot2)
```

In the following block of code the ranges of step length 2D and slope or set. These can be modified if other ranges are needed.

```
# range of step lengths in 2D
min_steplength2D <-  0
max_steplength2D <- 1000
increment_steplength2D <- 100
# range of slopes
min_slope <- 0
max_slope <- 60
increment_slope <- 0.1
```

Create sequences and empty lists that are needed in the rest of the script.

```
# sequences of step length and slope to calculate
steplength_2D <-  seq(min_steplength2D, max_steplength2D,
increment_steplength2D)
slope <- seq(min_slope, max_slope, increment_slope)
# list of a
# a <- (tan(45*pi/180)*steplength_2D)/(steplength_2D^2)
# make sure lists are empty
steplength_3D <- NULL
steplength_3D_concave <- NULL
diff_3D_2D <- NULL
diff_3D_concave <- NULL
diff_2D_concave <- NULL
x <- NULL
y <-  NULL
# define parametrized function
f <- function(t) c(t, a*t^2)
```

In the following block of code the actual calculations are conducted.

```
# calculate step length in 3D and the difference between 3D-2D for every
combination
for (step in steplength_2D) {
  for (angle in slope) {

    # concave slope = ax^2: calculate 'a' for step (x) and the angle (to
determine the height (y) at x by stable slope)
    a <- (tan(angle*pi/180)*step)/(step^2)

    # calculate step length on the straight slope
    D3 <- step/cos(angle*pi/180)

    # calculate step length on the concave slope
    concave <- arclength(f,0,step)$length

    # append to lists
    steplength_3D <- append(steplength_3D, D3)
```

```
    steplength_3D_concave <- append(steplength_3D_concave, concave)

    # calculate difference
    diff_3D_2D <- append(diff_3D_2D, D3 - step)
    diff_3D_concave <- append(diff_3D_concave, concave - D3)
    diff_2D_concave <- append(diff_2D_concave, concave - step)

    # append step and angle to lists for plotting
    x <- append(x, step)
    y <- append(y, angle)
  }
}
```

Finally, the relative difference is plotted, as is the difference between the constant and concave slopes.

```
# create dataframe
th_cncv <- data.frame("slope"=y, "di2D"=x, "di3D"=steplength_3D,
"di3D_cncv"=steplength_3D_concave, "didiff"=diff_3D_2D,
"diff3D_cncv"=diff_3D_concave, "diff2D_cncv"=diff_2D_concave)
th_cncv$constant=th_cncv$didiff/th_cncv$di2D
th_cncv$reldiff3D_cncv=th_cncv$diff3D_cncv/th_cncv$di2D
th_cncv$concave=th_cncv$diff2D_cncv/th_cncv$di2D
th_cncv2 <- th_cncv %>% tidyr::gather(key=Type, value=rell_diff, constant,
concave)
# plot
pl <- ggplot(data = th_cncv2, aes(x = slope, y = rell_diff, color = Type))
+ geom_point() + labs(x= 'Slope (degrees)', y="Relative difference 3D and
2D step length") + scale_x_continuous(breaks = scales::pretty_breaks(n =
13)) + scale_y_continuous(breaks = scales::pretty_breaks(n=10))
pl
pl2 <- ggplot(data = th_cncv, aes(x = slope, y = reldiff3D_cncv)) +
geom_point() + labs(x= 'Slope (degrees)', y="Relative difference 3D step
length on constant and concave slopes") + scale_x_continuous(breaks =
scales::pretty_breaks(n = 13)) + scale_y_continuous(breaks =
scales::pretty_breaks(n=10))
pl2
```

c. Difference between 3D and 2D step lengths at rugged slopes

In this script the difference between 2D and 3D step length on a rugged slope is calculated. This is done by generating a number of simulated profiles (default 1000, but this can be changed according to other needs) and calculate the step length by following these slopes (3D) and a flat surface (2D). the difference is calculated by subtracting the 2D step length from the 3D step length and dividing this difference by the 2D step length results in the relative difference.

```
library(zoo)
library(sp)
```

Change the ranges to your needs.

```
# range of step lengths in 2D
min_steplength2D <-  0
max_steplength2D <- 1000
increment_steplength2D <- 100
# other variables
Sd <- 20
mean <- 0
n_simulations <- 1000
moving_avg <- 300
resolution <- 1
```

Create sequences and empty lists that you need in the rest of the script.

```
# sequence of step lengths
steplength_2D <-  seq(min_steplength2D, max_steplength2D,
increment_steplength2D)
max_range_steplength <- max_steplength2D + moving_avg
# empty lists
dist <- NULL
slope <- NULL
di2D <- NULL
di3D_rgh <- NULL
Rq <- NULL
Rqgroup <- NULL
```

Function used in this script

```
distanceCalc <- function(profile_vector){
  # function to calculate the distance between every point on a line
  #
  # args:
  #   profile_vector: a vector with elevation at every point along the line
  #   Uses also max_range_steplength, moving_avg and resolution from the
main script
  #
  # returns:
  #   dist: vector with the distance between every point on the line

  dist[[1]] <- 0
  for (k in 2:(max_range_steplength-moving_avg+1)){
    dist[[k]] <- sqrt((profile_vector[k]-profile_vector[k-1])^2 +
resolution^2)
  }
  return(dist)
}
```

In this block the profiles are created and the slope, rugosity, 2D and surface step lengths are calculated.

```
# create elevation profiles
for (j in 1:n_simulations){
  Rq1 <- Sd
  x <- numeric(max_range_steplength)
  x[1] <- mean
  x[2] <- rnorm(1, x[1], Rq1)
  x[3] <- rnorm(1, (x[2]+x[1])/2, Rq1)
  for( i in 4:max_range_steplength ) {
    x[i] <- rnorm(1, (x[i-1]+x[i-2]+x[i-3])/3, Sd)
  }

  # take the moving average
  profile <- rollmean(x, moving_avg)

  # do a linear regression for different step lengths
  for (step in steplength_2D){
    di2D <- append(di2D, step)

    time = (1:(step))
    reg_lin = lm(profile[1:step+1]~time)

    # find slope
    slope <- append(slope, abs(atan(reg_lin$coefficients[2])/pi*180))

    # find Rq through the residuals as the profile
    residuals <-  resid(reg_lin)
    Rq <- append(Rq, sd(residuals))
    Rqgroup <- append(Rqgroup, floor(sd(residuals)))

    # calcualte distance for the profile and save in matrix
    dist <- distanceCalc(profile[1:step+1])
    cum_dist <- cumsum(dist)
    if (step == 0){di3D_rgh <- append(di3D_rgh, 0)}
    else{di3D_rgh <- append(di3D_rgh, cum_dist[step])}
  }
}
```

Finally, the results are plotted.

```
# create a data frame with all variables
th_rgh <- data.frame("slope"=slope, "di2D"=di2D, "di3D_rgh"=di3D_rgh,
"Rq"=Rq, "Rqgroup"=Rqgroup)
th_rgh$di3D <- th_rgh$di2D/cos(slope*pi/180)
th_rgh$didiff <- th_rgh$di3D-th_rgh$di2D
th_rgh$constant=th_rgh$didiff/th_rgh$di2D
th_rgh$diff2D_rgh <- th_rgh$di3D_rgh - th_rgh$di2D
th_rgh$diff3D_rgh <- th_rgh$di3D_rgh - th_rgh$di3D
th_rgh$reldiff3D_rgh <- th_rgh$diff3D_rgh/th_rgh$di2D
th_rgh$rugosity <- th_rgh$diff2D_rgh/th_rgh$di2D
th_rgh$constant_slope <- rep("constant slope", length(th_rgh$rugosity))
# remove slopes steeper than 60 degrees
th_rgh <- th_rgh[-which(th_rgh$slope>60),]
# sequenses needed for plotting
x = seq(0, 60, 0.1)
y = (((1000/cos(seq(0, 60, 0.1)*pi/180))-1000)/1000)
df = data.frame(slope = x, rugosity=y, constant_slope=rep("constant slope",
length(x)))
# plot
pl <- ggplot(data = th_rgh, aes(x = slope, y = rugosity, color = Rq, size =
constant_slope)) + geom_point(alpha = 0.2) + labs(x= 'Slope (degrees)',
y="Relative difference 3D and 2D step length") + scale_x_continuous(breaks
```

```
= scales::pretty_breaks(n = 13)) + scale_y_continuous(breaks =
scales::pretty_breaks(n=10)) + geom_line(data = df, aes(x=slope,
y=rugosity, color = constant_slope), color='black') +
scale_colour_gradientn(colors=terrain.colors(10), limit = c(0,60))
pl
```

d. Difference between 2D and surface step length for roe deer empirical data

In this script the difference between 2D and 3D step length of real roe deer movement data is calculated.

```
# libraries
library("RPostgreSQL")
library("lubridate")
library("sp")
library("rpostgis")
library("rgdal")
library("raster")
library("adehabitatLT")
library("amoter")
library("ggplot2")
```

Functions used in this script

```
distanceCalc_df <- function(x_vector, y_vector, z_vector=rep(0,10000000)){
  # function to calculate the distance between every point on a line
  #
  # args:
  #   x_vector: numerical array with the x_coordinates
  #   y_vector: numerical array with the y_coordinates
  #   z_vector: numerical array with the z_coordinates, default = zeros
  #
  # returns:
  #   dist: numerical array with the distance between every point on the
line

  dist <- NULL
  dist[[1]] <- 0
  for (k in 2:length(x_vector)){
    dist[[k]] <- sqrt((x_vector[k]-x_vector[k-1])^2+(y_vector[k]-
y_vector[k-1])^2+(z_vector[k]-z_vector[k-1])^2)
  }
  return(dist)
}
cumsumCalc <- function(segment, distance){
  # function to calculate the cumulative distance between every point on a
line, starts on zero with new segment.
  #
  # args:
  #   segment: a list with the index of the point on the segments
  #   distance: a list with the distance between every point on the line
(as calculated with distanceCalc_df)
  #
  # returns:
  #   cumsum: vector with the cumulative distance between every point on
the line, starting on zero at every new segment.

  cumdist <- NULL
  cumdist[[1]] <- cumsum(distance[segment[[1]]])
  for (i in 2:length(segment)){cumdist[[i]] <-
c(0,(cumsum(distance[segment[[i]]])))}
  return(cumdist)
}
terrainCalc <- function(cum_distance, elevation, segment){
  # function to calculate the rugosity and elevation for a tracking step
profile.
  #
```

```
  # args:
  #   cum_distance: cumulative distance between point on the trajectory,
starting on zero for every segment.
  #   elevation: elevation for every point on the trajectory.
  #   segment: a list with the index of the point on the segments
  #
  # returns:
  #   terrain: matrix with the the segment number, rugosity and slope for
every segment.

  # prepare matrix for output
  terrain <- matrix(nrow=0,ncol=3)
  colnames(terrain) <- c('seg','roughness','slope')

  for (i in 1:length(segment)){

    reg_lin <- lm(elevation[segment[[i]]]~cum_distance[[i]])
    residuals <- resid(reg_lin)
    stddev <- sd(residuals)
    slope <- abs(atan(reg_lin$coefficients[2])/pi*180)
    # add result to output
    terrain <- rbind(terrain, cbind(i ,stddev,slope))
    }
  return(terrain)

}
#' @examples
#' require(raster)
#' elev <- raster(ncol=30, nrow=30,ext=extent(0,30,0,30),crs=NA)
#' values(elev) <- runif(ncell(elev))*10
#' p1 <- cbind( x=c(1,8,14,18,23), y=c(3,5,11,4,17))
#' p2 <- cbind( x=c(1,3,4,15,21), y=c(3,12,21,24,16))
#
#  # elevation added, no resampling between observation points
#' p1e <- eal(p1,elev)
#' p2e <- eal(p2,elev)
#'
#' # elevation added with resampling between observation points
#' p1er <- eal(p1,elev,step=1)
#' p2er <- eal(p2,elev,step=1)
#' @export
eal_marrit <- function(co,elev,step=NULL,endpt=TRUE) {
#' elevation along a 2D line, adapted from package 'amoter' (van Loon,
2018)
#'
#' @param co a numeric 2-column matrix with the x-coordinate in the first
#' column and the y-coordinate in the second
#' @param elev a raster object which contains the elevation values for a
domain
#' which covers the x- and y-coordinates
#' @param step the length between the steps on the line
#' @param endpt a binary value to indicate whether the endpoints of each
line
#' segment should be kept in the resampled data (if set to TRUE, the
endpoints
#' are kept in the output)
#' @return a 4-column matrix with the first two columns
#' identical to co, and the z in the third column, the fourth column
contains
#' integers which refer to the line segments in co. The end-point of each
line
```

```
#' segment is also the starting point of the subsequent line segment, and
#' and a choice has been made to number these points by the id of the
#' new segment.
#' @examples
#' require(raster)
#' elev <- raster(ncol=30, nrow=30,ext=extent(0,30,0,30),crs=NA)
#' values(elev) <- runif(ncell(elev))*10
#' p1 <- cbind( x=c(1,8,14,18,23), y=c(3,5,11,4,17))
#' p2 <- cbind( x=c(1,3,4,15,21), y=c(3,12,21,24,16))
#
#  # elevation added, no resampling between observation points
#' p1e <- eal(p1,elev)
#' p2e <- eal(p2,elev)
#'
#' # elevation added with resampling between observation points
#' p1er <- eal(p1,elev,step=1)
#' p2er <- eal(p2,elev,step=1)
#' @export
  if(is.data.frame(co)){
    co<-as.matrix(co)
  }

  if(is.null(step)){
    seg <- c(1:nrow(co))
    return( cbind(co, z=raster::extract(elev,co), seg=seg) )
  }else{
    coi <- ed_resamp_marrit(co, step=step,endpt=endpt)
    zi <- raster::extract(elev,coi[,c(1,2)])
    coi <- cbind(x=coi[,1], y=coi[,2], z=zi, seg=coi[,3])
    return(coi)
  }
}
ed_resamp_marrit <- function(co, step=1, endpt=TRUE) {
#' find x,y coordinates at equal distances along a piecewise linear line,
adapted from package 'amoter' (van Loon, 2018)
#'
#' @param co a numeric 2-column matrix with the x-coordinates in the first
#' column and the y-coordinates in the second
#' @param step the length between the steps on the line
#' @param endpt a boolean value which indicates whether the nodes of the
#' piecewise linear line have to be included
#'
#' @return A 3-column matrix with in the first two columns x and y
coordinates
#' of the equidistant points and in the third columns an integer which
specifies
#' in which line-segment of co each point falls. The end-point of each line
#' segment is also the starting point of the subsequent line segment, and
#' and a choice has been made to number these points by the id of the
#' new segment. To keep this pattern consistent, the very last point of the
#' trajectory has a new id
#' @import stats
#' @examples
#' co <- cbind(x=c(1,3,7,8,5),y=c(12,7,6,10,14))
#'
#' coi1 <- ed_resamp(co, step=1, endpt=FALSE)
#' coi2 <- ed_resamp(co, step=2, endpt=FALSE)
#' plot(co[,1],co[,2],type='l',col='red')
#' points(coi1[,1],coi1[,2],pch=3)
#' points(coi2[,1],coi2[,2],pch=1,col='blue')
#'
```

```
#' coi1 <- ed_resamp(co, step=1, endpt=TRUE)
#' coi2 <- ed_resamp(co, step=2, endpt=TRUE)
#' plot(co[,1],co[,2],type='l',col='red')
#' points(coi1[,1],coi1[,2],pch=3)
#' points(coi2[,1],coi2[,2],pch=1,col='blue')
#' @export


  # for storing outputs
  coit <- matrix(nrow=0,ncol=3)
  colnames(coit) <- c('xi','yi','seg')
  xpart <- FALSE
  for(i in 1:(nrow(co)-1)){

    xpart <- FALSE
    # determine x-coordinate of last point
    if(i==1){
      pt_start <- co[1,]
      xi_start <- co[1,1]
    }else{
      if( coi$x[length(xi)] == co[i,1] | endpt){
        # last point falls on next node OR
        # explicit setting that each node should be part of interpolation
        pt_start <- co[i,]
        xi_start <- co[i,1]
      }else{
        xpart <- TRUE
        pt_start <- c(coi$x[length(xi)], coi$y[length(xi)])
        xi_start <- pt_start[1]
      }
    }
    # check if furthest point in next segment is more than 1 step
    # away from lastpoint, if not: move on to next segment
    # this part is only relevant if endpt == FALSE
    nextsegment <- co[c(i,i+1),]
    rngmax <- max( distrange_pl(co=nextsegment, pt=pt_start) )
    if((step>rngmax) & !endpt){next()}
    # if the remaining distance on the last line segment is smaller than
    # step, new starting x-value on the new segment is determined
    if(xpart){
      xi_start <- intpoint_pl(co=nextsegment, pt=pt_start, dist=step)[1]
      xpart <- FALSE
    }
    # the step length along the x-asis is calculated,
    # when the step length on the path is set to 'step'
    totstep <- diff( dal(nextsegment) )
    xstep <- diff(nextsegment[,1])
    pr_xstep <- step*xstep/totstep
    # the points on next segment are determined by linear interpolation,
if-statement MARRIT
    xi_end <- nextsegment[2,1]
    if (xi_start == xi_end){
      xi = 1
      coi <- data.frame(x=co[i,1], y=co[i,2])

    }else{
      xi <- seq(from=xi_start, to=xi_end, by=pr_xstep)
      coi <- approx(nextsegment, xout=xi)}

    # add result to output
    coit <- rbind(coit, cbind(coi$x,coi$y,rep(i,length(xi))))
```

```
  }

  # adding the very last coordinate of track at the end
  coit <- rbind(coit, c(nextsegment[2,],i+1))
  return(coit)
}
```

Connect to the database

```
# connect to database
drv <- dbDriver("PostgreSQL")
con <- dbConnect(drv, dbname = "eurodeer_db", host = "host", user =
"username", password = "password")

# check if postGIS is enabled on the database
pgPostGIS(con)
```

Retrieve the data from the database

```
# get the right projection of the DEM copernicus
proj4 <- dbGetQuery(con, "SELECT proj4text FROM spatial_ref_sys JOIN
(SELECT st_srid(rast) srid FROM env_data.dem_copernicus limit 1) a USING
(srid);")
# get boundary of studyareas
boundDE <- spTransform(pgGetBoundary(con, c("main", "study_areas"), clauses
= "WHERE study_areas_id = 15"), CRS(as.character(proj4)))
boundCH <- spTransform(pgGetBoundary(con, c("main", "study_areas"), clauses
= "WHERE study_areas_id = 25"), CRS(as.character(proj4)))
# get dem of studyareas
demDE <- pgGetRast(con, c("env_data", "dem_copernicus"), boundary =
boundDE)
demCH <- pgGetRast(con, c("env_data", "dem_copernicus"), boundary =
boundCH)
```

Get some metrics for the whole study area

```
# calculate the mean elevation for the studyareas
demDEMean <- cellStats(demDE, mean)
demCHMean <- cellStats(demCH, mean)
# calculate slope
slopeDE <- terrain(demDE, opt = 'slope', units = 'degrees')
slopeCH <- terrain(demCH, opt = 'slope', units = 'degrees')
mean_relslopeDE <- cellStats(slopeDE, mean)
mean_relslopeCH <- cellStats(slopeCH, mean)
mean_slopeDE <- mean_relslopeDE*100
mean_slopeCH <- mean_relslopeCH*100
```

Now extract the selected trajectories from the database

```
query15 <- "SELECT study_areas_id, animals_id, yyear, acquisition_time,
temp.marrit_de15.geom as geom, fixes_o, fixes_e, prop
FROM temp.marrit_de15
LEFT JOIN main.gps_data_animals USING (animals_id, acquisition_time)
WHERE gps_validity_code = 1
ORDER BY animals_id, acquisition_time;"

query25 <- "SELECT study_areas_id, animals_id, yyear, acquisition_time,
temp.marrit_ch25.geom as geom, fixes_o, fixes_e, prop
FROM temp.marrit_ch25
LEFT JOIN main.gps_data_animals USING (animals_id, acquisition_time)
WHERE gps_validity_code = 1
ORDER BY animals_id, acquisition_time;"
```

```
animals_de15 <- spTransform(pgGetGeom(con, query = query15),
CRS(as.character(proj4)))
animals_ch25 <- spTransform(pgGetGeom(con, query = query25),
CRS(as.character(proj4)))

# make ltraj class
tracksDE <- as.ltraj(coordinates(animals_de15), date =
animals_de15$acquisition_time, id = animals_de15$animals_id)
tracksCH <- as.ltraj(coordinates(animals_ch25), date =
animals_ch25$acquisition_time, id = animals_ch25$animals_id)
mean_slopeCH <- mean_relslopeCH*100
```

Plot the trajectories (if you want)

```
# check the trajectories Switzerland
plot(demCH, main = "Selected roe deer trajectories in the Swiss alps")
lines(tracksCH[[1]], col = 1)
lines(tracksCH[[2]], col = 2)
lines(tracksCH[[3]], col = 3)
lines(tracksCH[[4]], col = 4)
lines(tracksCH[[5]], col = 5)
lines(tracksCH[[6]], col = 6)
lines(tracksCH[[7]], col = 7)
lines(tracksCH[[8]], col = 'purple')
lines(tracksCH[[9]], col = 9)
lines(tracksCH[[10]], col = 10)
# check the trajectories Germany
plot(demDE, main = "Selected roe deer trajectories in South Germany")
lines(tracksDE[[1]], col = 1)
lines(tracksDE[[2]], col = 2)
lines(tracksDE[[3]], col = 'pink')
lines(tracksDE[[5]], col = 5)
lines(tracksDE[[6]], col = 6)
lines(tracksDE[[7]], col = 'red')
lines(tracksDE[[8]], col = 1)
lines(tracksDE[[9]], col = 'blue')
lines(tracksDE[[10]], col = 'purple')
lines(tracksDE[[4]], col = 'yellow')
```

Calculate for the Swiss dataset at every step the step length in 2D, surface step length, slope and roughness.

```
stepsCH <- NULL
for (i in 1:10){

  # determine elevation and distance along profile by linear interpolation.
  tr1 <- cbind(x=tracksCH[[i]]$x, y= tracksCH[[i]]$y)
  tr1e <- eal_marrit(tr1, demCH)
  tr1er <- eal_marrit(tr1, demCH, step = 25)
  tr1er_dist3D <- dalw(tr1er, 3)
  tr1er_dist2D <- dalw(tr1er, 2)

  # calculate the right indexes of the segments
  seg <- ptseg(tr1er)

  # take the double index of the segments (thus not endpoint == startpoint)
  segm <- seg
  for (j in 2:length(segm)){segm[[j]] <- segm[[j]][2:length(segm[[j]])]}
```

```
  # recalculate distance and cumulative distance to use in the terrainCalc
(incl. a linear regression)
  distance <-  distanceCalc_df(tr1er[,1], tr1er[,2])
  cumdist_2D <- cumsumCalc(segm, distance)
  terrain <- terrainCalc(cumdist_2D, tr1er[,3], seg)

  # create a data frame
  joined <-plyr::join(tracksCH[[i]][,c(1:3,6)],data.frame(tr1er),
type="left",by=c("x", "y"))

  # add NA to end to get same length
  tr1er_dist2D <- c(tr1er_dist2D, NA)
  tr1er_dist3D <- c(tr1er_dist3D, NA)
  terrain2 <- rbind(terrain, NA)
  # save results in data.frame
  tr1er_df <- (cbind(rep(id(tracksCH)[i],length(tr1er_dist2D)),joined,
tr1er_dist2D, tr1er_dist3D, (tr1er_dist3D-tr1er_dist2D),
terrain2[,'slope'], terrain2[,'roughness'], rep(25,length(tr1er_dist2D))))
  names(tr1er_df) <- c("id","x", "y", "date","dist", "z" , "seg", "di2D",
"di3D", "didiff", "slope", "roughness", "studyarea")
  # and save result in list
  stepsCH[[i]] <- tr1er_df
}
```

Clean data

```
sub_stepsCH <- NULL
for (i in 1:10){
  # remove steps more than 40' apart
  sub_stepsCH[[i]] <- stepsCH[[i]][-which(diff(stepsCH[[i]][,'date']) >
40),]

  # remove steps < 10m: can be GPS errors
  sub_stepsCH[[i]] <- sub_stepsCH[[i]][-which(sub_stepsCH[[i]][,"di2D"] <
15),]
}
# combine alle animals in one dataframe
sub_stepsCH_all <- rbind(sub_stepsCH[[1]], sub_stepsCH[[2]],
sub_stepsCH[[3]], sub_stepsCH[[4]], sub_stepsCH[[5]], sub_stepsCH[[6]],
sub_stepsCH[[7]], sub_stepsCH[[8]], sub_stepsCH[[9]], sub_stepsCH[[10]])
# plot
plot(sub_stepsCH_all$di2D,sub_stepsCH_all$didiff, ylab= "Difference (3D-2D
(m))", xlab = 'Step length 2D (m)')
plot(sub_stepsCH_all$slope,sub_stepsCH_all$didiff, ylab= "Difference (3D-2D
(m))", xlab = 'Slope (degrees)')
plot(sub_stepsCH_all$roughness,sub_stepsCH_all$didiff, ylab= "Difference
(3D-2D (m))", xlab = 'Roughness')
```

Calculate for every step the step length 2D, surface step length, slope and roughness. For the German dataset.

```
stepsDE <- NULL
for (i in 1:10){
  # determine elevation and distance along profiles by linear
interpolation.
  tr1 <- cbind(x=tracksDE[[i]]$x, y= tracksDE[[i]]$y)
  tr1e <- eal_marrit(tr1, demDE)
  tr1er <- eal_marrit(tr1, demDE, step = 25)
  tr1er_dist3D <- dalw(tr1er, 3)
  tr1er_dist2D <- dalw(tr1er, 2)

  # calculate the right indexes of the segments
```

```
  seg <- ptseg(tr1er)

  # take the double index of the segments (thus not endpoint == startpoint)
  segm <- seg
  for (j in 2:length(segm)){segm[[j]] <- segm[[j]][2:length(segm[[j]])]}

  # recalculate distance and cumulative distance to use in the terrainCalc
(incl. a linear regression)
  distance <-  distanceCalc_df(tr1er[,1], tr1er[,2])
  cumdist_2D <- cumsumCalc(segm, distance)
  terrain <- terrainCalc(cumdist_2D, tr1er[,3], seg)

  # create a data frame
  joined <-cbind(tracksDE[[i]][,c(1:3,6)],tr1e[,3:4])

  # add NA to end to get same length
  tr1er_dist2D <- c(tr1er_dist2D, NA)
    tr1er_dist3D <- c(tr1er_dist3D, NA)
    terrain2 <- rbind(terrain, NA)

  # save results in data.frame
  tr1er_df <- (cbind(rep(id(tracksDE)[i],length(tr1er_dist2D)),joined,
tr1er_dist2D, tr1er_dist3D, (tr1er_dist3D-tr1er_dist2D),
terrain2[,'slope'], terrain2[,'roughness'], rep(15,length(tr1er_dist2D))))
  names(tr1er_df) <- c("id","x", "y", "date","dist", "z" , "seg", "di2D",
"di3D", "didiff", "slope", "roughness", "studyarea")

  # save result in list
  stepsDE[[i]] <- tr1er_df
}
```

Clean data

```
sub_stepsDE <- NULL
for (i in 1:10){
  # remove steps more than 40' apart
  sub_stepsDE[[i]] <- stepsDE[[i]][-which(diff(stepsDE[[i]][,'date']) >
40),]

  # remove steps < 10m: can be GPS errors
  sub_stepsDE[[i]] <- sub_stepsDE[[i]][-which(sub_stepsDE[[i]][,"di2D"] <
15),]
}
# combine alle animals in one dataframe
sub_stepsDE_all <- rbind(sub_stepsDE[[1]], sub_stepsDE[[2]],
sub_stepsDE[[3]], sub_stepsDE[[4]], sub_stepsDE[[5]], sub_stepsDE[[6]],
sub_stepsDE[[7]], sub_stepsDE[[8]], sub_stepsDE[[9]], sub_stepsDE[[10]])
# plot
plot(sub_stepsDE_all$di2D,sub_stepsDE_all$didiff, ylab= "Difference (3D-2D
(m))", xlab = 'Step length 2D (m)')
plot(sub_stepsDE_all$slope,sub_stepsDE_all$didiff, ylab= "Difference (3D-2D
(m))", xlab = 'Slope (degrees)')
plot(sub_stepsDE_all$roughness,sub_stepsDE_all$didiff, ylab= "Difference
(3D-2D (m))", xlab = 'Roughness')
```

Combine Switzerland and Germany data

```
# create one dataframe for all data, add some variables
all <- rbind(sub_stepsCH_all, sub_stepsDE_all)
all$studyarea <- as.factor(all$studyarea)
levels(all$studyarea) <- c("Germany", "Switzerland")    #needed for legend
all$reldiff <- all$didiff/all$di2D      #relative difference (s3D-s2D)/s2D
```

```
all$groupRq <- floor(all$roughness)    # idea to plot the different groups
of rugosity
all$theoretical <- rep("constant slope", length(all$reldiff)) # needed as
for legend
all$rugosity <- all$roughness                                  # renamed
all$Rq <- all$roughness                # needed as for legend
```

Finally, make the different plots!

```
# different plots with ggplot
all_rel_slope_rq <- ggplot(data = all, aes(x = slope, y = reldiff*100,
color = Rq, size = theoretical)) + geom_point(alpha = 0.2) + labs(x= 'Slope
(degrees)', y="Relative difference 2D and surface step length ()%") +
geom_line(data = all, aes(x=slope, y = (((di2D/cos(slope*pi/180))-
di2D)/di2D)*100, color = theoretical), color='black') +
scale_colour_gradientn(colors=terrain.colors(10), limit = c(0,15)) +
scale_x_continuous(breaks = scales::pretty_breaks(n = 13)) +
scale_y_continuous(breaks = scales::pretty_breaks(n=10))
all_rel_slope_area <- ggplot(data = all, aes(x = slope, y = reldiff*100,
color = studyarea, size = theoretical)) + geom_point(alpha = 0.2) + labs(x=
'Slope (degrees)', y="Relative difference 2D and surface step length (%)")
+ geom_line(data = all, aes(x=slope, y = (((di2D/cos(slope*pi/180))-
di2D)/di2D)*100, color = theoretical), color='black') + guides(colour =
guide_legend(override.aes = list(alpha=1))) + scale_x_continuous(breaks =
scales::pretty_breaks(n = 13)) + scale_y_continuous(breaks =
scales::pretty_breaks(n=10))
all_rel_roughness  <- ggplot(data = all, aes(x = roughness, y =
reldiff*100, color = studyarea)) + geom_point(alpha = 0.1) + labs(x=
'Roughness', y="Relative difference 2D and surface step length (%)")
all_rel_slope_rq
all_rel_slope_area
all_rel_roughness
```

e. Theoretical difference 3D and 2D FPT at constant slopes

In this script the difference between 3D and 2D FPT on a constant slope is calculated. This is done for radius of FPT circles ranging from 50 to 700m and slopes ranging from 0 to 60 degrees. However, this can be changed according to other needs. The difference is calculated by subtracting the 2D FPT from the 3D FPT.

```
library(adehabitatLT)
library(spatstat)
library(maptools)
library(rgeos)
```

These variables can be changed according to the ranges you want to investigate

```
# variables
starttime <- ISOdatetime(2018,1,1,0,0,0)
steps <- 100
timelag <- 30     # in minutes
n_simulations <- 250
# range of radius
min_radius <-  50
max_radius <- 700
increment_radius <- 10
# range of slopes
min_slope <- 0
max_slope <- 85
increment_slope <- 5
```

Load the functions that are needed in this script

```
# load functions
passingFun <- function(trajectory, enclosure){
  # function to find step where a trajectory crosses an enclosed area
  #
  # args:
  #   trajectory: trajectory of class ltraj
  #   enclosure: spatial lines data frame enclosing an area
  #
  # returns:
  #   passing: object with the points before and after passing and the step
in which enclosure is passed

  points_out <- which(!point.in.polygon(trajectory[[1]]$x,
trajectory[[1]]$y, enclosure$x, enclosure$y))

  # if there are no points outside the enclosure, return NULL
  if(length(points_out) == 0){
    return(NULL)
  }

  # find the minimum of all points outside the polygon
  minimum <- min(points_out)

  # find both coordinates from first point outside enclosure and last point
before and the assosiated time
  index <- minimum
  x <- trajectory[[1]]$x[index]
  y <- trajectory[[1]]$y[index]
  time <- trajectory[[1]]$date[index]
  first_out <- list(index, x, y, time)
```

```
  names(first_out) <- c("index","x", "y", "time")

  index <- minimum-1
  x <- trajectory[[1]]$x[index]
  y <- trajectory[[1]]$y[index]
  time <- trajectory[[1]]$date[index]
  last_in <- list(index, x, y, time)
  names(last_in) <- c("index","x","y", "time")
  # make a spatial Line of passing step
  L <- Line(cbind(c(last_in$x, first_out$x), c(last_in$y, first_out$y)))
  L_l = Lines(list(L), ID="L")
  step = SpatialLines(list(L_l))

  # save everything in one object
  passing <- list(first_out, last_in, step)
  names(passing) <- c("first_out", "last_in", "step")

  # return
  return(passing)
}
findIntersectFun <- function(trajectory, passingStep, enclosure_sl,
timelag){
  # function to find the specific intersection point
  #
  # args:
  #   trajectory: trajectory of class ltraj
  #   passingStep: object with passingstep (obtained from passingFun)
  #   enclosure_sl: spatial lines data frame enclosing an area
  #   timelag: interval between the trajectory steps
  #
  # return:
  #   first intersecting point between enclosed area and the trajectory

  # make spatial lines df of trajectory
  trajectory_sldf <- ltraj2sldf(trajectory)

  # find the intersecting points
  intersects <- gIntersection(enclosure_sl, trajectory_sldf)

  # find the first intersection
  intersects_buff <- gBuffer(intersects, width=0.001)
  intersect_line <- gIntersection(passingStep$step, intersects_buff)
  intersect_point <- getSpatialLinesMidPoints(intersect_line)
  index_first_intersect <- match(round(intersect_point$x, digits = 2),
round(intersects$x, digits = 2))[1]
  if(is.na(index_first_intersect)){
    index_first_intersect <- match(round(intersect_point$y, digits = 2),
round(intersects$y, digits = 2))[1]
  }

  # save the coordinates of the first intersection
  x_first_intersect <- intersects$x[index_first_intersect]
  y_first_intersect <- intersects$y[index_first_intersect]

  # calculate length of lines
  length_in <- lineLengthFun(passingStep$last_in$x, x_first_intersect,
passingStep$last_in$y, y_first_intersect)
  length_tot <-
lineLengthFun(passingStep$last_in$x,passingStep$first_out$x ,
passingStep$last_in$y, passingStep$first_out$y)
```

```
  # calculate the proportion of the line to the intersect to the total step
and time associated to that proportion
  proportion <- length_in/length_tot
  minutes <- (proportion * timelag)
  time_crossing <- as.POSIXct(minutes*60, origin =
passingStep$last_in$time)

  # make object to return
  first_intersect <- list(index_first_intersect, x_first_intersect,
y_first_intersect, time_crossing)
  names(first_intersect) <- c("index", "x", "y", "time")

  return(first_intersect)

}
lineLengthFun <- function(x_coordinate1, x_coordinate2, y_coordinate1,
y_coordinate2){
  # function to calculate the length of a straigt line based on the
coordinates
  #
  # args:
  #   x_coordinate1: x coordinate of first point
  #   x_coordinate2: x coordinate of last point of straight line
  #   y_coordinate1: y coordinate of first point
  #   y_coordinate2: y coordiante of last point of straight line
  #
  # return:
  #   length of line
  # calculate the length of the x displacement
  x_length <- abs(max(x_coordinate1,x_coordinate2) -
min(x_coordinate1,x_coordinate2))

  # calculate the length of the y displacement
  y_length <- abs(max(y_coordinate1,y_coordinate2) -
min(y_coordinate1,y_coordinate2))

  # calculate the Euclidean distance between the two endpoints of the line
  Eucledian_length <- sqrt(x_length^2 + y_length^2)

  return(Eucledian_length)
}
```

In this block the other variables and parameters that should not be changed are created.

```
# setting up datetime vector
date_seq <- seq(from = starttime, by = paste(timelag, "min"), length.out =
steps)

# sequences of steplength and slope to calculate
radius <-  seq(min_radius, max_radius, increment_radius)
slope <- seq(min_slope, max_slope, increment_slope)

# empty lists
FPT_list3D <- NULL
FPT_list2D <- NULL
FPT_list_diff <- NULL
FPT_mean_diff <- NULL
FPT_min_diff <- NULL
FPT_max_diff <- NULL
x <- NULL
y <- NULL
count3D_list <- NULL
```

```
count2D_list <- NULL
count3D <- 0
count2D <- 0

# create seeds
seeds <- sample(1:100000000, n_simulations, replace=FALSE)
```

Here the actual calculations take place.

```
# running simulations and calculations
for (radi in radius) {
  for (angle in slope) {
    for (n in 1:n_simulations){
      set.seed(seeds[n])

      # create circle with specific radius
      circle <- ellipse(radi,radi, centre = c(0,0))
      circle_sp <- as(circle, "SpatialPolygons")
      circle_sl <- as(circle_sp, "SpatialLines")

      # create ellipse with specific radius and slope
      ellipse <- ellipse(radi,radi/cos(angle*pi/180), centre = c(0,0))
      ellipse_sp <- as(ellipse, "SpatialPolygons")
      ellipse_sl <- as(ellipse_sp, "SpatialLines")

      # simulate random walk
      random_walk <- simm.crw(date_seq, r = 0, burst = "RW r = 0")

      # cast to spatial lines data frame for later use
      random_walk_sldf <- ltraj2sldf(random_walk)

      # find out where the trajectory passes the circle or ellipse
      passing3D <- passingFun(random_walk, circle$bdry[[1]])
      passing2D <- passingFun(random_walk, ellipse$bdry[[1]])

      if (!is.null(passing3D)){
        count3D <- count3D + 1

        # find the first intersection
        intersection3D <- findIntersectFun(random_walk, passing3D,
circle_sl, timelag)

        # calculate first passage time
        FPT3D <- as.numeric(difftime(intersection3D$time, starttime, units
= "hours"))
        FPT_list3D <- append(FPT_list3D, FPT3D)
      }

      if (!is.null(passing2D)){
        count2D <- count2D + 1

        # find the first intersection
        intersection2D <- findIntersectFun(random_walk, passing2D,
ellipse_sl, timelag)

        # calculate first passage time
        FPT2D <- as.numeric(difftime(intersection2D$time, starttime, units
= "hours"))
        FPT_list2D <- append(FPT_list2D, FPT2D)
      }

      if (!is.null(passing2D) && !is.null(passing3D)){
```

```
        # calculate difference between 2D and 3D FPT
        diff_FPT <- FPT3D - FPT2D
        FPT_list_diff <- append(FPT_list_diff, diff_FPT)

        # save if this is the largest difference
        if(max(FPT_list_diff) == diff_FPT){
          largest_diff <- diff_FPT
          largest_traj <- random_walk
          largest_intersect3D <- intersection3D
          largest_intersect2D <- intersection2D
          largest_passing3D <- passing3D
          largest_passing2D <- passing2D
        }

        # save if this is the smallest difference
        if(min(FPT_list_diff) == diff_FPT){
          smallest_diff <- diff_FPT
          smallest_traj <- random_walk
          smallest_intersect3D <- intersection3D
          smallest_intersect2D <- intersection2D
          smallest_passing3D <- passing3D
          smallest_passing2D <- passing2D
        }
      }
    }
     # add values to a lists for plotting
    if(!is.null(FPT_list_diff)){
      FPT_mean_diff <- append(FPT_mean_diff, mean(FPT_list_diff))
      FPT_min_diff <- append(FPT_min_diff, min(FPT_list_diff))
      FPT_max_diff <- append(FPT_max_diff, max(FPT_list_diff))
      x <- append(x, radi)
      y <- append(y, angle)
    }
    # keep track of how many times the trajectory does not cross the FPT
circle
    count2D_list <- append(count2D_list, count2D)
    count3D_list <- append(count3D_list, count3D)

    # set list to NULL again
    FPT_list_diff <- NULL

    # set counts to 0
    count3D <- 0
    count2D <- 0
  }
}
```

Plot!

```
# create dataframe
th_FPT <- data.frame("slope"=y, "radius"=x, "FPTdiff"=FPT_mean_diff)
th_FPT$rel_diff=th_FPT$FPTdiff/th_FPT$radius

# plot
pl <- ggplot(data = th_FPT, aes(x = slope, y = FPTdiff, color = radius)) +
geom_point() + labs(x= 'Slope (degrees)', y="Difference 3D and 2D FPT (h)")
+ scale_x_continuous(breaks = scales::pretty_breaks(n=12), limit=c(0,60)) +
scale_y_continuous(breaks = scales::pretty_breaks(n=7), limit=c(0,14)) +
scale_colour_gradientn(colors=terrain.colors(10), limit=c(0,700))
pl
```