University of Zurich^UZH

Department of Geography

# 3-D Trajectory Simulation in Movement Ecology: Conditional Empirical Random Walk

GEO 620 Master's Thesis

**Author**
Merlin Unterfinger
11-927-894

**Supervisors**
Prof. Dr. Robert Weibel
Dr. Kamran Safi
(Max Planck Institute for Ornithology, 78315 Radolfzell, ksafi@orn.mpg.de)
Georgios Technitis

**Faculty Representative**
Prof. Dr. Robert Weibel

June 30, 2018
Department of Geography, University of Zurich

*“To raise new questions, new possibilities, to regard old problems from a new angle, requires creative imagination and marks real advance in science.”*

Einstein and Infeld (1938, p. 95)

# *Abstract*

A recently observed increase in the performance of tracking devices indicates that capturing positional data in three dimensions is about to become standard in movement ecology. To date, only a few methods exist that include the third dimension in the modeling and evaluation of movement behavior. This issue could be solved using simulations of three-dimensional trajectories. Simulated trajectories serve various purposes, such as the generation of null models for hypothesis testing, the deliver of the basis for resource selection models, or the quantifying of space use intensity. In terms of two-dimensional modeling, an algorithm for the simulation of empirically informed random trajectories between two given points, the eRTG algorithm, already exists. A three-dimensional version would be of great value in movement ecology, especially with regard to the investigation of tracking data of flying or diving organisms. Therefore, this master's thesis is dedicated to the development, implementation, and evaluation of the eRTG in three dimensions, followed by a use case that demonstrates the algorithm's capabilities.

The eRTG3D algorithm is based on a probabilistic movement model that reflects the mover's behavior from its perspective as well as its movement behavior in relation to the target. Therein, empirical distribution functions extracted from observed tracking data ensure that the simulated trajectories maintain a predefined geometry and arrive at the destination while remaining as random as possible. Since it is not trivial to move from two dimensions to three dimensions, a number of physical constraints, which further restrict the freedom of movement in the three-dimensional space, are necessary. An extended version of the eRTG3D algorithm allows the simulation of gliding and soaring trajectories of soaring birds on an uplift suitability map. The concluding demonstrator use case calculates the collision probabilities of soaring white storks during their fall migration with airplanes at Zurich Airport, based on high-resolution gliding and soaring simulations.

The major contribution of this thesis is a successfully verified algorithm that is capable of generating empirically informed random trajectories between two given points in the three-dimensional space. The simulated trajectories maintain the predefined geometry and are ecologically valid. Furthermore, the demonstrator use case at Zurich Airport showed the highest collision probabilities of birds and planes in the northern arrival and the eastern departure corridors.

**Keywords** 3-D; trajectory simulation; random walk; probabilistic model; movement ecology; tracking data; bird flight; gliding and soaring; bird-strike probability

# *Acknowledgements*

While working on this master's thesis, I was able to consistently rely on thematic, technical and personal support from a wide variety of people. In addition to orientation and critical feedback, this also included extremely motivating new ideas on what could further be realized in the context of this thesis. This motivated to do my best and, finally, made this thesis possible. Therefore I would like to sincerely thank:

- Prof. Dr. Robert Weibel, supervisor and faculty representative, and Dr. Kamran Safi, supervisor, for the many meetings, the critical feedback, the constant motivation to go one step further and to solve even more complex problems, and finally, for the fascination for this subject they have passed on to me.

- Georgios Technitis, co-supervisor, for an extensive theoretical introduction during the initial phase of this thesis and for the support when the programming work started.

- Dr. Halldór Janetzko, computational scientist, for the access to the database storing the flight trails and for the methodological input towards the end of the thesis.

- Melanie Zgraggen, English language teacher, for her grammatical proofreading of the thesis.

- My friends and family, especially Tomi Bricchi, Ester Unterfinger and Bruno Werder, who have supported me without reserve over many years.

- Tobias Meier, engineer and former roommate, for valuable and insightful technical discussions.

- And finally, Lina Torregroza, for her patience in listening to the same topic over and over again, for her continuous support and for making sure that I did not completely forget the world around me.

# Contents

# List of Figures

# List of Tables

# List of Listings

# Acronyms

| | | |
|---|---|---|
| **2-D** | Two-dimensional | 2 |
| **3-D** | Three-dimensional | 1 |
| **BBMM** | Brownian Bridge Movement Model | 19 |
| **BRW** | Biased Random Walk | 16 |
| **BCRW** | Biased Correlated Random Walk | 16 |
| **CMA** | Computational Movement Analysis | 11 |
| **CoT** | Cost of Transport | 24 |
| **CERW** | Conditional Empirical Random Walk | 39 |
| **CTRW** | Continuous-time Random Walk | 17 |
| **CRW** | Correlated Random Walk | 16 |
| **CRS** | Coordinate Reference System | 28 |
| **DB** | Database | 31 |
| **DEM** | Digital Elevation Model | 31 |
| **eRTG** | empirically Informed Random Trajectory Generator | 2 |
| **eRTG3D** | empirically Informed Random Trajectory Generator in 3-D | 3 |
| **GIS** | Geographic Information System | 28 |
| **GPS** | Global Positioning System | 1 |
| **HPC** | High Performance Computing | 45 |
| **HR** | Home Range | 10 |
| **HT** | Hyper Threading | 48 |
| **IDE** | Integrated Development Environment | 33 |
| **KDE** | Kernel Density Estimation | 10 |
| **LoS** | Line of Sight | 78 |
| **MKDE** | Movement-based Kernel Density Estimator | 10 |
| **MPO** | Moving Point Object | 11 |
| **PDF** | Probability Density Function | 34 |
| **PPA** | Potential Path Area | 18 |
| **PPS** | Potential Path Space | 18 |

| | | |
|---|---|---|
| **RTG** | Random Trajectory Generator | 19 |
| **RW** | Random Walk | 2 |
| **RRW** | Reinforced Random Walk | 18 |
| **SF** | Simple Feature | 51 |
| **SIRW** | Simple Isotropic Random Walk | 16 |
| **STP** | Space-time Prism | 18 |
| **SQL** | Structured Query Language | 31 |
| **UD** | Utilization Distribution | 3 |
| **UERW** | Unconditional Empirical Random Walk | 36 |
| **UTM** | Universal Transverse Mercator | 76 |

# Chapter 1

# Introduction

In the past decade, technologies to track moving objects have evolved rapidly. The accuracy of measurements and the handling and miniaturization of tracking devices has also made significant progress. Therefore, powerful tracking technologies, are increasingly spreading across various fields of research. Especially the Global Positioning System (GPS) has become the standard technology for the determination of positional data in high spatial and temporal resolution. Additionally, there is a growing trend to record not only the horizontal but also the vertical component of positions. Location data in three dimensions offer great potential to study many phenomena more comprehensively, because space is often not sufficiently represented as a flat surface. Additionally, the accuracy of Three-dimensional (3-D) measurements increases steadily, which further promotes their use (Byrne et al., 2017).

Nevertheless, many situations remain, where the temporal or spatial coverage with movement data in 3-D is insufficient for the study of certain phenomena. This is either due to high costs of tracking moving entities or a lack of technological possibilities. A common factor driving costs is the number of moving entities that need to be tracked. Tracking a flock of birds, for example, can quickly become very expensive or logistically impossible. In addition, high costs can also be caused by a difficult accessibility of the moving entity. Often it is also the tracking technology itself which restricts the data acquisition because of insufficient range or limited battery life. The limited battery capacity of GPS-tracking devices leads to the well-known trade-off between overall duration of data-collection time and temporal resolution (McMahon et al., 2017). Therefore, trajectory simulation is a necessity in many scientific applications, such as enhancing the temporal resolution of tracking data in retrospect by inserting appropriately simulated possible locations between the observed fixes, referred to as *bridging*, or deriving statements from movement data on an individual level about the behavior of a larger group on population level. A further important use of simulations is the completion of missing data in recorded trajectories. A particularity of the GPS is the need for access to sufficient satellite signals, so as to obtain a valid fixpoint. As a consequence of insufficient signals or other disturbing factors within the tracking device, missing fixes in the record can arise. Completing the missing sections with simulated trajectories ensures a continuous analysis of the data and is therefore a big advantage (Wentz, Campbell, and Houston, 2003).

## 1.1 Motivation and Goal

The scientific field, in which GPS tracking data typically forms the fundamental basis for the major part of research is movement ecology (Nathan et al., 2008; Nathan and Giuggioli, 2013). Especially, animal movement records are available for many species from different time periods with varying temporal and spatial resolution. A well-known data source is the *movebank*[1] repository hosted by the Max Planck Institute for Ornithology[2]. Among other animal tracking data sets, the repository also houses many GPS records of birds that contain the vertical position component. Even though large amounts of data are available, they are often insufficient to meet the requirements for analyzing the spatial behavior of birds in 3-D.

In animal ecology, a sufficient temporal resolution of the collected tracking data is often a major issue. Especially when observing migrating birds, which are the application focus of this thesis, handling the trade-off between full coverage and temporal resolution is of crucial importance. Thus, the use of simulations to overcome the limitations retrospectively is clearly understandable. Since the knowledge about the movement behavior of the observed animals is often incomplete, probabilistic movement-simulation models, such as Random Walks (RWs), are most suitable to simulate the needed trajectories (Technitis et al., 2015).

The empirically Informed Random Trajectory Generator (eRTG) is a powerful probabilistic movement-simulation algorithm (Technitis et al., 2016). The algorithm is capable of generating realistic random trajectories in a Two-dimensional (2-D) space between two given fixpoints in the Cartesian coordinate system. The trajectory simulation is based on empirical distribution functions extracted from observed trajectories (training data) and thus reflects the geometrical movement characteristics of a species or subgroup. Since they are as random as possible within the predefined movement behavior, the simulated trajectories can be used as null models for hypothesis testing, as a basis for resource-selection models, to enhance the temporal resolution retroactively, or to quantify space-use intensity. Also, for most populations, only a few individuals are tracked and the eRTG algorithm provides the possibility to make predictions on larger populations (Technitis et al., 2016).

As Van Toor et al. (2018) state, the eRTG algorithm offers great advantages over other movement-simulation models, since it is able to simulate individual trajectories between given points that reflect the geometric properties of template trajectories. They emphasize the usefulness of the algorithm in simulation tasks, in which the distance to be overcome is defined (given starting and endpoint) and the geometric properties of the simulated tracks are decisive. In this case, the eRTG algorithm ideally meets the conditions. Nevertheless, the eRTG clearly has the drawback that the trajectory generation is limited to the 2-D space and thus the algorithm is not suitable for the investigation of research problems in 3-D.

[1] www.movebank.org

[2] www.orn.mpg.de

Although many GPS devices already record 3-D location data with the longitude, latitude and height of tracked animals, the third dimension is widely neglected among many studies in movement ecology. Or, as Tracey et al. (2014, p. 1) address the problem regarding animal space use studies:

> "(...) progress has been limited by the inability of existing modeling techniques to take advantage of the three-dimensional (3-D) data sets, constraining estimates of animal space use to an often biologically unrealistic 2-D "Flatland" (...). Biologists are only beginning to recognize the theoretical and applied value of incorporating the vertical aspect into analyses of animal space use."

This indicates that the problem is not only insufficient access to movement data in 3-D, but that there is also a lack of appropriate scientific models and concepts to adequately consider the third dimension. In many research problems (Bras, Jouma'a, and Guinet, 2017; Cooper, Sherry, and Marra, 2014; Ferter et al., 2015; Weinzierl et al., 2016), not only the 2-D position on the Earth's surface, but also the height information of a location is of large interest. The necessity of the third dimension becomes especially apparent when the research is concerned with flying or diving animals, which are not confined to the Earth's surface. Even on uneven surfaces, the third dimension is hardly negligible. Ignoring the vertical component may considerably affect the understanding of the animals' behavior and lead to misinterpretations of their space use and energy expenditure. Hence, the simulation of empirically informed random trajectories of individuals that are moving freely through air or water between two given endpoints in the 3-D space could be very advantageous for many of these research aims. An important task could be the estimation of the 3-D Utilization Distribution (UD) of a bird species, based on a simulated migrating flock of birds in a certain region and during a given period. Such a 3-D UD could, for instance, allow the prediction of collision probabilities with starting and landing airplanes around airports located in the same area. A further task could be the study of the influence of meteorological factors and surface properties on the gliding and soaring behavior of birds, by the incorporation of uplift columns into the generation of empirically informed random trajectories.

As literature indicates, movement ecology requires a probabilistic movement simulation algorithm capable of generating trajectories in 3-D. This thesis therefore contributes to ongoing research in the field as its extends the functionality of the existing eRTG algorithm in a way that makes the algorithm capable of generating empirically informed random trajectories with a third dimension. It will further provide an illustrative application example of the newly crafted empirically Informed Random Trajectory Generator in 3-D (eRTG3D) algorithm to facilitate and promote the use of the algorithm across the field of movement ecology.

## 1.2 Research Questions

The extension of the eRTG under the conditions previously detailed suggests a tripartite division of this thesis, each part tackling one research question. The research questions are concerned themselves separately with the development, evaluation and application of the eRTG3D algorithm.

First, a simple approach for the incorporation of the third dimension into the existing eRTG algorithm is developed and implemented. In an iterative procedure, the complexity of the implementation is increased step by step. Hence, the first research question is dedicated to the development of the eRTG3D algorithm.

**RQ 1** Development – *How is the third dimension best modeled and implemented in the eRTG algorithm so that empirically informed random trajectories can be generated in 3-D space?*

Given that a functioning solution of the algorithm exists, the latter has to be examined for its reliability and representativeness. First, the probabilistic model is internally verified for correctness. Then, the validation of the output of the eRTG3D algorithm follows. It is assessed, whether the simulated trajectories represent the originally given movement behavior. The verification and validation requires the formulation of the second research question.

**RQ 2** Evaluation – *How representative are the random trajectories generated with the eRTG3D algorithm?*

After a successful evaluation of the algorithm, it is applied to real problems to illustrate some of its potential application areas. Specifically, the eRTG3D algorithm is used to estimate an UD in 3-D of white storks during fall migration in the Swiss Central Plateau. The resulting 3-D UD is then compared to the UD of airplanes, starting from and landing at Zurich Airport during this period. Comparing the two UDs results in a proxy for the bird-strike probabilities in the different arrival and departure corridors. Hence, the third research question focuses on an illustrative example application of the eRTG3D algorithm.

**RQ 3** Application – *What are the collision probabilities of white storks (Ciconia ciconia) and airplanes in the arrival and departure corridors at Zurich Airport (ZRH) during the birds' fall migration?*

Given these research questions, the overall aim of this thesis can be summarized as the development, implementation and evaluation of a probabilistic movement-simulation algorithm in 3-D, followed by a demonstration case study.

## 1.3 Outline

- In Chapter 2, the necessary background in the fields of movement ecology and computational movement analysis are given and the need for simulations in animal ecology is argued for. Subsequently, the relevant scientific research gaps are identified.

- Chapter 3 introduces the study area covered and the data used in this work.

- Chapter 4 explains the methodology for developing and evaluating the final version of the eRTG3D algorithm. In addition, the procedure for calculating bird-strike probabilities at Zurich Airport is explained.

- Chapter 5 presents the results with regard to the representativeness and validity of the new algorithm.

- Chapter 6 discusses the results and the applied methodology, and reviews the research questions.

- Chapter 7 offers conclusions and makes recommendations for future research.

# Chapter 2

# Background

## 2.1 Movement Ecology

The movement of individual organisms, defined as the change in location of an individual over time, is an intrinsic and characterizing property of most organisms encountered on Earth. Movement ecology, the research field that investigates the movements of animals, plants, or microorganisms, usually addresses the following four fundamental questions (Nathan and Giuggioli, 2013):

(I) Why? – Challenges the motivation for the movement, (II) how? – Investigates the nature of the motion, (III) when and where? – Examines spatial and temporal patterns, and (IV) what? – Assesses ecological and evolutionary consequences of the change in location. To assess these questions, various paradigms have evolved.

The *biomechanical paradigm* focuses on the physical properties of the motion itself; energy expenditure, mechanics, and physiology, all in relation to the motion of the individual, are the major point of interest (Vogel, 2003).

The *cognitive paradigm* sets focus on the movement-related decisions that motile individuals take. It is of primary interest, to understand the navigation mechanisms, which can be interpreted as rules influencing the spatial decisions (Poucet, Lenck-Santini, and Save, 2003).

The *optimality paradigm* investigates the relative effectiveness of different movement strategies to reach the same goal (e.g. energy gain or reproduction). The paradigm concentrates on ecological or evolutionary timescales to reason the dominance of a strategy in a given situation (Fretwell and Lucas, 1970).

The *random paradigm*, to which this thesis can mainly be assigned, investigates movement paths using probabilistic models that are grounded on theories such as Random Walks (RWs), diffusion or anomalous diffusion. The big advantage of this approach is its ability to overcome the often-encountered fact of missing cognitive and mechanistic information about the exact movement behavior of an organism, as it assumes that the components of movement are following certain probability distributions. Therefore, large-scale analyses without the necessity of detailed previous knowledge become possible (Technitis et al., 2015).

While the first two paradigms are strongly focused on short-term motion of individual organisms, the latter two are more concerned with long-term spatio-temporal patterns in the movement. To unite the different paradigms, Nathan et al. (2008) presented the following framework:

$$u_{t+1} = F(\Omega, \Phi, r_t, w_t, u_t) \tag{2.1}$$

In this framework, the potential new position $u_{t+1}$ of an organism is modeled as a function $F$ of the current place $u_t$, internal state $w_t$, motion capacity $\Omega$, navigation capacity $\Phi$, and environmental factors $r_t$. The crucial contribution of the framework of Nathan et al. (2008) is that it splits movement into parts, to which quantitative values can be assigned. This is a mandatory requirement for fitting probabilistic or deterministic models to movement behavior.

### 2.1.1 Perspectives on Movement

In general, there are two fundamentally different approaches of perceiving movement in space. On the one hand, the observer moves with the mover and repeatedly records information about the position in a predetermined time interval. This means that a time-stamped sequence of locations for the moving entity is created. This *Lagrangian* view of movement is widespread in movement ecology. GPS tracking, in which the tracking device is constantly at the location of the moving subject, is a typical example of the Lagrangian principle. The Lagrangian view is an internal, individual-based perspective (Both et al., 2013).

The *Eulerian* approach, on the other hand, is based on observation points that are permanently distributed in space and observe movement in its context. Therefore, it is an external, place-based perspective on movement. Entities that pass an observation are captured by recording the time of passing and the identity of the passing entity (Laube, 2014).

Recently, a shift from the Eulerian to the Lagrangian approach in movement ecology has been observed. This is mainly due to substantial technological progress in tracking devices that support the Lagrangian perspective. Nevertheless, the Eulerian approach still dominates the recording of movement of microorganisms, seeds or flying insects, since their movement is usually an externally powered transport and they are too small for carrying heavy tracking devices. On the other hand, the application of the Lagrangian approach on larger organisms allows the quantification of movement with a high spatio-temporal resolution and on sufficiently large scales.

From the different perspectives on movement, movement ecology has also established two different traditions. In studies, which are based on the Lagrangian view, movement is usually expressed with incremental movement parameters that describe properties of the motion between consecutive fixpoints (e.g. the distance and direction of movement), while in studies that are grounded on the Eulerian view, movement is generally modeled as a diffusion process (Smouse et al., 2010).

### 2.1.2 Spatio-temporal Scale of Movement

In studying movement behavior it is crucial to define certain terms, which refer to the different scales of organism movement. Segments of a movement trace of an individual that are linked to the same set of goals, are summarized as a *movement phase* (Nathan et al., 2008). Migration and foraging are the most prominent movement phases. A movement phase, in turn, can be composed of various types of motion, such as walking or swimming. These motion types are termed *canonical movement modes*. The sequence of all movement phases (and modes) together defines the *lifetime track* of an individual.

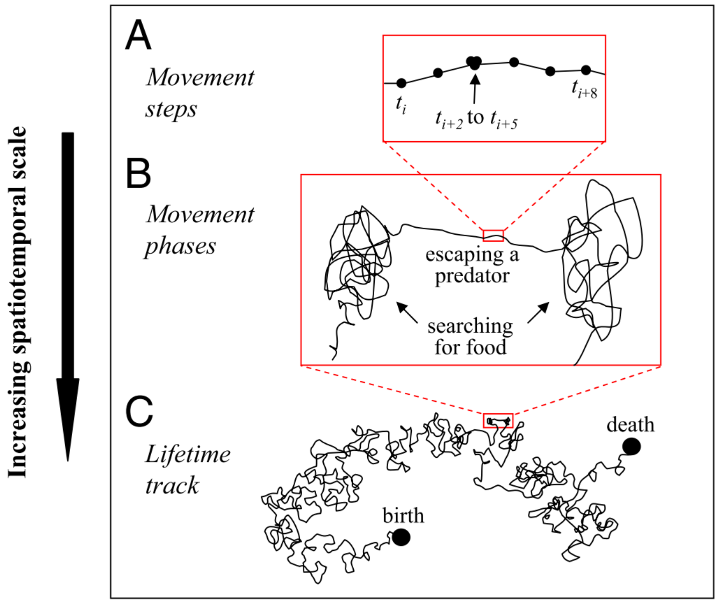


FIGURE 2.1: Spatio-temporal scale of movement. *Source: Nathan et al. (2008, p. 2)*

### 2.1.3 Utilization Distributions in Space and Time

A crucial concept in animal ecology are Utilization Distributions (UDs), which are used to examine animal movement behavior at an aggregated scale are in space and time. Van Winkle (1975) defines an UD as "*(...) the two-dimensional relative frequency distribution for the points of location of an animal over a period of time.*" According to this definition, a UD is the animal's relative frequency of occurrence in the 2-D space. Van Winkle's UDs are grounded on the concept of the space-time cube, in which movement is represented as a path in a 3-D space with $(x, y, t)$-dimensions. It is obvious that a definition that limits space use to a 2-D plane is not sufficient in virtually all cases. Even non-flying and non-diving animals use space in different layers of height. An ape, for example, would stay at the exact same position in the $(x, y)$-plane while climbing up a tree and therefore change the type of its nearby environment only by a shift in the $z$-dimension of the $(x, y, z)$-space. Because of this issue

Keating and Cherry (2009) extend the UD to all four dimensions of space and time $(x, y, z, t)$.

To quantify a UD over a period of time for an animal, the relative frequency of the visiting time for every location in the area of interest can be calculated, based on movement data. A simple method is to split space into regular units (e.g. cells, hexagons or voxels) and to sum up the time an animal stays in each unit. The probability of occurrence in a specific spatial unit is achieved by dividing the summed up time by the total time spent in the region.

The downside of this method is that the shape and scale of the splitting unit strongly influence the resulting UD. Also, the anchor point placement of the regular units in space affects the result. To at least partially avoid this issue, various techniques for the estimation of UDs have been developed. Most of the UD estimators in 2-D are grounded on the concept of Kernel Density Estimation (KDE), because it bypasses the distortions induced by the splitting into regular spatial units. The underlying equation is described by Gatrell et al. (1996):

$$\hat{\lambda}_t(s) = \sum_{i=1}^{n} \frac{1}{\tau^2} \cdot k \cdot \left( \frac{s - s_i}{\tau} \right) \tag{2.2}$$

In this equation, $k$ is the kernel weighting function and $\tau$ is the bandwidth, which is often also denoted as $h$. The KDE smooths the influence of each data point $s_i$ on the surrounding region, allowing an intensity estimate $\hat{\lambda}_t(s)$, even at a location $s$, where no data is observed. The result is a smooth UD in 2-D, which is easy to interpret. Tracey et al. (2014) go further and extend the the concept to a Movement-based Kernel Density Estimator (MKDE) that allows the estimation of UDs in 3-D. Nevertheless, some distortions still remain and some new uncertainties are introduced by setting the parameters $\tau$ and $k$ and by the choice of the underlying grid resolution to visualize the result.

Another concept that is intrinsically linked to UDs are Home Ranges (HRs) of animals. Burt (1943) denotes the HR as the space an animal traverses on a regular basis in order to pursue the normal activities of food gathering, mating and raising its young. Occasional movement outside the usual area, which might serve exploratory purposes, is not considered to be part of the HR. Although HRs are defined differently across the field of movement ecology, there was a shift to a more cognitive and less data-driven definition in recent years (Powell and Mitchell, 2012). The animal is familiar with resources, escape routes and potential mates in its HR and, therefore, HRs are closely related to cognitive maps the animal gathers about its environment. Studies have shown that the spatial knowledge provided by an HR affects the individual fitness of an animal (Blanco and Cortés, 2007).

UDs and HRs are two different concepts to represent a spatio-temporal pattern in animal movement behavior. But the motivation behind the concepts is clearly different: An animal of a migrating species may have two HRs, separated by a migration route. The route itself is not considered to be part of the HRs, while a UD could be defined exactly for the time period of migration, for the HRs or for the whole lifetime track of the animal. UDs specify only a probability of occurrence for a given time period, whereas HRs are linked to ecological properties of the environment.

Nevertheless, HRs often are derived from a certain probability contour of UDs, which represents the proportion of time the animal spends inside the contour. Since this approach has reached its limits, more sophisticated methods have been developed. They segment movement data into different time periods and define HRs over areas that provide vital resources for the different periods (Demšar et al., 2015).

## 2.2 Modeling Movement

In this section, necessary terms and concepts for the modeling of movement and the space containing the movement are introduced. These terms and concepts have their origins in Computational Movement Analysis (CMA), the field devoted to analyzing movement processes with methods from a range of fields including GIScience (Laube, 2014).

### 2.2.1 From Traces to Trajectories

An organism that is moving through space describes a movement trace that can be captured based on one of the two perspectives described in Section 2.1.1. The modeling is accomplished by representing the trace as a set of time-stamped locations in a 2-D $(x, y)$ or 3-D $(x, y, z)$ coordinate system (Laube, 2014). In general, no information about the mover's shape and extent is collected, the mover is therefore commonly represented as a Moving Point Object (MPO). The MPO ignores the actual dimensions of the the mover and the movement is reduced to a sequence of time-stamped observation points. It should be noted that the term movement also includes geometric changes of the object itself. A change in geometry, for example, is by no means negligible when analyzing the movement of tropical cyclones (Dodge, Weibel, and Lautenschütz, 2008).

The sequence of time-stamped points does not represent the full trace the mover covers, since movement between the fixpoints remains unknown. The path between two position fixpoints is usually approximated by a straight line. Such a sequence of time-stamped locations representing a trace is called a trajectory, where $(x, y, z)_{T_1}, \ldots, (x, y, z)_{T_n}$, are measurements at consecutive time steps $T_n$ (Gudmundsson, Laube, and Wolle, 2012).

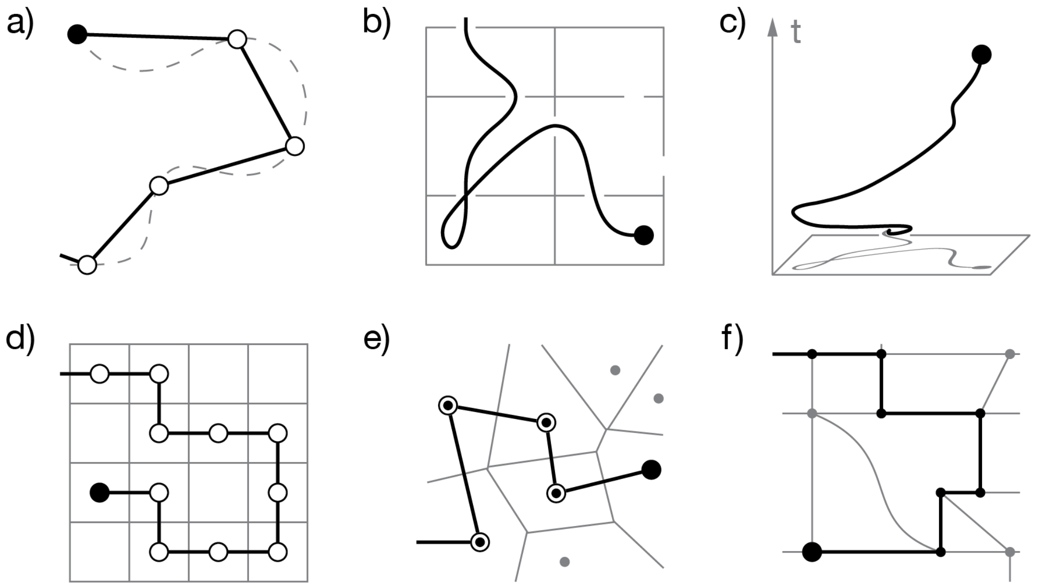
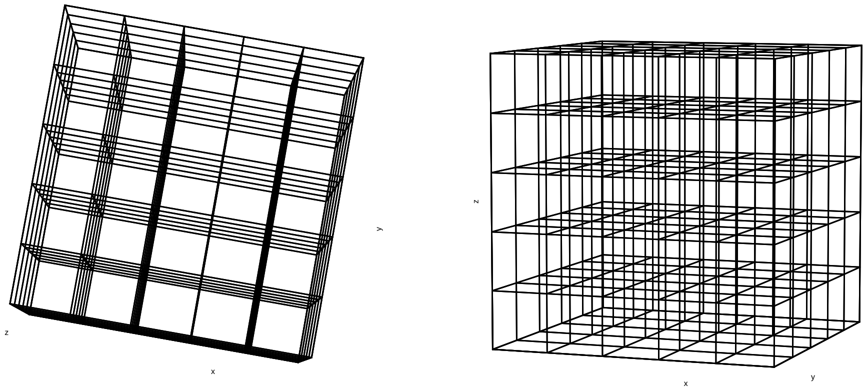


FIGURE 2.2: Movement spaces in 2-D: **a)** Euclidean homogeneous space, **b)** Constrained Euclidean space, **c)** Space-time aquarium, **d)** Heterogeneous field space, **e)** Irregular tessellation, **f)** Network space. *Based on: Laube (2014, p. 12)*

When modeling movement, not only traces are abstracted into trajectories, also the space in which movement takes place is modeled. Laube (2009) therefore describes movement spaces, which can be considered for the modeling of the 2-D space (Figure 2.2). The simplest case described is the *Euclidean homogeneous space*, in which a mover is able to move freely in all directions of space. The *constrained Euclidean space* limits the movement with insurmountable obstacles situated in the space. The *space-time aquarium* concept (also *space-time cube*) attempts to depict the temporal dimension as well. In each of these three concepts, a continuous movement of the object in space is possible. In contrast, the *heterogeneous field space* limits movement to individual cells. Location changes inside a field are discarded and movement is therefore no longer continuous, but takes place in discrete steps with abrupt cell changes. The *irregular tessellation* concept no longer divides space into uniform fields, but represents it using irregular polygons. In the *network space*, movement is captured on a predefined network of nodes and edges. Such a network can be given by, for example, a road network in the movement of cars, but also by the migration of animals between different habitats along known routes. The different movement spaces suggest certain ways of looking at movement. It seems more natural to associate a Euclidean homogeneous space with a Lagrangian view and to look at a network space from a Eulerian perspective. However, movement in a network space can also be recorded using the Lagrangian principle and be snapped onto the network afterwards.

Although these movement spaces are described for 2-D movement, they are easily extensible to the 3-D space. In particular, the constrained Euclidean space and heterogeneous field space in 3-D are important for this thesis. The 3-D constrained Euclidean space can be thought of as a continuous $(x, y, z)$-coordinate system with 3-D objects (e.g. the Earth's surface) in it that limit the freedom in movement. The heterogeneous field space becomes a *heterogeneous voxel space* in which the space is divided into regular cubes (voxels) between which the movement takes place (Figure 2.3).



(A) Top view

(B) Frontal view

FIGURE 2.3: Heterogeneous voxel space.

### 2.2.2 Quantitative Assessment of Trajectories

A trajectory in the form of a time-stamped fixpoint sequence, which is connected via straight paths, does not permit extensive statements about the movement behavior of an organism. Only purely spatial statements about preferred areas and possible routes connecting them can be derived. Often, however, statements about the mode of movement or the movement phases encountered are desired. From the exact position of a fixpoint and the time of the measurement, it is possible to calculate further properties, which quantitatively describe the relationship between two successive fixpoints. These *movement parameters* allow to solve more complex tasks, as for example the division of a trajectory into sections of different movement modes (Dodge, Weibel, and Lautenschütz, 2008). The derived movement parameters are calculated from the primitive parameters $(x, y, z, t)$. For example, the first derivative of the positions $(x, y, z)$ yields the parameters *direction* and *distance*. The *duration* between two fixpoints is obtained solely from the time $t$. Combining the first derivatives of the positions and the time results in the parameter *velocity*. The velocity is a vector, which includes the *direction* of movement and the current *speed* of the movement. The derivative of speed yields a secondary parameter called the *acceleration* (Giannotti and Pedreschi, 2008). Table 2.1 shows an overview of the movement parameters adapted for the 3-D case (Dodge, Weibel, and Lautenschütz, 2008).

| **Dimension** | **Primitive** | **Primary derivatives** | **Secondary derivatives** |
|---|---|---|---|
| **Spatial** | Position<br>$(x,y,z)$ | Distance<br>$f(x,y,z)$ | Spatial distribution<br>$f(distance)$ |
| | | Direction<br>$f(x,y,z)$ | Change of direction<br>$f(direction)$ |
| | | Spatial extent<br>$f(x,y,z)$ | Sinuosity<br>$f(distance)$ |
| **Temporal** | Instance<br>$(t)$ | Duration<br>$f(t)$ | Temporal distribution |
| | Interval<br>$(t)$ | Travel time<br>$f(t)$ | Change of duration<br>$f(duration)$ |
| **Spatio-temporal**<br>$(x,y,z,t)$ | – | Speed<br>$f(x,y,z,t)$ | Acceleration<br>$f(speed)$ |
| | | Velocity<br>$f(x,y,z,t)$ | Approaching rate |

TABLE 2.1: Movement parameters adjusted to 3-D, *Based on: Dodge, Weibel, and Lautenschütz (2008, p. 4)*

The primary derivative *direction* in 3-D consists of two components: the azimuth angle $\varphi$ and the polar angle $\theta$. As shown in Figure 2.4, $\varphi$ is defined as the angle between the $x$ axis and the orthogonal projection of the line segment covered on the $(x,y)$-plane. The distance of the point to the origin is called radius $r$. This combination of azimuth, polar, and radial coordinates $(\varphi,\theta,r)$ is the convention used in physics for spherical coordinates. In mathematics, $\varphi$, and $\theta$ are reversed in the definition. Given a triple of spherical coordinates $(\varphi,\theta,r)$, an azimuth reference direction and a reference plane, a point is uniquely determined in the Euclidean space (Arfken, Weber, and Harris, 2013; Weisstein, 2016).

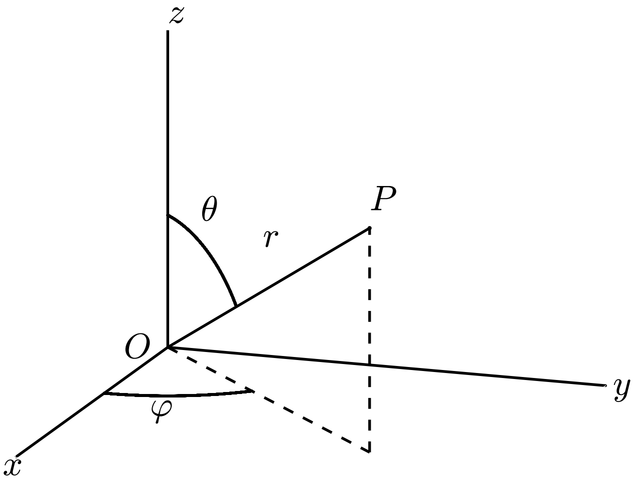


FIGURE 2.4: Spherical coordinate system according to the definition used in physics.

The conversion of points in the spherical coordinate system into a Cartesian coordinate system is described by the following equations:

$$x = r \cdot \sin(\theta) \cdot \cos(\varphi) \tag{2.3}$$

$$y = r \cdot \sin(\theta) \cdot \sin(\varphi) \tag{2.4}$$

$$z = r \cdot \cos(\theta) \tag{2.5}$$

Two consecutive fixpoints with the connecting line segment between them form an individual *step* of the trajectory. Therefore, the distance covered in one step is also called *step length $d$*. For every step, the change in azimuth (*turning angle $t$*) and the change in polar angle (*lift angle $l$*) can be calculated, which are secondary derivatives of the position. The movement from the mover's perspective is given by these three values $(t, l, d)$. A common practice to model the movement behavior of a moving organism in a stochastic manner is the extraction of distributions from turning angles $t_i$, lift angles $l_i$ and step lengths $d_i$, which describe the movement behavior (Smouse et al., 2010).

### 2.2.3 Movement Models

#### Random Walks

A widespread probabilistic movement model as introduced by Patlak (1953) are Random Walks (RWs). An RW is a stochastic process that describes a trajectory consisting of multiple random steps. In the context of ecology, RWs are frequently used to model animal movement, because they are very efficient if not much or no a priori knowledge is available about the animal's movement behavior. In contrast to deterministic models, which include many parameters, RW models involve strong simplifications of natural animal movement behavior (Bartumeus et al., 2005).

The foundation of RWs can be traced back to the Brownian motion of small particles. A very simple illustration of an RW is given by observing the irregular motion of small particles. In dependence on the absolute temperature $T$, particles move through space with velocity $v = \sqrt{kT/m}$, where $k$ is the Boltzmann constant and $m$ the mass of the particle. Since there are other particles present, there is a high probability that collisions with other particles occur. A single particle is forced to move in space and thus performs an RW. Every collision leads to a change in the direction of the movement and therefore defines a new step. In the example of many colliding particles, the starting point of all particles is at the same location. Due to the colliding particles' tendency to occupy the entire space, the probability of collisions decreases with the progressing dilution of the particles. This results in increasing step lengths over time. The expected step length at each time step is therefore dependent on time and the diffusion constant. The process of particles diluting from a given starting point is termed *diffusion* and can be approximated by RW models (Berg, 1993; Brown, 1828).

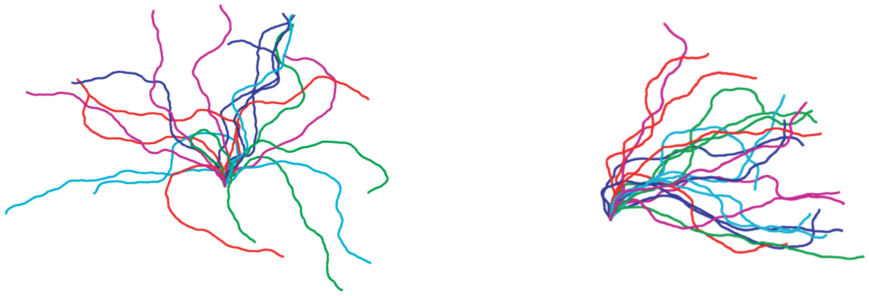
**Simple Isotropic Random Walk** The Simple Isotropic Random Walk (SIRW) used in early models of movement is uncorrelated and unbiased. Uncorrelated means that the direction and distance covered in a step are independent of the movement in the previous step. The new location is only dependent on the location of the previous step, but not on the previous movement properties itself (i.e. the process is Markovian regarding the location). Unbiased refers to the fact that the particles are not moving in a preferred global direction (Codling, Plank, and Benhamou, 2008). SIRWs therefore have a uniform distribution of the turning angle at each step, because every direction between 0 and $2\pi$ is equally likely to occur. The step length is usually modeled with a Gaussian or another exponentially decaying distribution (Bartumeus et al., 2005).

**Correlated Random Walk** In Correlated Random Walks (CRWs), each step chooses a direction which is correlated with the direction of the previous step. This means the direction of a step is dependent on the previous movement and propagates itself into the movement of the following steps, although its influence becomes smaller the further the trajectory moves from the initial step. The distribution of the step length is still Gaussian or exponentially decaying, but the angular distribution of the turning angles is now nonuniform (Figure 2.5 A).

In Biology, CRWs have frequently been used to describe the movement of animals, because animals have the general tendency to move forward and thus zig zag movement is less likely to occur. The tendency to continue in the same direction is known as persistence. It is important to state that the CRW is globally unbiased, since through the correlation of the motion, no overall preferred direction occurs in the generated walk. However, a so-called localized bias exists, which describes the tendency of the individual to persist in the present direction of movement. Therefore, the location at every step of a correlated random walk is no longer a Markov process (Kareiva and Shigesada, 1983).

**Biased Random Walk** If we take the above-mentioned example of the Brownian motion and assume a force acting on the particles (e.g. gravity), the collisions are still random, but there is a global tendency in the direction of movement, which is called drift. In this case, the RW is uncorrelated, but directionally biased, and therefore called Biased Random Walk (BRW). It is important to distinguish between the global directional bias (drift) in the BRW and the localized directional bias in the CRW.

**Biased Correlated Random Walk** In Biased Correlated Random Walks (BCRWs), the globally preferred direction (drift) and the direction of the last steps influence the movement of the next step (Figure 2.5 B). The drift is caused by an external force, which acts with the same intensity on all individuals in dependence of their location in space. The correlation is due to a biased choice of direction of an individual at each step (Codling, Bearon, and Thorn, 2010).



(A) Correlated Random Walk (CRW)

(B) Biased Correlated Random Walk (BCRW)

FIGURE 2.5: Random Walk (RW) examples. *Source: Codling, Plank, and Benhamou (2008, p. 817)*

**Continuous-time Random Walk and Lévy Flight** Continuous-time Random Walks (CTRWs) are used to model anomalous diffusion of particles that spread slower than the Brownian diffusion. The deceleration of the process is achieved by having the particles wait at every step for a random time interval before allowing them to perform the next step. In the case of particles that spread faster than in normal diffusion, the step length is modeled using the heavy-tailed Lévy distribution (power-law distribution) instead of a Gaussian or another exponentially decaying distribution. By doing so, large step lengths between steps in the movement of a particle are encouraged. Because of these huge jumps thus generated, this random walk model is called Lévy flight (Srokowski, 2008; Zaburdaev, Denisov, and Klafter, 2015).

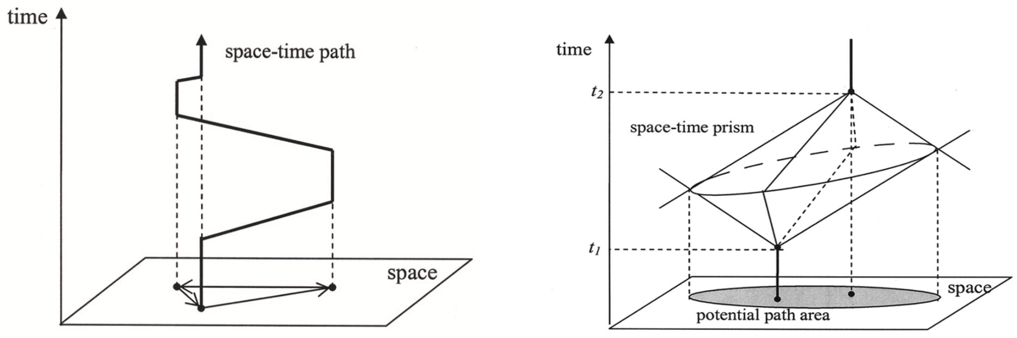
**Lévy Walk** Since particles (or animals) in the Lévy flight model can cover distances that are physically not possible in a single step, the maximum velocity of the movement has to be limited. By confining the heavy-tailed probability distribution to ballistic cones representing the maximum possible velocity, the Lévy flight model is constrained and now termed *Lévy walk*. The directions of the steps are still isotropic and random (uniform distribution of the turning angles), but the overall pattern of the random walk has changed significantly. Lévy walks consist of walk clusters connected by long and straight journeys between them. This behavior is observed in many different species, especially during foraging. It is suggested that Lévy walks are a more effective search strategy than Brownian searching, because they reduce oversampling (Bartumeus et al., 2005; Smouse et al., 2010; Zaburdaev, Denisov, and Klafter, 2015).

**Reinforced Random Walk** RWs in which the history of movement affects the subsequent movement of a mover are termed *reinforced* RWs. These walks prefer previously visited locations and the movement therefore becomes memory-based. The incorporation of a spatial memory in the Reinforced Random Walk (RRW) provides a crucial extension to the RW models (Foster, Grassberger, and Paczuski, 2009).

In movement ecology, it is very likely that many organisms are able to capture and store information about the landscape in which they live. Furthermore, it is assumed that the acquired spatial information influences the movement behavior of organisms. Memory-based movement is observable in organisms of varying degrees of complexity. Ants, for example, prefer paths on which other ants have secreted pheromones, inducing a chemotactic force. This attraction to previously visited places leads to the formation of a complex trail network in which hundreds of thousands of ants participate. Since the pheromones vanish over time, the shortest and therefore more frequented trails are preferred, while ineffective trails disperse. Similar memory-based effects on movement behavior have been observed in slime-producing myxobacteria, as the bacteria tend to follow their own or others' slime traces (Fontelos and Friedman, 2015; Stevens and Othmer, 1997).

### Space-time Prism

In contrast to the probabilistic RW models the Space-time Prism (STP) concept implements a deterministic rule based on the theory of time geography, as described by Hägerstrand (1970). The STP defines all the points that are possible to be reached by an individual in the three-dimensional space-time cube $(x, y, t)$, given a maximum speed, a starting and an endpoint (Figure 2.6). These points are denoted as Potential Path Space (PPS) in the space-time cube. The projection of the PPS on the $(x, y)$-plane defines the Potential Path Area (PPA), which includes all the locations where a visit is possible (Miller, 1991).



(A) Space–time path

(B) Space–time prism and potential path area

FIGURE 2.6: Space-time Prism (STP) concept. *Source: Yu (2006, p. 5)*

A visual representation of the STP for 3-D trajectories is more difficult to imagine, due to the fourth dimension of time. Nevertheless, a 3-D potential path space of possibly reachable points, given the maximum speed and time budget, is also applicable to the 3-D case.

### Brownian Bridge Movement Model

While RW models generate trajectories that start from a single point without a fixed endpoint, the Brownian Bridge Movement Model (BBMM) estimates the probability of visiting each location between the fixpoints of observed movement data. By fitting the BBMM to the observed data, a probability surface for the animal's occurrence is generated, covering the observation extent (Bullard, 1991; Horne et al., 2007). For this, the BBMM uses Brownian bridges, which are stochastic processes that assume Brownian motion of an individual between a starting and endpoint of successive location pairs (Chow, 2009). The uncertainty of the location of an individual is highest at the midpoint in time between the two points and zero at the exact times of the starting and endpoint. The probability density of a Brownian bridge is defined as the relative time spent in an area during the time interval between the two points. The BBMM combines all the successive Brownian bridges to one probability surface of occurrence for the overall time of observation. Because the probability density of the Brownian bridge is defined through a relative time spent in an area, the individual Brownian bridges are weighted with the proportion of the time interval regarding the total time. Potential application areas of the BBMM are estimating UDs and migration routes of animals or evaluating the influence of fine-scale resource selection on animal movement patterns (Calenge, 2006; Kranstauber et al., 2012).

### Random Trajectory Generator

The Random Trajectory Generator (RTG), proposed by Technitis et al. (2015), is an algorithm that combines the concepts of RWs, STPs and the BBMM to efficiently generate random trajectories in 2-D between given starting and endpoints, while minimizing the directional bias. The algorithm considers the maximum speed and the maximum available time for the completion of a movement as physical limitations. Like the BBMM, the RTG can be used to estimate possible positions between two observed fixes, but instead of yielding a probability surface of occurrence (UD), individual random trajectories connecting the starting with the endpoint are generated. The RTG overcomes many limitations of previous movement models, as it provides the opportunity to generate random walks that are conditioned to the fixes of observed data and, at the same time, pays respect to the time budget available and the maximum possible speed (Long, 2016). The RTG can enrich coarse movement data with random walks to obtain a higher resolution or to predict possible paths between two manually defined points when a time budget and maximum speed are set.

### Empirically Informed Random Trajectory Generator

The eRTG is a substantial extension to the RTG algorithm (Technitis et al., 2016). The eRTG connects two endpoints with random trajectories in 2-D, which are empirically informed and thus represent the movement characteristics of the template trajectory. In contrast to the RTG, in which all space-time reachable points (defined by the concept of STPs) have the same probability of being chosen, the eRTG integrates empirical movement parameters into the random trajectory generation. Although all the paths generated by the RTG are possible, the paths generated by the eRTG are ecologically more likely to occur in a given species or subgroup.

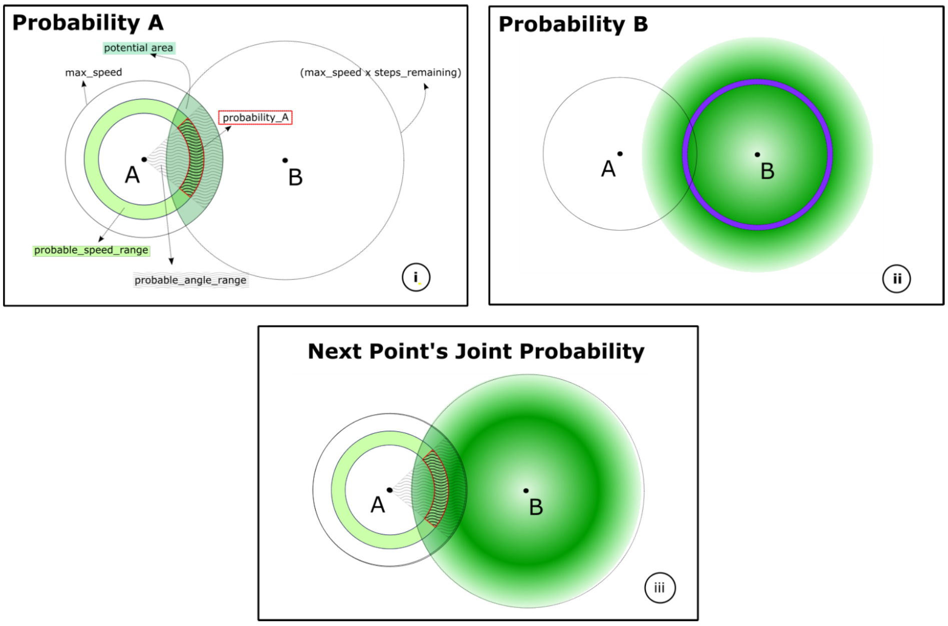


FIGURE 2.7: Concept of the eRTG: (i) shows the effect of the origin (ii) the effect of the destination and (iii) the combined effect. *Source: Technitis et al. (2016, p. 2)*

The eRTG generates trajectories in a stepwise procedure between two successive points (A and B) with discrete time-steps (Figure 2.7). The reachable area for the next step is defined by an intersection of the maximum speed circles from the two points. The radius of each maximum speed circle is given by the available travel time to reach the other point. From the starting point A a probability surface of possible next locations is created, based on the empirically observed dependency of turning angles and step lengths. Conversely, the probability surface of point B is a gravitational force, which is forcing the movement towards point B. Combining the probability surfaces of point A and B, and limiting it to the potential point area given by the maximum speed, the next point's joint probability surface is obtained. Therefore, the mover performs a CRW, which is constrained to the fixed endpoint B. This procedure is repeated for all $n$ time steps that are given by the ratio of the duration of walking and the user-defined time interval (Technitis et al., 2016).

### 2.2.4 Movement Simulation in Animal Ecology

In animal ecology, the causes and consequences of animal movement are of fundamental interest (Crist et al., 1992). The selection of resources, for example, is given by the physical capabilities and limitations in the movement of an animal (Forester, Im, and Rathouz, 2009). Since humanity harnesses almost the entire surface of the Earth for its own benefit, it is important to assess the potential impact of ongoing land use changes on wildlife. Studying animal movement is the only way of identifying and protecting areas that are of vital importance to animals (Cooke, 2008). In addition, possible conflict regions of humans and animals can be recognized, such as hunting grounds that overlap pastures, or bird migration corridors, which can prove as potentially dangerous for airplanes. For many animal species, movement data is not available in a sufficient coverage, which is why methods are needed to derive more information from the scarce data available for larger regions and animal populations. The simulation of movement data, based on a movement model is a promising approach to overcome the lack of data.

Simulations of trajectories, which are generated by estimating possible positions between given fixpoints of an observed trajectory, can also be used to fill in missing data or to up-sample low-resolution data. In addition, simulated trajectories, which were generated using certain assumptions, serve as null hypotheses in order to test observed trajectories against the simulated trajectories (Technitis et al., 2015).

The simulation of trajectories is always grounded on a movement model which makes assumptions about the movement behavior of an animal. These assumptions may either be deterministic rules or probabilistic distributions extracted from existing data. In the eRTG, probabilistic distributions propose points in space that are most likely to be visited, while an STP implements a deterministic rule that limits the reachable points, based on a maximum possible speed and a time budget available. More complex deterministic rules require a profound knowledge of the behavior of animals. However, sufficient previous knowledge is often not accessible, either due to its complexity or because of the lack of appropriate measuring devices. Also, exploratory and opportunistic movement is difficult to model with deterministic rules (but see Song and Miller (2014)). Therefore, in most cases, probabilistic movement models, based on the random paradigm (Section 2.1, p. 7), are used to simulate trajectories.

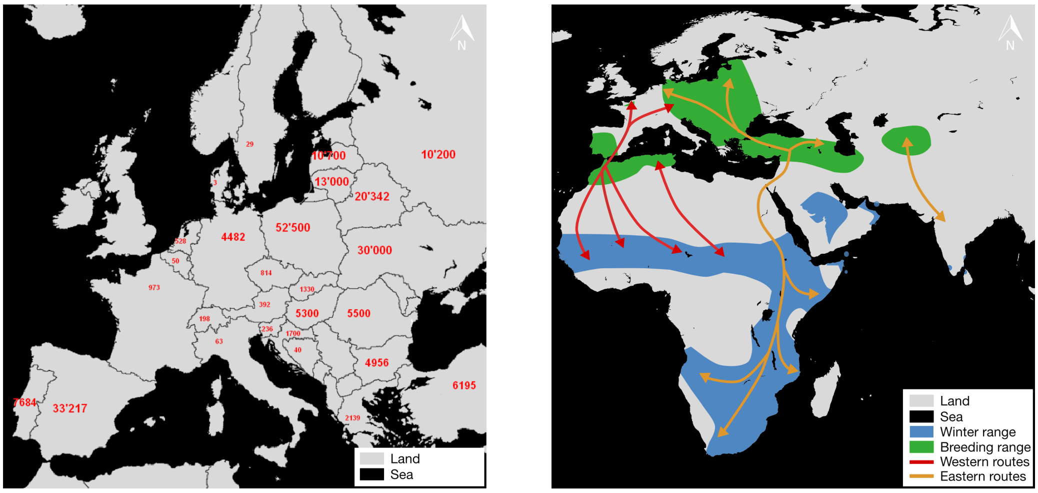
Van Toor et al. (2018) use the eRTG to simulate trajectories of bar-headed geese *Anser indicus* between range fragments. The ecological likelihood of the simulated trajectories is estimated in retrospect, based on properties of the crossed environment. Based on this likelihood, the connectivity in wintering areas and breeding areas is derived. This example illustrates how predictions on animal movement behavior can be made using simulations, even outside the spatial and temporal range of the available data.

## 2.3 White Stork Ecology

The white stork *Ciconia ciconia* is a bird species from the stork family *Ciconiidae*. White storks weigh 2.5 to 4.5 *kg* and are about 80 to 100 *cm* long. In adult animals, the wingspan typically ranges from 200 to 220 *cm*. Except for the black feathers on the wings, the plumage is pure white. The beak and the legs are colored reddish. White storks feed on small animals such as earthworms, insects, frogs, mice, rats, fish and on carrion. Due to the many sources of food that they are able to use, white storks are food opportunists, feeding on food that is abundant at a particular time. In search of prey, meadows and marshes are crossed and prey is quickly caught with the beak. Also, in shallow waters, water and bottom are searched for prey. Due to their hunting characteristics, white storks mainly inhabit open and semi-open landscapes. Wet and water-rich areas, such as floodplains and grasslands, are preferred.

The birds are active during the day and usually sleep at night. White storks can reach ages up to and above 35 years and sexually mature in their fourth year. They nest on objects that clearly exceed their surroundings in height, such as rocky outcrops, trees or buildings. Often, loose colonies with up to 30 couples are formed. White stork couples are faithful to their nest and, if possible, return to it every year. The nesting site is changed if the raising of offspring was not successful or if new pairs are formed (Elliott, 1992).

### 2.3.1 Population and Migration



(A) Population size per country

(B) Migration routes, breeding and wintering grounds

FIGURE 2.8: Estimated population sizes in 2004 and 2005 and migration routes of white storks. *Modified figures, based on: wikipedia.org*

In 2004 and 2005 the worldwide population size of white storks was estimated at approximately 700 000 individuals. Figure 2.8 (A) shows the estimated population size per country. The largest populations live in eastern Europe and in Spain during the breeding season; together they consist of 450 000 to 500 000 mature individuals. The overall trend is a growth in population, although there are some populations which are in decline. The decline is mainly due to land use change, which includes the drainage of swamps and the disappearance of grasslands (Waterbird Population Estimates, 2018).

White storks are Palearctic migrants (Walther, 2004) which use the assistance of uplift to cover long migration routes between their wintering and breeding grounds (Flack et al., 2016). Due to their reliance on updrafts, it is not possible for white storks to travel long distances over open water, which is why they prefer to travel over land. In order to reach their wintering grounds on the African continent, the European populations use the straits of Gibraltar or the Bosporus to cross the Sea. The European birds breed from February to April in the Palearctic and migrate southwards in the fall, when they form large flocks with hundreds to thousands of individuals (Flack et al., 2018). The main departure from the European breeding grounds takes place in August and lasts until the end of September. The major part arrives in the African wintering grounds early in October (Berthold et al., 2002; Berthold, Kaatz, and Querner, 2004).

Figure 2.8 (B) shows common migration routes, which can be separated into an eastern and a western corridor (Berthold et al., 2001). The choice of the corridor depends on the breeding location of the white stork population. Birds breeding in eastern Europe usually travel south using the eastern corridor. They pass the strait of the Bosporus, fly over Turkey, Syria, Israel and Egypt, from where they continue south following the Nile. Some birds even reach the southern tip of the African continent. The western corridor is used by birds that breed in central and western Europe. They travel via Germany, Switzerland, France and Spain, after which they cross the Mediterranean at the strait of Gibraltar. Once on the African continent, the Sahara desert poses the last obstacle to overcome before reaching the wintering grounds. The white storks that use the western corridor usually stay north of the equator, in the Sahel, since the African rain forest blocks their further journey to the south. During the migration phase, white storks depend on suitable stopover sites, where they can feed, drink and rest.

The major part of the population, around 500 000 individuals, uses the eastern corridor. Although the eastern corridor is twice as long as the western corridor, the storks need the same time to reach their wintering grounds. In spring, the white storks start their migration north and arrive in Europe in the end of March and in April (International BirdLife, 2016; Sanderson et al., 2006).

### 2.3.2 Flight Behavior

The predominant locomotion type of birds is flying. Bird flight can further be divided into different canonical movement modes. Flapping flight (powered flight) is the movement of birds induced through the flapping of the wings, which generates a forward-directed thrust. In gliding flight (unpowered flight), the loss of altitude is converted into a forward direction of movement. In soaring flight (externally powered), uplift is used to gain height, which is then again converted into a forward direction of movement by gliding (Alerstam, Gudmundsson, and Larsson, 1993; Hedenstrom, 1993; Hedenstrom and Alerstam, 1998; Videler, 2006; Shepard and Lambertucci, 2013).

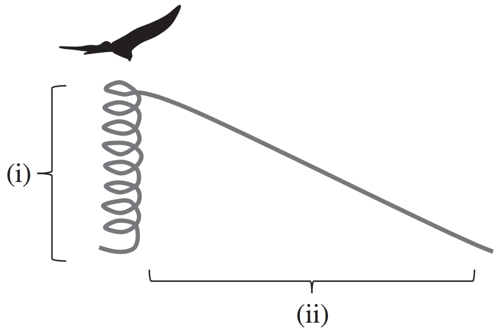


FIGURE 2.9: Soaring phase (i), where the bird gains height followed by a gliding phase (ii), where the potential energy is converted into kinetic energy. *Source: Shepard and Lambertucci (2013, p. 2)*

White storks are heavy birds and can therefore only use flapping flight for short distances. To cover large distances, they depend on vertically moving air masses that allow soaring flight (Figure 2.9). There are two types of uplift that soaring birds can use to gain height: *orographic* and *thermal* uplift. Orographic uplift is the rise in elevation of air masses above steep terrain, as for example over valley slopes and in front of mountain ridges. Controversy, the thermal uplift is induced by convection cells, caused by small-scale heterogeneity of solar heat absorption and storage of the Earth's surface. The current potential of a landscape with regard to soaring and gliding flight can be summarized as *energy landscape* (Péron et al., 2017). The energy landscape includes orographic uplift, caused by the topography, thermal uplift, induced by the weather conditions, as well as the distances between uplift zones and the current wind velocities. Soaring birds strive to minimize their Cost of Transport (CoT) by using uplift to gain height (Hedenstrom, 1993; Shepard and Lambertucci, 2013). The height gained while soaring is equivalent to the potential energy ($E_p = mgh$, where $m$ is the mass and $h$ is the position in height of the bird, and $g$ is the acceleration of free fall), which is converted into kinetic energy ($E_k = \frac{1}{2}mv^2$, where $m$ is the mass and $v$ is the speed of the bird) during gliding. Due to these physical dependencies and limitations, soaring birds are expected to adjust their flight behavior to the prevailing uplift conditions. Therefore, the energy landscape can be used to make assumptions about the flight height and the preferred route in a given landscape (Bohrer et al., 2012; Oloo, Safi, and Aryal, 2018).

## 2.4 Research Gaps

In the above review of theory and related work, whenever possible, concepts were presented that also apply to the 3-D case. Nevertheless, it became obvious that most of the current research restricts organism movement to a 2-D plane. In recent years, more and more 3-D movement data has become available, but due to the lack of appropriate movement models, the analysis is often still limited to the $x$ and $y$ coordinates. Especially when studying flying or diving animals, the incorporation of the height is of crucial importance. Probabilistic simulation models, such as the eRTG, are capable of reproducing trajectories that are empirically informed between given points, but due to the missing third dimension, the concepts are inadequate for many applications. Therefore the following research gaps need to be addressed:

- A probabilistic movement model, which represents the movement behavior in 3-D from the mover's perspective is clearly missing.
- A trajectory generator, based on such a probabilistic movement model, which is capable of producing individual empirically informed random trajectories in 3-D that are conditional on a given endpoint, does not exist.
- Most of the current approaches to analyze 3-D tracking data are restricted to the spatial extent of the observation and do not allow predictions in new, unseen areas, where no tracking data exist.
- Bird flight: To the author's knowledge, there is no model that allows the simulation of gliding and soaring trajectories based on an uplift suitability map between two given points.
- UDs are usually restricted to the 2-D space and no method is available to derive a UD in 3-D, based on a vast amount of simulated tracks in a new, unseen region.

# Chapter 3

# Study Area and Data

## 3.1 Study Area

The study area is the Swiss Central Plateau (green area in Figure 3.1), which extends from Lake Geneva in the southwest of Switzerland to Lake Constance in the northeast. In the south it is confined by the Alps and in the north the Jura mountain ridge forms a natural barrier. The altitude varies between 400 to 600 *m* above sea level and the landscape is hilly with several lakes and rivers.

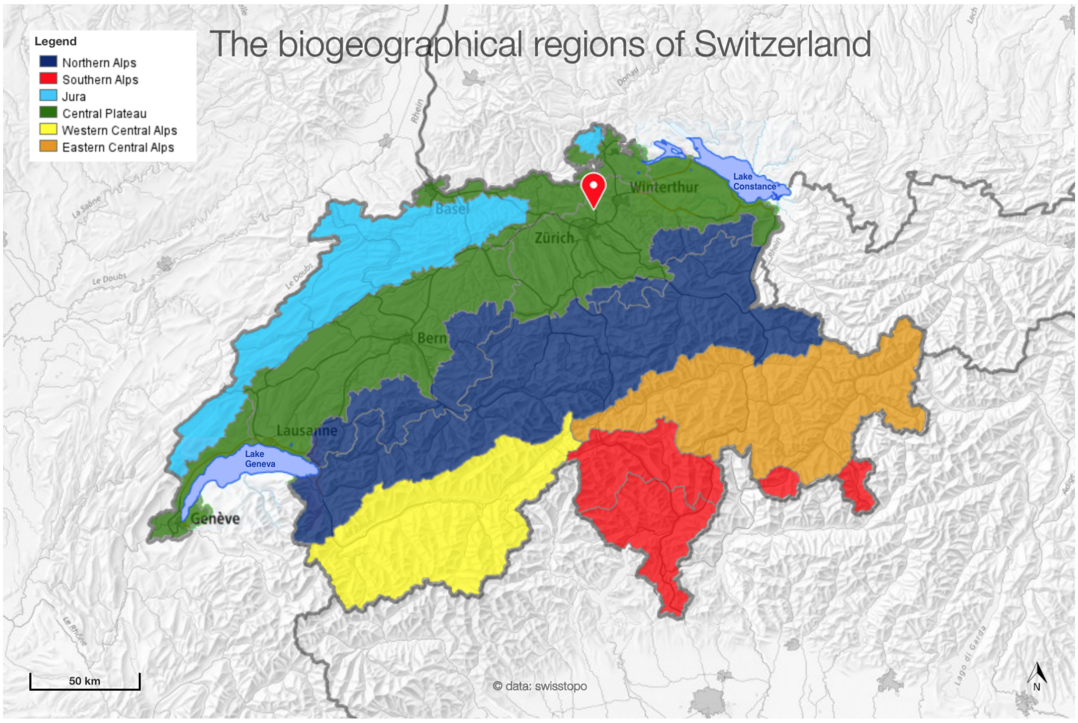


FIGURE 3.1: The biogeographical regions of Switzerland. *Modified figure, based on: swisstopo.ch*

Especially the region around Lake Constance is a breeding ground of white storks. The populations living there breed from spring to fall and begin their journey towards their wintering grounds in August and September. They prefer the western migration corridor and thus cross the Swiss Central Plateau towards the southeast.

There are several stop over sites in the Swiss Central Plateau, preferably lakes and associated wetlands. With the Alps in the south and the Jura in the north, topography dictates the white storks' migration corridor. Since the Swiss Central Plateau is a densely populated area with several major cities, the space use of humans and white storks overlaps. To the north of Zurich, there is Zurich Airport (red marker in Figure 3.1), which lies on the direct route of the birds migrating southeast.

For the high-resolution analysis of the movement behavior of white storks in the region near the airport, a buffer of 15 $km$ was drawn around it. All high-resolution simulations and airplane flight trails were masked to this buffer. The Cartesian projected coordinates of the airport in the Swiss projected coordinate system (CH1903+ / LV95) are: 2 684 039 $m$ east, 1 257 554 $m$ north, and 421.5 $m$ above sea level.

## 3.2 Data

### 3.2.1 Trajectories of White Storks

The GPS tracking data of white storks breeding near lake Constance stems from the study *Fall migration of white storks in 2014* (Movebank Study ID 332044860) and is a subset of the data in the study *LifeTrack White Stork SWGermany 2014-2017* (Movebank Study ID 21231406) on the movebank data repository. The data was published by Flack, Fiedler, and Wikelski (2017) and Weinzierl et al. (2016).

The data set contains the fall migration trajectories of 60 individuals from August until the end of September 2014. 7 153 159 GPS fixes were obtained during this period, by GPS devices attached to the birds. Beside the position also the height above the ellipsoid was recorded. Additionally, an accelerometer also collected data. The size of the GPS tracking data, without accelerometry is 1.7 $GB$. Due to limited battery life, the GPS tracking device only recorded data at regular sequences, with interruption intervals in between. The recorded sequences usually last 300 $s$ and have a high temporal resolution with a time lag of 1 $s$. Between the high-resolution sequences, the recording pauses for about 600 $s$.

The battery-saving recording technique that was used to record this data set is termed *timed sub-sampling*. The high-resolution sub-sampling can be scheduled based on regular intervals or be triggered by other sensors that consume less energy and run continuously, such as a 3-D-acceleration trigger. In case of solar energy-powered tracking devices, the sub-sampling starts whenever the battery power is sufficiently high. The advantage of this technique is that it provides a statistically valid picture of the movement behavior in high resolution and at the same time ensures a long-time coverage of the movement (Sherub et al., 2017).

After a visual inspection of the data in a Geographic Information System (GIS), the tracking data of white stork *Wibi 2 / DER AU044 (eobs 3943)* was chosen as it displays typical soaring and gliding flight behavior. The Coordinate Reference System (CRS) was transformed from latitude and longitude data (WGS84) to the Swiss projected reference system (CH1903+ / LV95). Then, the trajectory was cropped to the boundary of the Swiss Central Plateau. The remaining 4 800 GPS fixes were split into 16 high-resolution sequences interrupted by 15 breaks of 600 $s$ each.

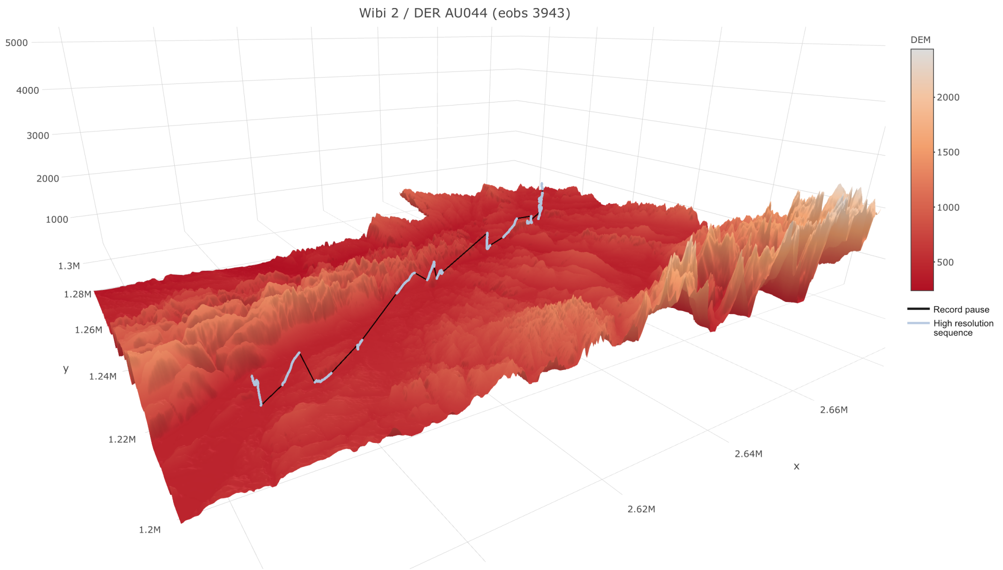


FIGURE 3.2: Fall migration of Wibi 2 crossing the Swiss Central Plateau, from the *Fall migration of white storks in 2014* data set.

According to Péron et al. (2017), the height above ellipsoid is typically more affected by measurement errors than the position in the $(x, y)$-plane. The accuracy of GPS tracking devices is largely depending on the satellite signal availability and therefore on the mover's position. At higher altitudes, surrounded by free sky, better accuracy often is achieved due to the lack of objects that interfere with the connection to the satellites. Higher sampling rates, as observed in this data set, lead to much more precise positioning data, since the connection to the satellites remains active between the fixpoints and has not to be re-established. Taylor and Blewitt (2006) provide an in-depth discussion of how satellite-based localization works.

For the evaluation of the algorithm, a low-resolution spring migration trajectory of the white stork *Niclas / DER AU053 (eobs 3341)*, from the same study (*LifeTrack White Stork SWGermany 2014-2017*), was used. Thereby a trajectory section, consisting of 31 fixpoints in the Swiss Central Plateau, was chosen by manual selection in a GIS. The average time lag between the fixpoints is 20 minutes. In this trajectory section from March 2017, Niclas crosses the Swiss Central Plateau during his migration northwards to the breeding grounds near Lake Constance. Again, the data has been transformed from latitude and longitude data to the Swiss projected reference system. Since the trajectory section does not contain any missing data, no further actions were necessary.

### 3.2.2 Digital Elevation Model

The digital height model *dhm25*, which serves as a reference for the flight height above ground and limits the constrained Euclidean space in which the simulated birds move, comes from *swisstopo*[1]. The raster data layer covers the entire territory of Switzerland with a cell size of 25 $m$. Swisstopo derives this matrix model from the vector-based National Map 1 : 25 000 by interpolation. Comparisons of height values from the raster with photogrammetrically measured control points show that in the Swiss Central Plateau and the Jura, the average error is 1.5 $m$; in the Prealps and Ticino it is 2 $m$, and in the Alps it is 3 to 8 $m$ (Swisstopo, 2018).

### 3.2.3 Uplift Suitability Map

Scacco et al. (in prep.) derived an uplift suitability map, based on observed soaring bird flight behavior and landscape features, which assesses the availability of uplift across Europe. The predictions of 10 models with a binary variable (0 = flapping flight, 1 = soaring flight) as response and different static environmental variables (elevation, land cover, etc.) as predictors were averaged. The resulting layer contains values from 0 to 1, which represents the likelihood of encountering uplifts that a bird can use for soaring. In Figure 3.3, the uplift suitability map was masked by the outline of Switzerland.

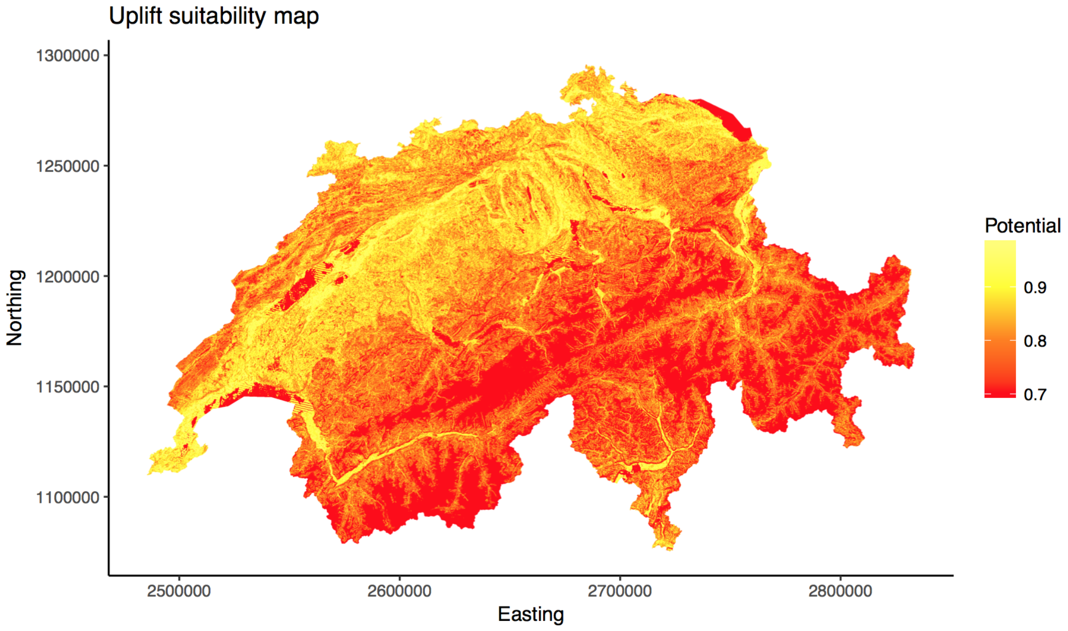


FIGURE 3.3: Uplift suitability map masked by the outline of Switzerland.

[1] www.swisstopo.admin.ch

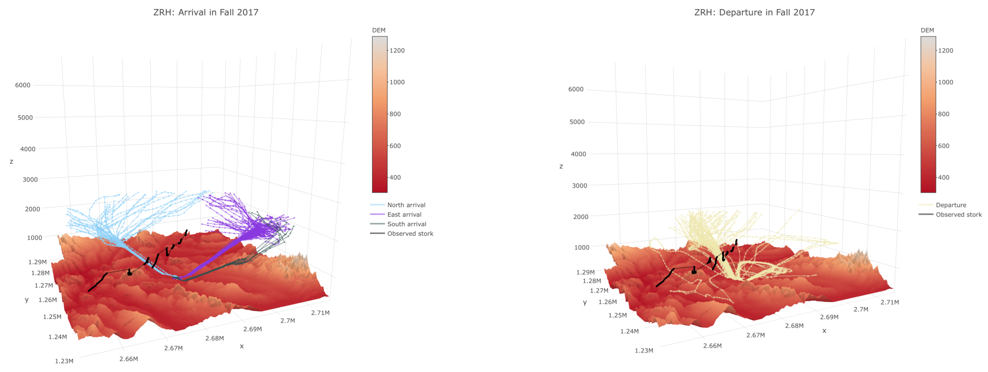
### 3.2.4 Air Traffic at Zurich Airport

In the past few years, Buchmüller et al. (2015) have recorded and preprocessed the complete flight traffic at Zurich Airport. The flight trails, which are stored as spatial *PostGIS*[2] objects in a *PostgreSQL*[3] Database (DB), are already classified into departures and arrivals. Arriving airplane trails are further divided into north, east and west flight corridor. The following Structured Query Language (SQL) query was used to extract and filter the data from the DB:

LISTING 3.1: SQL query: Accessing and filtering the flight trails

```
SELECT day, arrival, flighttrail::text FROM flights2
WHERE (EXTRACT(MONTH FROM day) = 9 OR EXTRACT(MONTH FROM day) = 8)
    AND EXTRACT(YEAR FROM day) = 2017
    AND NOT arrival = 'UAD'
    AND NOT arrival = 'Undefined'"
```

The retrieved position data of the flight trails is stored in latitude and longitude data (WGS84). Therefore, the CRS was changed from WGS84 to CH1903+ / LV95. Then the trails were cropped to the 15 *km* buffer around Zurich Airport. Often, when the planes were on the ground in the airport, height values below surface level were measured. To cope with this issue, these heights were mapped to the height of the surface at this position.



(A) Arrivals

(B) Departures

FIGURE 3.4: Flight traffic at Zurich Airport in August and September 2017.

Figure 3.4 shows a summary of the flight trails, the Digital Elevation Model (DEM) and the white stork trajectory in the 15 *km* buffer around Zurich Airport. The white stork Wibi 2 (black line) has come very close to the northern arrival and the eastern departure corridors. Although the two data sets were recorded in the same season of different years, it can be assumed that the white storks' migration corridors stay more or less constant (Van den Bossche et al., 2002).

[2] www.postgis.net

[3] www.postgresql.org

# Chapter 4

# Methodology

In this part of the thesis, the methodology that is used to realize, evaluate, and subsequently apply the eRTG3D is introduced. The chapter is divided into three sections, each dealing with one research question. Since the design and the implementation of the algorithm was an iterative procedure, in which the complexity of the versions continuously increased, only the methodology of the final version is described here. This version was also evaluated and applied in the demonstrator use case.

The implementation, evaluation, and application of the eRTG3D were written in *R*[1], a language and environment for statistical computing and graphics (R Development Core Team, 2008). The implementation started from an existing R script for the 2-D version of the eRTG (Technitis et al., 2016). *RStudio*[2] served as an open-source Integrated Development Environment (IDE) to create an R package containing the final version of the eRTG3D. In the description of the eRTG3D in the following sections, reference to the corresponding *R* functions will be made by means of footnotes.

## 4.1 Development and Implementation of the eRTG3D

Since different laws apply to the motion in the third dimension, a direct extension of the 2-D eRTG to 3-D is not feasible in one step. The strategy to cope with the increasing complexity arising from the inclusion of the third dimension was the elaboration of several versions of the eRTG3D. The complexity and functionality of the versions were increased gradually. First, a simple and unconstrained version in 3-D was elaborated, which afterwards was extended with vertical constraints, such as the Earth's surface and the maximum flight height. Then, probabilistic distributions, extracted from the *absolute* flight height over the ellipsoid and the *relative* flight height over ground, were incorporated. They promote the use of absolute heights and distances to the ground which are biologically more likely. To achieve fully representative simulated trajectories in 3-D, a probabilistic distribution of the gradient angle $g$ (polar angle $\theta$) is added to the final version. In addition, a slightly modified version was developed that allows bird flight simulations on an uplift suitability map. This version combines the movement modes *gliding* and *soaring* in one model.

[1] www.r-project.org

[2] www.rstudio.com

### 4.1.1 P Probability – The Mover's Behavior from its Perspective

**Turn-Lift-Step Histogram**

The starting point of the algorithm is the mover's behavior from its perspective. This perspective on the change of absolute orientation and position in space can be described by probabilistic distributions of the turning angle $t$, lift angle $l$, and step length $d$. To provide a sensible representation of the movement, these independent secondary derivative movement parameters have to be connected. If the turning angle, lift angle, and step length depend on each other, a sharp turning at high velocity, which means a long covered distance in one step, is less likely to occur than a smooth turn. To establish this dependency, a 3-D histogram, the *turn-lift-step histogram*, is extracted from the tracking data[3]. This histogram can be seen as a voxel cube with $(t, l, d)$-dimensions. Every combination of turning angle, lift angle, and step length is represented by a unique point that belongs to a voxel in the cube. The voxels represent a 3-D binning of the 3-D histogram, and their (bin-) size in each dimension is estimated by the Freedman–Diaconis rule:

$$binSize = 2 \cdot \frac{IQR(x)}{\sqrt[3]{n}} \tag{4.1}$$

In this rule, $IQR$ is the interquartile range of the data sample $x$ and $n$ is the number of observations (Freedman and Diaconis, 1981). Given the bin sizes in all three dimensions, a voxel space is set up that spans between the minimum and maximum value in each dimension. Then, the $(t, l, d)$-combination of every step is assigned to its corresponding voxel in the voxel space. Further, the occurrences per voxel are counted and normalized by the total number of observed steps. The resulting turn-lift-step histogram (or *tld-cube*) represents the probability of every $(t, l, d)$-combination in the voxel space to occur.

**Autodifferences**

The turn-lift-step histogram is not sufficient to fully describe the movement behavior from the perspective of the mover, since it reveals nothing about the autocorrelation of the $(t, l, d)$-combinations. However, since there is a consistency in the movement of animals, it is highly unlikely that a large step with a little change of direction is followed by a short one that includes an abrupt change of direction. The concept used in animal ecology to describe this dependency of steps on previous ones, is the calculation of *autodifferences*. Autodifferences are the differences in the turning angles, lift angles, and step lengths with a lag of 1, which maintains a minimal level of autocorrelation in each of the terms. A function is fitted to the kernel density estimates of each of the autodifferences in order to approximate the underlying Probability Density Functions (PDFs). The KDE uses a Gaussian kernel with the statistical property

[3] **Appendix A.2:** eRTG3D: *turnLiftStepHist()*

${\sigma^2}_k = \int t^2 k(t) \mathrm{d}t$ that is always equal to 1. Therefore, the bandwidth is the standard deviation of the kernel (Sheather and Jones, 1991; Venables and Ripley, 2003).

### Height Distributions

There are physical limitations that constrain the birds from moving freely in the 3-D space. Underground flying is, for obvious reasons, as impossible as flying at extreme altitudes, which can not be performed due to insufficient oxygen content, low temperatures and problems with air density. The distribution of the height values is therefore used to limit the degree of freedom in the third dimension.

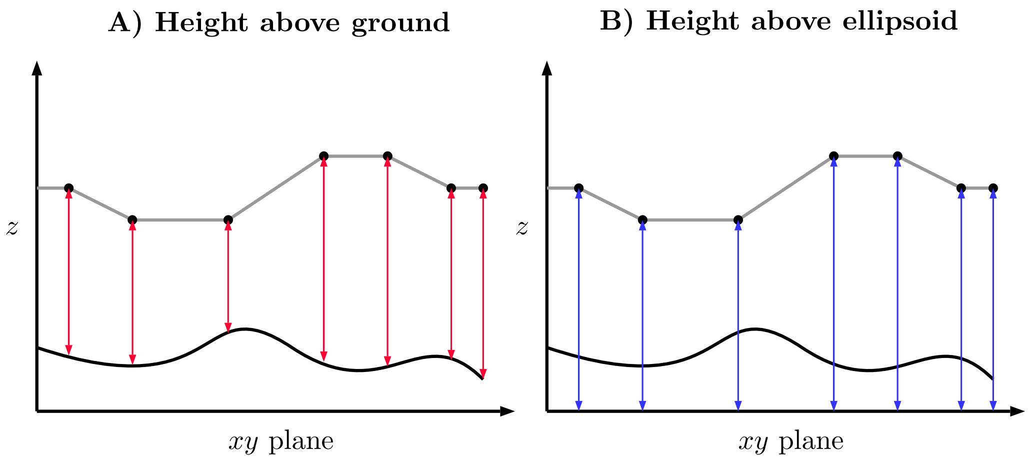


FIGURE 4.1: Height above ground and ellipsoid.

As Figure 4.1 indicates, there are two different perspectives on flight height: (A) the *height above ground*, which is the vertical distance between the bird and the Earth's surface, and (B) the *height above ellipsoid*, which is the vertical distance that is defined relatively to a reference ellipsoid. In general, the height above ellipsoid can be seen as the height above sea level, since the ellipsoid is usually fitted to the surface of the Earth at sea level. The distribution of the height above ellipsoid represents the preferred flight heights of a bird, which, for example, prevents it from crossing a mountain at an unrealistic altitude, whereas, the height above ground describes the behavior of the bird with regard to a changing topography, which prevents it from flying too close to the surface or even underground. With the goal of forcing the bird to fly at heights that are biologically more likely, the PDFs of both heights are estimated by fitting approximation functions to the KDEs of their empirically observed values.

### Gradient Distribution

A particularly important distribution, especially for the gliding and soaring behavior of birds, is that of the gradient $g$. Due to gravity, flying with a gradient between 0 and $\frac{\pi}{2}$ requires either uplift or the bird needs to generate a thrust by flapping its wings. In contrast, while descending with a gradient angle between $\frac{\pi}{2}$ and $\pi$, the potential energy of the altitude can be converted into kinetic energy and the bird moves forward. When modeling the different movement modes it becomes apparent that the choice between descending and ascending has to be considered. In soaring mode generally ascending gradients should be preferred over descending angles. Therefore, the PDF of the gradient is also approximated by fitting a function to the KDE of the gradient.

The combination of the turn-lift-step histogram with the approximations for the PDFs of the autodifferences, height and gradient, is termed the *P probability*[4]. The P probability fully describes the movement behavior from the mover's perspective in a stochastic manner. It is important to extract the P probability only from trajectories that have a constant time lag between the fixpoints. Otherwise the extracted probabilities are distorted and do not represent the movement behavior appropriately. In order to avoid this issue, missing data or delays between the fixpoints have to be removed from the data. To achieve a constant time lag, the trajectory can be split into proper trajectory sections[5], from which then the P probability is extracted[6].

### 4.1.2 Unconditional Empirical Random Walk

After the extraction of the P probability, an Unconditional Empirical Random Walk (UERW) in 3-D can be generated[7]. Since the walk is unconditional, no endpoint is given. This means that the mover moves away from a given starting point for a predefined number of steps. The constant time between the locations is identical to the time lag of the observed trajectory that the P probability was extracted from. The trajectory is generated in a stepwise procedure.

The trajectory generation needs start conditions. The initial orientation (azimuth $a_0$ and gradient $g_0$) of the mover and the starting point $(x_0, y_0, z_0)$ have to be set. Since the walk pays attention to the consistency in the movement behavior by considering the distributions of the autodifferences, an imaginary previous step has to be assumed. Therefore, a $(t, l, d)$-combination is sampled from the turn-lift-step histogram with regard to their probability of occurrence. The sampled combination is used to imitate the previous step. As a result, the initial values ($x_0$, $y_0$, $z_0$, $a_0$, $g_0$, $t_0$, $l_0$, $d_0$) are complete and the generation of the UERW can be undertaken.

At the given starting point of the walk, all possible combinations $(ts, ls, ds)$ for the next step $i + 1$ are taken from the turn-lift-step histogram and weighted with their probability $tldP$ of occurrence. Since only the middle value of the bins in each dimension can be selected, no continuous choice in the $(t, l, d)$-space is possible. To avoid this issue, a randomly sampled shift from the uniform distribution

[4] **Appendix A.4:** eRTG3D: *get.track.densities.3d()*
[5] **Appendix A.5:** eRTG3D: *track.split.3d()*
[6] **Appendix A.6:** eRTG3D: *get.section.densities.3d()*
[7] **Appendix A.7:** eRTG3D: *sim.uncond.3d()*

$U(-\frac{binSize}{2}, \frac{binSize}{2})$ is added to every bin midpoint, whereby the $binSize$ is adjusted to the corresponding dimension. For each step $i$ and dimension $(t, l, d)$, a new random shift ($tShift_i$, $lShift_i$, $dShift_i$) is sampled from the uniform distributions. Like this, all values in the range of a bin have the same probability to be selected. The gradients $gAll_{i+1}$ and azimuths $aAll_{i+1}$ for all possible next locations are calculated by:

$$aAll_{i+1} = a_i + ts + tShift_i \tag{4.2}$$

$$gAll_{i+1} = g_i + ls + lShift_i \tag{4.3}$$

The application of the approximation function of the gradient PDF $gDens()$ on all possible gradients of the next step yields their biological probability $gP_{i+1}$ of occurrence:

$$gP_{i+1} = gDens(gAll_{i+1}) \tag{4.4}$$

Next, the approximation functions ($autoT()$, $autoL()$, $autoD()$) of the autodifference PDFs are used to estimate the likelihood of the turning angle $atP_{i+1}$, lift angle $alP_{i+1}$, and step length $alD_{i+1}$ for the next step $i+1$, based on the $(t, l, d)$-combination of the previous step $i$:

$$atP_{i+1} = autoT(t_i - ts + tShift_i) \tag{4.5}$$

$$alP_{i+1} = autoL(l_i - ls + lShift_i) \tag{4.6}$$

$$adP_{i+1} = autoD(d_i - ds + dShift_i) \tag{4.7}$$

The probability $P_{i+1}$ for the selection of the next location, from the perspective of the mover, is achieved by multiplication of the different probabilities. To aggregate the autodifference probabilities into one probability, the cubic root of their multiplication is taken (see reasoning in Section 6.1 on combining probabilities, p. 76):

$$P_{i+1} = tldP \cdot gP_{i+1} \cdot \sqrt[3]{atP_{i+1} \cdot alP_{i+1} \cdot adP_{i+1}} \tag{4.8}$$

A sample with the probability $P_{i+1}$ on the voxel of the turn-lift-step histogram and the addition of the corresponding uniformly distributed shift terms results in the $(t, l, d)$-combination for the next step $(t_{i+1}, l_{i+1}, d_{i+1})$, from which the next location $(x_{i+1}, y_{i+1}, z_{i+1})$ in Cartesian coordinates is then derived:

$$x_{i+1} = x_i + d_{i+1} \cdot \sin\big(g_i + l_{i+1}\big) \cdot \cos\big(a_i + t_{i+1}\big) \tag{4.9}$$

$$y_{i+1} = y_i + d_{i+1} \cdot \sin\big(g_i + l_{i+1}\big) \cdot \sin\big(a_i + t_{i+1}\big) \tag{4.10}$$

$$z_{i+1} = z_i + d_{i+1} \cdot \cos\big(g_i + l_{i+1}\big) \tag{4.11}$$

By repeating these calculations for every step, a UERW in 3-D that has the prescribed empirical properties is successfully generated. The resulting trajectory can be used as unconditional walk or to seed the pull towards the target for the simulation of conditional walks.

### 4.1.3 Q Probability – The Pull Towards the Target

A conditional walk, connecting a starting with an endpoint by a predefined number of steps $n$, needs an attraction term: the *Q probability*[8]. This pull towards the target ensures that the endpoint is approached and hit. However, the influence of the Q probability on the walk is not constant, as the movement behavior of the mover relative to the endpoint varies over time. In order to calculate the Q probability, the distribution of turning angles, lift angles, and distances to target has to be known for every step in the walk. The distributions are ideally derived from empirical data or estimated from an unconditional process with the same properties, such as an UERW. The turning angle to target $tTarget_i$[9], lift angle to target $lTarget_i$[10] and the distance to target $dTarget_i$[11] are calculated for every step $i \in \{1, \ldots, n\}$:

$$tTarget_i = \text{atan2}(y_n - y_i, x_n - x_i) - a_i \tag{4.12}$$

$$lTarget_i = \text{atan2}(\sqrt{(x_n - x_i)^2 + (y_n - y_i)^2}, z_n - z_i) - g_i \tag{4.13}$$

$$dTarget_i = \sqrt{(x_n - x_i)^2 + (y_n - y_i)^2 + (z_n - z_i)^2} \tag{4.14}$$

In this, $(x_n, y_n, z_n)$ is the target location. To extract a *tTarget-lTarget-dTarget* histogram in the same way as described in Section 4.1.1, a sufficient amount of samples per step in each dimension need to be available. If a conditional walk with $n$ locations is to be generated, for every step, except for the last one, a histogram cube needs to be created. Therefore, $n - 1$ histogram cubes are needed. In order to have enough samples to ensure sound statistical distributions, a trajectory for the extraction of the Q probability has to be at least 1500 times longer than the conditional walk that is to be simulated. This lower bound was determined empirically, as it yields robust simulation results.

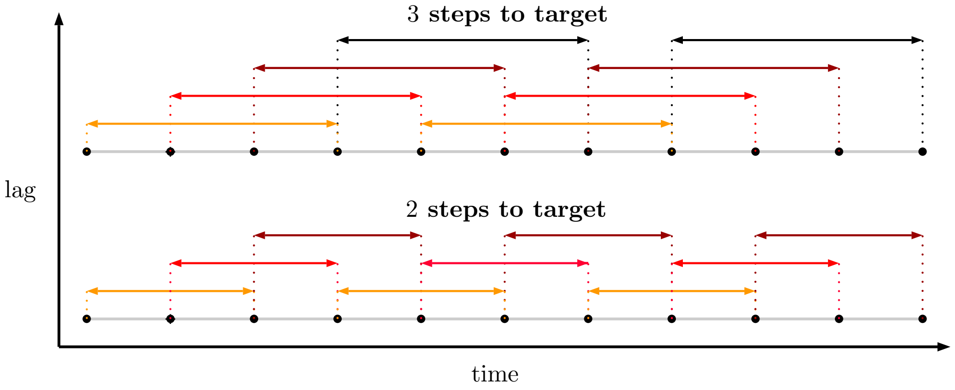


FIGURE 4.2: Moving window approach for the extraction of the Q probability from a trajectory.

[8] **Appendix A.8:** eRTG3D: *qProb.3d()*
[9] **Appendix A.9:** eRTG3D: *turn2target.3d()*
[10] **Appendix A.10:** eRTG3D: *lift2target.3d()*
[11] **Appendix A.11:** eRTG3D: *dist2target.3d()*

Using a moving window approach for every number of steps left to the target ($lag \in \{n, \dots, 2\}$), the relationship between the step $i = n - lag$ and the target $i = n$ is calculated. Figure 4.2 shows the moving window approach for the extraction of the Q probability for the steps $i = n - 3$ and $i = n - 2$ in the conditional walk.

### 4.1.4 Conditional Empirical Random Walk

Given the P and Q probabilities, a Conditional Empirical Random Walk (CERW) between two points is simulated[12]. For this purpose, the same start conditions as for the UERW and an additional endpoint are needed. A further important parameter is the number of steps provided to reach the endpoint. If a trajectory should be reproduced[13], usually the same number of steps is taken as the original trajectory consists of. If a trajectory is simulated between two new sites, the number of steps is estimated either based on the total available time or the average distance to target the mover can cover in one step. A time-based estimation of the number of steps is obtained by dividing the total time by the time lag between the steps:

$$n = \left\lceil \frac{t_n - t_0}{timeLag} \right\rceil \tag{4.15}$$

Hereby, the result of the division is rounded up to the nearest integer. To gather a distance-based estimation, the total Euclidean distance between the points is divided by the mean step to target $d_{mean}$:

$$n = \left\lceil \frac{\sqrt{(x_n - x_0)^2 + (y_n - y_0)^2 + (z_n - z_0)^2}}{d_{mean}} \right\rceil \tag{4.16}$$

The number of steps needed to connect the two points with a CERW is always defined before the construction of the walk. As for the UERW, the generation of the CERW is also a stepwise procedure. The selection of a next step starts with the calculation of the P probability $P_{i+1}$ at the current location, based on Equation 4.8. Then, the Cartesian coordinates of all possible next locations ($xAll_{i+1}$, $yAll_{i+1}$,$zAll_{i+1}$), defined by the current position $(x_i, y_i, z_i)$, possible next azimuths ($aAll_{i+1}$) and gradients ($gAll_{i+1}$) are calculated based on Equations 2.3, 2.4 and 2.5 (p. 15). For all possible next locations, the turning angle, lift angle and distance to the target are derived by Equations 4.12, 4.13 and 4.14. The result is a value triplet assigned to every possibility. From the Q probability, the *tTarget-lTarget-dTarget* histogram that belongs to the current step is used to assess the probability of occurrence of each of these value triplets. The probability is obtained by searching the voxel which contains the value triplet. If a value triplet is located outside the (*tTarget*,*lTarget*,*dTarget*) space of the histogram, a probability of zero is assigned to the triplet. The resulting probability

[12] **Appendix A.12:** eRTG3D: *sim.cond.3d()*

[13] **Appendix A.13:** eRTG3D: *reproduce.track.3d()*

$Q_{i+1}$ represents the likelihood of every possible next location to be picked, based on the movement behavior in relation to the endpoint.

The final probability $PQ_{i+1}$ is obtained by multiplying $P_{i+1}$ and $Q_{i+1}$. Additionally, the approximation functions of the relative and the absolute flight height PDFs are used to weight the probability of the next location. To extract the height of the Earth's surface $z_{i+1}^{topo}$ at all possible next locations, a DEM covering the extent of the simulation area is needed. An equal influence of the flight height above ground and the flight height above ellipsoid is achieved by multiplying them and taking the square root:

$$PQ_{i+1} = \underbrace{P_{i+1}}_{tldP \cdot gP_{i+1} \cdot \sqrt[3]{atP_{i+1} \cdot alP_{i+1} \cdot adP_{i+1}}} \cdot Q_{i+1} \cdot \sqrt{hDistEllipsoid(z_{i+1}) \cdot hDistTopo(z_{i+1} - z_{i+1}^{topo})} \tag{4.17}$$

The probability $PQ_{i+1}$ is now used to sample a voxel from the turn-lift-step histogram. The addition of the corresponding uniformly distributed shift terms to the sample results in the $(t, l, d)$-combination for the next step $(t_{i+1}, l_{i+1}, d_{i+1})$. Then, the next location $(x_{i+1}, y_{i+1}, z_{i+1})$ in Cartesian coordinates is derived by Equations 4.9, 4.10 and 4.11. This procedure is repeated until the penultimate step, which produces the data structure illustrated in Table 4.1.

| *step* | x | y | z | a | g | t | l | d | p |
|---|---|---|---|---|---|---|---|---|---|
| *i=0* | 0.0 | 0.0 | 10.0 | 1.0 | 2 | -0.2 | -0.2 | 1.1 | *NA* |
| *i=1* | 0.5 | 0.9 | 9.6 | 1.1 | 1.9 | 0.1 | -0.1 | 1.1 | 0.2 |
| *i=2* | 1 | 1.6 | 9.4 | 0.9 | 1.8 | -0.2 | -0.1 | 0.9 | 0.1 |
| ⋮ | | | | | | | | | |
| *i=n-1* | 11.1 | 16.7 | 1.8 | 0.9 | 2 | -0.2 | -0.2 | 1.1 | 0 |
| *i=n* | 11 | 17.5 | 1.4 | 1.0 | 1.9 | 0.1 | -0.1 | 0.9 | *NA* |

TABLE 4.1: Data structure of an example CERW.

In order to complete the walk, the penultimate location $(i = n - 1)$ is connected with the last location $(i = n)$, which thereby defines the $(t, l, d)$-combination of the last step. Finally, it is tested if this combination is possible in the P probability. If the probability of this combination is not zero, the mover has successfully performed a CERW. In case that the probability of the final combination is zero, a *dead end* is encountered and the walk ends without having reached its destination.

If at any other step during the walk all possible next steps have a probability of zero, the CERW ends abruptly in a dead end. The fact that $PQ_{i+1}$ is zero can have several causes; it is mostly due to an insufficient number of steps left to reach the endpoint. Since the CERW is based on a probabilistic model, not very probable movement combinations might occur as well. In such a case, the trajectory might grow into an undesirable direction in respect to the target. The influence of the autodifferences can further limit the freedom to reach the target. Other possibilities are dead ends caused by the topography or an improper extraction of the probabilities from the original trajectory.

When extracting the movement behavior of trajectories, it is very important that this only happens for one movement mode at a time. If different modes occur together in a trajectory, it must be split accordingly. Otherwise, the extracted turn-lift-step histogram and the corresponding PDF approximations of the autodifferences are no longer unique. A simulation based on these probabilities favors the occurrence of dead ends, since unrealistic combinations, such as the transition between the modes, are allowed to occur.

In addition, it is imperative that the time lag between the fixpoints in the original trajectory remains constant. Outliers distort the extracted density distributions, especially if the trajectory is rather short and therefore does not offer many samples. This issue can be circumvented by splitting the trajectory at steps with deviating time lags. Thereafter, the extraction of the probabilities is conducted on the clean sections.

### 4.1.5 Gliding and Soaring Simulation

Finally, an adapted version of the algorithm, which is capable of simulating gliding and soaring behavior[14], was developed. In this version, the simulation is grounded on a binary uplift raster layer, where a pixel value of 2 indicates an area that is suitable for soaring, while 1 denotes an absence of uplift, whereby gliding flight is enforced. The approach behind the simulation is the assumption that a bird approaches its target in (conditional) gliding mode and gains height in (unconditional) soaring mode, without a target direction. This seems to be reasonable, since in soaring mode only the gain in vertical distance matters, whereas in gliding mode the loss in height is converted into a mainly horizontal movement towards the target. Since gliding and soaring are two different movement modes of the same individual, also two P probabilities (one for each mode) must be extracted. In addition, the pull towards the target must be provided. Based on the assumption that the target is only approached in gliding mode, the Q probability has to be calculated based on pure gliding behavior. Therefore, the pull towards the target is either extracted from a long observed gliding trajectory or from an UERW based on the P probability of the gliding mode. The number of steps needed to hit the target is estimated using the distance-based approach.

The glide ratio $\epsilon$ limits the height a bird is allowed to gain in an uplift area. It defines the horizontal distance $\Delta s$ a bird is able to cover per vertical distance $\Delta h$ loss:

$$\epsilon = \frac{\Delta s}{\Delta h} \tag{4.18}$$

It can either be estimated by experts or empirically extracted from tracking data. For soaring birds this ratio typically is somewhere between 15 to 22. If a bird in the

[14] **eRTG3D:** *sim.glidingSoaring.3d()*

simulation encounters an uplift area (pixel value = 2), it gains height as long as the conversion of height into horizontal distance is insufficient to reach the endpoint. Otherwise, the soaring mode is only left if unrealistic heights are reached or the uplift area ends. The steps covered in soaring mode are unconditional (UERW) and are therefore not counted as steps in the actual CERW. As soon as the bird switches from soaring mode back to gliding mode, the pull towards the target is activated again. In gliding mode the bird ignores uplift areas as long as it has sufficient altitude to reach the endpoint.

As the bird moves unconditional while soaring, its orientation in space at the moment of a transition to gliding mode might not be pointed towards the target. For the calculation of the Q probabilities this issue is problematic, since a situation, in which the bird is pointed to the opposite direction in respect to the target is very likely to occur. Depending on the number of steps left, a turn towards the target in gliding mode might be unrealistic and the Q probability becomes zero. To avoid this problem, the transition is smoothed if the difference of the bird's azimuth and the azimuth towards the target is more than $\frac{pi}{12}$. The steps allowed for a smooth transition towards the target is based on the turning angle of the last step and is at maximum 9 steps. The number of smoothing steps is calculated by dividing the required turning angle to the target by the last turning angle. For the given number of smoothing steps the bird performs a correction-trajectory until its azimuth and gradient are pointed towards the target. In correct orientation, the trajectory continues in gliding mode until the target is reached or another uplift area is used.

### 4.1.6 Variations and Efficiency

The formally described versions of the algorithm include the distributions of the gradient, the absolute and the relative flight height. Therefore, these versions are tailored to the modeling of bird flight. However, the algorithm also works with a subset or without these distributions. Simulations without any of these distributions equate to a space, in which no physical constraints, such as gravity or obstacles, influence the movement behavior.

Although the focus was placed on the implementation of a properly functioning algorithm, efforts have been made to increase its efficiency. All processes that do not depend on previous steps, and are therefore independent, can be calculated in parallel. To achieve calculations that run on multiple cores, the capabilities of the *parallel* package to fork R sessions are used. The extraction of Q probabilities and the simulation of multiple CERWs can be run in parallel[15]. To further improve the efficiency of the trajectory generation, the maximum number of bins per dimension in the turn-lift-step histogram can be limited. The number of bins is set by default to a maximum of 25 in each dimension, but the value can be chosen freely. An unlimited number of bins is still possible by setting the value to infinity[16].

[15] **eRTG3D:** multicore = TRUE

[16] **eRTG3D:** maxBin = Inf

## 4.2 Evaluation of the Algorithm

The evaluation of the algorithm is carried out on the previously described version for one movement mode, using synthetic trajectories and real-life data. For the internal verification of the algorithm, 3-D CRWs are generated[17] for correlation values between 0.1 and 0.99, since the initially set conditions for their production are preserved. Then, these trajectories are reproduced by the eRTG3D algorithm. In a next step, the simulations are statistically compared with the CRWs that served as input for the simulations. Due to the fact that the trajectories do not necessarily share the same extent and area, relative quantities that describe the movement behavior have to be compared. Therefore, the distributions of the turning angle, lift angle, and step length of the simulated and the original trajectory are tested against each other. Two different approaches are applied for testing the distributions.

In order to access the validity of the algorithm, an empirically observed trajectory is reproduced multiple times by the eRTG3D algorithm. The simulations are then inspected visually in their spatial domain by 2-D[18] and 3-D plots[19], and in the statistical domain by plotting distributions of their geometric properties[20]. Additionally, a sensitivity analysis of the parameter that limits the maximum number of bins per dimension in the turn-lift-step histogram is performed. The results are verified statistically and validated through visual inspection.

### 4.2.1 Two-sample Kolmogorov–Smirnov Test

The first approach for the verification is a two-sample Kolmogorov–Smirnov test that is applied to the corresponding distributions of turning angle, lift angle, and step length and their autodifference distributions[21]. The Kolmogorov–Smirnov statistic quantifies a distance between the empirical distribution functions of two samples. Therefore the hypotheses of the test are:

$H_0 : F_X(x) = F_Y(x)$ – *The probability distributions do not differ significantly.*
$H_1 : F_X(x) \neq F_Y(x)$ – *The probability distributions differ significantly.*

To ensure that the simulated trajectory represents the original trajectory, significance is not allowed for any test at a significance level of $\alpha = 0.05$. If this condition holds, the trajectories are considered as statistically not differing. The approach states that as long as no significant deviations are detected between the distributions of the trajectories, the trajectories can not be classified as unequal movement behavior. This test is very powerful since it looks at the distributions as a whole and does not just compare the locations of their means. Nevertheless, its sensitivity to sample size can

[17] **Appendix A.14:** eRTG3D: *sim.crw.3d()*
[18] **eRTG3D:** *plot2d()*
[19] **eRTG3D:** *plot3d()*
[20] **eRTG3D:** *plot3d.densities()*
[21] **Appendix A.15:** eRTG3D: *test.verification.3d(test = "ks")*

be considered its downside (Wasserstein and Lazar, 2016). At sufficiently large sample sizes, even the smallest differences between the distributions become significant. Especially when dealing with simulations that may quickly become very large in size, a more robust procedure would be desirable.

### 4.2.2 Mean Difference and One-sample t-Test

The basis of the second statistical test approach, is the assumption that, if two trajectories are identical, the mean difference of many randomly sampled pairs of turning angles, lift angles, and step lengths should be zero[22]. The smaller number of steps of the compared trajectories is taken as sample size. Then, random pairs are drawn from both trajectories for turning angles, lift angles, and step lengths. The differences of the pairs build three distributions. The mean of these distributions is expected to be zero if the trajectories are grounded on the same movement behavior. Therefore, the distributions are tested against an expected value of zero, by applying a two-sided one-sample t-test, with the following hypotheses:

$H_0 : \mu = 0$ – *The mean of the distribution differs not significantly from zero.*
$H_1 : \mu \neq 0$ – *The mean of the distribution differs significantly from zero.*

If two trajectories represent the same movement behavior, the tests should not be significant at a significance level of $\alpha = 0.05$.

### 4.2.3 Visual Comparison and Sensitivity Analysis

In order to test the ecological validity of the eRTG3D algorithm, an observed white stork trajectory is reproduced and then plotted. In addition to the simulations, the 2-D and 3-D plots also contain the original trajectory and are provided with different backgrounds, such as an uplift suitability map or a DEM. This allows a visual validation of the simulations in the living environment of the animal. Thereby, the focus is primarily set on a similar geometry of the simulated trajectories in comparison to the original, which also includes equal behavior in height and gradient choice. In addition, the consistency in movement is compared and obvious violations are checked. Underground flights, unrealistically high altitudes or extreme turning and lift angles are considered to be impossible.

To estimate the influence of the bin limitation in the turn-lift-step histogram, a sensitivity analysis is conducted on this parameter. An observed trajectory is reproduced with different bin limitations, ranging from only 10 to an infinite number of possible dimensions in the tld-cube. The resulting trajectories are tested statistically against each other with the approaches described in Section 4.2.1 and 4.2.2, respectively. In addition, the results are also compared visually.

[22] **Appendix A.15:** eRTG3D: *test.verification.3d(test = "ttest")*

## 4.3 Bird-strike Probability at Zurich Airport

In this section, the methodology behind the demonstrator use case of the eRTG3D algorithm is introduced. The focus is put on the collision probabilities of migrating white storks and airplanes, starting and landing at Zurich Airport, in fall. Therefore, the general approach is to derive UDs for white storks and airplanes in a buffer of 15 $km$ around the airport. The UDs are based on a vast amount of simulated white stork migration trajectories and the tracked airplane flight trails in August and September of 2017. The simulations are based on the movement behavior of the white stork Wibi 2. To ensure as much randomness as possible in the stork's choice of a route to cross the Swiss Central Plateau, low-resolution simulations between the lake Constance and Neuchâtel are conducted. Then they are masked by the buffer around the airport and up-sampled to high-resolution gliding and soaring behavior. From these high-resolution gliding and soaring trajectories, the final UD is extracted. This UD quantifies the space use of the white storks during the fall migration around Zurich Airport, taking into account the average uplift conditions in the region.

Due to the heavy computational costs for the simulation of a large number of high-resolution gliding and soaring trajectories, the calculations are exported to the *Draco*[23] High Performance Computing (HPC) cluster of the Max Planck Society. Therefore, the work flow is adjusted in a way that the generation of the trajectories can be distributed across the nodes of the cluster.

### 4.3.1 Preprocessing Data

After all data sets have been defined and transformed to the same CRS, data preparation is undertaken. First, the airplane flight trails are checked for missing height values. Missing height information is filled in by linear interpolation and height values below ground are set to the level of the topography at this location. This procedure is omitted for the white stork trajectory of Wibi 2, since no missing values occur. Then, time-constant sections of the white stork trajectory are identified. This is done by setting an expected time lag of 1 second, and by allowing a tolerance of 0.5 seconds. Steps with time lags that deviate more than the set tolerance from the expected time lag are used to place a split. The result is a list of time-constant trajectory sections that serve as input for further processing.

Since the gliding and soaring simulations require a binary uplift raster layer, the continuous uplift suitability map is classified into two categories. The first category is assigned the value 1 and forces the bird to gliding mode, while the second category has the value 2, which allows the bird to fly in soaring mode. The selection of the biologically sensible threshold of 0.9 is based on a binary classification test conducted on independent data by Scacco et al. (in prep.). In this test, the sensitivity is equal to the number of correctly classified soaring steps, and the specificity represents the number of correctly classified gliding steps. Thresholds between 0.88 and

[23] www.mpcdf.mpg.de/services/computing/draco

0.92 were found to give the best results. In Figure 4.3 the chosen threshold is marked by a vertical dotted black line.

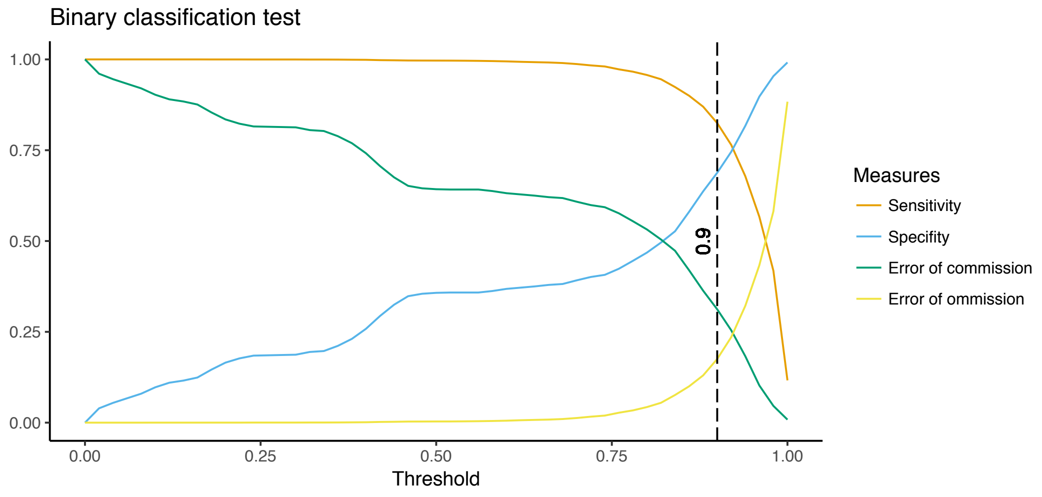


FIGURE 4.3: Binary classification test for setting a threshold in the uplift suitability map. *Based on: Scacco et al. (in prep.)*

### 4.3.2 Low-resolution Simulation on a Large Scale

Since high-resolution gliding and soaring simulations crossing the Swiss Central Plateau are computationally too expensive to realize, the approach is to simulate only inside the buffer of 15 *km* around the airport in a high resolution. In order to ensure an ecologically sensitive, but random distribution of starting and endpoints on the buffer's outline, low-resolution simulations crossing the Swiss Central Plateau are generated. In a next step, their part inside the buffer is up-sampled to high-resolution gliding and soaring simulations between the low-resolution fixpoints.

120 seconds are chosen as time lag for the low-resolution simulations. This time lag provides a good compromise between a sufficient number of detailed trajectories and satisfactory efficiency. First, the proper high-resolution sections of the observed white stork trajectory are down-sampled to the time lag of 120 seconds. Then, the P probability is extracted from the down-sampled sections. The starting and endpoints for the simulation of the CERWs are derived from the observed white stork trajectory of Wibi 2. In order to allow as much freedom as possible, two lines perpendicular to the direct connection of starting and endpoint of Wibi 2 are drawn with a length of 12 *km* (Figure 4.4). Regarding the width of the Swiss Central Plateau, 12 *km* are an appropriate length so as to obtain an ecologically sound distribution of the white storks. For every simulated CERW a starting and endpoint is randomly sampled from these lines. The number of steps it takes the CERWs to connect the randomly sampled starting and endpoints is estimated using the distance-based approach (Equation 4.16, p. 39). Subsequently, the Q probability is extracted from a

long UERW that is simulated based on the P probability with a time lag of 120 seconds. Given the P and Q probabilities, and starting and endpoints, the simulation of the low-resolution CERWs crossing the Swiss Central Plateau can be conducted.

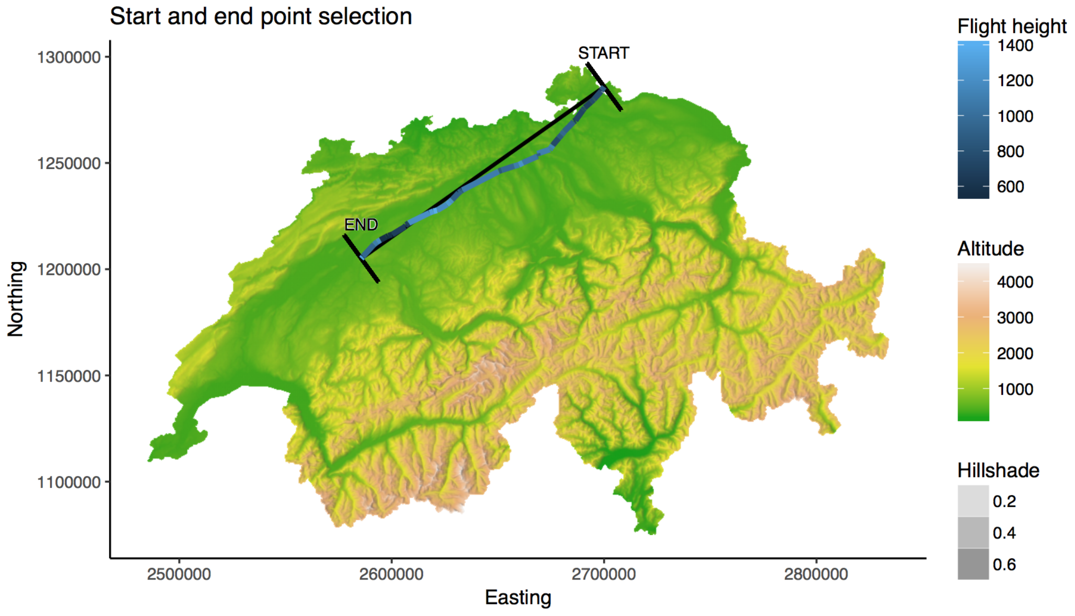


FIGURE 4.4: Random sampling of starting and endpoints for low-resolution simulations from two lines of 12 *km* length, perpendicular to the direct connection of the starting and endpoint of Wibi 2.

### 4.3.3 High-resolution Gliding and Soaring Simulation

To generate the high-resolution simulations on the sections of the low-resolution trajectories within the buffer, the probabilities for the gliding and soaring flight modes must be extracted from the original white stork trajectory. For this purpose, the clean sections of the high-resolution trajectory are further divided into sections of gliding and soaring. All steps that show a positive change in height are classified as soaring steps. Steps with a negative height change are classified as gliding steps. To avoid an extreme fragmentation, a moving window median[24] is used to smooth the resulting classification. The size of the windows was defined in comparison to the number of steps that typical sections consist of. In a first step, the classification is smoothed with a wide median window covering 11 steps, whereby larger gaps in long segments are closed. Since this window size is not sensitive to small gaps in short segments, small gaps in short sections are closed with a further median window covering 5 steps. Now the P probabilities for the gliding and soaring mode are extracted separately from the classified sections.

Then, the glide ratio $\epsilon$, as defined in Equation 4.18 (p. 41), is extracted from the gliding sections. Therefore, the difference in height is divided by the difference in

[24] **Appendix A.16:** eRTG3D: *movingMedian()*

horizontal distance between the starting and endpoint of every gliding section. In order to minimize the influence of outliers, the median of the glide ratios of all gliding sections is taken and set as the glide ratio for the simulation.

The up-sampling to high-resolution trajectories is conducted between each pair of successive fixpoints of the low-resolution trajectory within the buffer. This has the advantage that not the entire distance within the buffer is up-sampled all at once, which drastically reduces the required computing power. Since the number of maximum possible steps between the successive fixpoints is defined in advance, the Q probability is calculated only once and is then reused for the simulations of the gliding and soaring sections. For each section that has to be up-sampled, the number of simulation steps for connecting the successive points is estimated using Equation 4.16 (p. 39). Then, the Q probability is truncated to this number of steps.

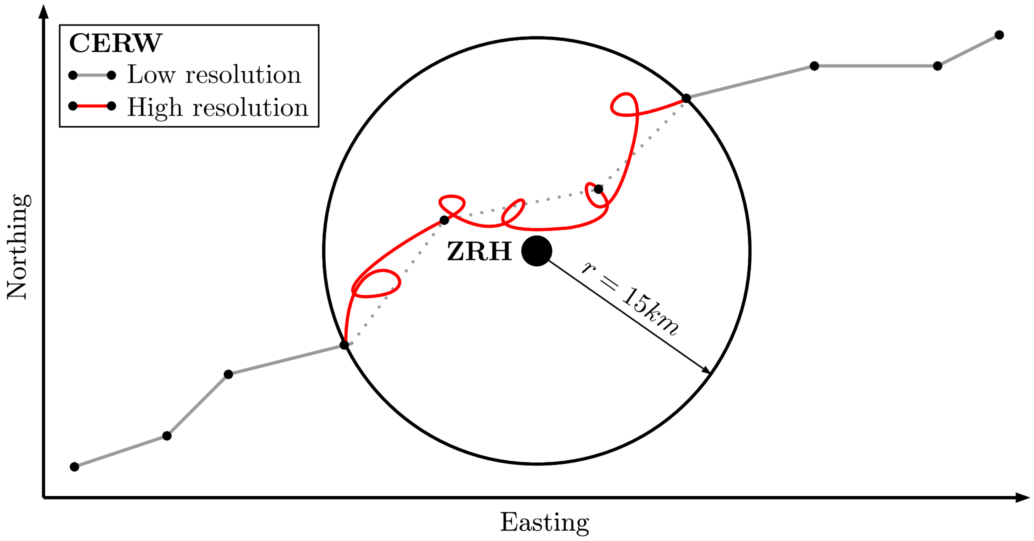


FIGURE 4.5: Up-sampling procedure of a low-resolution trajectory crossing Zurich Airport to a high-resolution gliding and soaring trajectory.

Figure 4.5 illustrates the up-sampling procedure. In a first step, the low-resolution trajectory crossing the Swiss Central Plateau is generated. Afterwards, the fixpoints inside the 15 $km$ buffer around the airport are identified and connected by high-resolution gliding and soaring CERWs. This procedure is repeated until 100 000 white stork trajectories have been simulated in the region of Zurich Airport.

### 4.3.4 High Performance Computing Cluster

The Max Planck Computing and Data Facility hosts several HPC systems, including the Draco cluster, which is used in this case. Most of its compute nodes have a main memory of 128 $GB$ and 64 cores if Hyper Threading (HT) is enabled (Table 4.2). For an efficient simulation of the trajectories on the HPC cluster of the Max Planck Society, the procedure is adjusted in a scalable and modular manner. Since the cluster is accessed via a batch system, the *Slurm Workload Manager*[25], batch jobs have to be defined. These batch jobs are then submitted to the batch queue on the cluster.

| *type* | **nodes** | **cores** | **clock** | **memory** |
|---|---|---|---|---|
| *compute* | 768 | 64 (HT) | 2.3 GHz | 128 GB |
| | 1 | 64 (HT) | 2.3 GHz | 256 GB |
| | 4 | 64 (HT) | 2.3 GHz | 512 GB |
| | 64 | 80 (HT) | 2.2 GHz | 256 GB |
| *gpu* | 102 | 64 (HT) | 2.3 GHz | 128 GB |
| | 4 | 64 (HT) | 2.3 GHz | 256 GB |
| *total* | 943 | 61 376 (HT) | 1.12 PetaFlop/s | 131 072 GB |

Table 4.2: System configuration of the Draco HPC cluster of the Max Planck Society. *Based on: www.mpcdf.mpg.de*

Every batch job has to define the resources and maximum time that it needs beforehand. Based on the amount of resources ordered, the priority is determined in the batch queue. To avoid a low priority in the queue, the generation of the 100 000 trajectories is split into smaller parts. Therefore, only 1 000 trajectories per batch job are simulated. In addition, the batch job is limited to one node of the cluster, where it can use all cores and the total memory of the node. Following from this, each batch job is simultaneously generating 64 trajectories on one node until the total 1 000 are simulated. Due to its small size, the batch job is treated with high priority and gets the required resources assigned faster. To achieve 100 000 trajectories, the batch job is submitted 100 times to the Draco cluster via the Slurm Workload Manager, using the following instructions:

Listing 4.1: Slurm: Instructions for the cluster

```
#SBATCH --job-name=simTracks
#SBATCH --output=log/simTracks_%a.o.log
#SBATCH --time=12:00:00
#SBATCH --nodes=1
#SBATCH --array=1-100
#SBATCH --ntasks=64
#SBATCH --partition=general
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=1024
```

Since the batch job is executed 100 times and thus 100 times the same input data sets are needed for the simulation, it is possible to prepare the data sets beforehand and save them in a single file. This file, which includes the P and Q probabilities for the different resolutions, the uplift suitability map, and the DEM, is loaded into the node's memory whenever a batch job is started on a node.

[25] www.slurm.schedmd.com

### 4.3.5 Collision Probability

Once the high-resolution white stork trajectory simulation and the filtering of the airplane flight trails are completed, UDs for birds and planes are extracted. For this purpose, a voxel space with a voxel resolution of $100 \times 100 \times 100\,m$ is defined for the spatial extent of the $15\,km$ buffer. The height of the voxel space is delimited by the lowest topography value and the maximum flight height observed in the simulated bird trajectories. Since the time lag between the fixpoints in the simulations is constantly 1 second, the total time the bird spends in a voxel can thus be calculated by adding up the fixpoints per voxel[26]. A relative measure for the expected time spent in a voxel is obtained by dividing each voxel count value by the total number of fixpoints in the voxel space. Thus, a probability of occurrence of white storks is calculated for each voxel during fall migration. Since the time lag is also constant in the airplane data set, the same procedure is applied on the flight trails to obtain a probability of occurrence of airplanes around Zurich Airport in August and September of 2017.

The two voxel spaces, which represent the UDs of the white storks and airplanes, have the same anchor point and orientation in space and are therefore congruent with each other. This constellation mathematically allows a multiplication of the two UDs that yields the likelihood of appearing in the same voxel during the same period. Under some simplifying assumptions, as for example a static mean weather and a constant spatial distribution over time, the resulting voxel space represents the collision probabilities of white storks and airplanes at Zurich Airport during fall migration.

[26] **Appendix A.17:** eRTG3D: *voxelCount()*

# Chapter 5

# Results

## 5.1 Empirically Informed Random Trajectory Generator in 3-D

The major contribution of this thesis is the eRTG3D, an algorithm capable of generating empirically informed random trajectories in the 3-D space between two given fixpoints. The trajectory generation is based on the methodology introduced in the preceding chapter and provides robust and traceable simulations that meet the initial requirements. The algorithm has a modular structure, which splits the simulation procedure into smaller parts. This allows the reuse of structures that are needed multiple times for simulations with the same conditions. Furthermore, it makes the algorithm well-extensible and adjustable to complex scientific problems. In contrast, a slimmed-down version of the algorithm can be used for simple applications to maintain a high level of computational efficiency.

The methodology was implemented in R and is available via the R package *eRTG3D*[1]. Beyond the basic functionality of the eRTG3D, the package also provides functions for testing and visualizing the results. Methods that facilitate the preprocessing of the input data, such as the transformation of the CRS in 3-D or the cleaning of outliers from trajectories, extend the package. Furthermore, wrapper functions that combine multiple functions of the algorithm together guarantee an easy *out-of-the-box* use of the algorithm. Finally, the package supports *move* objects from the *move*[2] package and comes with a linkage to the *sf*[3] package. The sf package delivers a framework to encode spatial vector data, the so-called Simple Features (SFs). It binds to *GDAL*[4] for reading and writing spatial data, to *GEOS*[5] for spatial operations, and to *Proj.4*[6] for CRS transformations. The eRTG3D package comes with functions that convert the observed and simulated trajectories to simple features. The resulting *sf, data.frame* objects enable the use of the full functionality of the sf package, which also includes the access to spatial databases, such as *PostGIS*[7].

[1] www.github.com/munterfinger/eRTG3D

[2] www.cran.r-project.org/package=move

[3] www.cran.r-project.org/package=sf

[4] www.gdal.org

[5] www.osgeo.org/geos

[6] www.proj4.org

[7] www.postgis.net

The algorithm has been implemented such that always the same inputs must be provided in advance. Only then, the simulation of trajectories can be conducted. Therefore, a standard procedure has been established. First, the P and Q probabilities are extracted either from suitable trajectories or from clean trajectory sections. Then, the start and end conditions are set. Thus, these structures remain constant and any number of CERWs can be generated. The following block of code illustrates the standard procedure of the algorithm by which a bird's trajectory is reproduced:

LISTING 5.1: Standard procedure for reproducing a trajectory

```r
library(eRTG3D)

trackSections <- track.split.3d(bird, timeLag)
P <- get.section.densities.3d(trackSections, DEM = dem)

n.locs <- nrow(bird)
start <- Reduce(c, bird[1, 1:3])
end <- Reduce(c, bird[n.locs, 1:3])
a0 <- bird$a[1]
g0 <- bird$g[1]

factor <- 1500
UERW <- sim.uncond.3d(n.locs = n.locs*factor, start = start,
                      a0 = a0, g0 = g0, densities = P)

Q <- qProb.3d(UERW, n.locs = n.locs)

CERW <- sim.cond.3d(n.locs = n.locs, start = start, end = end,
                    a0 = a0, g0 = g0, densities = P,
                    qProbs = Q, DEM = dem)
```

To generate more than one CERW (e.g. 100), all available cores of the computer can be addressed with the following code:

LISTING 5.2: Reproduce a trajectory multiple times in parallel

```r
CERWs <- n.sim.cond.3d(n.sim = 100, multicore = TRUE,
                       n.locs = n.locs, start = start, end = end,
                       a0 = a0, g0 = g0, densities = P,
                       qProbs = Q, DEM = dem)
CERWs <- filter.dead.ends(CERWs)
```

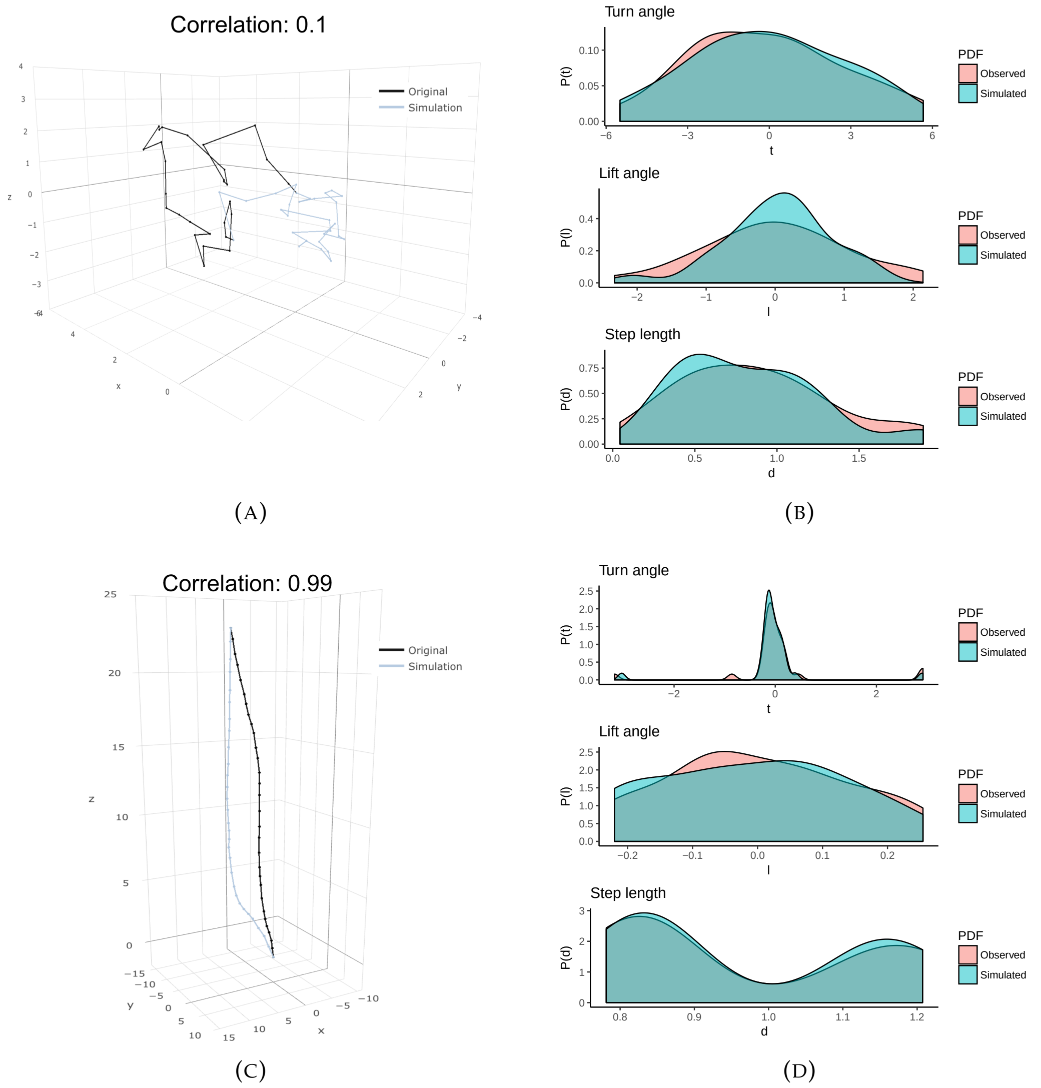
An easier way to get the same results as above is provided by a wrapper function:

LISTING 5.3: Wrapper function for the reproduction of trajectories

```r
CERWs <- reproduce.track.3d(bird, DEM = dem,
                            n.sim = 100, multicore = TRUE)
```

## 5.2 Verification

For the verification of the algorithm, 10 CRWs with correlation values between 0.1 and 0.99 were reproduced 30 times each by the eRTG3D algorithm. The limitation of the bins in the turn-lift-step histogram was deactivated, which means that an infinite number of bins was possible in each dimension. Additionally, the gradient distribution was not incorporated in the generation of the CERWs, since the CRWs were not constrained by any physical limitations and thus unconstrained movement in space was allowed. Furthermore, simulations that have encountered a dead end, were ignored. Figure 5.1 shows two CRWs using a correlation value of 0.1 and 0.99, respectively, with their reproductions by the eRTG3D algorithm and the PDFs of the turning angles, lift angles, and step lengths.



(A) (B) (C) (D)

FIGURE 5.1: CRWs with a correlation of 0.1 and 0.99 and their reproductions by the eRTG3D algorithm. **A)** Correlation of 0.1: 3-D plot, **B)** Correlation of 0.1: Densities, **C)** Correlation of 0.99: 3-D plot, **D)** Correlation of 0.99: Densities.

The reproductions were then tested against the original CRWs by applying the approach described in Section 4.2.1 (p. 43). Table 5.1 lists the p-values of the Kolmogorov–Smirnov tests:

| *cor* | **turning angle** | **lift angle** | **step length** | **auto turn** | **auto lift** | **auto step** |
|---|---|---|---|---|---|---|
| *0,1* | 0,942 | 0,639 | 0,919 | 0,627 | 0,272 | 0,887 |
| *0,2* | 0,968 | 0,223 | 0,942 | 0,638 | 0,132 | 0,369 |
| *0,3* | 0,588 | 0,358 | 0,629 | 0,711 | 0,026 | 0,486 |
| *0,4* | 0,925 | 0,79 | 0,48 | 0,985 | 0,259 | 0,457 |
| *0,5* | 0,993 | 0,32 | 0,568 | 0,994 | 0,125 | 0,476 |
| *0,6* | 0,972 | 0,68 | 0,66 | 0,944 | 0,505 | 0,68 |
| *0,7* | 0,884 | 0,861 | 0,919 | 0,828 | 0,638 | 0,995 |
| *0,8* | 0,919 | 0,518 | 0,991 | 0,828 | 0,791 | 0,741 |
| *0,9* | 0,96 | 0,69 | 0,912 | 0,627 | 0,093 | 0,345 |
| *0,99* | 0,952 | 1 | 0,639 | 0,386 | 0,894 | 0,515 |

TABLE 5.1: p-values of the Kolmogorov–Smirnov tests.

Following the methodology described in Section 4.2.2 (p. 44), the mean pair-wise differences between the distributions of the trajectories were tested against zero (Table 5.2):

| *cor* | **turning angle** | **lift angle** | **step length** |
|---|---|---|---|
| *0,1* | 0,952 | 0,716 | 0,811 |
| *0,2* | 0,813 | 0,621 | 0,566 |
| *0,3* | 0,742 | 0,841 | 0,957 |
| *0,4* | 0,895 | 0,852 | 0,903 |
| *0,5* | 0,966 | 0,531 | 0,759 |
| *0,6* | 0,567 | 0,889 | 0,222 |
| *0,7* | 0,76 | 0,429 | 0,689 |
| *0,8* | 0,264 | 0,998 | 0,643 |
| *0,9* | 0,405 | 0,566 | 0,964 |
| *0,99* | 0,754 | 0,328 | 0,811 |

TABLE 5.2: p-values of the t-tests for the mean differences.

At a significance level of $\alpha = 0.05$, no test results of both approaches are significant, which means that no reproduction is significantly different from its original CRW. To determine a possible trend in the reproduction quality with respect to the correlation of the CRWs, Figure 5.2 was created. It indicates a weak trend in the results of the t-tests, whereby reproductions of CRWs with low correlations tend to have higher p-values. In contrast, the p-values of the Kolmogorov–Smirnov tests do not show a dependency on the correlation values of the initial CRWs. However, the autodifferences distributions of the lift angles have the lowest p-values, while at a high correlation of 0.99 all scores are good.

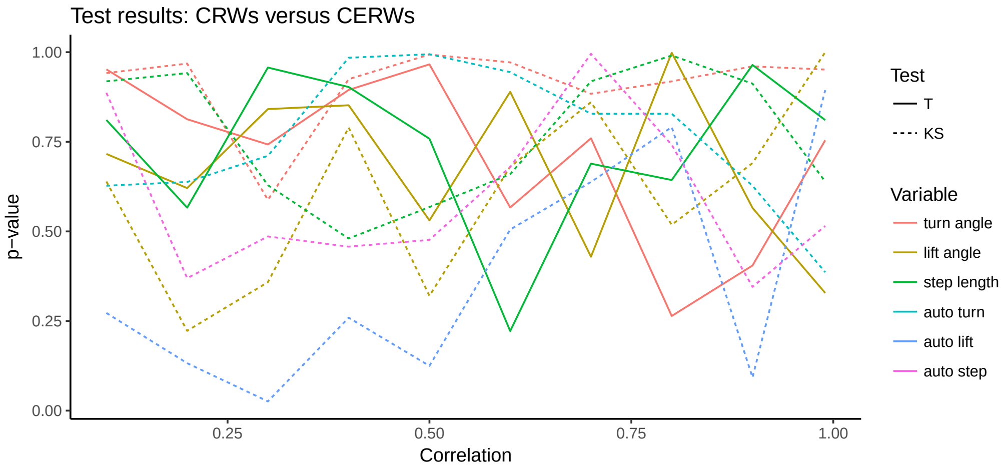


FIGURE 5.2: p-values of the statistical tests, conducted on the distributions of the CRWs versus the CERWs.

In addition, the autocorrelation of the azimuth and gradient was calculated at a lag of one for every CRW and the corresponding CERW. Since each CRW was reproduced 30 times, the mean of the autocorrelation values of the CERWs was taken. Since the CRWs' turning and lift angles were sampled from a wrapped normal distribution, correlated by a concentration parameter, the calculated autocorrelations of the CRWs are not exactly equal to the initially set correlation values for their production. A linear model was fitted to the autocorrelations in the azimuth and gradient. The $r^2$ value is 0.605 for the azimuth and 0.708 for the gradient. This means that approximately two-thirds of the variance in the autocorrelations is explainable.

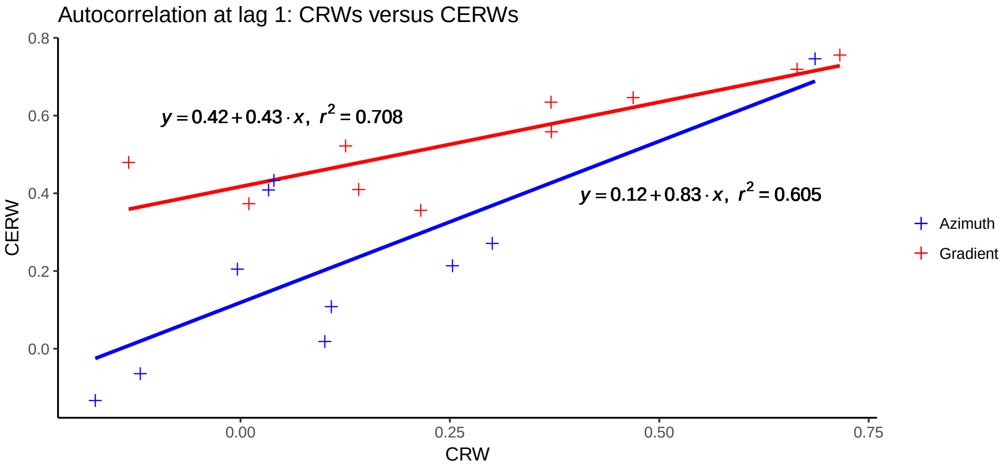


FIGURE 5.3: Linear regressions between the autocorrelation values of the CRWs and the mean correlation of the corresponding 30 CERWs.

## 5.3 Validation

The validation of the eRTG3D algorithm in terms of its biological and ecological validity is based on 30 reproductions of the low-resolution trajectory of the white stork ‘Niclas’ by the eRTG3D algorithm. Niclas traversed the Swiss Central Plateau during his migration northwards. As with the verification of the algorithm in the previous section, the simulation of the trajectories allowed an unlimited number of bins in each dimension of the turn-lift-step histogram. Trajectories that did not arrive at the destination were ignored. Figure 5.4 shows the original trajectory together with the simulations in their topographical environment. Each trajectory consists of 31 fixpoints.

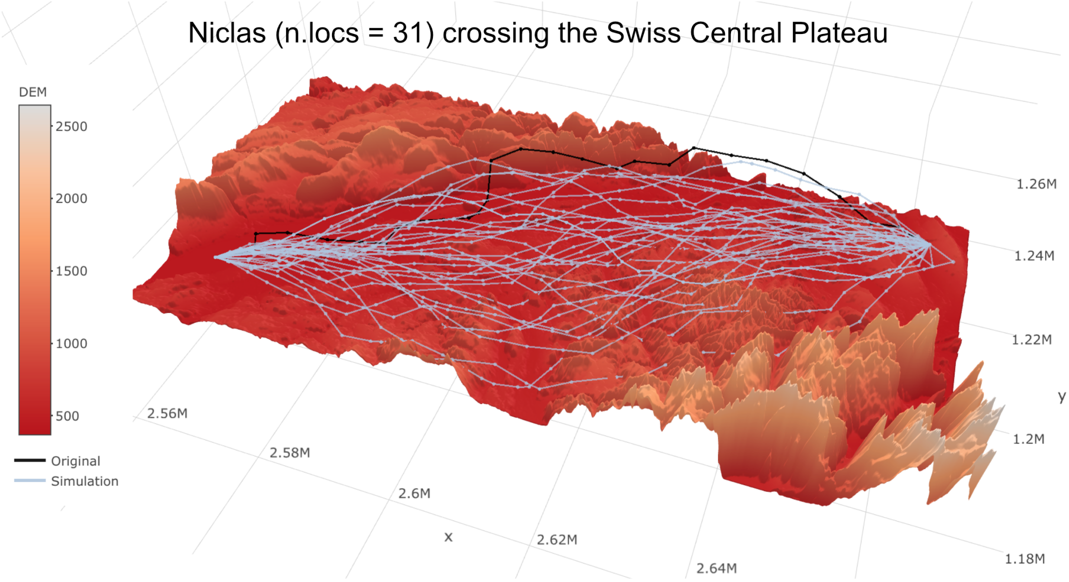


FIGURE 5.4: White stork Niclas crossing the Swiss Central Plateau (black line), was reproduced 30 times by the eRTG3D algorithm (light blue lines).

Most of the simulated trajectories are concentrated around the path of Niclas and remain between the Jura Mountains in the north and the Alps in the south. However, a few trajectories also use valleys in the southern foothills of the Alps for their northward-directed migration. In addition, there are some rare cases in which the simulated trajectory passes through mountain ranges, which is not possible. In general, the flight height of Niclas varies slightly more than in most simulated trajectories. Therefore, also the distribution of the gradient seems to vary less within the simulations, which makes them appear smoother.

As the simulation of the trajectories involved the distributions of the gradient and the absolute, as well as the relative flight height of Niclas, their properties should be reflected in the simulations. Figure 5.5 plots the densities of gradient, relative and absolute flight height of the original trajectory in comparison to the simulated trajectories. It is important to note that the densities were obtained from samples of different sizes. The simulations offered 30 times more fixpoints for the extraction of the densities.

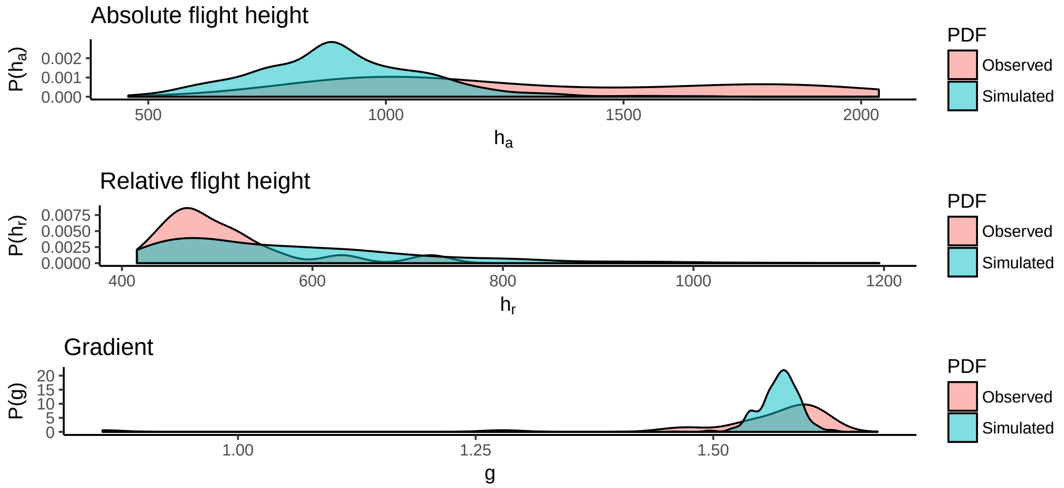


FIGURE 5.5: Densities of gradient, relative and absolute flight height of Niclas' trajectory and the simulated trajectories.

The density plots indicate that, in the simulations, the lower absolute heights are overrepresented and the lower relative heights are underrepresented. The density of the gradient of the simulations is narrower and has the mode (most frequently occurring value) at a slightly smaller value than the gradient density of Niclas.

## 5.4 Sensitivity Analysis

The sensitivity analysis of the maximum bin parameter was also conducted on the movement behavior of Niclas. For different limits of the maximum number of bins in the turn-lift-step histogram, the trajectory of Niclas was reproduced 30 times by the eRTG3D algorithm.

| *maxBin* | **turning angle** | **lift angle** | **step length** | **auto turn** | **auto lift** | **auto step** |
|---|---|---|---|---|---|---|
| *12* | 0,858 | 0,493 | 0,916 | 0,499 | 0,171 | 0,869 |
| *15* | 0,874 | 0,325 | 0,991 | 0,68 | 0,091 | 0,952 |
| *25* | 0,903 | 0,228 | 0,916 | 0,619 | 0,044 | 0,985 |
| *50* | 0,874 | 0,058 | 0,866 | 0,639 | 0,001 | 0,96 |
| *100* | 0,833 | 0,124 | 0,881 | 0,721 | 0,002 | 0,899 |
| *150* | 0,7 | 0,15 | 0,934 | 0,629 | 0,002 | 0,968 |
| *250* | 0,797 | 0,098 | 0,866 | 0,731 | 0,001 | 0,931 |
| *500* | 0,72 | 0,098 | 0,797 | 0,731 | 0,001 | 0,919 |
| *Inf* | 0,7 | 0,095 | 0,903 | 0,608 | 0,001 | 0,936 |

TABLE 5.3: Sensitivity analysis of the maximum number of bins: p-values of the Kolmogorov–Smirnov tests.

The same statistical tests that have been used in the verification process of the algorithm were applied in the sensitivity analysis comparing the simulations with the original trajectory of Niclas. The p-values of the tests of both approaches are listed in Tables 5.3 and 5.4, respectively. At a significance level of $\alpha = 0.05$, only the Kolmogorov–Smirnov tests of the autodifferences of the lift angle are significant for reproductions that are based on turn-lift-step histograms that allow more than 25 bins per dimension. Furthermore, the tests of the lift angle have rather low p-values, but they are not significant. Most of the other p-values are close to 1.

| ***maxBin*** | **turning angle** | **lift angle** | **step length** |
|---|---|---|---|
| *12* | 0,532 | 0,861 | 0,592 |
| *15* | 0,922 | 0,371 | 0,711 |
| *25* | 0,798 | 0,947 | 0,192 |
| *50* | 0,889 | 0,93 | 0,315 |
| *100* | 0,9 | 0,916 | 0,944 |
| *150* | 0,929 | 0,97 | 0,347 |
| *250* | 0,501 | 0,994 | 0,46 |
| *500* | 0,726 | 0,774 | 0,88 |
| *Inf* | 0,832 | 0,826 | 0,739 |

TABLE 5.4: Sensitivity analysis of the maximum number of bins: p-values of the t-tests for the mean pair-wise difference.

In general, the p-values of the t-tests are higher than the p-values of the Kolmogorov–Smirnov tests. Furthermore, no t-test result is significant. In contrast to the first approach, in which the p-values of the step lengths are the highest, the p-values of the step lengths in the second approach tend to be lowest.

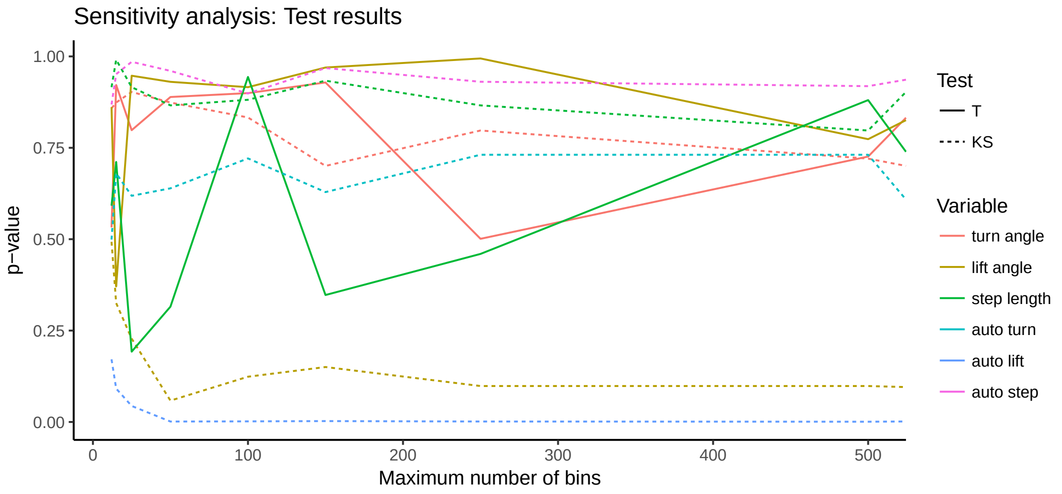
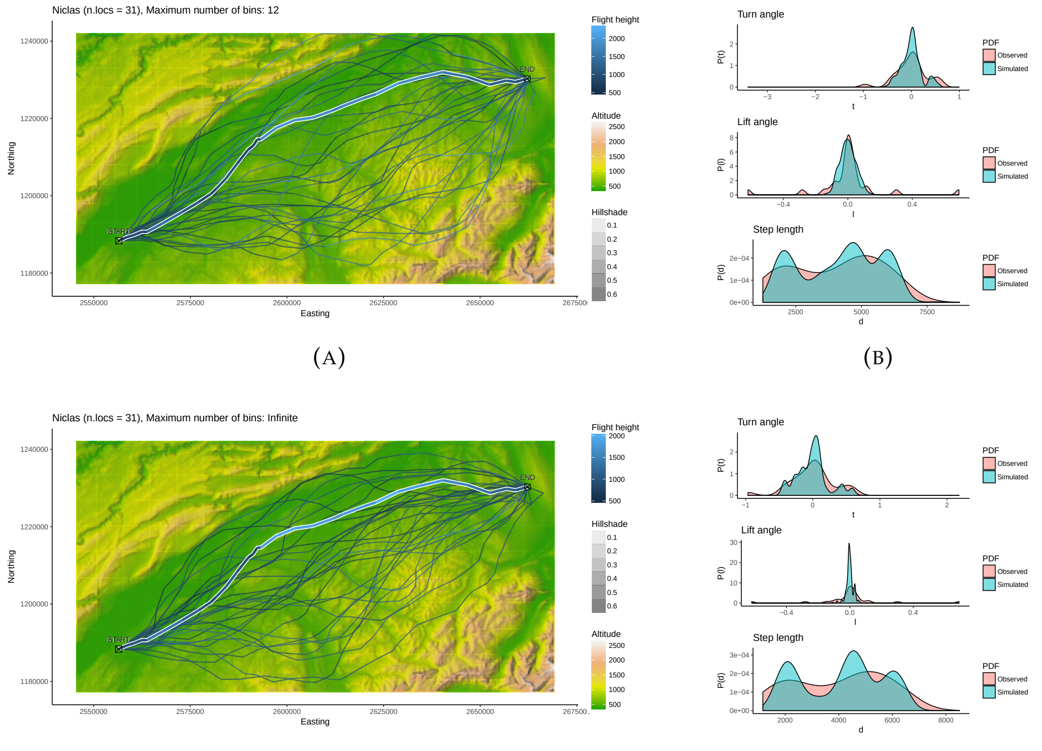


FIGURE 5.6: p-values of the sensitivity analysis, conducted on the distributions of the observed trajectory versus the CERWs, with varying maximum number of bins allowed in the turn-lift-step histogram.

In Figure 5.6, the results of the t-tests indicate the trend that the values tend to be closer to one for high numbers of maximum possible bins. This trend is less pronounced in the results of the Kolmogorov–Smirnov tests; and the trend is even reversed for the p-values of the lift angles and the corresponding autodifferences, where the p-values are sometimes even significant ($\alpha = 0.05$) for a large number of maximum possible bins. Besides the p-values of the lift angles and the corresponding autodifferences, the results of the Kolmogorov–Smirnov tests seem to be less affected by the number of allowed bins.



(A) (B)

(C) (D)

FIGURE 5.7: A limit of maximum 12 bins per dimension versus an unlimited number of bins. **A)** Maximum of 12 bins: 2-D plot, **B)** Maximum of 12 bins: Densities, **C)** Infinite number of bins possible: 2-D plot, **D)** Infinite number of bins possible: Densities.

The visual comparison of the simulations with a maximum limit of 12 bins per dimension with simulations based on an unlimited turn-lift-step histogram shows that the spatial variance is stronger in simulations with the lower bin limitation. These trajectories are distributed more widely in space and follow the original trajectory of Niclas (white deposited line in Figure 5.7 A and C) less closely. Furthermore, it can be observed that the variation at the beginning of the simulations, that is near the starting point, is significantly greater. In addition, the simulations with an unlimited histogram seem to better follow the topography, as they do not traverse the Jura Mountains in the north, and as they follow the course of valleys in the south more precisely.

## 5.5 Bird-strike Probability at Zurich Airport

In this section, the results of the demonstrator use case of the eRTG3D algorithm are presented. All the intermediate steps, which were necessary to eventually obtain the joint collision probability of migrating white storks and airplanes at Zurich Airport in fall, are outlined by various figures and tables.

### 5.5.1 P and Q Probabilities of the Different Modes

In order to extract the P and Q probabilities for the low-resolution simulations, the trajectory of Wibi 2 was split into clean sections. Thereby, a time lag of one second was ensured with a tolerance of 0.05 seconds. The clean sections were then down-sampled to a time lag of 120 seconds. From these down-sampled sections, the P probability was extracted. Then, a UERW, 1500 times longer than the trajectory of Wibi 2, served as input for the calculation of the Q probability.

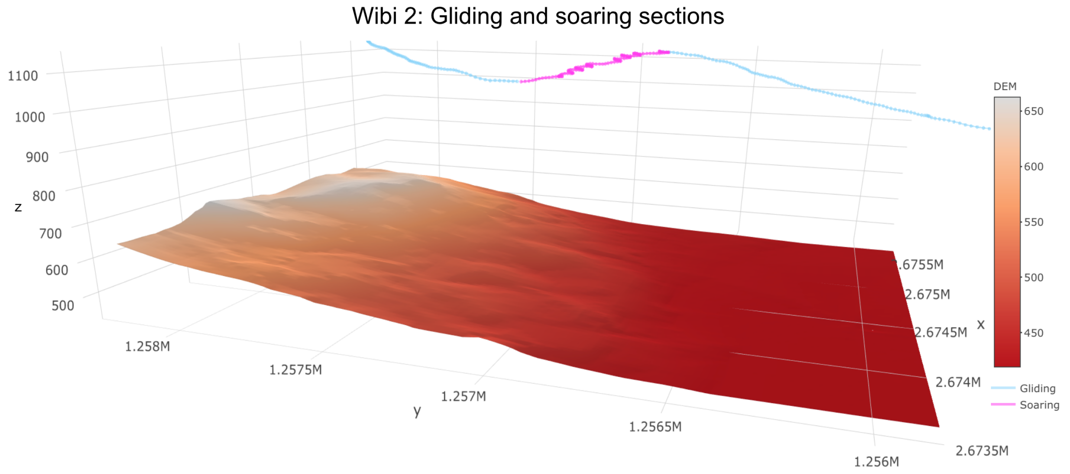


FIGURE 5.8: Gliding (light blue line) and soaring (pink line) classification of a section of the trajectory of Wibi 2.

Starting again from the trajectory of Wibi 2, the clean high-resolution sections further were split into gliding and soaring sections. Therefore, the steps were classified based on positive or negative change in height into gliding or soaring steps. Afterwards, the moving median approach, described in Section 4.3.3 (p. 47), was applied on the binary classification. The classification result was visually checked and 3 gliding and 7 soaring sections were omitted, since they were either too short or did not reflect a clear flying behavior. Figure 5.8 shows a clean example section of the trajectory of Wibi 2 and its classification into gliding and soaring.

From the classified sections, turn-lift-step histograms were extracted for the gliding and soaring mode, respectively. Illustrations of these histograms are shown in

Figures 5.9 and 5.10, whereby their tld-cubes were mapped to the spatial dimensions, starting from the origin $(0,0,0)$. While the turn-lift-step histogram of the soaring mode is strongly scattered with an upward directed tendency, the movement in gliding mode is more focused towards the front and the step length is much longer.

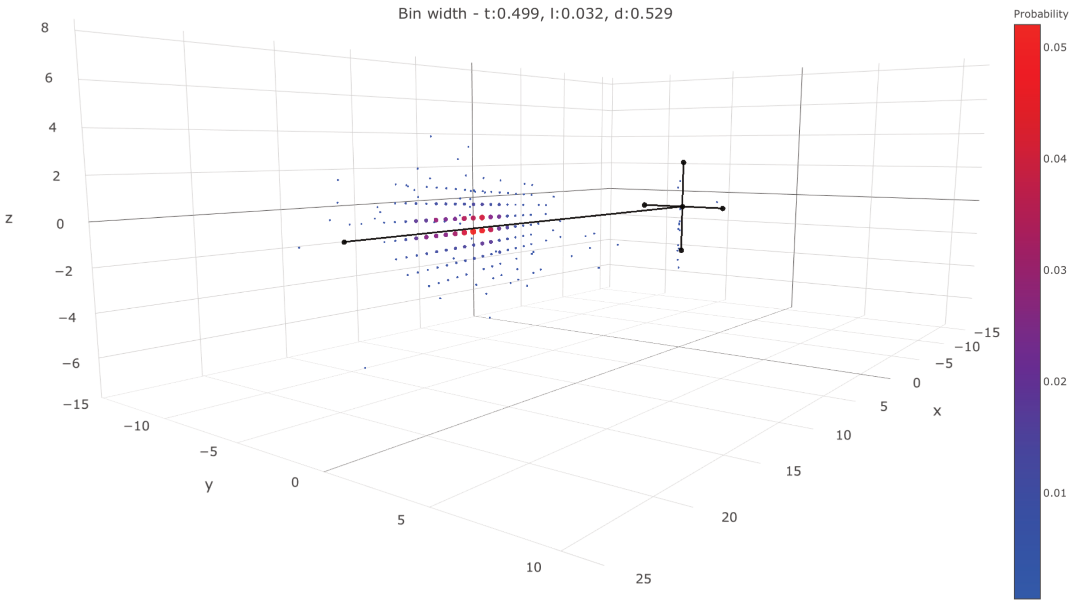


FIGURE 5.9: Turn-lift-step histogram, representing the gliding mode.

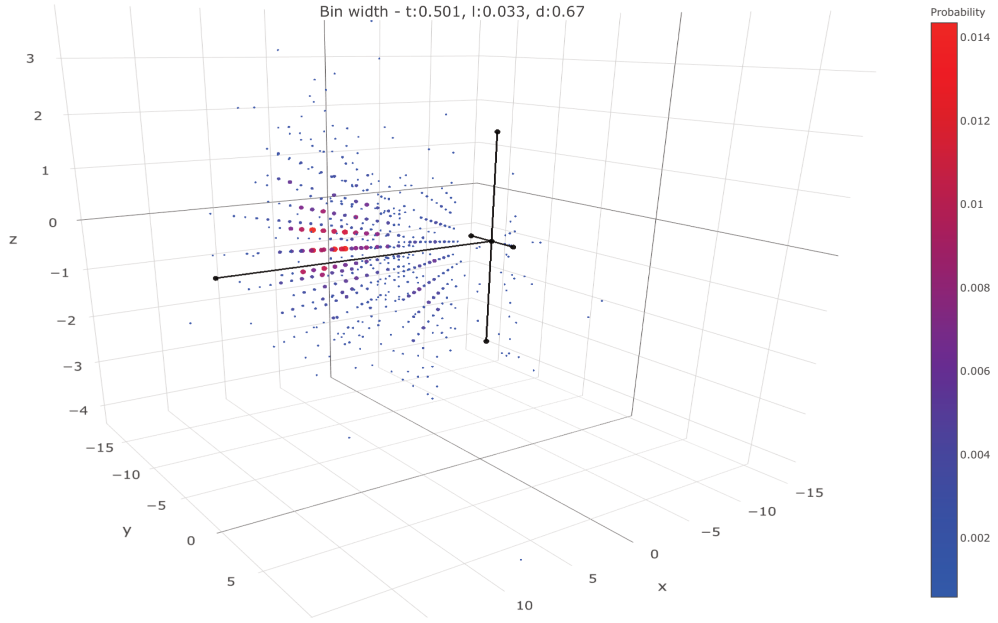


FIGURE 5.10: Turn-lift-step histogram, representing the soaring mode.

Furthermore, the density approximations of the gradient, as well as absolute and relative flight heights, were gathered for the two modes. In Figure 5.11 the difference in the distribution of the gradient between the two modes becomes obvious. Since an angle of $\frac{\pi}{2}$ represents an exactly horizontal flight, this threshold nicely separates the two densities. The gradient in soaring mode varies more strongly than in gliding mode, because soaring depends on the encountered uplift conditions, while the gliding behavior is more independent from the environment and therefore more constant.

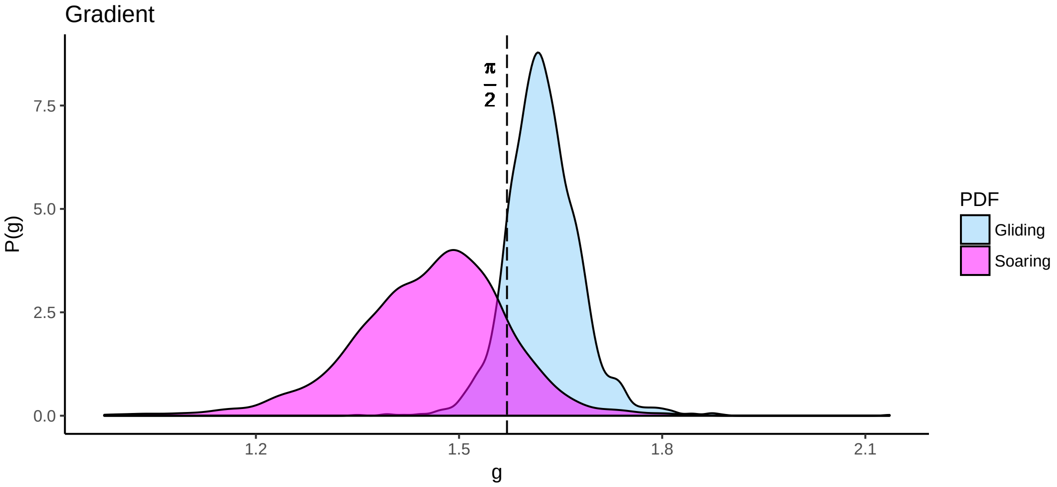


FIGURE 5.11: Gradient densities of gliding (light blue) and soaring (pink) mode, extracted from the classified sections.

As Table 5.5 indicates, the approach of the destination mostly takes place in gliding flight mode, with a mean step towards the target value of −16.88. Further, the negative median glide ratio of −5.05 in soaring mode quantifies the height gain, which means Wibi 2 gains 1 *m* of height per 5 *m* horizontal movement towards the target. In contrast the bird converts 1 *m* of height loss into almost 18 *m* of horizontal movement.

| *mode* | **sections** | **steps** | **mean dz** | **mean step2target** | **median glide ratio** |
|---|---|---|---|---|---|
| *gliding* | 25 | 2218 | -0,98 | -16,88 | 17,67 |
| *soaring* | 20 | 1764 | 1,19 | -3,99 | -5,09 |

TABLE 5.5: Gliding and soaring classification statistics.

To assess the extracted P probabilities, a gliding and a soaring section was reproduced that are shown in Figures 5.12 and 5.13, respectively. Thereafter, the simulations were tested statistically against the original sections. Except for the step length, no significant deviation occurred[8].

[8] **Appendix C.1 and C.2**: p-values of the tests

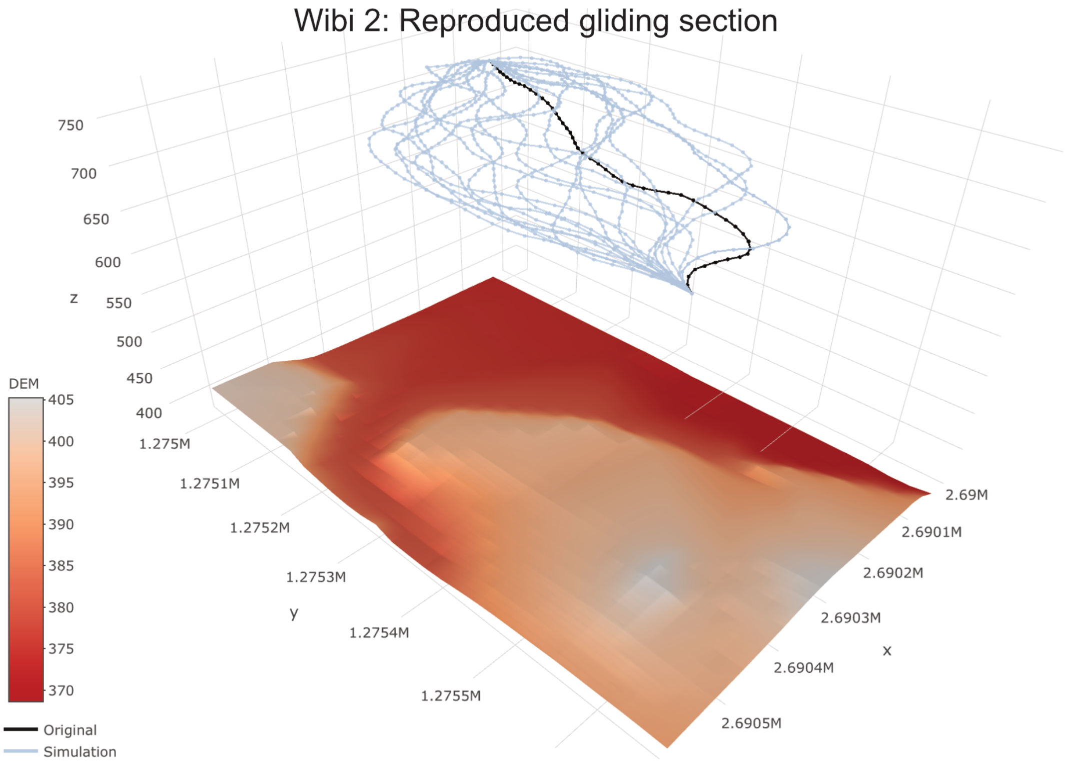


FIGURE 5.12: Reproductions (light blue lines) of a gliding section (black line).

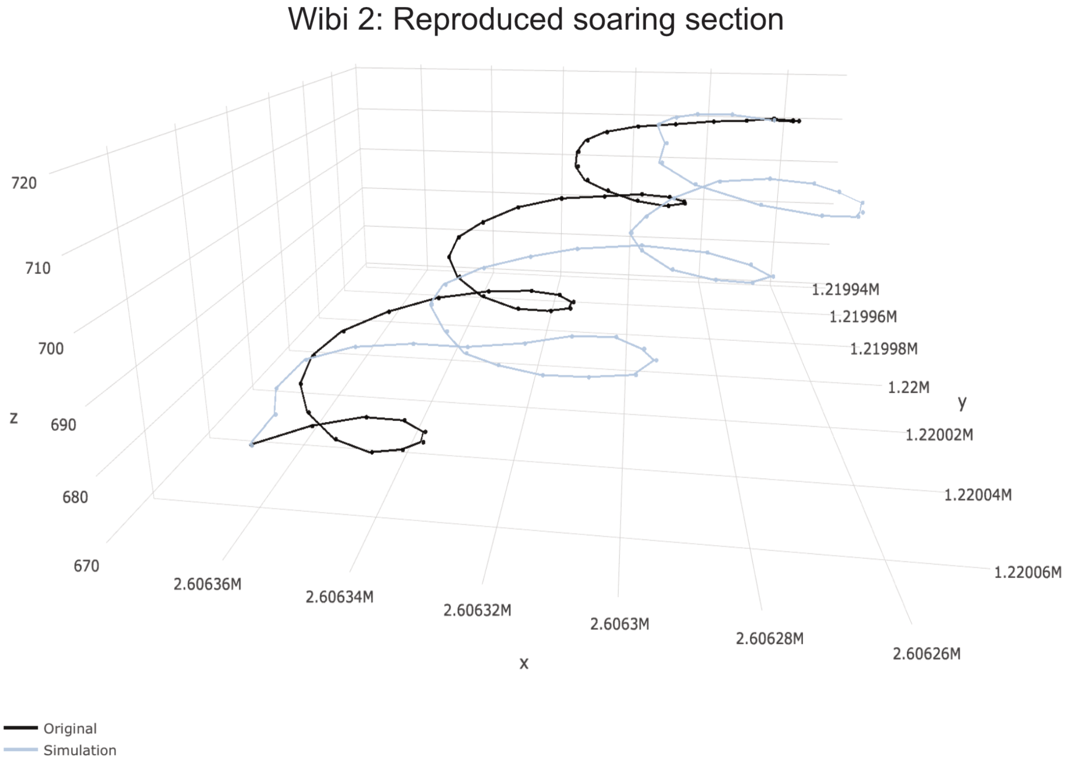


FIGURE 5.13: Reproduction (light blue line) of a soaring section (black line).

### 5.5.2 Gliding and Soaring in One Model

Given stable probabilities, the two modes were tested combined in one model. Therefore, a trajectory section of Wibi 2 (white deposited line in Figure 5.14), consisting of one soaring and two gliding sections, was reproduced 9 times based on the two modes and a binary uplift background layer. Due to the tendency of approaching the target in gliding mode, the Q probability was extracted from an UERW that was simulated based on the P probability of the gliding mode.

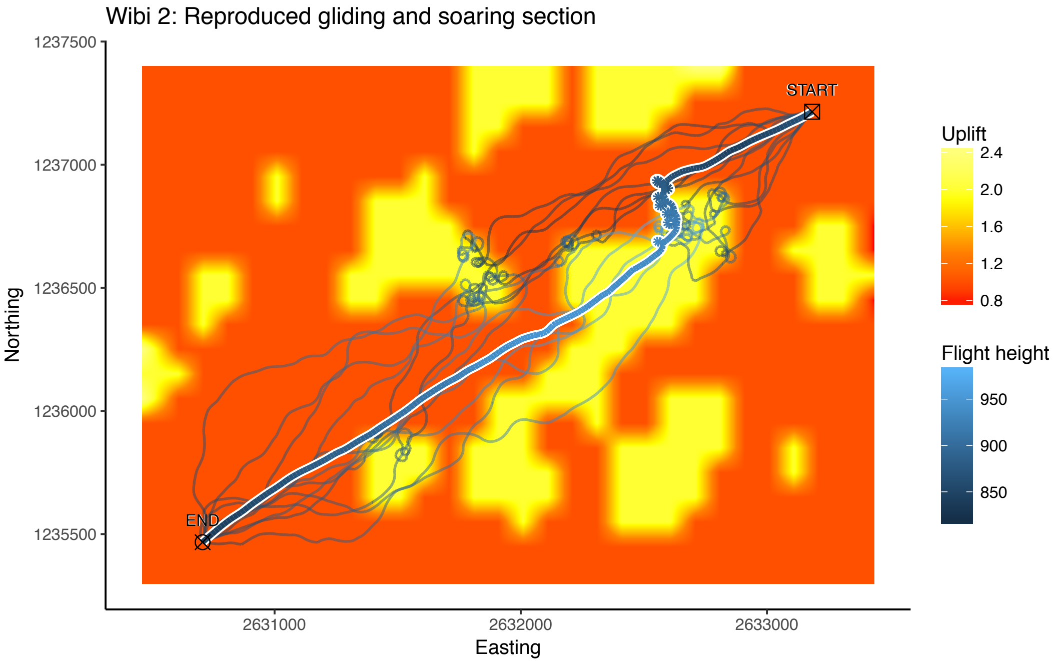


FIGURE 5.14: 2-D plot of the gliding and soaring reproductions and the uplift suitability map (yellow = uplift, red = no uplift).

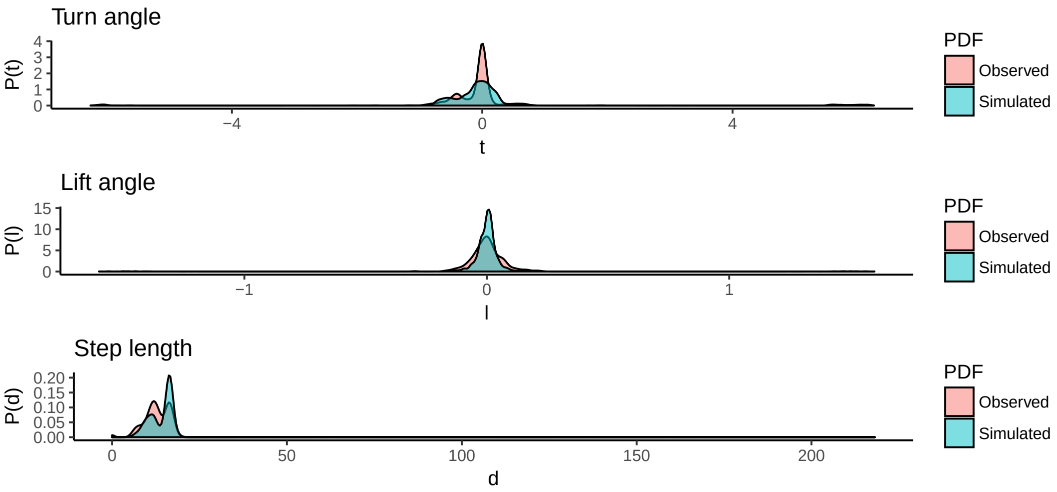


FIGURE 5.15: Densities of the gliding and soaring reproductions, and original trajectory, respectively.

About 70 percent of the simulated CERWs encountered a dead end. This rate is significantly higher than the dead end rate of the algorithm version that simulates trajectories based on one movement mode, which is usually about 15 percent.

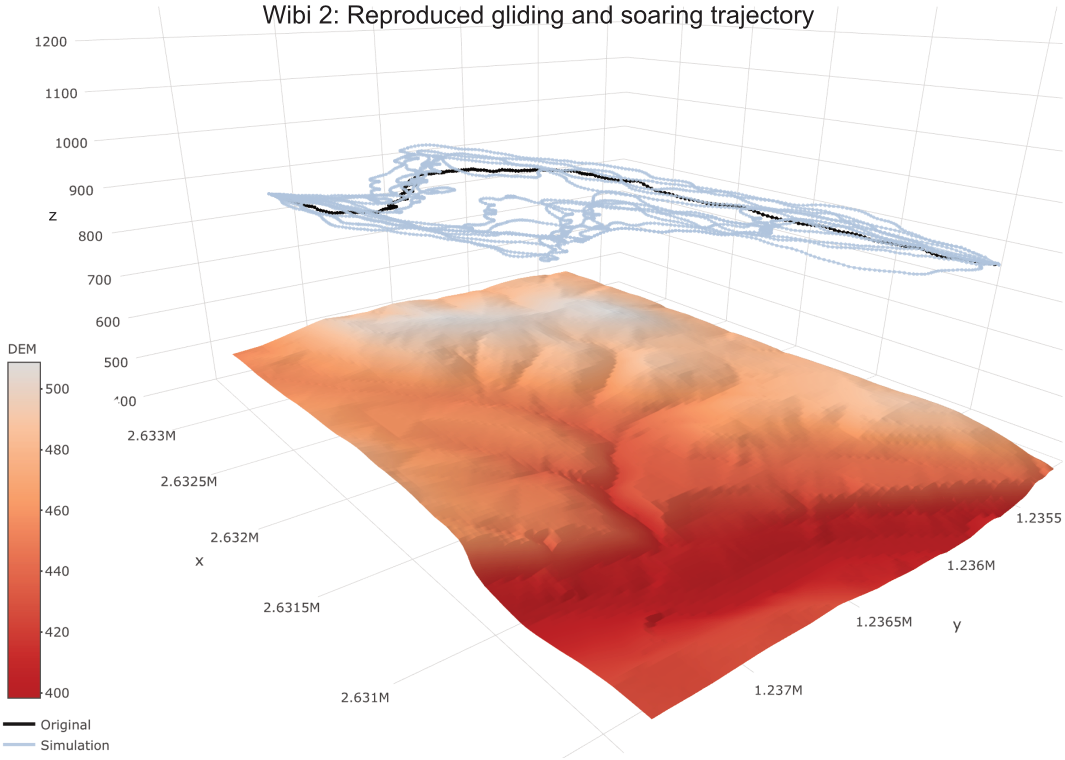


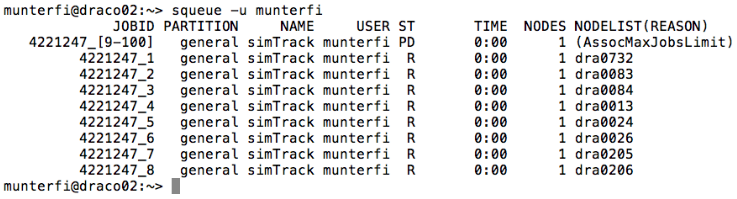
FIGURE 5.16: 3-D plot of the gliding and soaring reproductions (light blue lines) and the original trajectory section (black line).

Figure 5.16 shows the original and the simulated trajectories in 3-D. It is well recognizable that the simulated birds use the uplift areas to gain height until they have enough potential energy accumulated to reach the target. They also ascend about to the same height as the original trajectory and follow its path. There are also CERWs that use the uplift areas several times, because they could not gain enough height on the first climb.

After this successful combination of gliding and soaring behavior in one model, the Q probability for the maximum number of gliding steps in the final simulations was calculated. Then, all input data sets for the simulation were exported to an *RData* file for the cluster. In addition to the P and Q probabilities for the various modes, the file also contains the raster files of the DEM and the binary uplift suitability map. Furthermore, the two vectors for the random sampling of the starting and endpoints of the low-resolution simulation and the buffer around Zurich Airport are included. Finally, parameters such as the glide ratio or the mean step towards the target in gliding mode are part of the file. With these preparatory steps in place, the large-scale simulations on the cluster can be performed.

### 5.5.3 Simulations on the HPC Cluster

As soon as all input data sets were prepared for the cluster, the simulations on the HPC cluster could begin. The simulation batch job[9] was submitted 100 times to the Draco cluster via the Slurm Workload Manager, by the script shown in Listing 4.1 (p. 49). Since the requested times and resources of the submitted jobs were rather small, they were treated with high priority in the queue of the cluster. Figure 5.17 shows the simulation jobs in the queue of the account *munterfi* that was used.



```
munterfi@draco02:~> squeue -u munterfi
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
     4221247_[9-100]   general simTrack munterfi PD       0:00      1 (AssocMaxJobsLimit)
         4221247_1   general simTrack munterfi  R       0:00      1 dra0732
         4221247_2   general simTrack munterfi  R       0:00      1 dra0083
         4221247_3   general simTrack munterfi  R       0:00      1 dra0084
         4221247_4   general simTrack munterfi  R       0:00      1 dra0013
         4221247_5   general simTrack munterfi  R       0:00      1 dra0024
         4221247_6   general simTrack munterfi  R       0:00      1 dra0026
         4221247_7   general simTrack munterfi  R       0:00      1 dra0205
         4221247_8   general simTrack munterfi  R       0:00      1 dra0206
munterfi@draco02:~> █
```

FIGURE 5.17: Screenshot of the simulation batch jobs in the queue on the Draco cluster.

Approximately three days later, 100 000 high-resolution gliding and soaring trajectories in the 15 *km* buffer around Zurich Airport were generated. They were saved in portions of 1 000 trajectories each, as *RData* files to the file system of the cluster. The size of all simulated trajectories together is nearly 9 *GB*.

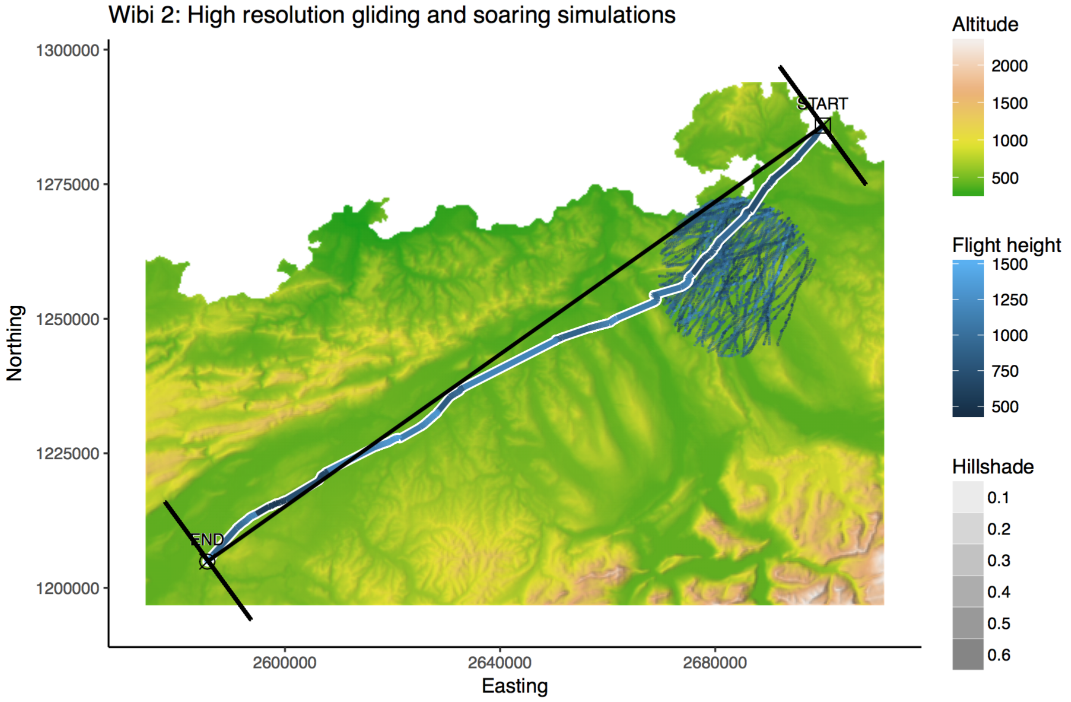


FIGURE 5.18: 100 high-resolution gliding and soaring simulation examples in relation to the original trajectory of Wibi 2.

[9] **Appendix A.18**: *simTracks.R*

Figure 5.18 places 100 example simulations and the original trajectory of Wibi 2 into their spatial context of the Swiss Central Plateau. The black lines in the figure indicate the locations from which the starting and endpoints were sampled. As soon as a low-resolution simulation has crossed the buffer, the up-sampling to high-resolution gliding and soaring trajectories began.

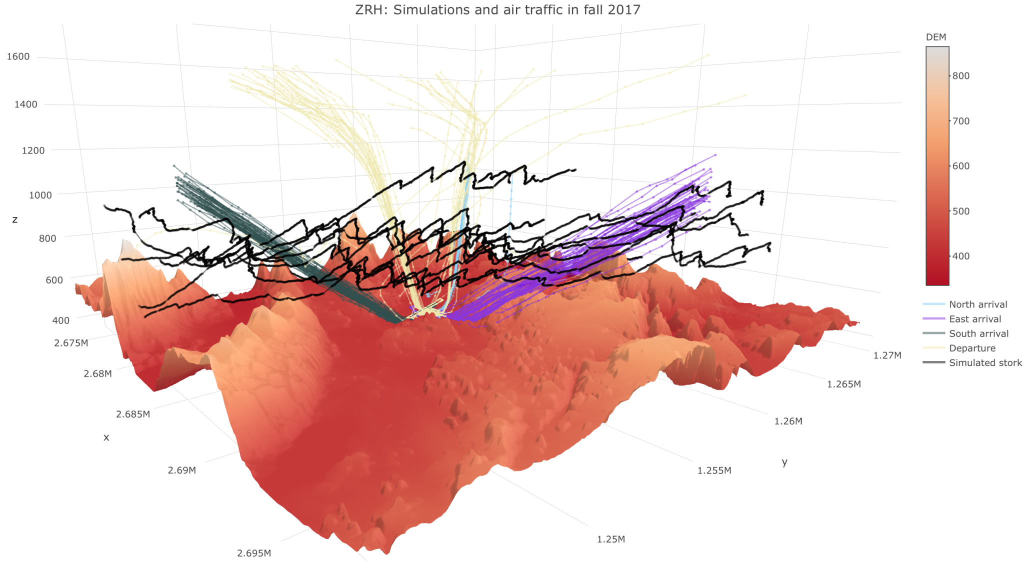


FIGURE 5.19: 10 simulation examples and a subset of 50 flight trails per flight corridor at Zurich Airport.

The statistical comparison of 1 000 random samples of the 100 000 simulations with the original trajectory revealed significant deviations for all Kolmogorov–Smirnov tests, while only the distributions of the step lengths differed significantly in the t-tests[10].

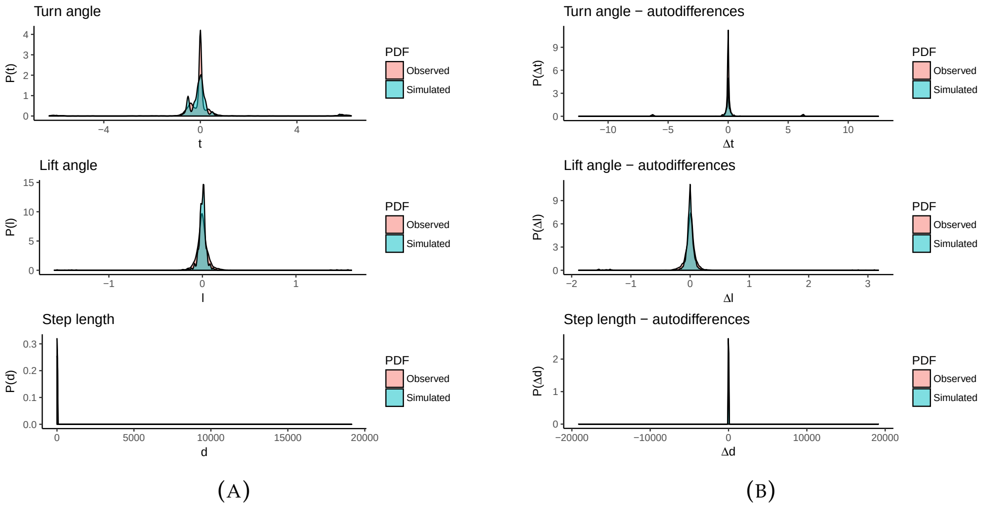


FIGURE 5.20: Densities (**A**) and autodifferences densities (**B**) of the high-resolution gliding and soaring simulations.

[10] **Appendix C.3**: p-values of 1 000 reproduced gliding and soaring CERWs

### 5.5.4 Joint Utilization Distribution

Based on a voxel space of 100 × 100 × 100 *m*, the UDs of airplanes and white storks were calculated by counting the fixpoints per voxel, followed by a subsequent standardization over the entire number of fixpoints. Then, the two voxel spaces were multiplied. The resulting voxel space represents the probability of a common occurrence per voxel in the 3-D space, and therefore is a *joint* UD of birds and airplanes. Figures 5.22, 5.23 and 5.24, respectively, show the voxel spaces as horizontal image slices in bands of 100 *m* altitude each. The lower and upper bounds of each band are recorded in the title of each image slice (e.g. *m500.600* designates the height band between 500 and 600 *m* altitude).

The simulated white storks fly mainly at altitudes between 700 and 1000 *m*. They spend more time in areas with good uplift conditions. Since the flight of the birds is directed towards the south, the northern edges of the areas with good uplift conditions are preferably used for soaring. In total, the probability of occurrence is highest along the northern edges at altitudes between 800 and 900 *m*. In the UD of the airplanes, the individual flight corridors are clearly recognizable. The northern corridor is strongly favored for arrivals, while the western direction is preferred for departures. The departure corridors are spatially more spread out than the arrival corridors, whereby the spread increases above an altitude of 1000 *m*. However, the arrival corridors proceed at a shallower angle than the departure corridors. The joint UD indicates the highest collision probabilities in the northern arrival corridor between 600 and 900 *m* altitude. The western departure corridor also has a higher collision probability at altitudes between 800 and 1100 *m*.

Figure 5.21 shows a 3-D rendering of the overlaid UDs of the birds (yellow) and the airplanes (blue), and the collision probability (red) at Zurich Airport.

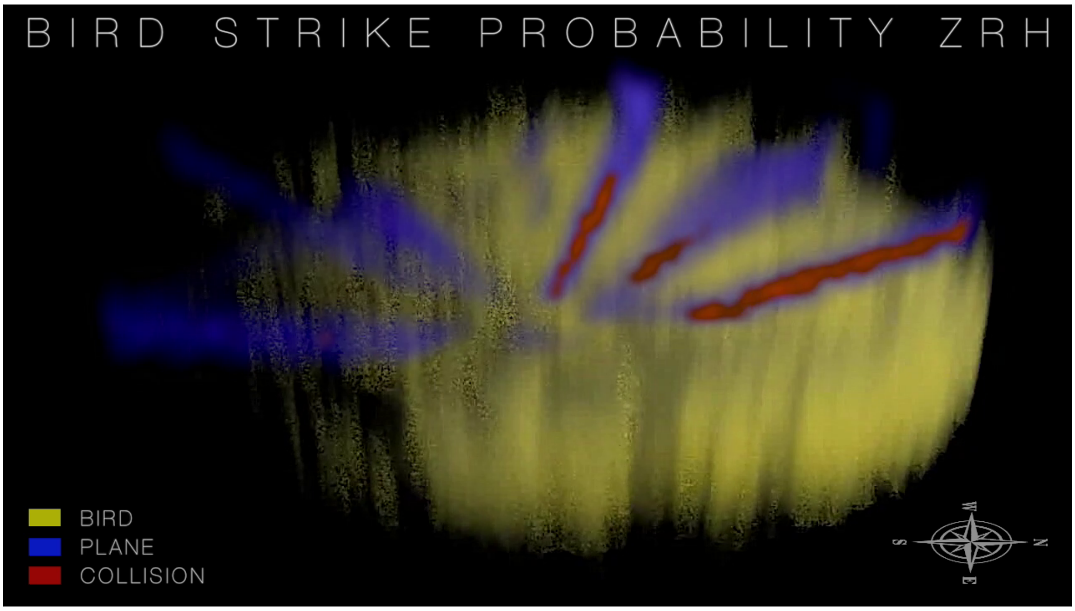


FIGURE 5.21: 3-D rendering of the bird-strike probability.

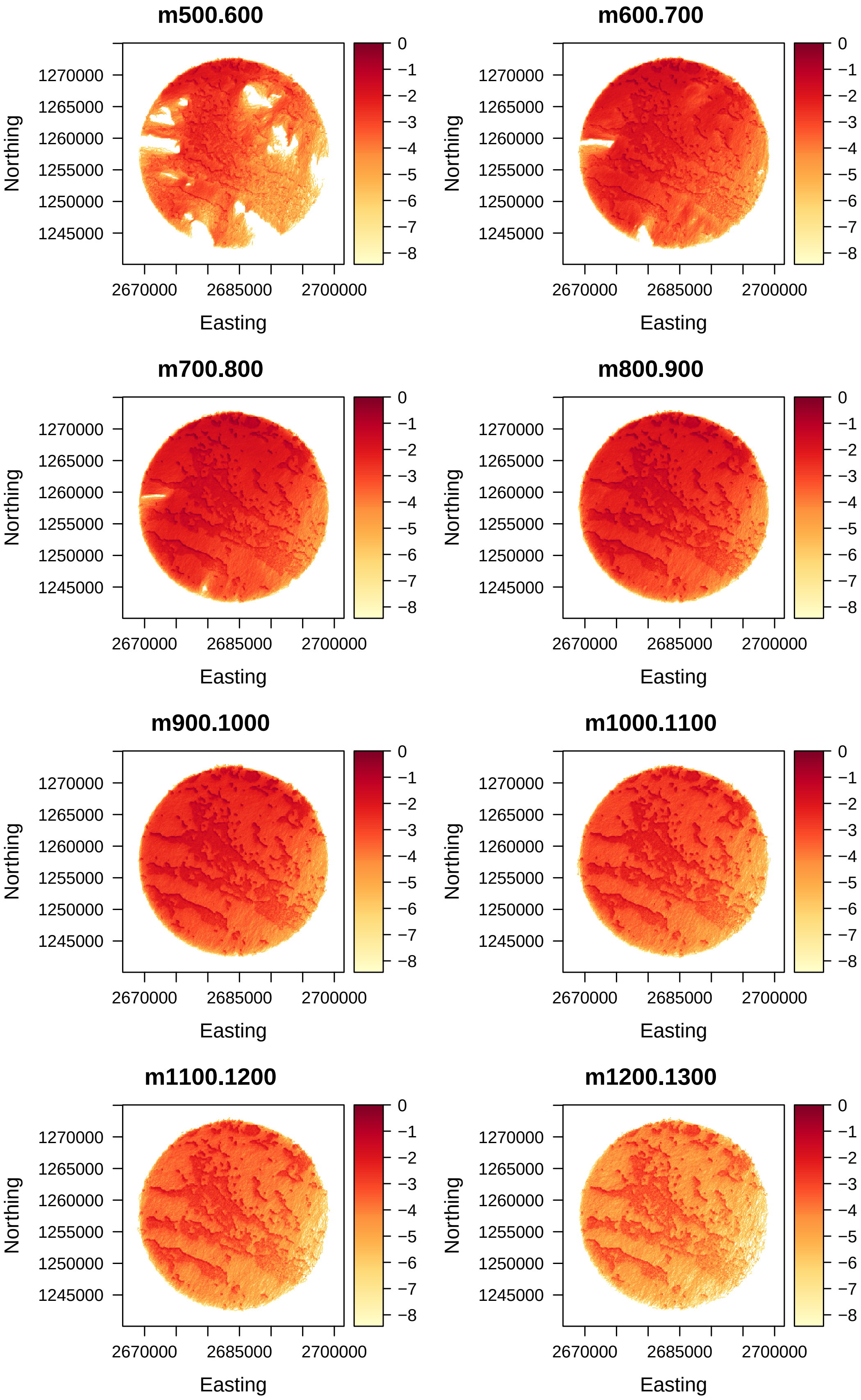


FIGURE 5.22: UD of white storks. Scale: logarithmic.

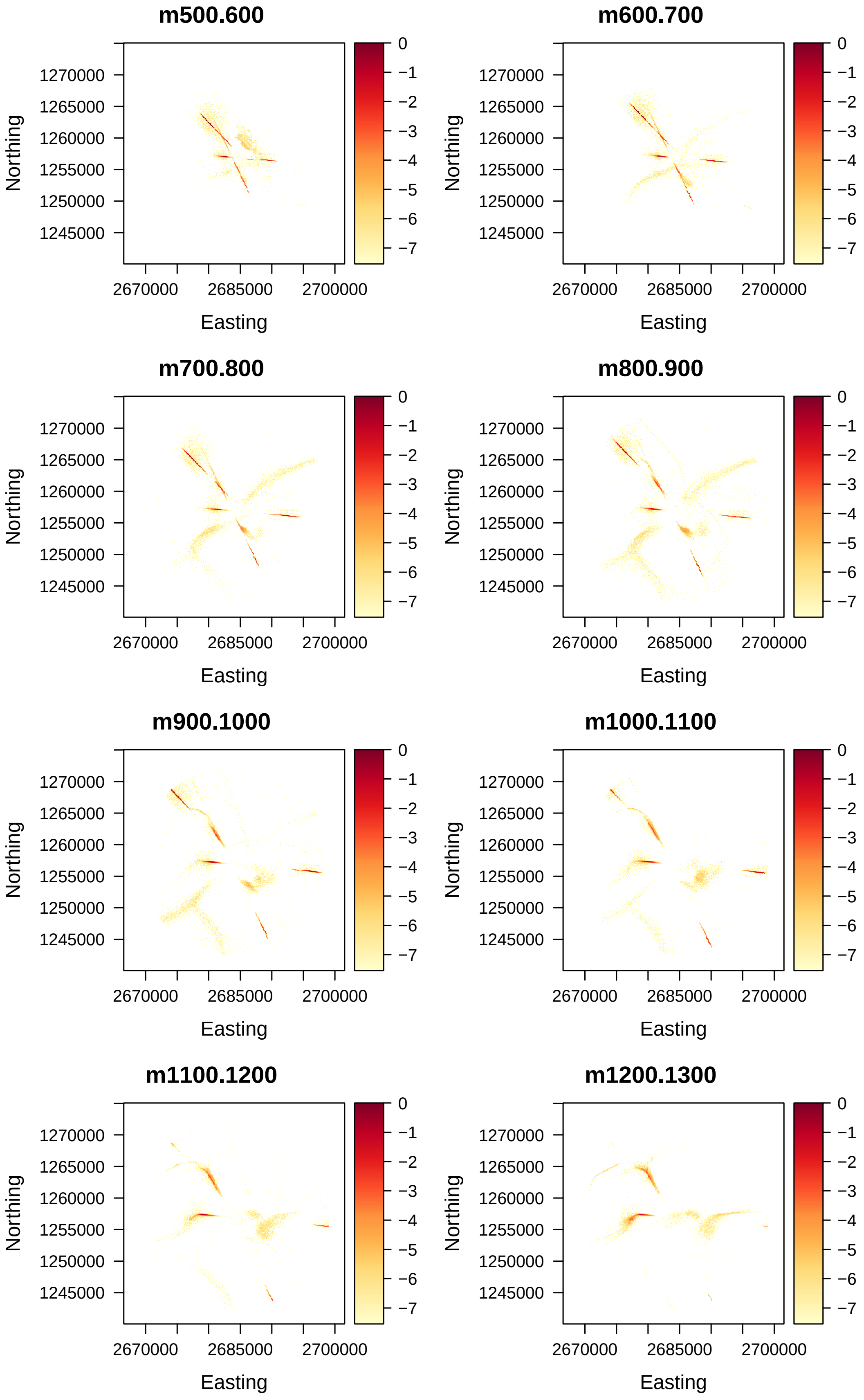


FIGURE 5.23: UD of airplanes. Scale: logarithmic.

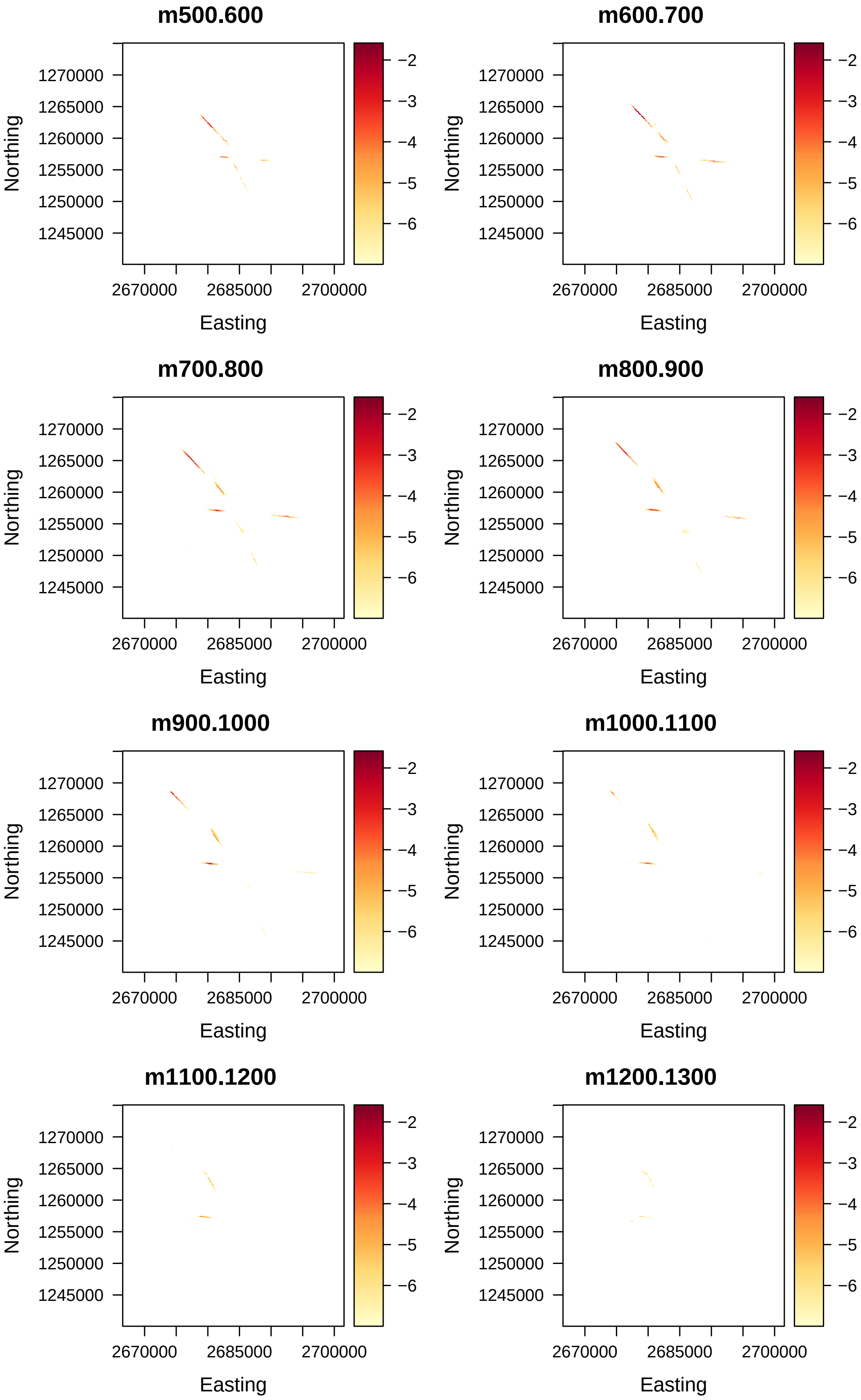


FIGURE 5.24: Collision probabilities of white storks with airplanes during fall migration. Scale: logarithmic.

# Chapter 6

# Discussion

In this chapter, the results of the last chapter are critically discussed and the findings placed in the context of the research questions posed at the outset of this thesis. Therefore, each research question is dealt with individually in a separate section. For reasons of comparability, each section uses the same structure.

## 6.1 RQ 1 – Development and Implementation

The first part of this thesis was devoted to the development and implementation of a probabilistic movement model that allows the simulation of trajectories in the 3-D space. The development started from the existing eRTG algorithm, as introduced by Technitis et al. (2016), which generates empirically informed random trajectories in the 2-D space. Following from this, the crucial criterion was to ensure that the simulated 3-D trajectories mimic a previously determined movement behavior and have a predefined starting and endpoint. These important characteristics were used to guarantee that the eRTG3D algorithm is applicable for many different tasks in the field of movement ecology. Situations, for example, in which an insufficient amount of 3-D movement data is available, can be bridged by means of simulation. In addition, the new model should help to fill in missing data in 3-D trajectories as well as to enable up-scaling the temporal resolution of low-resolution 3-D trajectories. Another application area of the algorithm was set to be the creation of null models for the expected movement in 3-D space. Therefore, the first research question was posed as follows:

**RQ 1** Development – *How is the third dimension best modeled and implemented in the eRTG algorithm so that empirically informed random trajectories can be generated in 3-D space?*

Research Question 1 forms the core of this thesis. All following steps and procedures depend on the successful development and implementation of the eRTG3D. Hence, the major part of the total time was invested into the development of the methodology for Research Question 1. The goal was to maintain the modular concept of the 2-D eRTG and to develop the resulting eRTG3D algorithm as comprehensively as possible. Although the application part of the thesis is fully devoted to the simulation of bird trajectories, the core of the algorithm was developed in such a way that the 3-D movement is defined in a generic manner and not just limited to bird flight.

The basic concept behind the trajectory simulation is a stepwise procedure that is grounded on two probabilities, which are extracted from an observed input trajectory in 3-D. The first probability, the *P probability*, describes the movement behavior from the mover's perspective, while the second probability, the *Q probability*, represents the attraction towards the target. From this modular construction principle it follows that, with the eRTG3D algorithm, not only conditional but also unconditional trajectories, so-called UERWs, can be generated. The UERWs serve a further purpose since the Q probability can be extracted from them if the original input trajectory does not consist of a sufficient number of fixpoints. The core of both probabilities is a multidimensional histogram, a voxel space, that represents either $(t, l, d)$-combinations for one step (P probability) or $(t, l, d)$-combinations for multiple steps in respect to the target (Q probability). Therefore, a new concept has been developed: the *turn-lift-step histogram* or *tld-cube*. Further, the consistency in movement was considered by incorporating density approximations of the autodifferences in each of the $(t, l, d)$-dimensions. The concept of fitting approximation functions to density estimates of empirical distributions was then applied to further movement parameters, such as the absolute and relative flight height and the gradient. Finally, in the stepwise simulation of a CERW, the probability of every possible next location is assessed by the extracted histograms and the approximation functions. Based on this probability, the next location is sampled. This procedure is repeated until the trajectory either encounters a dead end or reaches the target.

### Interpretation of the Resulting Algorithm

By extending the eRTG to the third dimension, the algorithm is now based on a 3-D probabilistic movement model. This makes it possible to simulate empirically informed trajectories in 3-D space. They can either be unconditional (UERW) or conditional on a given endpoint (CERW). Further, the 3-D space in which the simulations take place, may be characterized in various ways. Frequently used heights above ground and ellipsoid, or preferred gradients, are used to limit the freedom in the movement of the simulations. Thereby, physical constraints in the 3-D space, such as obstacles or gravity, may be accounted for.

The eRTG3D algorithm is a very robust algorithm that requires virtually no parameters. Further, it is computationally scalable and adaptable to a variety of problems. The simulations can be performed independently of the original spatial extent, since they only require start conditions, an endpoint and a movement behavior represented in the empirical distributions of movement parameters of given trajectories. This means that the simulations can take place in new and unobserved regions that are not congruent with the extent of the original trajectory. While most of the current approaches to analyze 3-D tracking data extract summarizing characteristics, which describe the distribution of occurrences in the given 3-D space of the observation (Cooper, Sherry, and Marra, 2014; Tracey et al., 2014; Bras, Jouma'a, and Guinet, 2017), the eRTG3D is not bound to the spatial extent of the data collection. This allows to make predictions about a probable movement behavior of an animal in a region where it has not been tracked before. This circumstance opens up a multitude of new research opportunities.

In comparison to other 3-D trajectory simulation algorithms, such as the 3-D RRW of Foster, Grassberger, and Paczuski (2009), the endpoint that the empirically informed random trajectories have to reach may be defined. This property is a unique feature of the eRTG3D algorithm within the field of animal movement simulation. Furthermore, due to the modular design of the algorithm, most of the RW types can be imitated and thereby turned into a conditional version. For example, by the inclusion of a background layer that reinforces already visited places, a conditional RRW version could easily be realized by the eRTG3D.

### Uncertainties and Limitations

Although great efforts have been invested in the development of a suitable methodology for implementing the third dimension, some uncertainties have nevertheless arisen. They either are a direct consequence of the properties of the input data, or they have emerged with the choice of a particular way to solve a problem, even if other approaches were available. In the following, the uncertainties concerning Research Question 1 are discussed and their handling is justified.

**Data** If the algorithm is not applied to synthetic data, but to real movement data most likely GPS tracking data will be used. Since the accuracy of the GPS devices depends on the quality of the satellite link, it can not be assumed that the positioning error is constant. The quality of the data is largely dependent on the geometry of the current recording situation. In an open sky situation, the record will be more accurate than in a covered terrain, where the direct contact to the satellites could be disturbed. Such disturbances may result in a delay in the recording time interval or even in missing data. In the case of the eRTG3D, besides the position in the $(x, y)$-plane, the height information is of crucial importance. It should be noted that the $z$-component of GPS tracking data is often more affected by errors than the $(x, y)$-position. Bouten et al. (2013) tested a light-weight GPS device on white storks. At a measurement interval of $6\,s$, the positional mean error was $2.45\,m$ and the mean altitude error was $2.77\,m$. At larger time intervals between the fixpoints, the accuracy in the position and height measurements decreases. This is due to the fact that in long intervals the satellite connection is not maintained and thus at the time of the fixpoint recording less satellites are usually available. In addition, the success rate for obtaining fixpoints decreases, which in the end leads to irregular time lags between the fixpoints. In order to automatically split trajectories into clean sections with a constant time lag, a tolerance of ±0.5 times the standard deviation of all time lags, as maximum variation from the expected time lag, showed good segmentation results.

**Cartesian coordinates** Since the formulae behind the algorithm are based on the Cartesian $(x, y, z)$-coordinate system, latitude and longitude coordinates are not supported. They have to be transformed into a Cartesian projected CRS. It follows that the curvature of the Earth's surface is not taken into account in the calculations. This is negligible for simulations on a small spatial scale, but if large-scale simulations are planned, the curvature of the earth must be considered. A solution is to divide the simulation into several sections and then simulate them individually in the corresponding Cartesian CRS. Then, the sections are converted to latitude and longitude data and stitched together. Thereby, it is helpful to follow the zones of the Universal Transverse Mercator (UTM) coordinate system, which approximates the slope of the Earth with a system of projected CRSs with local projection parameters.

**Bin size estimation** Since the Freedman–Diaconis rule is very robust and well established in scientific literature, it was chosen to estimate the bin size per dimension in the turn-lift-step histogram. But there are further formulae and rules of thumb for the estimation of the optimal bin size available, as for example the *Sturges' formula*, *Rice Rule*, *Scott's normal reference rule*, and many others (Lane et al., 2011). In order to better support the choice of a method, it would be conceivable to implement the most promising methods in the eRTG3D algorithm and compare the resulting simulations with each other.

**Combining probabilities** To obtain the probability $P_{i+1}$ for each individual step, several probabilities must be combined. This is done by multiplying the different probability values at any next location (Equation 4.8, p. 37). By just multiplying the individual terms, no weighting takes place. In the case of the probabilities for the autodifferences in each dimension, the probability is concerned about the same property. Therefore, in their combination they have to be weighted as less important than the probability of the turn-lift-step histogram, so as to maintain enough freedom in movement. To achieve this, the probabilities of the autodifferences are multiplied with each other and then the cubic root is taken. Then, the individual probabilities are united into one probability value per location and can be multiplied with the turn-lift-step histogram probability. Figure 6.1 illustrates the issue based on three empirical distribution functions (blue lines) extracted from values that were sampled from normal distributions with varying $\mu$ and $\sigma^2$ parameters. By multiplying the three distributions, the probabilities decrease drastically; by applying the cubic root, the probabilities rise again. In the lower plot of the figure, the two resulting distributions are shown standardized between 0 and 1. The distribution, where the third root was taken, is wider and thus allows more freedom in the choice of the next location than the simple multiplication. Therefore, the combined influence of the autodifferences is of equal importance as the probabilities of the turn-lift-step histogram in the final probability $P_{i+1}$.

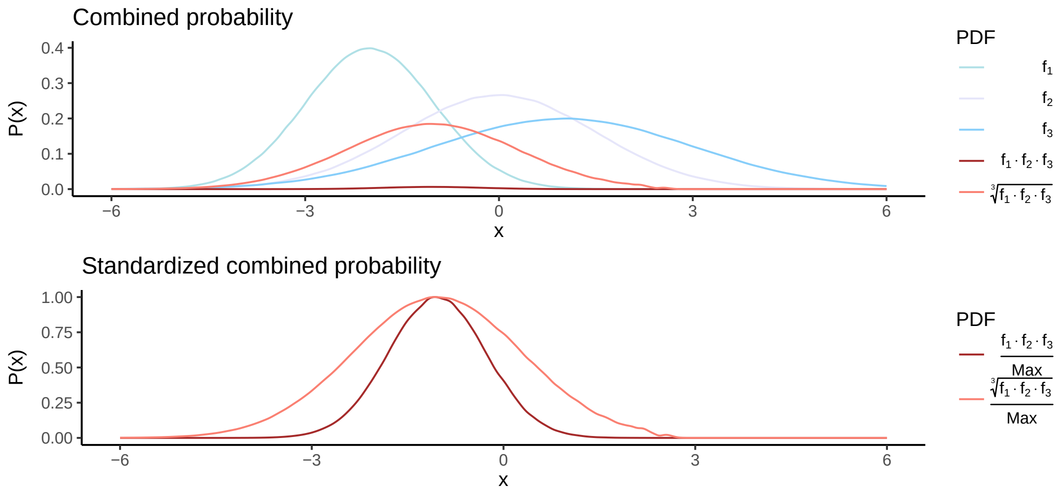


FIGURE 6.1: Combining probabilities: Multiplication versus multiplication and applying the cubic root.

The same approach is applied in Equation 4.17 (p. 40), where the two probabilities for the absolute and relative flight height are multiplied and the square root is taken to unite them into one probability. Another approach to avoid this issue could be to weight the probabilities individually. But this would introduce a variety of new problems and an extensive sensitivity analysis on the definition of the weights would become necessary. Various other approaches have been explored in this work, such as omitting the root or dividing by the number of probabilities. All alternative approaches increased the dead ends encountered by the CERWs.

**Dead ends** Since the eRTG3D algorithm is grounded on a probabilistic movement model, dead ends are not necessarily a bad thing. The CERWs should reflect the geometric characteristics of the movement behavior of the original trajectories and, at the same time, should be as random as possible within this limitation. Hence, the CERW may maneuver itself into a dead end because the algorithm does not foresee more than one step and thus does not schedule its entire route. The downside of the dead ends is that they alone can not reveal why the CERWs got stuck, since a dead end is encountered if the sum of one probability term in the Equation 4.17 (p. 40) is zero. Thus, it is difficult to detect whether the initially extracted probabilities are distorted or the given topography favors many dead ends. By saving the individual probability terms for every step, this issue could be solved, since this would allow their analysis in retrospect, and possibly backtracking. Additionally, further conclusions can be drawn about the (movement) suitability of the environment, by studying the locations in the 3-D space where the dead ends occurred.

**Line of sight** Since the simulation is based on fixpoints which are arranged in a sequence in order to form a trajectory, the actual path between the fixpoints is ignored. When simulating trajectories in a 3-D space that contains obstacles, such as the Earth's surface, this concept can be problematic. If the temporal resolution of the simulation is smaller than the spatial scale of the objects in space, a trajectory might pass through them, since during the simulation only the subsequent fixpoint can not be inside or below the object. In Figure 6.2, for example, the connecting straight line between two subsequent fixpoints passes through the Earth's surface, even though both endpoints are above the surface. To overcome this issue, it would be necessary to calculate the Line of Sight (LoS) for every path between all steps. Since this is a computationally demanding task in which all height differences from the line to each grid cell of the DEM passing the line are calculated, it is not possible to solve the problem this way. A simple workaround is to use a sufficiently high temporal resolution in the simulations and thus eliminate situations in which an obstacle is traversed between two subsequent fixpoints.

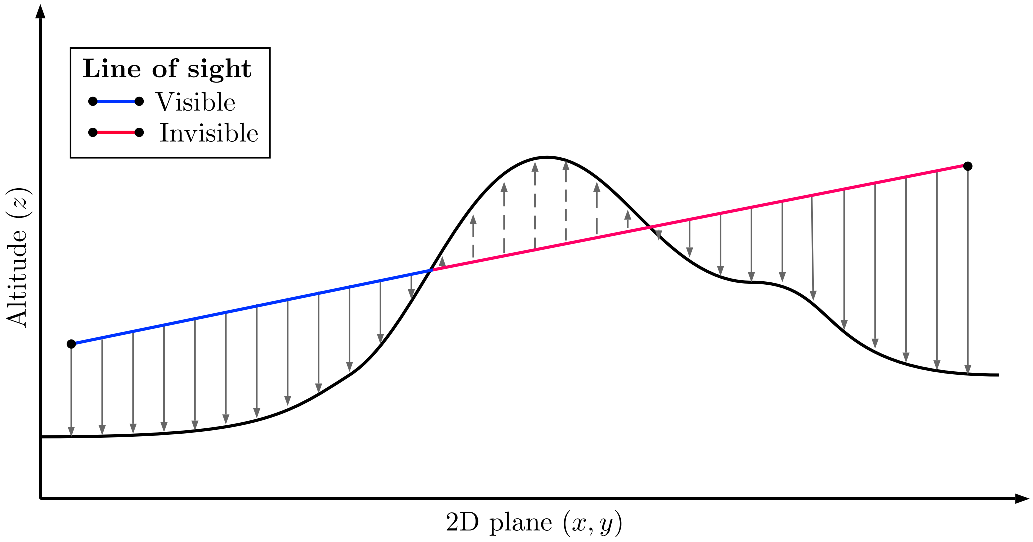


FIGURE 6.2: Line of sight between two subsequent fixpoints.

## Reflections on RQ1

As an answer to Research Question 1, it can be stated that the methodology for the simulation of empirically informed random trajectories in 3-D largely retains the modular concept of the initial 2-D eRTG, consisting of a P and a Q probability, respectively. However, the individual components of the concept and the stepwise trajectory generation itself were extended to the third dimension. Due to the great freedom of movement in 3-D space, the introduction of further physical constraints became necessary. As a consequence, a methodological concept was developed that appropriately reproduces movement behavior in the 3-D space. Additionally, this methodology was implemented in the R package *eRTG3D*.

## 6.2 RQ 2 – Evaluation

The second part of the thesis was dedicated to the evaluation of the newly developed and implemented algorithm. Not only the internal correctness of the developed probabilistic movement model, but also the ecological validity of the final implementation has to be inspected. Therefore, the second research question was formulated as follows:

**RQ 2** Evaluation – *How representative are the random trajectories generated with the eRTG3D algorithm?*

In order to answer Research Question 2, the evaluation of the eRTG3D algorithm was split into an internal verification, a validation in terms of its environmental validity and a sensitivity analysis of the parameter that limits the maximum number of bins in the turn-lift-step histogram. All test results have been interpreted at a significance level of 5 percent ($\alpha = 0.05$).

For the internal verification, two statistical test procedures have been developed. In each case, an initial trajectory was compared with 30 simulated reproduction trajectories that were based on the movement characteristics extracted from the initial trajectory. In order to guarantee the independence of the tests from the spatial extent of the trajectories, properties, which describe the steps from the mover's view (turning angle, lift angle and step length), were tested against each other. The first statistical test procedure is based on two-sample Kolmogorov-Smirnov tests, whereby the distributions of all available sample pairs are tested against each other. Due to the susceptibility of the Kolmogorov–Smirnov test to the sample size, a second approach was developed. Here, the distributions of the differences in turning angle, lift angle and step length of a limited number of sample pairs were tested on two sides against an expected value of 0, using a one-sample t-test. The limit of the sample pairs was set to the length of the shortest trajectory in the test. Both methods have in common that they do not test equality in itself, but assume that as long as no significant deviation can be determined, the trajectories can be regarded as originating from the same movement behavior.

In order to assess the ecological validity of the eRTG3D algorithm, an empirically observed trajectory of a white stork was reproduced 30 times by the eRTG3D algorithm. Then, the geometrical properties and the choice of flight path of the reproductions were compared visually with the original trajectory. The visual comparison was based on various 2-D and 3-D plots, in which also background properties, such as the DEM, had been visualized.

For the sensitivity analysis, the same white stork trajectory as used for the ecological validation was reproduced 30 times each for 9 different values of the dimension limiting parameter in the turn-lift-step histogram. Thereafter, the reproductions were tested statistically and compared visually against the original.

### Interpretation of the Evaluation Results

#### Verification

The verification of the algorithm was conducted on synthetic trajectories. Testing the CRWs, which were created for different correlation values, against their reproductions revealed no significant results (Table 5.1 and 5.2, p. 54). Since the initial CRWs were reproduced 30 times each, it can be assumed that the results are constant and reproducible. Furthermore, only a weak trend was identifiable in the p-values of the t-tests in dependency of the correlation values of the initial CRWs (Figure 5.2, p. 55). Although most p-values are close to 1, their interpretation remains tricky due to the influence of the sample size, a problem that is addressed in the next section. Moreover, they do not allow drawing direct conclusions in terms of the equality of the reproductions, since the p-values represent only the probability of obtaining a result as it is, in the case when the null hypothesis is true. However, it is important to note that, independently of the correlation value for the production of the CRWs (consistency in movement), no reproduction was significantly different than its initial CRW. The algorithm therefore is considered to be robust against different variations in the consistency in movement of trajectories. This finding is further underlined by Figure 5.3 (p. 55), where the autocorrelations of both, the CRWs and their reproduction CERWs, are calculated retrospectively. The autocorrelation in step length was omitted in this figure, since in the production of the CRWs the length of each step was sampled from a predefined normal distribution and therefore was not correlated with the step length of the preceding step. In any case, the eRTG3D was able to reconstruct CRWs with trajectories that had no significantly different distributions of turning angle, lift angle and step length and their autocorrelation values resembled the values of the original CRWs.

#### Validation

The validation of the eRTG3D algorithm in terms of its biological and ecological validity was conducted based on 30 reproductions of the low-resolution trajectory of a white stork crossing the Swiss Central Plateau. Figure 5.4 (p. 56) indicates that most of the simulated CERWs are biologically feasible. There are two trajectories that traverse a mountainous area. Here, the *line of sight*-problem, which was previously mentioned in this chapter, becomes apparent, since the big time lag used in the simulations allows passing through the small southern foothills of the Alps. This is clearly a methodological problem of the algorithm, but it can easily be avoided by increasing the temporal resolution of the simulations. Besides this problem, all simulations appear to be ecologically plausible, as they avoid unrealistically high altitudes and flybys close to the ground. Further, the consistency in movement resembles the original white stork trajectory. Furthermore, the majority of the simulations choose a path that leads to the target just as directly as the original trajectory. A few reproductions are slightly more scattered and hold north or south in regard to the original trajectory, but they remain within the Swiss Central Plateau. Due to the random component of CERWs, the spatial scattering is also desired. The scattering can further be explained by the fact that, besides the DEM, no background, which would further limit the freedom in the movement, such as the uplift suitability map, was used in these simulations.

### Sensitivity Analysis

Apart from the autodifferences of the lift angle, no test result of the sensitivity analysis was significant (Table 5.3 and 5.4, p. 57). It can be assumed that the influence of the gradient distribution and the height distributions distort the choices with regard to the lift angle and that, therefore, the autodifferences in the lift angle deviate significantly from the original. This statement is underlined by the fact that no other distributions of the autodifferences deviate significantly from the original trajectory. Figure 5.6 (p. 58) indicates that the larger the allowed number of bins becomes, the more robust the result is. In this case, *robust* means that the variation of the p-values in the different dimensions (turning angle, lift angle, step length and the corresponding autodifferences) is smaller and they are closer to 1. This circumstance is well traceable, as even a strongly limited turn-lift-step histogram allows for a continuous choice in every dimension, since a uniform distributed shift term is added to each bin midpoint. Thus, as the number of allowed bins in the histogram increases, the probability to draw an extreme sample due to the shift term will decrease, since the voxel bin is smaller in each dimension. The visual comparison (Figure 5.7, p. 59) of the results is also very clear. The simulations that are based on an unlimited turn-lift-step histogram respond better to the changing topography. They follow the course of valleys more exactly and better use topographically suitable regions. Moreover, they avoid an unrealistic outburst from the Swiss Central Plateau to the north.

## Uncertainties and Limitations

Regarding the methodology that was used to cope with Research Question 2, some uncertainties and limitations need to be addressed.

**P-value** Although the p-value is a widely used statistical measure that underpins many evaluations of research results, it often leads to false interpretations and misuse of test results. In the case of this thesis, where the p-values are used to estimate whether two trajectories originate from the same movement behavior, it is important to note that the p-value says nothing about the probability of the null hypothesis ($H_0$) itself. It only represents the probability to receive exactly the same result as observed in the sample, given the null hypothesis is true (Goodman, 2008; Greenland et al., 2016). Since, at a significance level of 5 percent ($\alpha = 0.05$), the correct null hypothesis is discarded (type I error) every $20^{th}$ time on average, the p-value should be calculated several times by repeating the experiment (Head et al., 2015).

**Sample size** A fundamental limitation of the statistical significance analysis is the susceptibility of the p-value to the sample size, or, as Wasserstein and Lazar (2016, p. 132) adress the problem:

> "Any effect, no matter how tiny, can produce a small p-value if the sample size or measurement precision is high enough, and large effects may produce unimpressive p-values if the sample size is small or measurements are imprecise."

In the context of this thesis, this implies that, when short trajectories are compared against each other, the test results have the tendency to overlook actually important deviations between the distributions. By contrast, long trajectories or a large number of short simulations increase the sample size and thus ensure that even the slightest deviations between the distributions of turning angle, lift angle and step length lead to significant test results.

**Distribution of height** The visual inspection of the densities of the absolute and relative flight height and the gradient in Figure 5.5 (p. 57) revealed that the lower absolute heights are overrepresented and the lower relative heights are underrepresented. Due to the influence of the topography on the relative flight height in the simulations, a variation from the original flight height is to be expected to a certain extent, but the trend seems to be systematic. An explanation for this could be that both height probabilities (absolute and relative) have the same weight during the creation of the CERWs. In real life, it may happen that one height is more important than the other. From the systematic over- and underrepresentation it can be concluded that the altitude in the simulations is generally somewhat more constant than in the original trajectory. The same constancy is also evident in the density of the gradient, which is slightly narrower and has the mode at a slightly lower value than the original gradient density. It can be assumed that the stronger constancy in the heights and gradient originates from the influence of the autodifferences, that limit the variation in movement. The white stork trajectory chosen for the validation has a low temporal resolution of 20 minutes. This may not be an ideal condition for the extraction of the autodifference approximations, since the variance is expected to be extremely high.

### Reflections on RQ2

With respect to the initially posed Research Question 2, it can be concluded that, due to the successful verification, the reliable validation and the robust sensitivity analysis, the trajectories generated with the eRTG3D are internally correct, ecologically valid and thus able to adequately reproduce 3-D movement behavior. From the sensitivity analysis it further follows that the choice of a low maximum number of bins is only justified if a high computational efficiency is of maximum concern.

## 6.3 RQ 3 – Application

In the third part of the thesis, the capabilities of the eRTG3D algorithm were shown by means of a demonstrator use case. As an example use case, the estimation of probabilities of migrating white storks colliding with airplanes near Zurich Airport was chosen. Thus, the third research question was posed as follows:

**RQ 3** Application – *What are the collision probabilities of white storks (Ciconia ciconia) and airplanes in the arrival and departure corridors at Zurich Airport (ZRH) during the birds' fall migration?*

The idea behind Research Question 3 was to demonstrate and cover as many applications of the eRTG3D algorithm as possible in one use case. In order to achieve this, the standard functionality of the eRTG3D algorithm was used for the low-resolution simulation of white stork trajectories crossing the Swiss Central Plateau. Then, the modularity of the algorithm was used for its extension by the gliding and soaring functionality. Thereby, the uplift suitability map, as a further physical constraint, was included into the simulations. Finally, the low-resolution trajectories were up-sampled to high-resolution gliding and soaring trajectories based on the gliding and soaring version of the algorithm. The temporal up-sampling of trajectories is a further application that the eRTG3D algorithm was designed for. Outsourcing the calculations to a cluster demonstrated the scalability of the algorithm, which is based on its modularity and ability to parallelize the simulation processes.

To complete the application part, UDs were derived from the simulated trajectories of the white storks and the observed airplane flight paths. Since the calculation of the UDs was not part of the core task of this master's thesis, the reasonably simple approach of voxel counting was chosen. As a last step, the collision probability was obtained by multiplication of the two UDs.

### Interpretation of Results

The final high-resolution gliding and soaring trajectories in the area around Zurich Airport look very plausible (Figure 5.19, p. 67). As the distributions of the flight height and the gradient as well as the consistency in movement resemble the movement characteristics that were observed in the trajectory of Wibi 2. Furthermore, the gliding and soaring appears to be very natural, since no extreme heights are gained and no unrealistically long gliding phases occur.

It has to be pointed out that all statistical results of the Kolmogorov–Smirnov tests are significant. Also in the t-tests, the distribution of the step length in the simulations varies significantly from the observed distribution in Wibi 2. These negative statistical test results have three causes. Firstly, there are two movement modes in the simulations and in the original trajectory. However, the proportion of time spent in each mode is not defined beforehand, since it is mostly depending on the flight path chosen and the uplift suitability encountered while following the path.

Therefore, this proportion varies from trajectory to trajectory, which leads to different distributions across the simulations. Secondly, the transition between soaring and gliding phases is smoothed. Thereby, the bird is oriented towards the target. Since the movement covered in this correction path is not sampled from the P and Q probabilities, the final distributions of the simulations are slightly distorted. Finally, the third reason is the vast size of the samples that are compared, which affects the robustness of the p-value. Every simulated trajectory consist of more than 1 000 steps and 1 000 simulated trajectories were tested against the original trajectory of Wibi 2.

Although the statistical test results of the final simulations are not as positive as expected, it can nevertheless be assumed that the simulated trajectories are ecologically valid and represent plausible white stork migration routes. This conclusion is based on the better statistical results, which were obtained when the movement modes were tested separately (Table C.1 and C.2, p. xlv), and the visual inspection of the densities of the turning angle, lift angle, step length, and the corresponding autodifferences in Figure 5.20 (p. 67).

From the observed trajectory of Wibi 2 it became clear that the white storks pass through the airspace above Zurich Airport during their fall migration. The calculated UD of white storks passing above the airport further reveals that most of the birds use the space to the north of the airport for their journey at altitudes between 700 and 1 000 $m$. This is mainly due to the fact that white storks encounter good uplift conditions there and the route to their destination in the southwest is more direct. A similar trend in space use is observed in the UD of the airplanes, where the northern arrival corridor and the eastern departure corridor are preferred and used in similar height bands that the white storks use too. Buchmüller et al. (2015) come to similar results in their analyses of the air traffic at Zurich Airport (Figure 6.3).

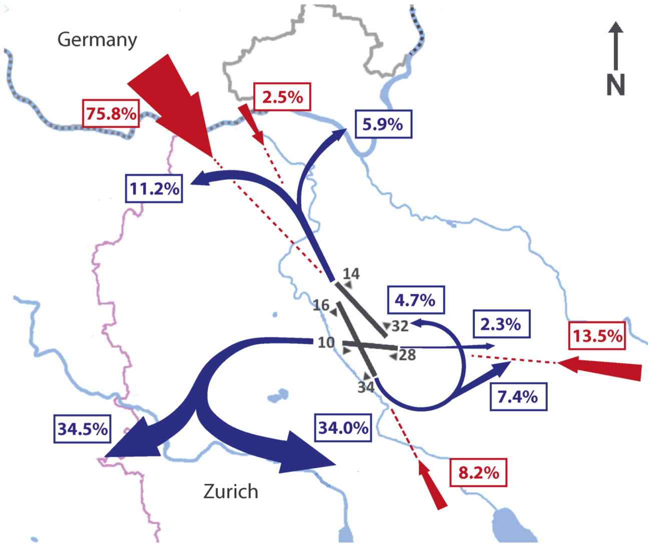


FIGURE 6.3: Air traffic corridors for arrivals (red) and departures (blue) at Zurich Airport. *Source: Buchmüller et al. (2015, p. 181)*

### Uncertainties and Limitations

In the methodology that was applied to cope with the third research question, there are some points that have to be considered critically. These points are discussed in the following.

**Individuality** The probabilities (P and Q) for the simulations were, for the sake of simplicity, only extracted from the trajectory of one white stork. However, as the available data basis in the movebank data repository is very large, and due to its modular structure, the eRTG3D algorithm would be very well suited to perform simulations based on multiple individuals. To achieve this, several white stork trajectories would have to be prepared accordingly and then, several probabilities could be extracted from them. By means of a random sampling for each simulation of a trajectory, the movement behavior of an individual white stork could hence be drawn from the previously extracted probabilities. Overall, this would most likely lead to ecologically more representative simulations.

**Binary uplift suitability** Since the uplift suitability map is classified into a binary layer, the edge areas of uplift zones are consistently and intensely used for soaring. If, instead of a rigid binary classification, a continuous uplift suitability map with intensity values between 0 and 1 was used, the transition between the zones would be less pronounced. It is also important to note that the uplift suitability map represents an average weather situation, which is independent of the time of day and the day of the year.

**Voxel resolution** The resolution of the voxel cells of $100 \times 100 \times 100\,m$ was chosen in regard to the average size of airplanes. A plane is only represented as a point in the analysis, but it has a much larger extension. Consequently, the calculation of the collision probabilities is based on the assumption that, whenever an airplane and a bird meet in such a voxel, they inevitably collide.

**Distribution over time** When calculating the UDs, the spatial distribution of birds and airplanes is considered to be constant over time. This means that it is not distinguished at what time a bird passes the airspace above the airport. The same applies to the UD of the airplanes.

### Reflections on RQ3

In order to answer Research Question 3 it can be concluded that the highest calculated risk of collisions during the fall migration of white storks is located in the northern arrival corridor at altitudes between 600 and $900\,m$. In addition, the collision probability is also higher in the eastern departure corridor at altitudes between 800 and $1\,100\,m$. These results seem plausible, since a migration route to the north of the airport is more favorable for the white storks, while at the same time the air traffic is the most intense in this region. Due to the underlying assumption that the spatial distributions of birds and airplanes are constant over time, the calculated probabilities can not be considered as absolute. They rather show the danger for bird strikes relatively between the different flight corridors, while good uplift conditions can drastically increase the duration that white storks spend in an airspace. Thus, the danger of collisions of soaring birds with airplanes could be estimated by measuring the uplift conditions in the flight corridors, as these measurements could be used as a proxy for estimating the bird-strike probability in real time.

# Chapter 7

# Conclusion

## 7.1 Summary

The work on this thesis started off with a comprehensive review of literature with the objective to identify important research gaps in the field of 3-D animal movement simulation. In order to address the identified research gaps, three research questions have been posed. The first research question was dedicated to the development and implementation of the eRTG3D algorithm in order to enable the simulation of empirically informed random trajectories in 3-D space. The second research question tackled the evaluation of the algorithm, which was subdivided into an internal verification of the algorithm, an ecological validation of the generated trajectories, and a sensitivity analysis. The third research question was concerned with the demonstration of the capabilities of the eRTG3D algorithm. To this end, an illustrative example application was carried out, whereby the collision probabilities of migrating white storks with airplanes at Zurich Airport were calculated. In the subsequent discussion the obtained results were interpreted and the methodology applied was critically examined. Finally, the results were reflected with regard to the initially posed research questions and thereby the research questions were answered.

## 7.2 Contributions

Although the thesis was tripartite due to the formulation of three research questions, the most important knowledge gains are summarized in one comprehensive list that presents the outcome of this thesis as a whole:

- An extensive literature review was conducted, which summarized the existing movement simulation models in the field of movement ecology. Thereby, crucial research gaps in the field of 3-D movement simulation were identified.
- Thereafter, starting from the existing concept of the eRTG, a new probabilistic movement model that represents the movement behavior of a mover from its perspective in 3-D was developed, termed *eRTG3D*.
- The eRTG3D algorithm was then implemented, based on this 3-D probabilistic movement model, which enables the simulation of unconditional (UERWs)

and conditional (CERWs) empirically informed random trajectories in the 3-D space.

- Subsequently, the new algorithm was successfully evaluated. The trajectories simulated by the eRTG3D algorithm are statistically reliable and ecologically valid.

- The successful evaluation allows the use of the eRTG3D algorithm for modeling and studying movement phenomena in 3-D, based on observed tracking data. It should be noted that the regions within which the simulations take place must not be congruent with the initial spatial extent of the original tracking data that was used training the algorithm. Hence, the eRTG3D algorithm even enables movement analyses outside the original area of observation in new regions.

- Furthermore, a version of the algorithm reproducing the movement behavior of soaring birds was developed and implemented. To the author's knowledge, no other algorithm exists, which is capable of simulating gliding and soaring flight behavior of birds in 3-D, based on an uplift suitability map.

- The modular and efficient design of the algorithm, its independence of the initial spatial extent and the fact that actual trajectories are generated between two given points introduce a multitude of new possibilities tackle movement-related research questions in 3-D. These new possibilities include the generation of null hypotheses for movement in 3-D under given assumptions, the temporal up-sampling of 3-D trajectories, the bridging of missing data in 3-D, or the derivation of statements about the 3-D movement behavior of a larger group based on the observed movement behavior of an individual, to name but a few.

- A demonstrator use case was conducted. The estimation of collision probabilities of soaring white storks and airplanes at Zurich Airport successfully demonstrated the use of the different capabilities of the eRTG3D algorithm.

- The demonstrator use case further revealed that the highest collision probability of white storks with airplanes at the airport Zurich during their fall migration is located in the northern arrival corridor ($600 - 900\, m\, a.s.l.$) and in the eastern departure corridor ($800 - 1\,100\, m\, a.s.l.$). Furthermore, the crucial influence of the prevailing uplift conditions on the space use of soaring birds was highlighted.

- Finally, the eRTG3D algorithm is available as an R package on *www.github.com*, which makes it easily accessible, facilitates its use among the field of movement ecology, and favors its further development.

## 7.3 Future Research

Further research related to the eRTG3D algorithm can be split into two types: On the one hand, the new algorithm, as it exists now, enables the study of a variety of new movement phenomena of flying animals by means of simulations in 3-D. Furthermore, the investigation of the gliding and soaring behavior of birds is made possible to a new extent. On the other hand, the eRTG3D algorithm itself can be further developed and extended. The methodology of the current eRTG3D version features some critical points that have been addressed in Chapter 6. It would be important to further explore these issues, and, for example, to find a methodological solution to the *line of sight* problem. In addition to the immediate improvement of the existing methodology, there are almost unlimited possibilities of extending the eRTG3D algorithm with exciting new capabilities. Due to the modular methodological concept of the eRTG3D algorithm, the development and implementation of such extensions should be straightforward. In the following, some interesting extensions worth thinking about are proposed.

A capability that could be implemented with relatively little effort is the extension of the algorithm for diving animals. Hence, only the water surface would need to be introduced as a confining top layer and the already integrated DEM layer could be replaced by a bathymetric layer, which represents the bottom of the water body. As a consequence, the simulation of trajectories that are limited to a water body would become possible. Another useful enhancement of the algorithm could be the merging of the 2-D with the 3-D version. Thus, it should be possible to switch seamlessly between the versions during the simulation, e.g. using the 2-D version for simulating long-haul, large-scale migration, while using the 3-D version for more detailed and more local movements. It would further be beneficial to allow the inclusion of multiple movement modes per individual in a more general manner. The switch between modes could then be coordinated by additional physical constraints, such as energy expenditure or the time of day. In addition, the integration of social attractors and networks into the algorithm would enable simulations of the movement behavior of individuals with respect to other individuals or even swarm behavior. Finally, the development could even be directed towards an agent-based model, in which entire lifespans of individuals, including their predators or competing species, as well as available resources, could be included so as to reflect an entire ecosystem. In this regard, the main property of the algorithm that unconditional and conditional empirically informed random trajectories can be generated, is particularly useful, since exploratory as well as targeted movement behavior could by this manner be included in the agent-based model.

As this concluding section has highlighted, the possibilities of the eRTG3D algorithm are far from being exhausted, on the contrary, further exciting research topics and versatile applications have just been made possible through the eRTG3D algorithm.

# Bibliography

Alerstam, T., Gudmundsson, G. A., and Larsson, B. (1993). “Flight Tracks and Speeds of Antarctic and Atlantic Seabirds: Radar and Optical Measurements”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 340.1291, pp. 55–67. ISSN: 0962-8436. DOI: 10.1098/rstb.1993.0048.

Arfken, G. B., Weber, H. J., and Harris, F. E. (2013). *Mathematical Methods for Physicists*. ISBN: 9780123846549. DOI: 10.1016/C2009-0-30629-7.

Bartumeus, F., Da Luz, M. G. E., Viswanathan, G. M., and Catalan, J. (2005). “Animal search strategies: A quantitative random-walk analysis”. In: *Ecology* 86.11, pp. 3078–3087. ISSN: 00129658. DOI: 10.1890/04-1806.

Berg, H. C. (1993). “Diffusion: Microscopic Theory”. In: *Random Walks in Biology*. Ed. by Berg, H. C. New Jersey: Princeton University Press. Chap. 1, pp. 5–16. ISBN: 9780691000640.

Berthold, P., Bossche, W. V. D., Jakubiec, Z., Kaatz, C., Kaatz, M., and Querner, U. (2002). “Long-term satellite tracking sheds light upon variable migration strategies of White Storks (Ciconia ciconia)”. In: *Journal of Ornithology* 143.4, pp. 489–495. ISSN: 00218375. DOI: 10.1046/j.1439-0361.2002.02044.x.

Berthold, P., Kaatz, M., and Querner, U. (2004). “Long-term satellite tracking of white stork (Ciconia ciconia) migration: constancy versus variability”. In: *Journal of Ornithology* 145.4, pp. 356–359. ISSN: 0021-8375. DOI: 10.1007/s10336-004-0049-2.

Berthold, P., Bossche, W. V. D., Fiedler, W., Kaatz, C., Kaatz, M., Leshem, Y., Nowak, E., and Querner, U. (2001). “Detection of a new important staging and wintering area of the White Stork Ciconia ciconia by satellite tracking”. In: *Ibis* 143.4, pp. 450–455. ISSN: 00191019. DOI: 10.1111/j.1474-919X.2001.tb04946.x.

Blanco, J. C. and Cortés, Y. (2007). “Dispersal patterns, social structure and mortality of wolves living in agricultural habitats in Spain”. In: *Journal of Zoology* 273.1, pp. 114–124. ISSN: 0952-8369. DOI: 10.1111/j.1469-7998.2007.00305.x.

Bohrer, G., Brandes, D., Mandel, J. T., Bildstein, K. L., Miller, T. A., Lanzone, M., Katzner, T., Maisonneuve, C., and Tremblay, J. A. (2012). “Estimating updraft velocity components over large spatial scales: contrasting migration strategies of golden eagles and turkey vultures”. In: *Ecology Letters* 15.2, pp. 96–103. ISSN: 1461023X. DOI: 10.1111/j.1461-0248.2011.01713.x.

Both, A., Duckham, M., Laube, P., Wark, T., and Yeoman, J. (2013). “Decentralized Monitoring of Moving Objects in a Transportation Network Augmented with Checkpoints”. In: *The Computer Journal* 56.12, pp. 1432–1449. ISSN: 0010-4620. DOI: 10.1093/comjnl/bxs117.

Bouten, W., Baaij, E. W., Shamoun-Baranes, J., and Camphuysen, K. C. J. (2013). "A flexible GPS tracking system for studying bird behaviour at multiple scales". In: *Journal of Ornithology* 154.2, pp. 571–580. ISSN: 2193-7192. DOI: 10.1007/s10336-012-0908-1.

Bras, Y. L., Jouma'a, J., and Guinet, C. (2017). "Three-dimensional space use during the bottom phase of southern elephant seal dives". In: *Movement Ecology* 5.1, p. 18. ISSN: 2051-3933. DOI: 10.1186/s40462-017-0108-y.

Brown, R. (1828). "XXVII. A brief account of microscopical observations made in the months of June, July and August 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies". In: *Philosophical Magazine Series 2* 4.21, pp. 161–173. ISSN: 1941-5850. DOI: 10.1080/14786442808674769.

Buchmüller, J., Janetzko, H., Andrienko, G., Andrienko, N., Fuchs, G., and Keim, D. A. (2015). "Visual Analytics for Exploring Local Impact of Air Traffic". In: *Computer Graphics Forum* 34.3, pp. 181–190. ISSN: 01677055. DOI: 10.1111/cgf.12630.

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

Burt, W. H. (1943). "Territoriality and Home Range Concepts as Applied to Mammals". In: *Journal of Mammalogy* 24.3, pp. 346–352. DOI: 10.2307/1374834.

Byrne, M. E., Holland, A. E., Bryan, A. L., and Beasley, J. C. (2017). "Environmental conditions and animal behavior influence performance of solar-powered GPS-GSM transmitters". In: *The Condor* 119.3, pp. 389–404. ISSN: 0010-5422. DOI: 10.1650/CONDOR-16-76.1.

Calenge, C. (2006). "The package "adehabitat" for the R software: A tool for the analysis of space and habitat use by animals". In: *Ecological Modelling* 197.3-4, pp. 516–519. ISSN: 03043800. DOI: 10.1016/j.ecolmodel.2006.03.017.

Chow, W. C. (2009). "Brownian bridge". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 1.3, pp. 325–332. ISSN: 19395108. DOI: 10.1002/wics.38.

Codling, E. A., Bearon, R. N., and Thorn, G. J. (2010). "Diffusion about the mean drift location in a biased random walk". In: *Ecology* 91.10, pp. 3106–3113. ISSN: 00129658. DOI: 10.1890/09-1729.1.

Codling, E. a., Plank, M. J., and Benhamou, S. (2008). "Random walk models in biology". In: *J. R. Soc. Interface* 5.25, pp. 813–834. ISSN: 1742-5689. DOI: 10.1098/rsif.2008.0014.

Cooke, S. (2008). "Biotelemetry and biologging in endangered species research and animal conservation: relevance to regional, national, and IUCN Red List threat assessments". In: *Endangered Species Research* 4.1-2, pp. 165–185. ISSN: 1863-5407. DOI: 10.3354/esr00063.

Cooper, N. W., Sherry, T. W., and Marra, P. P. (2014). "Modeling three-dimensional space use and overlap in birds". In: *The Auk* 131.4, pp. 681–693. ISSN: 0004-8038. DOI: 10.1642/AUK-14-17.1.

Crist, T. O., Guertin, D. S., Wiens, J. A., and Milne, B. T. (1992). "Animal Movement in Heterogeneous Landscapes: An Experiment with Eleodes Beetles in Shortgrass Prairie". In: *Functional Ecology* 6.5, pp. 536–544. DOI: 10.2307/2390050.

Demšar, U., Buchin, K., Cagnacci, F., Safi, K., Speckmann, B., Van de Weghe, N., Weiskopf, D., and Weibel, R. (2015). "Analysis and visualisation of movement: an interdisciplinary review". In: *Movement Ecology* 3.1, p. 5. ISSN: 2051-3933. DOI: 10.1186/s40462-015-0032-y.

Dodge, S., Weibel, R., and Lautenschütz, A.-K. (2008). "Towards a taxonomy of movement patterns". In: *Information Visualization* 7.3-4, pp. 240–252. ISSN: 1473-8716. DOI: 10.1057/palgrave.ivs.9500182.

Einstein, A. and Infeld, L. (1938). *The Evolution of Physics*. Ed. by Snow, C. P. London: Cambridge University Press, p. 319.

Elliott, A. (1992). "Family Ciconiidae (Storks)". In: *Handbook of the Birds of the World, Volume 1: Ostrich to Ducks*. Ed. by Del Hoyo, J., Elliot, A., and Sargatal, J. 1st ed. Barcelona: Lynx Edicions, p. 696. ISBN: 8487334091.

Ferter, K., Hartmann, K., Kleiven, A. R., Moland, E., and Olsen, E. M. (2015). "Catch-and-release of Atlantic cod (Gadus morhua): post-release behaviour of acoustically pretagged fish in a natural marine environment". In: *Canadian Journal of Fisheries and Aquatic Sciences* 72.2. Ed. by Jech, J. M., pp. 252–261. ISSN: 0706-652X. DOI: 10.1139/cjfas-2014-0290.

Flack, A, Fiedler, W, and Wikelski, M (2017). *Data from: Wind estimation based on thermal soaring of birds. Movebank Data Repository.* DOI: 10.5441/001/1.bj96m274.

Flack, A., Fiedler, W., Blas, J., Pokrovsky, I., Kaatz, M., Mitropolsky, M., Aghababyan, K., Fakriadis, I., Makrigianni, E., Jerzak, L., Azafzaf, H., Feltrup Azafzaf, C., Rotics, S., Mokotjomela, T. M., Nathan, R., and Wikelski, M. (2016). "Costs of migratory decisions: A comparison across eight white stork populations". In: *Science Advances* 2.1, pp. 1–8. ISSN: 2375-2548. DOI: 10.1126/sciadv.1500931.

Flack, A., Nagy, M., Fiedler, W., Couzin, I. D., and Wikelski, M. (2018). "From local collective behavior to global migratory patterns in white storks". In: *Science* 360.6391, pp. 911–914. ISSN: 0036-8075. DOI: 10.1126/science.aap7781.

Fontelos, M. A. and Friedman, A. (2015). "A PDE model for the dynamics of trail formation by ants". In: *Journal of Mathematical Analysis and Applications* 425.1, pp. 1–19. ISSN: 0022247X. DOI: 10.1016/j.jmaa.2014.12.030.

Forester, J., Im, H., and Rathouz, P. (2009). "Accounting for animal movement in estimation of resource selection functions: Sampling and data analysis". English (US). In: *Ecology* 90.12, pp. 3554–3565. ISSN: 0012-9658. DOI: 10.1890/08-0874.1.

Foster, J. G., Grassberger, P., and Paczuski, M. (2009). "Reinforced walks in two and three dimensions". In: *New Journal of Physics* 11.2, p. 023009. ISSN: 1367-2630. DOI: 10.1088/1367-2630/11/2/023009.

Freedman, D. and Diaconis, P. (1981). "On the histogram as a density estimator:L 2 theory". In: *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 57.4, pp. 453–476. ISSN: 0044-3719. DOI: 10.1007/BF01025868.

Fretwell, S. D. and Lucas, H. L. (1970). “On territorial behavior and other factors influencing habitat distribution in birds”. In: *Acta Biotheoretica* 19, pp. 16–36.

Gatrell, A. C., Bailey, T. C., Diggle, P. J., and Rowlingson, B. S. (1996). “Spatial point pattern analysis and its application in geographical epidemiology”. In: *Transactions of the Institute of British Geographers* 21.1, pp. 256–274.

Giannotti, F. and Pedreschi, D. (2008). *Mobility, data mining and privacy: Geographic knowledge discovery*, pp. 1–410. ISBN: 9783540751762. DOI: 10.1007/978-3-540-75177-9.

Goodman, S. (2008). “A Dirty Dozen: Twelve P-Value Misconceptions”. In: *Seminars in Hematology* 45.3, pp. 135–140. ISSN: 00371963. DOI: 10.1053/j.seminhematol.2008.04.003.

Greenland, S., Senn, S. J., Rothman, K. J., Carlin, J. B., Poole, C., Goodman, S. N., and Altman, D. G. (2016). “Statistical tests, P values, confidence intervals, and power: a guide to misinterpretations”. In: *European Journal of Epidemiology* 31.4, pp. 337–350. ISSN: 0393-2990. DOI: 10.1007/s10654-016-0149-3.

Gudmundsson, J., Laube, P., and Wolle, T. (2012). *Springer Handbook of Geographic Information*. Ed. by Kresse, W. and Danko, D. M. 1st ed. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 423–438. ISBN: 978-3-540-72678-4. DOI: 10.1007/978-3-540-72680-7.

Hägerstrand, T. (1970). “What about people in Regional Science?” In: *Papers of the Regional Science Association* 24.1, pp. 6–21. ISSN: 1056-8190. DOI: 10.1007/BF01936872.

Head, M. L., Holman, L., Lanfear, R., Kahn, A. T., and Jennions, M. D. (2015). “The Extent and Consequences of P-Hacking in Science”. In: *PLOS Biology* 13.3, pp. 1–15. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1002106.

Hedenstrom, A. (1993). “Migration by Soaring or Flapping Flight in Birds: The Relative Importance of Energy Cost and Speed”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 342.1302, pp. 353–361. ISSN: 0962-8436. DOI: 10.1098/rstb.1993.0164.

Hedenstrom, A. and Alerstam, T. (1998). “How Fast Can Birds Migrate?” In: *Journal of Avian Biology* 29.4, p. 424. ISSN: 09088857. DOI: 10.2307/3677161.

Horne, J. S., Garton, E. O., Krone, S. M., and Lewis, J. S. (2007). “Analyzing Animal Movements Using Brownian Bridges”. In: *Ecology* 88.9, pp. 2354–2363. ISSN: 0012-9658. DOI: 10.1890/06-0957.1.

International BirdLife (2016). *Ciconia ciconia*. URL: http://www.iucnredlist.org/details/22697691/0 (visited on 04/22/2018).

Kareiva, P. M. and Shigesada, N. (1983). “Analyzing insect movement as a correlated random walk”. In: *Oecologia* 56.2-3, pp. 234–238. ISSN: 00298549. DOI: 10.1007/BF00379695.

Keating, K. and Cherry, S. (2009). “Modeling utilization distributions in space and time”. In: *Ecology* 90.7, pp. 1971–1980. ISSN: 00129658. DOI: 10.1890/08-1131.1.

Kranstauber, B., Kays, R., LaPoint, S. D., Wikelski, M., and Safi, K. (2012). “A dynamic Brownian bridge movement model to estimate utilization distributions for heterogeneous animal movement". In: *Journal of Animal Ecology* 81.4, pp. 738–746. ISSN: 00218790. DOI: 10.1111/j.1365-2656.2012.01955.x.

Lane, D. M., Scott, D., Hebl, M., Guerra, R., Osherson, D., and Zimmer, H. (2011). *Online Statistics Education: A Multimedia Course of Study*. URL: http://www.onlinestatbook.com (visited on 03/14/2018).

Laube, P. (2009). "Progress in Movement Pattern Analysis". In: *Behaviour Monitoring and Interpretation, BMI, Smart Environments*. Ed. by Gottfried, B. and Aghajan, H. 3rd ed. Amsterdam: IOS Press, pp. 43–71. ISBN: 978-1-60750-048-3. DOI: 10.3233/978-1-60750-048-3-43.

— (2014). *Computational Movement Analysis*. SpringerBriefs in Computer Science. Cham: Springer International Publishing. ISBN: 978-3-319-10267-2. DOI: 10.1007/978-3-319-10268-9.

Long, J. A. (2016). "Kinematic interpolation of movement data". In: *International Journal of Geographical Information Science* 30.5, pp. 854–868. ISSN: 1365-8816. DOI: 10.1080/13658816.2015.1081909.

McMahon, L. A., Rachlow, J. L., Shipley, L. A., Forbey, J. S., Johnson, T. R., and Olsoy, P. J. (2017). "Evaluation of micro-GPS receivers for tracking small-bodied mammals". In: *PLOS ONE* 12.3. Ed. by Crowther, M. S., p. 19. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0173185.

Miller, H. J. (1991). "Modelling accessibility using space-time prism concepts within geographical information systems". In: *International journal of geographical information systems* 5.3, pp. 287–301. ISSN: 0269-3798. DOI: 10.1080/02693799108927856.

Nathan, R. and Giuggioli, L. (2013). "A milestone for movement ecology research". In: *Movement Ecology* 1.1, p. 1. ISSN: 2051-3933. DOI: 10.1186/2051-3933-1-1.

Nathan, R., Getz, W. M., Revilla, E., Holyoak, M., Kadmon, R., Saltz, D., and Smouse, P. E. (2008). "A movement ecology paradigm for unifying organismal movement research". In: *Proceedings of the National Academy of Sciences of the United States of America* 105.49, pp. 19052–19059. ISSN: 10916490. DOI: 10.1073/pnas.0800375105.

Oloo, F., Safi, K., and Aryal, J. (2018). "Predicting Migratory Corridors of White Storks, Ciconia ciconia, to Enhance Sustainable Wind Energy Planning: A Data-Driven Agent-Based Model". In: *Sustainability* 10.5, p. 1470. ISSN: 2071-1050. DOI: 10.3390/su10051470.

Patlak, C. S. (1953). "Random walk with persistence and external bias". In: *The Bulletin of Mathematical Biophysics* 15.3, pp. 311–338. ISSN: 0007-4985. DOI: 10.1007/BF02476407.

Péron, G., Fleming, C. H., Duriez, O., Fluhr, J., Itty, C., Lambertucci, S., Safi, K., Shepard, E. L., and Calabrese, J. M. (2017). "The energy landscape predicts flight height and wind turbine collision hazard in three species of large soaring raptor". In: *Journal of Applied Ecology* 54.6, pp. 1895–1906. ISSN: 13652664. DOI: 10.1111/1365-2664.12909.

Poucet, B., Lenck-Santini, P.-P., and Save, E. (2003). "Drawing parallels between the behavioural and neural properties of navigation". In: *The Neurobiology of Spatial Behaviour*. Ed. by Jeffery, K. J. February 2014. Oxford: Oxford University Press. Chap. 10, pp. 187–198. ISBN: 9780198515241. DOI: 10.1093/acprof.

Powell, R. A. and Mitchell, M. S. (2012). "What is a home range?" In: *Journal of Mammalogy* 93.4, pp. 948–958. ISSN: 0022-2372. DOI: 10.1644/11-MAMM-S-177.1.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: http://www.r-project.org.

Sanderson, F. J., Donald, P. F., Pain, D. J., Burfield, I. J., and Van Bommel, F. P. (2006). "Long-term population declines in Afro-Palearctic migrant birds". In: *Biological Conservation* 131.1, pp. 93–105. ISSN: 00063207. DOI: 10.1016/j.biocon.2006.02.008.

Scacco, M., Duriez, O., Wikelski, M., and Safi, K. (in prep.). *Birds flight behaviour and static landscape features can map uplift availability across Europe*.

Sheather, S. J. and Jones, M. C. (1991). "A reliable data-based bandwidth selection method for kernel density estimation". In: *Journal of the Royal Statistical Society Series B (Statistical Methodology)* 53.3, pp. 683–690. ISSN: 0035-9246. DOI: 10.2307/2345597.

Shepard, E. L. C. and Lambertucci, S. A. (2013). "From daily movements to population distributions: weather affects competitive ability in a guild of soaring birds". In: *Journal of The Royal Society Interface* 10.88, pp. 20130612–20130612. ISSN: 1742-5689. DOI: 10.1098/rsif.2013.0612.

Sherub, S., Fiedler, W., Duriez, O., and Wikelski, M. (2017). "Bio-logging, new technologies to study conservation physiology on the move: a case study on annual survival of Himalayan vultures". In: *Journal of Comparative Physiology A* 203.6-7, pp. 531–542. ISSN: 0340-7594. DOI: 10.1007/s00359-017-1180-x.

Smouse, P. E., Focardi, S., Moorcroft, P. R., Kie, J. G., Forester, J. D., and Morales, J. M. (2010). "Stochastic modelling of animal movement". In: *Philosophical Transactions: Biological Sciences* 365.1550, pp. 2201–2211. ISSN: 09628436. DOI: 10.2307/25699241.

Song, Y. and Miller, H. J. (2014). "Simulating visit probability distributions within planar space-time prisms". In: *International Journal of Geographical Information Science* 28.1, pp. 104–125. ISSN: 1365-8816. DOI: 10.1080/13658816.2013.830308.

Srokowski, T. (2008). "Lévy flights in nonhomogeneous media: Distributed-order fractional equation approach". In: *Physical Review E* 78.3, p. 031135. ISSN: 1539-3755. DOI: 10.1103/PhysRevE.78.031135.

Stevens, A. and Othmer, H. G. (1997). "Aggregation, Blowup, and Collapse: The ABC's of Taxis in Reinforced Random Walks". In: *SIAM Journal on Applied Mathematics* 57.4, pp. 1044–1081. ISSN: 0036-1399. DOI: 10.1137/S0036139995288976.

Swisstopo (2018). *Digital height model, dhm25*. URL: https://www.swisstopo.admin.ch/en/home/products/height/dhm25.html (visited on 04/29/2018).

Taylor, G. and Blewitt, G. (2006). *Intelligent Positioning*. Chichester, UK: John Wiley & Sons, Ltd, p. 200. ISBN: 9780470035665. DOI: 10.1002/0470035668.

Technitis, G., Othman, W., Safi, K., and Weibel, R. (2015). "From A to B, randomly: a point-to-point random trajectory generator for animal movement". In: *International Journal of Geographical Information Science* 29.6, pp. 912–934. ISSN: 1365-8816. DOI: 10.1080/13658816.2014.999682.

Technitis, G., Weibel, R., Kranstauber, B., and Safi, K. (2016). "An algorithm for empirically informed random trajectory generation between two endpoints". In: *GIScience 2016: Ninth International Conference on Geographic Information Science*. Vol. 9. s.n., online. DOI: 10.5167/uzh-130652.

Tracey, J. A., Sheppard, J., Zhu, J., Wei, F., Swaisgood, R. R., and Fisher, R. N. (2014). "Movement-Based Estimation and Visualization of Space Use in 3D for Wildlife Ecology and Conservation". In: *PLoS ONE* 9.7. Ed. by Sueur, C., e101205. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0101205.

Van den Bossche, W., Berthold, P., Kaatz, M., Nowak, E., and Querner, U. (2002). "Eastern European White stork populations: migration studies and elaboration of conservation measures". In: *BfN - Skripten* 66, pp. 1–197.

Van Toor, M. L., Kranstauber, B., Newman, S. H., Prosser, D. J., Takekawa, J. Y., Technitis, G., Weibel, R., Wikelski, M., and Safi, K. (2018). "Integrating animal movement with habitat suitability for estimating dynamic migratory connectivity". In: *Landscape Ecology* 33.6, pp. 879–893. ISSN: 0921-2973. DOI: 10.1007/s10980-018-0637-9.

Van Winkle, W. (1975). "Comparison of Several Probabilistic Home-Range Models". In: *The Journal of Wildlife Management* 39.1, pp. 118–123.

Venables, W. N. and Ripley, B. D. (2003). "Modern Applied Statistics With S". In: *Technometrics* 45.1, pp. 111–111. ISSN: 0040-1706. DOI: 10.1198/tech.2003.s33.

Videler, J. J. (2006). "Bird flight modes". In: *Avian Flight*. Oxford University Press, pp. 122–160. DOI: 10.1093/acprof:oso/9780199299928.003.0006.

Vogel, S (2003). *Comparative Biomechanics: Life's Physical World*. 2nd ed. Princeton: Princeton University Press, p. 640. ISBN: 9780691112978.

Walther, B (2004). *List of Western Palearctic bird species migrating within Africa*. URL: http://www.zmuc.dk/verweb/staff/bawalther/migratoryBirdsList.htm (visited on 06/20/2018).

Wasserstein, R. L. and Lazar, N. A. (2016). "The ASA's Statement on p -Values: Context, Process, and Purpose". In: *The American Statistician* 70.2, pp. 129–133. ISSN: 0003-1305. DOI: 10.1080/00031305.2016.1154108.

Waterbird Population Estimates (2018). *Ciconia ciconia*. URL: http://wpe.wetlands.org (visited on 04/22/2018).

Weinzierl, R., Bohrer, G., Kranstauber, B., Fiedler, W., Wikelski, M., and Flack, A. (2016). "Wind estimation based on thermal soaring of birds". In: *Ecology and Evolution* 6.24, pp. 8706–8718. ISSN: 20457758. DOI: 10.1002/ece3.2585.

Weisstein, E. W. (2016). *Spherical Coordinates*. URL: http://mathworld.wolfram.com/SphericalCoordinates.html (visited on 04/19/2018).

Wentz, E. A., Campbell, A. F., and Houston, R. (2003). "A comparison of two methods to create tracks of moving objects: linear weighted distance and constrained random walk". In: *International Journal of Geographical Information Science* 17.7, pp. 623–645. ISSN: 1365-8816. DOI: 10.1080/1365881031000135492.

Yu, H. (2006). "Spatio-temporal GIS Design for Exploring Interactions of Human Activities". In: *Cartography and Geographic Information Science* 33.1, pp. 3–19. ISSN: 1523-0406. DOI: 10.1559/152304006777323136.

Zaburdaev, V., Denisov, S., and Klafter, J. (2015). "Lévy walks". In: *Reviews of Modern Physics* 87.2, pp. 483–530. ISSN: 0034-6861. DOI: 10.1103/RevModPhys.87.483.

# Appendix A

# Code

## R Package 'eRTG3D'

In this section of Appendix A the key functions of the *eRTG3D* package are listed. As the development of the package will be further continued, this code represents the state of the functions used to answer the research questions. Listing A.1 gives an overview on the specifications of the package at the time of submission of this thesis.

LISTING A.1: eRTG3D: *DESCRIPTION*

```
Package: eRTG3D
Title: Generate Empirically Informed Random Trajectories in 3-D
Version: 0.5.2
Authors@R: c(
    person("Merlin", "Unterfinger", email = "info@munterfinger.ch", role = c("aut", "cre")),
    person("Kamran", "Safi", email = "ksafi@orn.mpg.de", role = "aut"),
    person("George", "Technitis", email = "george.technitis@gmail.com", role = "aut"),
    person("Robert", "Weibel", email = "robert.weibel@geo.uzh.ch", role = "aut"))
URL: https://github.com/munterfinger/eRTG3D
Description: The empirically informed random trajectory generator in three dimensions (eRTG3D)
             is an algorithm to generate realistic random trajectories in a 3-D space
             between two given fix points in space. The trajectory generation is based on
             empirical distribution functions extracted from observed trajectories (training data)
             and thus reflects the geometrical movement characteristics of the mover.
Depends: R (>= 3.5.0)
Imports: CircStats (>= 0.2-4),
         doParallel (>= 1.0.11),
         ggplot2 (>= 2.2.1),
         gridExtra (>= 2.3),
         raster (>= 2.6-7),
         rasterVis (>= 0.45),
         parallel (>= 3.5.0),
         pbmcapply (>= 1.2.5),
         plyr (>= 1.8.4),
         plotly (>= 4.7.1),
         sp (>= 1.2-7)
License: GPL (>= 1.2-7)
Encoding: UTF-8
LazyData: true
RoxygenNote: 6.0.1
Suggests: knitr,
    rmarkdown,
    sf (>= 0.6-3)
VignetteBuilder: knitr
```

The current version of the package can be accessed as follows:

- Install package from inside R:
  `devtools::install_github("munterfinger/eRTG3D")`
- Download source: *www.github.com/munterfinger/eRTG3D*

LISTING A.2: eRTG3D: *stepTurnLiftHistogram()*

```
turnLiftStepHist <- function(turn, lift, step, printDims = TRUE, rm.zeros = TRUE, maxBin = 25)
{
  nx <- min(max(floor(2 * pi / .fd.bw(turn)), 12), maxBin)
  ny <- min(max(floor(2 * pi / .fd.bw(lift)), 12), maxBin)
  nz <- min(max(floor(max(step) / .fd.bw(step)), 12), maxBin)
  if(printDims){message("  |TLD cube dimensions: ", nx, " x ", ny, " x ", nz)}
  tCuts <- .cutMidpoints(turn, nx); lCuts <- .cutMidpoints(lift, ny); dCuts <- .cutMidpoints(step, nz)
  h <- list(turn=tCuts[[1]],
           lift=lCuts[[1]],
           step=dCuts[[1]])
  h <- do.call(data.frame, h)
  h <- as.data.frame(table(h))
  tRes <- tCuts[[2]]; lRes <- lCuts[[2]]; dRes <- dCuts[[2]];
  colnames(h)[4] <- "prob"
  if (rm.zeros) {h <- h[!h$prob==0, ]}
  h$prob <- h$prob/sum(h$prob)
  h[1:3] <- lapply(h[1:3], function(x) {as.numeric(levels(x))[x]})
  return(list(values = h, tRes = tRes, lRes = lRes, dRes = dRes))
}
```

LISTING A.3: eRTG3D: *get.densities.3d()*

```
get.densities.3d <- function(turnAngle, liftAngle, stepLength, deltaLift, deltaTurn, deltaStep,
                             gradientAngle = NULL, heightEllipsoid = NULL, heightTopo = NULL, maxBin = 25)
{
  cubeTLD <- turnLiftStepHist(turn = turnAngle, lift = liftAngle, step = stepLength, maxBin = maxBin)
  autoT <- approxfun(density.default(deltaTurn))
  autoL <- approxfun(density.default(deltaLift))
  autoD <- approxfun(density.default(deltaStep))
  if (!is.null(gradientAngle)) {
    gDens <- approxfun(density.default(gradientAngle[gradientAngle > 0 & gradientAngle < pi]))
  } else {
      gDens <- function(x){return(as.numeric(x > 0 & x < pi))}
  }
  if (!is.null(heightEllipsoid)) {
    hDistEllipsoid <- approxfun(density.default(heightEllipsoid))
  } else {
      hDistEllipsoid <- function(x){1}
  }
  if (!is.null(heightTopo)) {
    hDistTopo <- approxfun(density.default(heightTopo))
  } else {
      hDistTopo <- function(x){1}
  }
  return(list(tldCube = cubeTLD, autoT = autoT, autoL = autoL, autoD = autoD, gDens = gDens,
             hDistEllipsoid = hDistEllipsoid, hDistTopo=hDistTopo))
}
```

LISTING A.4: eRTG3D: *get.track.densities.3d()*

```
get.track.densities.3d <- function(track, gradientDensity = TRUE, heightDistEllipsoid = TRUE,
                                   DEM = NULL, maxBin = 25)
{
  .is.df.xyz(track)
  track <- track.properties.3d(track)
  turnAngle <- track$t[2:nrow(track)]; liftAngle <- track$l[2:nrow(track)]; stepLength <- track$d[2:nrow(track)]
  deltaTurn <- diff(turnAngle); deltaLift <- diff(liftAngle); deltaStep <- diff(stepLength)
  if (gradientDensity) {
    gradientAngle <- track$g} else {gradientAngle <- NULL}
  if (heightDistEllipsoid) {heightEllipsoid <- track$z} else {heightEllipsoid <- NULL}
  if (!is.null(DEM)) {
    .check.extent(DEM = DEM, track = track)
    heightTopo <- track$z - raster::extract(DEM, track[,1:2])
  } else {heightTopo <- NULL}
  return(get.densities.3d(turnAngle = turnAngle, liftAngle = liftAngle, stepLength = stepLength,
                          deltaLift = deltaLift, deltaTurn = deltaTurn, deltaStep = deltaStep,
                          gradientAngle = gradientAngle, heightEllipsoid = heightEllipsoid,
                          heightTopo = heightTopo, maxBin = maxBin))
}
```

LISTING A.5: eRTG3D: *track.split.3d()*

```
track.split.3d <- function(track, timeLag, lag = NULL, tolerance = NULL)
{
  .is.df.xyz(track)
```

```r
  if (any(is.na(timeLag))) stop("TimeLag is not allowed to contain NAs.")
  if (is.null(lag)) {
    m <- mean(timeLag)
  } else {m <- lag}
  if (is.null(tolerance)) {
    tolerance <- 0.5 * sd(timeLag)
  }
  splitRows <- which(abs(m-timeLag) > tolerance)
  trackSections <- split(track, cumsum(1:nrow(track) %in% (splitRows+2)))
  nSplits <- length(splitRows); nChange <- round(sum(timeLag[splitRows]/m-1));
  message(paste("  |Mean time lag: ", round(m,2) , ", tolerance: ", round(tolerance,2),
                ", number of splits: ", nSplits, ", proposed change in steps: ", nChange, sep=""))
  return(trackSections)
}
```

LISTING A.6: eRTG3D: *get.section.densities.3d()*

```r
get.section.densities.3d <- function(trackSections, gradientDensity = TRUE, heightDistEllipsoid = TRUE,
                                     DEM = NULL, maxBin = 25)
{
  trackSections <- lapply(X=trackSections, FUN= function(X) track.properties.3d(X)[2:nrow(X), ])
  deltaTurn <- Reduce(c, lapply(X = trackSections, FUN = function(X) diff(X$t)))
  deltaLift <- Reduce(c, lapply(X = trackSections, FUN = function(X) diff(X$l)))
  deltaStep <- Reduce(c, lapply(X = trackSections, FUN = function(X) diff(X$d)))
  trackSections <- do.call(rbind, trackSections)
  turnAngle <- trackSections$t; liftAngle <- trackSections$l; stepLength <- trackSections$d
  if (gradientDensity) {gradientAngle <- trackSections$g} else {gradientAngle <- NULL}
  if (heightDistEllipsoid) {heightEllipsoid <- trackSections$z} else {heightEllipsoid <- NULL}
  if (!is.null(DEM)) {
    .check.extent(DEM = DEM, track = trackSections)
    heightTopo <- trackSections$z - raster::extract(DEM, trackSections[,1:2])
  } else {heightTopo <- NULL}
  return(get.densities.3d(turnAngle = turnAngle, liftAngle = liftAngle, stepLength = stepLength,
                          deltaLift = deltaLift, deltaTurn = deltaTurn, deltaStep = deltaStep,
                          gradientAngle = gradientAngle, heightEllipsoid = heightEllipsoid,
                          heightTopo = heightTopo, maxBin = maxBin))
}
```

LISTING A.7: eRTG3D: *sim.uncond.3d()*

```r
sim.uncond.3d <- function(n.locs, start=c(0,0,0), a0, g0, densities, error = TRUE)
{
  # progress bar and time
  message(paste("  |Simulate UERW with ", n.locs, " steps", sep = ""))
  start.time <- Sys.time()
  pb <- txtProgressBar(min = 0, max = n.locs, style = 3)
  ui <- floor(n.locs/20)+1
  # get coordinates of the tldCube
  ts <- densities$tldCube$values$turn
  ls <- densities$tldCube$values$lift
  ds <- densities$tldCube$values$step
  # get probs for each turn-lift-distance combination
  tldProbs <- densities$tldCube$values$prob
  sCond <- sample(1:nrow(densities$tldCube$values), 1, prob=tldProbs)
  # "x" "y" "z" "a" "g" "t" "l" "d" "p"
  # "1" "2" "3" "4" "5" "6" "7" "8" "9"
  RTG <- matrix(0, n.locs, 9)
  RTG[1,] <- c(start[1], start[2], start[3], a0, g0, ts[sCond], ls[sCond], ds[sCond], NA)
  # create random noise if error is TRUE (uniform distributed)
  if (error) {
    tShift <- runif(n.locs, -densities$tldCube$tRes / 2, densities$tldCube$tRes / 2)
    lShift <- runif(n.locs, -densities$tldCube$lRes / 2, densities$tldCube$lRes / 2)
    dShift <- runif(n.locs, -densities$tldCube$dRes / 2, densities$tldCube$dRes / 2)
  } else {
    tShift <- lShift <- dShift <- numeric(n.locs)
  }
  for (i in 2:n.locs)
  {
    # get influence of current autodifferences
    atProbs <- densities$autoT(RTG[i-1, 6] - ts + tShift[i])
    alProbs <- densities$autoL(RTG[i-1, 7] - ls + lShift[i])
    adProbs <- densities$autoD(RTG[i-1, 8] - ds + dShift[i])
    atProbs[is.na(atProbs)] <- 0
    alProbs[is.na(alProbs)] <- 0
    adProbs[is.na(adProbs)] <- 0
    atProbs <- atProbs / sum(atProbs)
    alProbs <- alProbs / sum(alProbs)
    adProbs <- adProbs / sum(adProbs)
    # multiply and take the third squareroot
    pProbs <- tldProbs * (atProbs * alProbs * adProbs)^(1/3)
    # apply gradient distribution or if gradientDensity is set to FALSE in get.densities.3d(),
    # limit gradient to 0-pi otherwise.
    gAll <- (RTG[i-1, 5] + ls + lShift[i])
    gProbs <- densities$gDens(gAll)
    gProbs[is.na(gProbs)] <- 0
    gProbs <- gProbs / sum(gProbs)
    pProbs <- pProbs * gProbs
    # sample on turnLiftStepHist = tldCube and add shifts
    rP <- sample(1:nrow(densities$tldCube$values), size = 1, prob = pProbs)
    t <- ts[rP] + tShift[i]
    l <- ls[rP] + lShift[i]
    d <- ds[rP] + dShift[i]
    p <- pProbs[rP]
    # absolute spherical orientation, wrap angles around -pi-0 & 0-pi
    a <- .wrap(RTG[i-1, 4] + t)
    g <- .wrap(RTG[i-1, 5] + l)
    # new coordinates of the next step
    x <- (d * sin(g) * cos(a)) + RTG[i-1, 1]
    y <- (d * sin(g) * sin(a)) + RTG[i-1, 2]
    z <- (d * cos(g)) + RTG[i-1, 3]
    # "x" "y" "z" "a" "g" "t" "l" "d" "p"
    RTG[i,] <- c(x, y, z, a, g, t, l, d, p)
    # update progress bar
    if ((i %% ui) == 0) {setTxtProgressBar(pb, i)}
  }
  rownames(RTG) <- c()
  colnames(RTG) <- c("x", "y", "z", "a", "g", "t", "l", "d", "p")
  # close progress bar
  setTxtProgressBar(pb, n.locs)
  close(pb)
  message(paste("  |Runtime: ", round(as.numeric(Sys.time()) - as.numeric(start.time), 2), " secs", sep = ""))
  return(as.data.frame(RTG))
}
```

LISTING A.8: eRTG3D: *qProb.3d()*

```
qProb.3d <- function(sim, n.locs, multicore = FALSE, maxBin = 25)
{
  if (multicore) {
    if(.Platform$OS.type == "unix") {return(.qProb.3d.unix(sim, n.locs, maxBin = maxBin))}
    if(.Platform$OS.type == "windows") {
      return(suppressWarnings(.qProb.3d.windows(sim, n.locs, maxBin = maxBin)))}
  } else {
    start.time <- Sys.time()
    message(paste("  |Extracting Q probabilities for ", n.locs, " steps", sep = ""))
    # steps minus 2
    nSteps <- n.locs - 2
    # progress bar
    pb <- txtProgressBar(min = 0, max = nSteps, style = 3)
    # lift angles to target as a function of number of steps
    cubeList <- lapply(1:nSteps, function(x) {
      # update progressbar
      setTxtProgressBar(pb, x)
      # turn angle, lift angles and distance to target as a function of number of steps
      t <- .wrap(atan2(diff(sim$y, lag = x), diff(sim$x, lag = x)) - sim$a[1:(length(sim$a) - x)])
      l <- .wrap(atan2(sqrt(diff(sim$x, lag = x) ^ 2 + diff(sim$y, lag = x) ^ 2),
                       diff(sim$z, lag = x)) - sim$g[1:(length(sim$g) - x)])
      d <- sqrt(diff(sim$x, lag = x) ^ 2 + diff(sim$y, lag = x) ^ 2 + diff(sim$z, lag = x) ^ 2)
      # the Qprob is thinned to the lag that suggests breaking off of the autocorrelation
      # of the turning angle to target, the lift angle to target and the distance to target
      # for the relevant number of steps. This is mainly to reduce redundancy introduced
      # by the sliding window approach adopted in estimating the relationships
      k <- max(head(which(acf(t, lag.max = nSteps, plot = FALSE)$acf < 0.05),1)-1,
               head(which(acf(l, lag.max = nSteps, plot = FALSE)$acf < 0.05),1)-1,
               head(which(acf(d, lag.max = nSteps, plot = FALSE)$acf < 0.05),1)-1)
      t <- t[seq(1, length(t), by = k)]
      l <- l[seq(1, length(l), by = k)]
      d <- d[seq(1, length(d), by = k)]
      # get stepTurnLiftHistograms
      return(turnLiftStepHist(turn=t, lift=l, step=d, printDims = FALSE, rm.zeros = TRUE, maxBin = maxBin))
    })
    setTxtProgressBar(pb, nSteps)
    close(pb)
    message(paste("  |Runtime: ", round(as.numeric(Sys.time()) - as.numeric(start.time), 2), " secs", sep = ""))
    return(rev(cubeList))
  }
}
```

LISTING A.9: eRTG3D: *turn2target.3d()*

```
turn2target.3d <- function(track)
{
  .is.df.xyz(track = track)
  track <- track.properties.3d(track)
  target <- Reduce(c, track[nrow(track), 1:3])
  .wrap(atan2(target[2]-track$y, target[1] - track$x) - track$a)
}
```

LISTING A.10: eRTG3D: *lift2target.3d()*

```
lift2target.3d <- function(track)
{
  .is.df.xyz(track = track)
  track <- track.properties.3d(track)
  target <- Reduce(c, track[nrow(track), 1:3])
  .wrap(atan2(sqrt((target[1]-track$x) ^ 2 + (target[2]-track$y) ^ 2),
              (target[3]-track$z)) - track$g)
}
```

LISTING A.11: eRTG3D: *dist2target.3d()*

```
dist2target.3d <- function(track)
{
  .is.df.xyz(track = track)
  target <- Reduce(c, track[nrow(track), 1:3])
  sqrt((target[1]-track$x) ^ 2 + (target[2]-track$y) ^ 2 + (target[3]-track$z) ^ 2)
}
```

LISTING A.12: eRTG3D: *sim.cond.3d()*

```
sim.cond.3d <- function(n.locs, start=c(0,0,0), end=start, a0, g0, densities, qProbs,
                        error = FALSE, DEM = NULL, BG = NULL)
{
  start.time <- Sys.time()
  if(!is.null(DEM)) {
    .check.extent(DEM = DEM, track = data.frame(rbind(start, end)))
  }
  if(!is.null(BG)) {
    .check.extent(DEM = BG, track = data.frame(rbind(start, end)))
  }
  # progress bar and time
  message(paste("  |Simulate CERW with ", n.locs, " steps", sep = ""))
  pb <- txtProgressBar(min = 0, max = n.locs-2, style = 3)
  ui <- floor(n.locs/20)+1
  # replace the probability distribution for step length 1 by the one from
  # the qProbs since that one relies on more samples derived from sim
  densities[[1]] <- tail(qProbs,1)[[1]]
  # get the coordinates of the step length and turning angle bin centres
  names(start) <- c("x", "y", "z")
  names(end) <- c("x", "y", "z")
  # get coordinates of the tldCube
  ts <- densities$tldCube$values$turn
  ls <- densities$tldCube$values$lift
  ds <- densities$tldCube$values$step
  # get probs for each combination
  tldProbs <- densities$tldCube$values$prob
  # sample one randomly to set the initial conditions
  # for the previous to first turn and previous to first step
  # for the start point, as this is needed to inform the auto-difference
  # likelihood
  sCond <- sample(1:nrow(densities$tldCube$values), 1, prob=tldProbs)
  # "x" "y" "z" "a" "g" "t" "l" "d" "p"
  # "1" "2" "3" "4" "5" "6" "7" "8" "9"
  RTG <- matrix(0, n.locs, 9)
  RTG[1, ] <- c(start[1], start[2], start[3], a0, g0, ts[sCond], ls[sCond], ds[sCond], NA)
  # Create random noise if error is TRUE
  if (error) {
    tShift <- runif(n.locs - 2, -densities$tldCube$tRes / 2, densities$tldCube$tRes / 2)
    lShift <- runif(n.locs - 2, -densities$tldCube$lRes / 2, densities$tldCube$lRes / 2)
    dShift <- runif(n.locs - 2, -densities$tldCube$dRes / 2, densities$tldCube$dRes / 2)
  } else {
    tShift <- lShift <- dShift <- numeric(n.locs - 2)
  }
  # start creating the track step for step
  for (i in 1:(n.locs - 2))
```

```
{
  # get the auto-difference probability for turning angle
  atProbs <- densities$autoT(RTG[i, 6] - ts + tShift[i])
  # get the auto-difference probability for lift angle
  alProbs <- densities$autoL(RTG[i, 7] - ls + lShift[i])
  # get the auto-difference probability for step length
  adProbs <- densities$autoD(RTG[i, 8] - ds + dShift[i])
  # set NAs to zero probability
  atProbs[is.na(atProbs)] <- 0
  alProbs[is.na(alProbs)] <- 0
  adProbs[is.na(adProbs)] <- 0
  # standardize the probabilities to sum to one
  atProbs <- atProbs / sum(atProbs)
  alProbs <- alProbs / sum(alProbs)
  adProbs <- adProbs / sum(adProbs)
  # calculate the probability to make a step forward. The auto-difference probabilities are
  # calculated as one jointly contributing probability and therefore square rooted befor
  # multiplication with the two dimensional probability distribution
  P <- (tldProbs) * (atProbs * alProbs * adProbs)^(1/3)
  # calculate the azimuth
  a <- .wrap(RTG[i, 4] + ts + tShift[i])
  # calculate the gradient
  g <- .wrap(RTG[i, 5] + ls + lShift[i])
  # convert the coordinates from step length turning angle dimension
  x1 <- ((ds + dShift[i]) * sin(g) * cos(a)) + RTG[i, 1]
  y1 <- ((ds + dShift[i]) * sin(g) * sin(a)) + RTG[i, 2]
  z1 <- ((ds + dShift[i]) * cos(g)) + RTG[i, 3]
  # calculate the distances of the cell centers in the spatial domain
  # to the target (last location of the empirical track)
  endD <- as.numeric(sqrt((end[1] - x1) ^ 2 + (end[2] - y1) ^ 2 + (end[3] - z1) ^ 2))
  # calculate the azimuth of the cell centres to the target and substract from it the direction of arrival
  # resulting in turning angle towards target
  endT <- as.numeric(.wrap(atan2(as.numeric(end[2] - y1), as.numeric(end[1] - x1)) - a))
  # calculate the gradient of the possibilite steps to the target and substract from it the angle of arrival
  # resulting in turning angle towards target
  endL <- as.numeric(.wrap(atan2(as.numeric(sqrt((end[1] - x1) ^ 2 + (end[2] - y1) ^ 2)),
                                 as.numeric(end[3] - z1))) - g)
  # get the probabilities of making it distance and turning angle wise
  # which is derived from the two dimensional probability distribution for the
  # appropriate step being modelled
  # get possible coordinates
  qCube <- qProbs[[i]]
  tVal <- unique(qCube$values$turn)
  lVal <- unique(qCube$values$lift)
  dVal <- unique(qCube$values$step)
  # find closest coordinates
  tCoords <- unlist(lapply(endT, function(x) tVal[which.min(abs(tVal-x))]))
  lCoords <- unlist(lapply(endL, function(x) lVal[which.min(abs(lVal-x))]))
  dCoords <- unlist(lapply(endD, function(x) dVal[which.min(abs(dVal-x))]))
  # extract Q
  Q <- unlist(lapply(1:length(tCoords), function(x){
    test <- (qCube$values$turn == tCoords[x] &
             qCube$values$lift == lCoords[x] &
             qCube$values$step == dCoords[x]);
    if(any(test==TRUE)) {return(qCube$values$prob[test])} else {return(0)}
  }))
  # the overall probability is the product of the probability
  # of making a step forward and the probability of making it to the
  # target. The weight of the target probability needs to be adjusted
  # by division by end distance, because the number of cells to choose from
  # are increasing with distance to target, which needs to be accounted
  # for prior to sampling based on overall probability
  Probs <- P * Q / endD
  # apply gradient distribution or if gradientDensity is set to FALSE in get.densities.3d(),
  # limit gradient to 0-pi otherwise.
  gProbs <- densities$gDens(g)
  gProbs[is.na(gProbs)] <- 0
  gProbs <- gProbs / sum(gProbs)
  Probs <- Probs * gProbs
  # account for probable flight height, if a DEM is provided the relative flight height is taken
  # otherwise only the absolute ellipsoid height.
  if(!is.null(DEM))
  {
    surface <- raster::extract(DEM, cbind(x1, y1))
    demP <-  densities$hDistTopo(z1 - surface) * as.numeric(z1 >= surface)
    demP[is.na(demP)] <- 0
    demP <- demP / sum(demP)
    hProb <- densities$hDistEllipsoid(z1)
    hProb[is.na(hProb)] <- 0
    hProb <- hProb / sum(hProb)
    Probs <- Probs * sqrt(demP * hProb)
    } else {
      hProb <- densities$hDistEllipsoid(z1)
      hProb[is.na(hProb)] <- 0
      hProb <- hProb / sum(hProb)
      Probs <- Probs * hProb
    }
    # use the coordinates of the spatial grid for which the probabilities are calculated
    # and use it to overlay a background raster for example to avoid walks in certain areas
    if(!is.null(BG))
    {
      bgP <- raster::extract(BG, cbind(x1, y1))
      Probs <- Probs * bgP
    }
    # make sure we have no missing nor negative probabilities
    Probs[is.na(Probs)] <- 0
    Probs[Probs <= 0] <- 0
    # check whether the run might have ended up in a dead-end,
    # which will set the zero probability status to TRUE
    if(all(Probs==0)){
      RTG <- NULL
      close(pb)
      message(paste("|Runtime: ", round(as.numeric(Sys.time()) - as.numeric(start.time), 2), " secs", sep = ""))
      warning("Dead end encountered.")
      return(RTG)
    }else{
      # draw a point randomly based on the probability
      rP <- sample.int(nrow(densities$tldCube$values), size = 1, prob = Probs)
      # "x" "y" "z" "a" "g" "t" "l" "d" "p"
      # "1" "2" "3" "4" "5" "6" "7" "8" "9"
      RTG[i + 1, ] <- c(x1[rP], y1[rP], z1[rP], a[rP], g[rP], ts[rP], ls[rP], ds[rP], Probs[rP])
      # update progress bar
      if ((i %% ui) == 0) {setTxtProgressBar(pb, i)}
    }
  }
  # the track is forced to target location and the appropriate distance is added
  RTG[1, 8] <- NA
  RTG[n.locs,] <- c(end[1], end[2], end[3], NA, NA, NA, NA, NA, NA)
  RTG[n.locs, 8] <- sqrt((RTG[n.locs, 1] - RTG[n.locs-1, 1])^2 +
                         (RTG[n.locs, 2] - RTG[n.locs-1, 2])^2 +
                         (RTG[n.locs, 3] - RTG[n.locs-1, 3])^2)
  # Stop if the step length of the last step is larger than the largest possible step
  if(RTG[n.locs, 8] > max(densities$tldCube$values$step, na.rm = TRUE)*sqrt(2)) {
    RTG <- NULL
    close(pb)
    message(paste("|Runtime: ", round(as.numeric(Sys.time()) - as.numeric(start.time), 2), " secs", sep = ""))
    warning("Dead end encountered in last step.")
    return(RTG)
  }
  rownames(RTG) <- c()
  colnames(RTG) <- c("x", "y", "z", "a", "g", "t", "l", "d", "p")
  # close progress bar
  setTxtProgressBar(pb, i)
  close(pb)
  message(paste("  |Runtime: ", round(as.numeric(Sys.time()) - as.numeric(start.time), 2), " secs", sep = ""))
  return(as.data.frame(RTG))
}
```

LISTING A.13: eRTG3D: *reproduce.track.3d()*

```
reproduce.track.3d <- function(track, n.sim = 1, multicore = FALSE, error = TRUE,
                               DEM = NULL, BG = NULL, filterDeadEnds = TRUE, plot2d = FALSE,
                               plot3d = FALSE, maxBin = 25, gradientDensity = TRUE)
{
  .is.df.xyz(track = track)
  track <- track.properties.3d(track)
  n.locs <- nrow(track)
  if (n.locs>1500) stop("Track is too long (>1500 steps).")
  turnAngle <- track$t[2:nrow(track)]; liftAngle <- track$l[2:nrow(track)]; stepLength <- track$d[2:nrow(track)]
  deltaTurn <- diff(turnAngle); deltaLift <- diff(liftAngle); deltaStep <- diff(stepLength)
  heightEllipsoid <- track$z
  if (gradientDensity) {gradientAngle <- track$g} else {gradientAngle <- NULL}
  if (!is.null(DEM)) {
    .check.extent(DEM = DEM, track = track)
    heightTopo <- track$z - raster::extract(DEM, track[,1:2])
  } else {heightTopo <- NULL}
  D <- get.densities.3d(liftAngle = liftAngle, turnAngle = turnAngle, stepLength = stepLength,
                        deltaLift = deltaLift, deltaTurn = deltaTurn, deltaStep = deltaStep,
                        heightEllipsoid = heightEllipsoid, heightTopo = heightTopo,
                        gradientAngle = gradientAngle, maxBin = maxBin)
  uerw <- sim.uncond.3d(n.locs*1500, start = c(track$x[1],track$y[1],track$z[1]),
                        a0 = track$a[1], g0 = track$g[1], densities = D, error = error)
  Q <- qProb.3d(uerw, n.locs, multicore = multicore, maxBin = maxBin)
  cerwList <- suppressWarnings(n.sim.cond.3d(n.sim = n.sim, n.locs <- n.locs,
                                             start=c(track$x[1],track$y[1],track$z[1]),
                                             end=c(track$x[n.locs],track$y[n.locs],track$z[n.locs]),
                                             a0 = track$a[1], g0 = track$g[1], densities=D, qProbs=Q,
                                             error = error, multicore = multicore, DEM = DEM, BG = BG))
  if(filterDeadEnds){cerwList <- filter.dead.ends(cerwList)}
  if(plot2d){print(plot2d(origTrack = track, cerwList = cerwList, DEM = DEM))}
  if(plot3d){plot3d(origTrack = track, cerwList = cerwList, DEM = DEM)}
  return(cerwList)
}
```

LISTING A.14: eRTG3D: *sim.crw.3d()*

```
sim.crw.3d <- function(nStep, rTurn, rLift, meanStep, start = c(0,0,0))
{
  # correlated angles
  t <- CircStats::rwrpnorm(n = nStep - 2, mu = 0, rho = rTurn)
  a <- .wrap(cumsum(c(runif(1, 0, 2 * pi), t)))
  l <- CircStats::rwrpnorm(n = nStep - 2, mu = 0, rho = rLift)
  g <- abs(.wrap(cumsum(c(runif(1, 0, pi), l))))
  f <- abs(scale(CircStats::rwrpnorm(n = nStep - 1, mu = 0, rho = (rTurn+rLift)/2))[,1])
  d <- rep(meanStep, nStep-1) * f
  # deltas in all 3 directions
  dx <- (d * sin(g) * cos(a))
  dy <- (d * sin(g) * sin(a))
  dz <- (d * cos(g))
  # generate track
  t <- data.frame(
    x = cumsum(c(start[1], dx)),
    y = cumsum(c(start[2], dy)),
    z = cumsum(c(start[3], dz))
  )
  return(t)
}
```

LISTING A.15: eRTG3D: *test.verification.3d()*

```
test.verification.3d <- function(track1, track2, alpha = 0.05, plot = FALSE, test = "ks")
{
  if (!any(test == c("ks", "ttest"))) stop("The variable 'test' must either be 'ks' or 'ttest'.")
  if (!is.list(track1) || !is.list(track2)) stop("Track input has to be of type list or data.frame.")
  if (is.list(track1) && is.data.frame(track1)) {track1 <- list(track1)}
  if (is.list(track2) && is.data.frame(track2)) {track2 <-list(track2)}
  track1 <- filter.dead.ends(track1); track2 <- filter.dead.ends(track2)
  # track(s) 1
  track1 <- lapply(track1, function(x){track.properties.3d(x)[2:nrow(x), ]})
  difftrack1 <- do.call("rbind", lapply(track1,function(x){data.frame(diffT = diff(x$t),
                                                                      diffL = diff(x$l),
                                                                      diffD = diff(x$d))}))
  track1 <- do.call("rbind", track1)
  t1 <- track1$t; l1 <- track1$l; d1 <- track1$d;
  diffT1 <- difftrack1$diffT; diffL1 <- difftrack1$diffL; diffD1 <- difftrack1$diffD;
  # track(s) 2
  track2 <- lapply(track2, function(x){track.properties.3d(x)[2:nrow(x), ]})
  diffTrack2 <- do.call("rbind", lapply(track2, function(x){data.frame(diffT = diff(x$t),
                                                                       diffL = diff(x$l),
                                                                       diffD = diff(x$d))}))
  track2 <- do.call("rbind", track2)
  t2 <- track2$t; l2 <- track2$l; d2 <- track2$d;
  diffT2 <- diffTrack2$diffT; diffL2 <- diffTrack2$diffL; diffD2 <- diffTrack2$diffD;
  if (test == "ks"){
    message("  |*** Two-sample Kolmogorov-Smirnov test ***")
    message("  |H0: Probability distributions do not differ significantly")
    message("  |H1: Probability distributions differ significantly")
    # turn
    turnT <- suppressWarnings(ks.test(t1, t2, alternative = "two.sided"))
    diffTurnT <- suppressWarnings(ks.test(diffT1, diffT2, alternative = "two.sided"))
    # lift
    liftT <- suppressWarnings(ks.test(l1, l2, alternative = "two.sided"))
    diffLiftT <- suppressWarnings(ks.test(diffL1, diffL2, alternative = "two.sided"))
    # step
    stepT <- suppressWarnings(ks.test(d1, d2, alternative = "two.sided"))
    diffStepT <- suppressWarnings(ks.test(diffD1, diffD2, alternative = "two.sided"))
    # print results
    message(paste("  |Turn angle - ", .test2text(turnT, alpha), ", autodifferences - ",
                  .test2text(diffTurnT, alpha), sep=""))
    message(paste("  |Lift angle - ", .test2text(liftT, alpha), ", autodifferences - ",
                    .test2text(diffLiftT, alpha), sep=""))
    message(paste("  |Step length - ", .test2text(stepT, alpha), ", autodifferences - ",
                    .test2text(diffStepT, alpha), sep=""))
    if (plot) {
      suppressWarnings(plot3d.multiplot(
        .plot3d.density(t1, t2, titleText = "Turn angle"),
        .plot3d.density(l1, l2, titleText = "Lift angle"),
        .plot3d.density(d1, d2, titleText = "Step length"),
        cols = 1
      ))
    }
    return(list(turnT, liftT, stepT, diffTurnT, diffLiftT, diffStepT))
  }
  if (test == "ttest"){
    message("  |*** One Sample t-test ***")
    message("  |H0: Difference between tracks does not differ significantly from 0")
    message("  |H1: Difference between tracks differs significantly from 0")
    nSample <- min(nrow(track1), nrow(track2))
    # turn
    turnT <- suppressWarnings(t.test(diffT <- (sample(t1, nSample)-sample(t2, nSample)), mu = 0,
                                     alternative = "two.sided"))
    # lift
    liftT <- suppressWarnings(t.test(diffL <- (sample(l1, nSample)-sample(l2, nSample)), mu = 0,
                                     alternative = "two.sided"))
    # step
    stepT <- suppressWarnings(t.test(diffD <- (sample(d1, nSample)-sample(d2, nSample)), mu = 0,
                                     alternative = "two.sided"))
    # print results
    message(paste("  |Turn angle  - ", .test2text(turnT, alpha), sep=""))
    message(paste("  |Lift angle  - ", .test2text(liftT, alpha), sep=""))
    message(paste("  |Step length - ", .test2text(stepT, alpha), sep=""))
    if (plot) {
      suppressWarnings(plot3d.multiplot(
        .plot3d.density(diffT, titleText = "Mean difference turn angle"),
        .plot3d.density(diffL, titleText = "Mean difference Lift angle"),
        .plot3d.density(diffD, titleText = "Mean difference Step length"),
        cols = 1
      ))
    }
    return(list(turnT, liftT, stepT))
  }
}
```

LISTING A.16: eRTG3D: *movingMedian()*

```
movingMedian <- function(data, window){
  if(!(window %% 2 == 0)) {window <- floor(window/2)} else{stop("Window must be an uneven number.")}
  total <- length(data)
  result <- vector(length = total)
  for(i in (window+1):(total-window)){
    result[i] <- median(data[(i-window):(i+window)])
  }
  result[1:window] <- median(data[1:window])
  result[(total-window):total] <- median(data[(total-window):total])
  return(result)
}
```

LISTING A.17: eRTG3D: *voxelCount()*

```
voxelCount <- function(points, extent, xyRes, zRes = xyRes, zMin, zMax, standartize = FALSE){
  rTem <- raster::raster(extent, res=xyRes)
  rTem[] <- 0
  rStack <- raster::stack()
  for (i in 1:round((zMax-zMin)/zRes)) {
    cat('\r', paste("  |Counting points in Voxels for height: ", zMin+(i-1)*zRes, "m - ",
                    (zMin+i*zRes), "m ...", sep = ""))
    flush.console()
    p <- points[points[,3] > (zMin+(i-1)*zRes) & points[,3] < (zMin+i*zRes), ]
    if (!nrow(p) == 0) {
      p <- sp::SpatialPoints(coords = cbind(p[,1], p[,2]))
      r <- raster::rasterize(p, rTem, fun='count')
      r[is.na(r[])] <- 0
      rStack <- raster::stack(rStack, r)
    } else {
      rStack <- raster::stack(rStack, rTem)
    }
    names(rStack)[i] <- c(paste("m", zMin+(i-1)*zRes, "-", (zMin+i*zRes), sep = ""))
  }
  if(standartize){
    maxR <- max(raster::maxValue(rStack))
    minR <- min(raster::minValue(rStack))
    for(i in 1:length(rStack@layers)) {
      rStack@layers[[i]] <- (rStack@layers[[i]] - minR) / (maxR - minR)
    }
  }
  cat('\r', "   |Done.\n")
  flush.console()
  return(rStack)
}
```

# High Performance Computing Cluster

LISTING A.18: HPC cluster: *simTracks.R*

```
#!/usr/bin/env /mpcdf/soft/SLES122/common/R/3.4.2/gcc-6.3_mkl-2017/bin/Rscript

#SBATCH --job-name=simTracks
#SBATCH --output=log/simTracks_%a.o.log
#SBATCH --error=log/simTracks_%a.e.log
#SBATCH --time=12:00:00
#SBATCH --nodes=1
#SBATCH --array=1-100
#SBATCH --ntasks=64
#SBATCH --partition=general
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=1024

#####################################################################################
##    Simulation of soaring gliding tracks of white storcks (CERWs) crossing ZRH    ##
##                            Master's Thesis - eRTG3D                               ##
##                                 19/3/2018                                         ##
##            M. Unterfinger, Geography UZH (merlin.unterfingeruzh.ch)               ##
#####################################################################################

# Load library and set up data needed
library(eRTG3D)
load(file = "data/clusterData.RData")

# Set random seed
seed <- sample(1000000:9999999, 1)
set.seed(seed)

# Set simulation parameters
n.sim <- 1000
limit <- 25
nCores <- 64

# Simulate high resolution soaring gliding tracks of white storcks (CERWs) crossing ZRH
cat("Simulation of ", n.sim, " tracks on ", nCores, " Cores, with seed = ", seed, "\n" , sep = "")
t <- Sys.time()
glidingSoaringList <- parallel::mclapply(X = 1:n.sim, FUN = function(X){
  simTrack <- NULL
  while(is.null(simTrack)){
    start <- startVec.120[sample(1:nrow(startVec.120), 1), ]
    end <- endVec.120[sample(1:nrow(endVec.120), 1), ]
    capture.output(simTrack <- suppressMessages(sim.cond.3d(sim.locs.120, start = start, end = end,
                              a0 = a0.120, g0 = g0.120, densities = D.120,
                              qProbs = Q.120, error = TRUE, DEM = dem, BG = thermals)))
    if(!is.null(simTrack)) {
      inside <- dist2point.3d(simTrack, ZRH, groundDistance = FALSE) < maxDist
      if(!any(inside)) {simTrack <- NULL} else {
        # Simulate 'gliding&soaring' between lowres track points
        simTrack <- simTrack[inside, ]
        if (nrow(simTrack) < 2) {simTrack <- NULL} else {
          glidingSoaring <- simTrack[1, ]
          for (i in 1:(nrow(simTrack)-1)){
            start <- Reduce(c, simTrack[i, 1:3])
            end <- Reduce(c, simTrack[i+1, 1:3])
            a0 <- simTrack$a[i]
            g0 <- simTrack$g[i]
            sim.locs <- round((sqrt(sum((end-start)^2))/meanStep2target.gliding))
            Q <- tail(qGliding, sim.locs-2)
            count <- 0
            part <- NULL
            while(is.null(part) & count < limit){
              capture.output(part <- suppressMessages(sim.glidingSoaring.3d(MODE = modeZRH,
                                           dGliding = D_gliding, dSoaring = D_soaring,
                                           qGliding = Q, start = start, end = end,
                                           a0 = a0, g0 = g0, error = TRUE,
                                           smoothTransition = TRUE,
                                           glideRatio = glideRatio,
                                           DEM = demZRH, BG = NULL)))
              count <- count + 1
            }
            if (is.null(part)) {simTrack <- NULL; glidingSoaring <- NULL; break} else {
              glidingSoaring <- rbind(glidingSoaring, part[2:nrow(part), ])
            }
          }
          simTrack <- glidingSoaring
        }
      }
    }
  }
  return(simTrack)
}, mc.cores = nCores)
Sys.time()-t

# Save simulated tracks
fName <- paste("results/simTracks_n", n.sim, "_s", seed,".RData", sep = "")
save(glidingSoaringList, file = fName)
cat("'", fName, "' created\n", sep = "")
```

LISTING A.19: HPC cluster: Logs of completed simulation batch jobs

```
Simulation of 1000 tracks on 64 Cores, with seed = 7569187
Time difference of 4.050283 hours
'results/simTracks_n1000_s7569187.RData' created

Simulation of 1000 tracks on 64 Cores, with seed = 9693961
Time difference of 4.267797 hours
'results/simTracks_n1000_s9693961.RData' created

Simulation of 1000 tracks on 64 Cores, with seed = 4771287
Time difference of 4.152869 hours
'results/simTracks_n1000_s4771287.RData' created

Simulation of 1000 tracks on 64 Cores, with seed = 9732833
Time difference of 4.172582 hours
'results/simTracks_n1000_s9732833.RData' created

Simulation of 1000 tracks on 64 Cores, with seed = 6179125
Time difference of 4.007906 hours
'results/simTracks_n1000_s6179125.RData' created

Simulation of 1000 tracks on 64 Cores, with seed = 3786342
Time difference of 3.926022 hours
'results/simTracks_n1000_s3786342.RData' created

Simulation of 1000 tracks on 64 Cores, with seed = 4964122
Time difference of 3.819284 hours
'results/simTracks_n1000_s4964122.RData' created

Simulation of 1000 tracks on 64 Cores, with seed = 2811817
Time difference of 4.271228 hours
'results/simTracks_n1000_s2811817.RData' created

Simulation of 1000 tracks on 64 Cores, with seed = 6374207
Time difference of 3.81878 hours
'results/simTracks_n1000_s6374207.RData' created

Simulation of 1000 tracks on 64 Cores, with seed = 8481337
Time difference of 3.951675 hours
'results/simTracks_n1000_s8481337.RData' created

Simulation of 1000 tracks on 64 Cores, with seed = 7450931
Time difference of 4.03747 hours
'results/simTracks_n1000_s7450931.RData' created

Simulation of 1000 tracks on 64 Cores, with seed = 4131877
Time difference of 4.062701 hours
'results/simTracks_n1000_s4131877.RData' created

Simulation of 1000 tracks on 64 Cores, with seed = 3940392
Time difference of 4.409196 hours
'results/simTracks_n1000_s3940392.RData' created
```

# Appendix B

# Software

| Software | Description, area of application |
|---|---|
| Affinity Photo 1.76 | Professional image editing software; used to edit raster-based figures |
| Blender 2.79 | Professional, free and open-source 3-D computer graphics software toolset; used for 3-D rendering the voxel data of the UDs and collision probability |
| Final Cut Pro X 10.3.3 | Professional non-linear video editing application; used to create an animation of the 3-D rendering with title and legend (Figure 5.21) |
| git 2.14.3 | Free and open source distributed version control system; used for source code (.R) management of the R package and to keep track of changes in the scripts (.Rmd) and the final thesis (.tex) |
| LaTeX 3.04 | High-quality typesetting system; used to render the documentation of the eRTG3D and the thesis itself |
| LaTeXDraw 3.3.9 | Vector drawing editor for LaTeX; used to create vector-based figures in the thesis |
| macOS 10.12.6 | Graphical operating system for Mac; used as operating system for most tasks during the work on the thesis |
| MATLAB 9.3 | Multi-paradigm numerical computing environment and proprietary programming language; used to create graphics |
| Mendeley 1.19 | Free reference manager; used to manage the literature |
| PostGIS | Provides spatial objects for the PostgreSQL database; used to enable support of the spatial data types in the database |

| Software | Description, area of application |
|---|---|
| PostgreSQL 9.6 | Open source object-relational database system that uses and extends the SQL language; used to access and filter the flight trails of airplanes at Zurich Airport |
| Postico 1.3.2 | Modern PostgreSQL client for macOS; used to gain a first overview of the database |
| QGIS 3.0.3 | Free and open source GIS; used to visually check spatial data sets prior to their actual use |
| R 3.5.0 | Language and environment for statistical computing and graphics; used for the implementation of the eRTG3D, data preprocessing, further analyses and the visualization of results |
| RStudio 1.1.453 | Free and open-source IDE for R; used as the main development environment for the package and further scripts |
| Slurm 17.11 | Free and open-source job scheduler for Linux and Unix-like kernels; used to submit the simulation jobs to the Draco HPC of the Max Planck Society |
| texmaker 5.0.2 | Free cross-platform LaTeX editor; used to write the thesis |
| TextMate 2.0-rc.4 | General-purpose GUI text editor for macOS; used to edit various types of text files |

TABLE B.1: Software used in this thesis.

# Appendix C

# Additional Results

| *test* | **turning angle** | **lift angle** | **step length** | **auto turn** | **auto lift** | **auto step** |
|---|---|---|---|---|---|---|
| *KS* | 0.179 | 0.396 | 0.003 | 0.383 | 0.536 | 0.17 |
| *T* | 0.913 | 0.935 | 0.022 | | | |

TABLE C.1: p-values of reproduced gliding CERWs tested against the original gliding section.

| *test* | **turning angle** | **lift angle** | **step length** | **auto turn** | **auto lift** | **auto step** |
|---|---|---|---|---|---|---|
| *KS* | 0.112 | 0.717 | 0.002 | 0.17 | 0.705 | 0.17 |
| *T* | 0.958 | 0.916 | 0.001 | | | |

TABLE C.2: p-values of reproduced soaring CERWs tested against the original soaring section.

| *test* | **turning angle** | **lift angle** | **step length** | **auto turn** | **auto lift** | **auto step** |
|---|---|---|---|---|---|---|
| *KS* | 0 | 0 | 0 | 0 | 0 | 0 |
| *T* | 0.73 | 0.066 | 0.002 | | | |

TABLE C.3: p-values of 1 000 reproduced gliding and soaring CERWs tested against the original trajectory of Wibi 2.

# Declaration of Authorship

I, Merlin Unterfinger, hereby declare that the submitted thesis is the result of my own, independent work. All external sources are explicitly acknowledged in the thesis.

Signed: M. Unterfinger