

Matthias Uden

SNP Analytics – Ein Werkzeug zur visuellen Analyse des Besucherverhaltens im Schweizerischen Nationalpark

Betreuer: Prof. Dr.-Ing. habil. Monika Sester, Prof. Dr. Jonathan Raper
Prüfer: Prof. Dr.-Ing. habil. Monika Sester

Zusammenfassung

Die Entwicklung positionsbezogener Anwendungen für mobile Geräte gewinnt heutzutage zunehmend an Bedeutung. *Smartphones* mit Positionierungsfähigkeit und Internetzugang sind immer weiter verbreitet und ersetzen herkömmliche Mobiltelefone. Die permanente Kenntnis des eigenen Aufenthaltsortes und die Abfrage von Informationen über die Umgebung wird immer wichtiger. Das Anwendungsspektrum solcher *Location-Based Services* ist sehr breit gefächert. Auch der Schweizerische Nationalpark hat das Potential dieser Technologie für seine Zwecke erkannt. Seit einigen Jahren ist dort der digitale Wanderführer *WebPark* im Einsatz. Die Besucher können *PDA*s ausleihen, die sie mit ortsabhängigen Informationen bei ihrer Wanderung versorgen. Einige der Daten wurden in den letzten Jahren aufgezeichnet, um das raumzeitliche Verhalten der Besucher zu analysieren. Jeder Funktionsaufruf, den ein Wanderer mit seinem Gerät tätigt, wird zusammen mit der aktuellen Position abgespeichert. Dies führt zu umfangreichen Datensätzen, deren Analyse speziell im Hinblick auf den Einfluss des Wanderführers auf die raumzeitliche Bewegung der Besucher interessant ist.

Einige existierende Arbeiten zeigen bereits aufschlussreiche Verhaltensmuster, jedoch fehlt es noch an einer einheitlichen Auswertestrategie. In dieser Arbeit wird eine neue Methode zur umfassenden Untersuchung des Besucherverhaltens vorgestellt. Diese basiert auf der visuell gestützten Analyse der Daten von 2007 und 2008 in der interaktiven Benutzeroberfläche *SNP Analytics*. Die Daten werden zunächst einer umfangreichen Vorverarbeitung unterzogen, in der lineare Bezugssysteme eingesetzt werden, um den Ort jeder Besucheraktion entlang der festen Wanderrouen im Park auszudrücken. Die Unsicherheiten in den Daten können dadurch verringert und eine solide Basis für die Analyse geschaffen werden. Anschließend wird die Programmoberfläche entwickelt, welche die visuelle Analyse der Daten beliebiger Jahre auf flexible Art und Weise erlaubt. Sie beinhaltet verschiedene Statistiken und zeichnet sich durch einen hohen Grad an Interaktivität aus.

Mithilfe der Software konnten neue Erkenntnisse über das Besucherverhalten gewonnen werden. Durch das Konzept der *Regions of Interest* wurde beispielsweise (un-)typisches räumliches Verhalten an bestimmten Punkten des Parks gefunden. Die Extraktion verschiedener Besuchertypen anhand häufig genutzter Funktionsaufrufe ist ein weiteres Beispiel.

In Zukunft muss die komplexe Vorverarbeitungsmethodik verbessert und automatisiert werden, um neue Datensätze einfacher zu verarbeiten. Die Bedienung von *SNP Analytics* ist stellenweise noch umständlich, darüber hinaus gibt es Ideen für Erweiterungen. Die Anwendung der Software auf Daten anderer Nationalparks mit ähnlichen Systemen ist noch nicht möglich, aber prinzipiell denkbar. Die Analysesoftware besitzt das Potential, zukünftig ein fester Bestandteil der Auswertung der *WebPark*-Daten im Schweizerischen Nationalpark zu werden.

Abstract

The development of location-aware applications for mobile devices is rapidly gaining in importance. Smartphones equipped with positioning technique and internet access are widely distributed and are replacing traditional mobile phones. The permanent knowledge of your own position and the request of information about the surroundings is getting more and more important.

Such *location-based services* have a wide range of applications. The Swiss National Park also recognised the potential of this technology for their purposes. Since a few years, the *WebPark* digital hiking guide is successfully being deployed. The visitors can rent a *PDA* which provides location-based information during their hike. Parts of the data have been recorded during the last years, in order to analyse the visitor behaviour. Each function call that is made by the hiker is being stored together with the current position. This leads to comprehensive data sets, whose analysis is particularly interesting in the context of the influence the hiking guide has on the spatio-temporal movement of the visitors.

Recent work has already revealed some behaviour patterns. However, there still exists a lack of a consistent analysis strategy. This thesis introduces a new method for examining the visitor behaviour. It is based on the visually supported analysis of the 2007 and 2008 data with the interactive user interface SNP Analytics. The data has to pass through a preprocessing first, in which linear referencing techniques are used in order to relate each visitor log to one of the fixed hiking trails in the park. Thus, the uncertainty in the data can be reduced and a solid basis for the analysis is created. The graphical user interface SNP Analytics is developed subsequently. It allows to visually analyse the data of arbitrary years in a flexible way. It contains several statistics and is characterised by a high degree of interaction.

With this software, some new insights regarding the visitor behaviour could be found. The concept of the *Regions of Interest* for example allowed to find (a-)typical spatial behaviour on certain points of the park. The extraction of different visitor types is another example.

Future work includes improving the complex preprocessing methodology, in order to make the processing of new datasets easier. At some points, working with SNP Analytics is still inconvenient. Also, there exist several ideas for possible extensions. The software cannot be applied to the data of other national parks with similar systems yet, but it is generally conceivable. SNP Analytics has the potential to become an integral part of the analysis of the *WebPark* data in the Swiss National Park.

Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Werken wörtlich oder sinngemäß übernommenen Gedanken sind unter Angabe der Quellen gekennzeichnet. Ich versichere, dass ich bisher keine Prüfungsarbeit mit gleichem oder ähnlichem Thema bei einer Prüfungsbehörde oder anderen Hochschule vorgelegt habe.

Hannover, den 29. Oktober 2010

Matthias Uden

Declaration

I hereby declare that I have generated this thesis on my own and without use of any other than the acknowledged sources of help. Where I have consulted the published work of others, this is always clearly attributed. I assure that none of this work has previously been submitted for a degree or any other qualification at any other university.

Hannover, 29th October 2010

Matthias Uden

Danksagung

Ich möchte verschiedenen Personen danken, die mich bei der Anfertigung dieser Arbeit maßgeblich unterstützt haben. Der größte Dank gilt meiner Betreuerin Monika Sester am Institut für Kartographie und Geoinformatik in Hannover, dafür, dass sie meinen Auslandsaufenthalt ermöglicht und mich bei diesem Vorhaben so tatkräftig unterstützt hat. Weiterhin gilt mein Dank mehreren Personen am *giCentre* in London: Vesna Brujic-Okretic für ihre Unterstützung und die Möglichkeit, am dortigen Institut zu arbeiten. Jonathan Raper und David Mountain, die mich für die Arbeit mit den *WebPark*-Daten begeistert haben und hilfreich zur Seite standen. Schließlich bin ich Antonia Eisenhut und Ruedi Haller vom Schweizerischen Nationalpark sehr dankbar dafür, dass sie die Daten für meine Arbeit bereitgestellt haben.



Acknowledgements

I would like to thank several people who supported me with this work. Firstly, I want to thank my supervisor Monika Sester at the Institute for Cartography und Geoinformatics in Hannover for making my stay in London possible and for the continued guidance. Also, I would like to express my gratitude to the following people at the *giCentre* in London: Vesna Brujic-Okretic for having me at the department and giving me the great opportunity to work there. Jonathan Raper and David Mountain for their motivation to work on the *WebPark* data and their helpful advice. Finally, I am also grateful to Ruedi Haller and Antonia Eisenhut from the Swiss National Park for providing the data.

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
Tabellenverzeichnis	iv
1 Einleitung	1
1.1 Motivation	1
1.2 Aufbau und Konzept der Arbeit	2
2 WebPark – Ein Location-Based Service im Schweizerischen Nationalpark	4
2.1 Grundlagen der Location-Based Services	4
2.1.1 Definition	4
2.1.2 Anforderungen	6
2.1.3 Komponenten	7
2.1.4 Anwendungen	8
2.2 Der Schweizerische Nationalpark	9
2.3 Das WebPark-Projekt	11
2.3.1 Motivation	11
2.3.2 System und Software	12
2.3.3 Praxiseinsatz	14
3 Einordnung in den Forschungskontext	15
3.1 Location-Based Services	15
3.2 Map Matching	19
3.3 Analyse raumzeitlicher Bewegungsdaten	20
3.3.1 Spatial Data Mining	21
3.3.2 Visuell gestützte Analyse und Geovisualisierung	23
3.4 Existierende Arbeiten mit den WebPark-Daten	26
4 Technische Grundlagen	30
4.1 Geoprozessierung in ArcGIS	30
4.1.1 ArcGIS	30
4.1.2 Python	31
4.1.3 Lineare Bezugssysteme	32
4.2 Datenbanken und (Postgre)SQL	33
4.2.1 Allgemeines	34
4.2.2 SQL	35
4.2.3 PostgreSQL und PostGIS	37
4.3 GUI-Programmierung mit Java und Swing	38
4.3.1 Java und GUIs	38
4.3.2 Aspekt der Nebenläufigkeit	40

5	Vorverarbeitung der Daten	41
5.1	Datenbeschreibung und Vorgehen	41
5.2	Konvertierung der Rohdaten	44
5.3	Verarbeitung in ArcGIS	45
5.3.1	Vorbereitung	46
5.3.2	Herausfiltern hoher Geschwindigkeiten	49
5.3.3	Projektion mittels linearer Bezugssysteme	50
5.3.4	Lösen mehrdeutiger Projektionen	52
5.3.5	Tabellenverknüpfung und Shapefile-Export	53
5.4	Erstellen der PostGIS-Datenbank	54
5.4.1	Begründung	54
5.4.2	Import der ArcGIS-Geodaten	55
5.4.3	Weitere Relationen und das Konzept der Regions of Interest	56
6	Das Programm SNP Analytics	59
6.1	Ziele und Aufbau des Programmes	59
6.2	Herstellen der Datenbankverbindung	62
6.3	Der Filter	63
6.4	Die Views	67
6.5	Weitere Funktionen	77
7	Ergebnisse der visuellen Analyse	79
7.1	Zusammenhang raumzeitliches Verhalten – Verwendung des Wanderführers	79
7.2	(Un-)typisches räumliches Besucherverhalten	81
7.3	Unterschiedliche Benutzertypen	84
7.4	Dauer und Abstand der Gerätenutzung	85
7.5	Verwendung der Geo-Bookmark Funktion	86
7.6	Weitere Ergebnisse	88
8	Diskussion	91
8.1	Vorverarbeitung	91
8.2	Evaluation von SNP Analytics	98
8.3	Methodik und Ergebnisse der Analyse	101
9	Zusammenfassung und Ausblick	108
	Literatur	111
A	Anhang	121
A.1	Action Codes 2007 und 2008	121
A.2	Übersicht aller Regions of Interest	123

Abbildungsverzeichnis

1	LBS als Schnittmenge verschiedener Disziplinen.	5
2	Komponenten eines LBS.	7
3	Der Schweizerische Nationalpark mit den 21 Wanderwegen.	10
4	Ausschnitte aus der <i>iPhone</i> -Version der <i>WebPark</i> -Software.	13
5	Verschiedene Bereiche der LBS.	16
6	Drei verschiedene Ansätze zur visuell gestützten Analyse von Geodaten.	24
7	Berechnung der Dichteverteilung aller Abfragen in einer Vorarbeit.	27
8	Das Prinzip linearer Bezugssysteme.	33
9	Aufbau eines Datenbanksystems.	34
10	Oberfläche von <i>pgAdmin</i>	38
11	Die <i>Swing Demo</i> gibt einen Überblick über alle Swing-Komponenten.	39
12	Ausschnitt aus den Daten von 2007.	42
13	Projektion der Punkte auf den Wanderpfad.	43
14	Ablauf der Vorverarbeitung	44
15	Ausreißerglättung durch getrimmtes, gleitendes Mittel.	48
16	Herausfiltern von Einträgen mit einer Durchschnittsgeschwindigkeit über 15 km/h.	50
17	Histogramm der Punktentfernungen zu Route 17.	52
18	Mehrdeutigkeiten im Kreuzungsbereich zweier Routen.	53
19	Das Konzept der <i>Regions of Interest</i>	57
20	Screenshots der beiden Programme, die Vorbild für SNP Analytics sind.	60
21	Die Oberfläche von SNP Analytics.	62
22	Die sechs Komponenten des Filters.	64
23	Die interaktive Karte beim Bewegen des Mauszeigers über eine <i>RoI</i>	68
24	Der <i>Route2D-View</i>	70
25	Die Routenprofil-Ansicht.	71
26	Zwei verschiedene Gesamtnutzungsstatistik- <i>Views</i>	72
27	Statistik zur Verteilung der <i>Action Codes</i>	73
28	Detaillierte <i>RoI</i> -Analyse.	75
29	Die beiden <i>Views</i> zur Analyse der Aktionsketten.	76
30	Beispiel zum Zusammenhang von räumlichem Verhalten und Gerätenutzung.	82
31	Die <i>RoI</i> -Analyse zeigt die Sonderstellung der Regionen Zernez und Il Fuorn.	83
32	Die räumliche Verteilung der Abfragen von Besucher 314.	85
33	Statistik zur Dauer der Aktionsketten.	86
34	<i>Tag cloud</i> mit dem Inhalt der <i>Geo-Bookmarks</i>	87
35	Beispiel zur negativen Korrelation von Geschwindigkeit und Anzahl der Abfragen.	89
36	Punktanhäufung abseits des Pfades.	95
37	Im Kreuzungsbereich treten mehrdeutige Projektionen auf.	97
38	Problem der geringen Grundgesamtheit bei der Analyse der <i>RoI</i> -Sequenzen.	103
39	Unterschiede in der Struktur der <i>WebPark</i> -Software und den <i>Action Codes</i>	106

Tabellenverzeichnis

1	Ausgangstabelle des SQL-Beispiels.	36
2	Ergebnisrelation des ersten SQL-Beispiels.	36
3	Ergebnisrelation des zweiten SQL-Beispiels.	37
4	Liste aller 13 Wanderrouen und ihrer Länge.	49
5	Anzahl der Programmaktionen innerhalb der Picknickplätze und im Rest des Parks.	79
6	Anzahl der Informationsabrufe über jede der Routen, abhängig von der eigenen Position.	80
7	Vier verschiedene Nutzertypen und ihre Top-Vertreter im Jahr 2007.	85
8	Konfidenzintervalle der Punktabstände zu den Routen im Jahr 2007.	94

1 Einleitung

1.1 Motivation

Innerhalb der letzten Jahre ist eine rasante Entwicklung im Bereich mobiler Geräte und Internet-technologien zu beobachten. *Smartphones* mit Positionierungsfähigkeit und Internetzugang sind immer weiter verbreitet und ersetzen herkömmliche Mobiltelefone. Der eigene Aufenthaltsort und der Raumbezug allgemein gewinnen in unserer Gesellschaft zunehmend an Bedeutung. Die Abfrage positionsabhängiger Informationen über die eigene Umgebung ist mit den Geräten an beliebigen Orten möglich. Die Zahl der verfügbaren Applikationen für *Smartphones*, welche die eigene Position einbeziehen, wächst stetig. Diese *Location-Based Services* versorgen den Nutzer mit verschiedensten Informationen über seine aktuelle Umgebung.

Das Anwendungsspektrum solcher Dienste ist sehr breit gefächert. Ein möglicher Einsatzbereich sind auch Nationalparks. Diese haben mittlerweile neben dem Naturschutz ebenso das Ziel, ihren Besuchern reichhaltige Informationen über Flora und Fauna zu vermitteln und damit ihr Naturbewusstsein zu fördern. Über klassische Medien wie Karten oder Informationstafeln hinaus bieten *Location-Based Services* die Chance, den Nutzer individuell und umfassend über die Natur in seiner Umgebung zu informieren. Die Fragen, die sich dem Besucher während seiner Wanderung stellen, können direkt an Ort und Stelle durch einen digitalen Wanderführer beantwortet werden. Dieses Potential hat der Schweizerische Nationalpark erkannt und war Teil des dreijährigen *WebPark*-Forschungsprojektes, welches im Jahr 2001 begann. In diesem wurde der Einsatz positionsbezogener Anwendungen zur Vermittlung geographisch relevanter Informationen für Nutzer speziell in Naturschutzgebieten untersucht. Ergebnis ist eine Anwendung für mobile Geräte, welche im Schweizerischen Nationalpark seit einigen Jahren erfolgreich eingesetzt wird. Die Besucher können *PDA*s ausleihen, die ihnen während der Wanderung permanent vielfältige Informationen über ihre Umgebung liefern.

Neben der Attraktivität aus touristischer Hinsicht besteht mit dem *WebPark*-System auch eine einzigartige Möglichkeit, das raumzeitliche Verhalten der Besucher im Park zu untersuchen. Der Nationalpark zeichnete für diesen Zweck in den letzten Jahren, unter dem Einverständnis der Besucher, anonym einige der Bewegungsdaten auf. Bei jeder Interaktion des Wanderers mit dem Gerät werden Details über die Art der gewählten Programmfunktion sowie seine aktuelle Position abgespeichert. Diese Daten besitzen sehr viel Potential für Analysen, die die raumzeitliche Bewegung des Besuchers mit der Nutzung des Wanderführers in Verbindung bringen. Für solche Untersuchungen existiert bislang allerdings noch keine einheitliche Auswertestrategie. Einige existierende Arbeiten zeigen mit den Daten des Jahres 2007 bereits, dass interessante Verhaltensmuster extrahiert werden können.

Das Ziel dieser Arbeit ist es, einen rundum neuen Ansatz zur Auswertung der *WebPark*-Daten beliebiger Jahre zu entwickeln. Dieser beruht auf der visuellen Analyse mit der Programmoberfläche *SNP Analytics*. Damit sollen tiefgehende Erkenntnisse über das Besucherverhalten gewonnen werden. Der Schwerpunkt wird dabei auf die in Kapitel 1.2 aufgeführten Fragen gelegt.

1.2 Aufbau und Konzept der Arbeit

Im Anschluss an die Einleitung beginnt die Arbeit in Kapitel 2 mit der Beschreibung der Herkunft der zugrunde liegenden Daten. Dazu wird zunächst Grundlagenwissen über *Location-Based Services* vermittelt, bevor der Schweizerische Nationalpark sowie konkret das *WebPark*-System im Detail erläutert werden.

Anschließend wird in Kapitel 3 das Thema dieser Arbeit in den Forschungskontext eingeordnet. Hier spielen aktuelle Aspekte aus verschiedenen Bereichen eine Rolle: Es beginnt mit dem Forschungsfeld der *Location-Based Services*, gefolgt von einem Einblick in das *Map Matching*. Weiterhin werden unterschiedliche Themen aus dem Bereich der Analyse raumzeitlicher Bewegungsdaten behandelt, zum einen mittels *Spatial Data Mining*-Techniken und zum anderen durch die visuell gestützte Analyse und Geovisualisierung. Abschließend werden bereits existierende Arbeiten mit den *WebPark*-Daten dargelegt.

Kapitel 4 beschäftigt sich darauffolgend mit den technischen Grundlagen einiger Werkzeuge, die im Verlauf der Arbeit zur Anwendung kommen und deren Erarbeitung daher für das weitere Verständnis notwendig ist. Dies sind im einzelnen die Geoprozessierung mit dem Programm *ArcGIS* und der Skriptsprache *Python*, Grundlagen der Datenbanken und des speziell genutzten Datenbankmanagementsystems *PostgreSQL* sowie die Programmierung interaktiver Benutzeroberflächen mit *Java*.

Nach den Grundlagenkapiteln folgt die Erläuterung der eigenen Arbeiten. Diese orientieren sich an fünf elementaren Schritten jeder Datenanalyse, die von Andrienko und Andrienko (2006, 1) beschrieben werden. Nach der anfänglichen visuellen Inspektion besteht der erste Schritt in der Formulierung von Fragen. Diese sollen festlegen, welche Phänomene interessant erscheinen und welche Aspekte einer näheren Untersuchung unterzogen werden sollen. Entsprechend werden im Folgenden einige solcher Fragen aufgezählt, welche im Zusammenhang mit den *WebPark*-Daten und unter Berücksichtigung der bereits existierenden Arbeiten entstanden sind.

Die wichtigste Frage, die über allen anderen steht, lautet:

1. ***Wie hängt das raumzeitliche Verhalten der Besucher mit der Verwendung der digitalen Wanderführer zusammen?***

Es soll untersucht werden, *wo* die Besucher *welche* Funktionen der *WebPark*-Software benutzen und *wie* die dadurch gewonnenen Informationen ihr Bewegungsverhalten beeinflusst.

Es schließen sich weitere, speziellere Fragen an:

2. *Wie sieht (un-)typisches räumliches Verhalten der Besucher an markanten Punkten des Parks, z. B. Picknickplätzen oder Wegkreuzungen, aus?*
3. *Gibt es Unterschiede in der Art und Weise, den Wanderführer zu benutzen? Welche „Besuchertypen“ sind vorhanden?*
4. *Wie lange und in welchen Abständen benutzen die Besucher ihr Gerät? Von welchen Einflussfaktoren ist dies abhängig?*

5. *An welchen Orten des Parks wird speziell die „Geo-Bookmark“-Funktion verwendet? Was ist der textliche Inhalt dieser?*

Die Fragen sind, abgesehen von der ersten, nicht in einer bestimmten Rangfolge zu sehen. Sie sind alle im Vorfeld der durchgeführten Arbeiten entstanden und sollen durch diese beantwortet werden.

Der zweite Schritt der Datenanalyse besteht laut Andrienko und Andrienko (2006, 1) aus der Wahl der Methoden zur Durchführung. Nach Betrachtung der existierenden Ansätze im Zusammenhang mit den *WebPark*-Daten ergibt sich die Methodik, welche durch den Titel der Arbeit ausgedrückt wird: Es soll ein interaktives Werkzeug zur **visuellen Analyse** entwickelt werden. Bezüglich dieser Methodik sind folgende Fragen entstanden:

6. *Können interaktive Visualisierungstechniken die Extraktion von Verhaltensmustern ermöglichen und das Verständnis für die Daten erhöhen? Sind diese generisch und auch auf andere Szenarien anwendbar?*

Nachdem somit die relevanten Forschungsfragen aufgestellt sind und das Analysewerkzeug festgelegt ist, folgt ein unumgänglicher dritter Schritt: Die Vorbereitung der Daten, um sie für die gewählten Methoden verfügbar zu machen. Das ist auch ein Hauptbestandteil der Arbeit, welcher in Kapitel 5 erläutert wird. Dort werden die vorliegenden Daten sowie alle notwendigen Schritte zur Vorverarbeitung beschrieben.

Anschließend werden laut Andrienko und Andrienko (2006, 1) die gewählten Analysemethoden auf die Daten angewendet. Dafür muss zunächst das Analysewerkzeug SNP Analytics entwickelt werden. Dies ist neben der Vorverarbeitung der Hauptteil der Arbeit und im sechsten Kapitel ausführlich dargelegt.

Kapitel 7 beinhaltet einige beispielhafte Analyseergebnisse, welche mit der Software erarbeitet wurden. Es wird dort versucht, die oben gestellten Fragen mit dem entstandenen Werkzeug zu beantworten.

Der abschließende Schritt jeder Datenanalyse ist die Interpretation und Bewertung der Ergebnisse. Der Fokus liegt in dieser Arbeit jedoch nicht allein auf den Analyseergebnissen. Daher werden in Kapitel 8 ebenfalls die Stärken und Schwächen der angewendeten Vorverarbeitungsmethodik sowie der entwickelten Programmoberfläche erörtert.

Die Arbeit schließt mit der Zusammenfassung der gewonnenen Erkenntnisse und dem Ausblick auf mögliche Folgearbeiten in Kapitel 9.

2 WebPark – Ein Location-Based Service im Schweizerischen Nationalpark

Dieses Kapitel beschäftigt sich mit der Herkunft der zu untersuchenden Daten: dem digitalen Wanderführer *WebPark* im Schweizerischen Nationalpark. Da dieser ein *Location-Based Service* auf einem mobilen Gerät ist, werden zunächst Grundlagen positionsbezogener Anwendungen erarbeitet. Anschließend wird näher auf den Ursprungsort, den Schweizerischen Nationalpark, sowie vor allem das *WebPark*-Projekt eingegangen. Letzteres ermöglichte erst die Einführung des Wanderführers im Park und somit die Erfassung der Bewegungsdaten.

2.1 Grundlagen der Location-Based Services

2.1.1 Definition

Die Entwicklung von Anwendungen für mobile Geräte wie *Smartphones* oder *TabletPCs*, häufig auch kurz *Apps* genannt, hat in den vergangenen Jahren durch die zunehmende Verbreitung solcher Geräte stark zugenommen. Dabei müssen spezielle technische Einschränkungen und die im Vergleich zu fest installierten Geräten veränderte Anwendungsumgebung berücksichtigt werden. Als wichtige Begrenzungen sind vor allem die Bildschirmgröße, die Rechenleistung sowie die Interaktionsmöglichkeiten zu nennen. Außerdem ist die typische Nutzung von *Smartphones* im Kontrast zu beispielsweise Heimcomputern durch regelmäßige Einsätze für jeweils einen kurzen Zeitraum gekennzeichnet. Der mobile Nutzer möchte unterwegs möglichst schnell Lösungen für bestimmte auftretende Probleme finden. Mountain und MacFarlane (2007) nennen in diesem Zusammenhang das Forschungsfeld des *Mobile Computing*, welches durch die oben beschriebene Entwicklung immer mehr an Bedeutung gewinnt.

Aufgrund der Mobilität des Nutzers besitzen die aufkommenden Fragestellungen meist einen konkreten Raumbezug. Als Beispiel dient das alltägliche Szenario eines nicht ortskundigen Besuchers in einer fremden Stadt, der auf der Suche nach dem nächst gelegenen italienischen Restaurant oder der nächsten Bushaltestelle ist. Er zieht eine passende Anwendung für sein *Smartphone* zurate, die ihm Informationen liefern kann, sofern diese als grundlegenden Input seine derzeitige Position kennt. Die heutzutage weit verbreitete Fähigkeit der Geräte, diese zu liefern, führt seit einigen Jahren zu immer mehr solcher Dienste. Wenn die aktuelle Position des Nutzers bei der Informationsbereitstellung einbezogen wird, kann im Allgemeinen von positionsbezogenen Diensten oder Englisch *Location-Based Services*, kurz LBS, gesprochen werden. Eine von vielen existierenden Definitionen liefert Mabrouk (2005, 4):

„A wireless-IP service that uses geographic information to serve a mobile user. Any application service that exploits the position of a mobile terminal.“

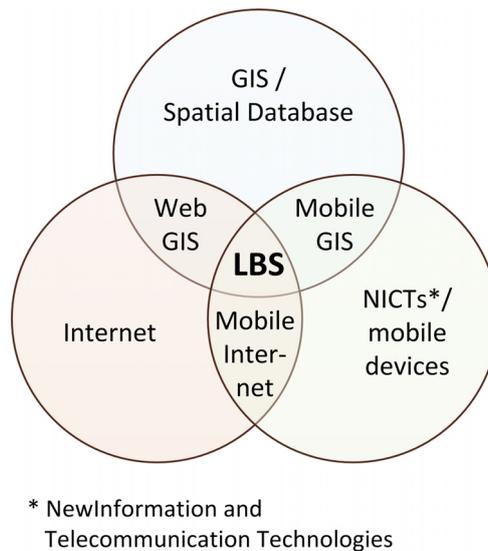


Abbildung 1: LBS als Schnittmenge verschiedener Disziplinen (nach Brimicombe (2002)).

Diese Definition beschreibt den Kern eines LBS – die Berücksichtigung der Position des Nutzers. Hinter dieser einfachen Aussage verbirgt sich laut Steiniger u. a. (2006) der fundamentale Unterschied zwischen LBS und herkömmlichen Informationsmedien, die auf Karten oder dem Internet basieren. Sie sind in vielerlei Hinsicht einzigartig, da sie *kontextbezogen* arbeiten und abhängig von Position und Umgebung des Nutzers die dargestellten Inhalte anpassen. Bei konventionellen Diensten müssen die individuell wichtigen Informationen selbst heraus gesucht werden, LBS hingegen filtern diese automatisch und ersparen somit wertvolle Zeit.

Über die grundlegende Definition hinaus ist es laut Raper u. a. (2007) allerdings schwierig, einen konkreten Rahmen für LBS festzulegen. Der Grund dafür ist, dass diese neue Technologie einen breiten Anwendungsbereich in verschiedenen Disziplinen hat und ebenfalls auf kommerzieller Ebene für hohes Interesse sorgt.

Abbildung 1 zeigt, dass LBS prinzipiell als Schnittmenge folgender drei Bereiche gesehen werden kann:

- **Mobile Geräte:** LBS sind für mobile Geräte gedacht und müssen an deren Eigenschaften angepasst werden.
- **Internettechnologie:** LBS tauschen die benötigten Informationen bezüglich der Nutzerabfrage mit dem Dienstanbieter in der Regel über eine Internetverbindung aus.
- **GIS:** LBS arbeiten per Definition mit geographisch relevanten Informationen, die in den meisten Fällen mithilfe einer Karte angezeigt werden. Somit fungiert das mobile Gerät als geographisches Informationssystem (GIS).

Obwohl ein wichtiger Teil eines LBS demnach auf GIS basiert, sind diese laut Virrantaus u. a. (2001) nicht aus dem GIS-Bereich hervorgegangen. Die ersten LBS wurden nicht etwa von etablierten GIS-Firmen entwickelt. Vielmehr waren hierfür die Hersteller mobiler Geräte verant-

wortlich, die durch die Positionierungsfähigkeit durch GPS-Chips das große Potential für diese Art von Diensten schnell erkannt und die Entwicklung selbiger nach vorne getrieben haben.

Die Besonderheit von LBS im Vergleich zu anderen mobilen Anwendungen besteht vor allem in der Berücksichtigung des *Kontextes* zur Generierung individuell abgestimmter Informationen für den Nutzer. Die Position des Nutzers ist offensichtlich die wichtigste Kontextinformation und grundlegende Voraussetzung. Darüber hinaus nennen Nivala und Sarjakoski (2003) verschiedene weitere Kontextbestandteile, die bei LBS von Bedeutung sein können: Es spielt nicht nur eine Rolle, wo sich der Nutzer befindet, sondern beispielsweise auch, wie seine unmittelbare Umgebung aussieht, wann und warum er den Dienst nutzt. Außerdem kann beispielsweise sein Verhalten in früheren, vergleichbaren Fällen wichtig sein sowie seine persönlichen Interessen oder die soziokulturelle Situation.

2.1.2 Anforderungen

LBS sollten in der Lage sein, möglichst viele der genannten Kontextvariablen einzubeziehen und dementsprechend die angezeigten Informationen anzupassen.

Trotz der oben beschriebenen Vielfalt in Bezug auf die involvierten Disziplinen lässt sich der typische Anwendungskontext von LBS laut Steiniger u. a. (2006) recht klar formulieren: Der Nutzer möchte generell wissen, *wo* er ist und *was* sich in seiner Umgebung befindet, das für ihn von Interesse ist. Die Anwendung von LBS erfolgt in den meisten Fällen in einer für den Nutzer unbekanntem Umgebung und sein Verhalten ist daher gut vorhersagbar. Es wird von dem Bedürfnis an raumbezogenen Informationen dominiert, die sein Problem lösen können. Die Festlegung typischer Verhaltensweisen des Nutzers hilft bei der Abgrenzung von Anforderungen, die der LBS erfüllen sollte. Reichenbacher (2004) nennt fünf elementare *Aktionen* des mobilen Nutzers, die ihn während seiner Suche nach Informationen kennzeichnen:

- **Positionierung:** Die Kenntnis des eigenen Standortes ist die grundlegende Voraussetzung für alle weiteren Aktionen.
- **Suche:** Der Nutzer sucht nach bestimmten Orten, Personen oder Ereignissen in seiner Umgebung, die ihn interessieren. Diese werden häufig als *Points of Interest (POI)* bezeichnet.
- **Navigation:** Der Nutzer möchte wissen, auf welchem Wege er am schnellsten den von ihm gewünschten Ort erreicht.
- **Identifikation:** Hiermit ist die Abfrage von Eigenschaften der POIs in seiner Umgebung gemeint, um genauere Informationen über diese zu erhalten und diese zu „identifizieren“.
- **Ereignisüberprüfung:** Die Abfrage von aktuellen Ereignissen und dem aktuellen Stand von Informationen.

Das Ziel eines vollständigen LBS ist die Bereitstellung von Informationen verschiedener Art, um den Nutzer bei diesen Aktionen zu unterstützen: Für Positionierung und Navigation sind dies reine Geoinformationen, für die anderen genannten Punkte nennen Steiniger u. a. (2006)

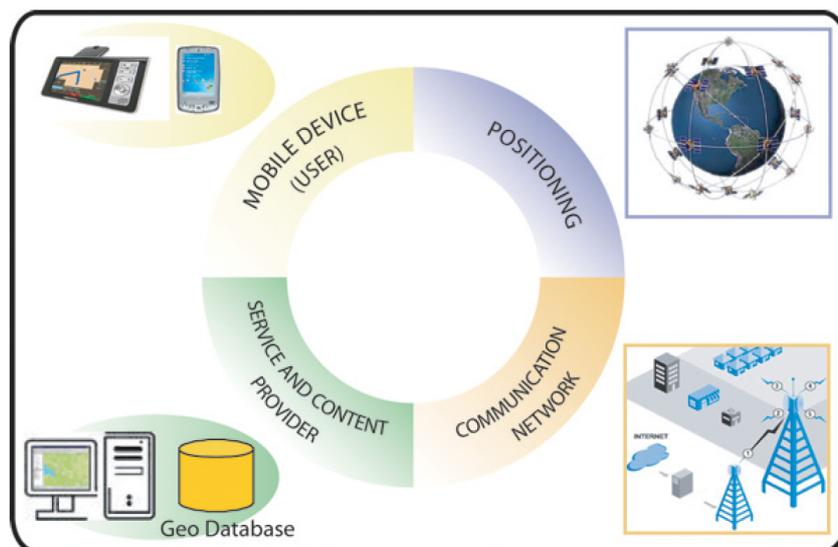


Abbildung 2: Komponenten eines LBS (aus Steiniger u. a. (2006)).

aber auch andere Arten. Dazu zählen allgemeine statische Informationen über die POIs, wie beispielsweise Name, Öffnungszeiten, Telefonnummer, Beschreibung etc. Diese Art „Gelbe Seiten“ bleiben über einen längeren Zeitraum konstant. Als dynamisch oder topologisch werden hingegen Informationen bezeichnet, die sich abhängig von der Nutzerposition und der Zeit stetig ändern. Dazu zählen beispielsweise aktuelle Verkehrsbehinderungen oder die Wettervorhersage. Zusätzlich sind Sicherheitsinformationen über mögliche Gefahren (z. B. Lawinengefahr im Gebirge) und Anlaufstellen, das heißt nahe Krankenhäuser oder Ähnliches relevant. Schließlich ist die Einbeziehung persönlicher Informationen des Nutzers, der aktiv mit dem LBS interagieren soll, sinnvoll. Hierzu zählt die Veröffentlichung von persönlichen Kommentaren und Bewertungen über POIs, die später für andere nützlich sein kann.

Um den Anforderungen gerecht zu werden, ist eine sorgfältige und eingehende Analyse der Nutzerbedürfnisse im Vorfeld jeder LBS-Entwicklung von essentieller Bedeutung. Darüber hinaus müssen vielfältige und ständig aktuelle Informationsquellen bereitgestellt werden.

2.1.3 Komponenten

Steiniger u. a. (2006) nennen fünf allgemeine Komponenten eines LBS, die für die Umsetzung benötigt werden (vgl. Abbildung 2):

Als erstes ist das **mobile Gerät** zu nennen, welches dem Nutzer die Informationen anzeigen soll. In den meisten Fällen ist dies ein *Smartphone*, es können aber ebensogut Laptops, *TabletPCs* oder Fahrzeugnavigationssysteme sein.

Die zweite Komponente besteht in der Technik zur **Positionierung** des Gerätes. Es gibt verschiedene Möglichkeiten zur Umsetzung: Die am häufigsten genutzte Methode ist die GPS-Positionierung. Entsprechende Chips sind üblicherweise Bestandteil jedes *Smartphones* und lie-

fern eine Genauigkeit von einigen Metern. Die Nutzung von GPS ist jedoch keinesfalls unangefochten und es gibt viele Überlegungen zu alternativen Methoden, zum Beispiel die Verwendung von Mobilfunksignalen oder der Inertialnavigation (siehe Kapitel 3.1).

Weiterhin wird eine **Kommunikations-Infrastruktur** benötigt, um den Informationsaustausch zwischen dem Nutzer und dem Anbieter des Dienstes zu realisieren. Durch existierende GSM¹- bzw. UMTS²-Netzwerke des Mobilfunkanbieters und moderne Übertragungstechnologien wie HSDPA³ ist dies in der Regel schon vorhanden.

Die beiden letzten notwendigen Komponenten sind schließlich ein **Anbieter des Dienstes** sowie ein Anbieter der zugrunde liegenden **Daten**. Ersterer ist dafür verantwortlich, dass die Anfrage des Nutzers verarbeitet und nach einer Lösung gesucht wird. Davon getrennt ist üblicherweise der Datenanbieter, der auch das Kartenmaterial, falls benötigt, verwaltet und zur Verfügung stellt.

Für einen vollständigen LBS ist jede der aufgeführten Komponenten unverzichtbar. Folgendes Beispiel soll die Aufgabe jeder Komponente in der Prozesskette der Nutzung eines LBS verdeutlichen:

Angenommen, ein Nutzer befindet sich in einer fremden Stadt und besitzt ein *mobiles Gerät*. Er möchte wissen, wo sich die nächste U-Bahn-Station Richtung Hauptbahnhof befindet. Er hat zwar einen Stadtplan, weiß jedoch nicht genau, wo er sich darauf befindet. Außerdem sind keine U-Bahn-Stationen eingezeichnet. Einen Fahrplan mit aktuellen Abfahrtszeiten besitzt er nicht. Auf seinem Gerät befindet sich allerdings eine geeignete *App*, die von einem *Dienstanbieter* heruntergeladen wurde. Mittels im Gerät integrierter *Positionierungstechnik* ist die Nutzerposition auf einige Meter bekannt. Die Anwendung übermittelt durch die bestehende Internetverbindung (*Kommunikations-Infrastruktur*) die Nutzeranfrage an den *Dienstanbieter*. Dieser sucht nach Ergebnissen in entsprechenden *Datenbanken*, die beispielsweise der lokale Nahverkehrsbetrieb bereitstellt, und fordert zusätzlich Kartenmaterial von anderen Anbietern an. Das Ergebnis, ist eine Karte der Umgebung der Person, in dem seine eigene Position sowie die entsprechend geeigneten U-Bahn-Stationen eingezeichnet sind. Es wird als Antwort auf die Anfrage an das Gerät zurückgeschickt. Typischerweise kann der Nutzer in der Karte navigieren und ebenfalls zusätzliche statische und dynamische Informationen über die Stationen abfragen, wie zum Beispiel Abfahrtszeiten und im Idealfall auch aktuelle Verzögerungen.

2.1.4 Anwendungen

Mögliche Einsatzgebiete von LBS sind sehr breit gefächert, da sie für viele verschiedene Szenarien und Nutzerbedürfnisse entwickelt werden können. In den vorangegangenen Abschnitten wurden bereits einige typische Anwendungsfälle erläutert. Steiniger u. a. (2006) versuchen, das breite Feld in einzelne Kategorien zu unterteilen:

¹Global System for Mobile Communications

²Universal Mobile Telecommunications System

³High Speed Downlink Packet Access

- **Notfallsituationen:** Dienste, die dem Nutzer in Gefahrensituationen zur Seite stehen und sicherheitsrelevante Aktionen übernehmen können. Zum Beispiel das automatische Absetzen von Notrufen inklusive der Nutzerposition.
- **Navigation:** Anwendungen, die dem Nutzer dabei helfen, von A nach B zu kommen, beispielsweise in der Fahrzeugnavigation.
- **Information:** Anwendungen, die reichhaltige Hintergrundinformationen über gewünschte POIs bereitstellen, z. B. digitale Touristen- oder Restaurantführer.
- **Werbung und Abrechnung:** Orts- und/oder personenbezogene Werbung, automatische ortsbezogene Gebührenabrechnung, z. B. in einem Maut-System.
- **Verfolgung:** Anwendungen, die die Bewegung von Personen, Fahrzeugen oder Produkten verfolgen.
- **Management:** Anwendungen, die in der Verwaltung von Unternehmen eingesetzt werden können. Oft im Polizei- oder Feuerwehrbereich anzutreffen, z. B. Flottenmanagement.
- **Spiel und Freizeit:** So genannte *Location-Based Games*, welche die aktuelle Position und die Umgebung einbeziehen, z. B. Geocaching.

Diese Unterteilung bedeutet nicht, dass sich jeder LBS in eine der Kategorien zwingen lässt. Vielmehr sind hier verschiedene Bereiche aufgelistet, in denen LBS eingesetzt werden können. Es existieren Überlappungen und viele Anwendungen decken mehrere der Kategorien ab.

Die Art der Umgebung sowie die erforderliche Genauigkeit der Position variieren je nach Anwendungskategorie sehr stark. Von Indoor- bis Outdooranwendungen, von eher niedriger bis möglichst hoher benötigter Positionierungsgenauigkeit finden sich überall Beispiele. So auch im Schweizerischen Nationalpark, in dem ebenfalls ein LBS für die Besucher angeboten wird, der die Daten für diese Arbeit liefert.

2.2 Der Schweizerische Nationalpark

Der Schweizerische Nationalpark (SNP, 2010) wurde im Jahr 1914 gegründet und ist der älteste Nationalpark Mitteleuropas. Er umfasst eine Fläche von gut 17000 Hektar und befindet sich im äußersten Osten des Landesgebietes im Kanton Graubünden in einer Höhenlage zwischen 1400 und 3200 Metern.

Abbildung 3 zeigt die Lage sowie die 21 festgelegten Routen des Parks, auf denen die Besucher wandern können. Diese gehen teilweise auch über das Parkgebiet hinaus. Das Verlassen der Wege ist laut Artikel 2 der Nationalparkordnung aus Naturschutzgründen untersagt (vgl. SNP, 1983). Die Routen unterscheiden sich in Länge, Höhendifferenz und Geländebeschaffenheit und werden in verschiedene Schwierigkeitsgrade eingeteilt. Das Nationalparkzentrum ist der Ausgangspunkt der meisten Touren und befindet sich in Zernez. Innerhalb des Parks gibt es mehrere Rastplätze sowie zwei Unterkünfte: die Blockhütte „Chamanna Cluozza“ sowie das Hotel „Parc Naziunal II Fuorn“.

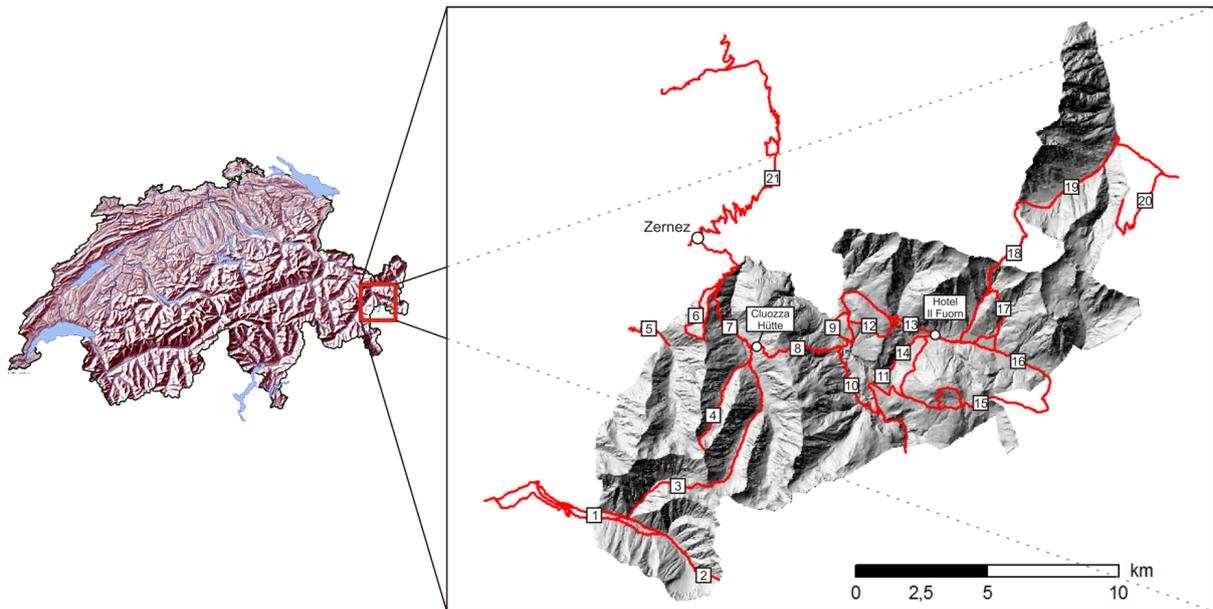


Abbildung 3: Der Schweizerische Nationalpark mit den 21 Wanderwegen.

Der Schweizerische Nationalpark verfolgt allgemein einige für Nationalparks typische Ziele, beispielsweise den Naturschutz oder die Bereitstellung von Informationen über Flora und Fauna für die Besucher. Der SNP im Speziellen legt darüber hinaus traditionell ebenfalls großen Wert auf wissenschaftliche Forschung, um die komplexen Prozesse der Natur besser verstehen und dokumentieren zu können. Forscher aus verschiedenen Disziplinen beschäftigen sich mit den Eigenheiten der Natur im Park. Ein Teil der Aktivitäten ist auch im Bereich raumbezogener Daten und Geographischer Informationssysteme (GIS) angesiedelt. Erste Bemühungen um ein eigenes GIS gab es bereits 1992 in Zusammenarbeit mit der Universität Zürich. Seit dem Jahr 2000 existiert eine eigene Abteilung „Rauminformationen und GIS“. Das umfangreiche GIS ist für die Parkbetreiber ein wichtiges Instrument zur Erfassung, Verarbeitung, Analyse und Präsentation aller raumbezogener Daten. Der Park besitzt mittlerweile große Datenbestände an Raster- und Vektordaten. Diese reichen von topographischen Karten über Luftbilder bis hin zu digitalen Geländemodellen und werden in einem eigenen Datenzentrum verwaltet (vgl. DataCenter, 2010). Die umfangreichen GIS-Aktivitäten werden auch für den Besucher der Parkwebseite (SNP, 2010) ersichtlich, der den Park dort beispielsweise mittels einer interaktiven Karte erkunden kann.

Ein Resultat der Bemühungen ist die Einführung des digitalen Wanderführers im Jahr 2005, den die Besucher für ihre Wanderung ausleihen können. Dieser *Location-Based Service* bietet dem Nutzer umfangreiche positionsabhängige Informationen über den Nationalpark. Während der Wanderung können diese abgerufen und in einer Umgebungskarte angezeigt werden.

Die Entwicklung dieses Systems erfolgte nicht vom Nationalpark allein. Ermöglicht hat dies ein Forschungsprojekt mit mehreren Kooperationspartnern namens *WebPark*, welches später im Nationalpark praktisch umgesetzt wurde. Im Folgenden wird dieses Projekt sowie die daraus entstandene Software mit einigen der grundlegenden Funktionen beschrieben.

2.3 Das WebPark-Projekt

2.3.1 Motivation

WebPark ist ein von der Europäischen Union finanziertes Forschungs- und Entwicklungsprojekt. Während der Laufzeit von Oktober 2001 bis Oktober 2004 arbeiteten verschiedene Universitäten und Organisationen zusammen, um ein System für *Location-Based Services* für die Besucher und Betreiber von Erholungs- und Naturschutzgebieten und Nationalparks zu entwickeln und umzusetzen. Das Projekt ist nicht allein auf den Schweizerischen Nationalpark bezogen. Dieser ist lediglich einer der Parks, in denen das System praktisch eingesetzt wird (siehe Kapitel 2.3.3).

Die Motivation für *WebPark* kann laut Dias u. a. (2004a) von zwei Seiten erfolgen: Erstens die Bedürfnisse seitens der Besucher des Naturschutzgebiets/Parks und zweitens die der Betreiber und Mitarbeiter. In der Frühphase des Projekts führten Abderhalden u. a. (2002) umfassende Befragungen und Analysen durch, in denen unter anderem genau diese Bedürfnisse untersucht wurden. Somit konnte anschließend ein für beide Seiten attraktives System entwickelt werden. Die Umfragen auf Besucherseite sind vor allem wichtig, um direkt von den potentiellen Nutzern zu erfahren, welche Funktionen in der Software für sie interessant wären. Dies ist in erster Linie die Möglichkeit, während der Wanderung im Park an beliebigen Orten individuell zugeschnittene Informationen über beispielsweise Pflanzen, Tierarten und weitere Dinge in der eigenen Umgebung abrufen zu können. Klassische Papiermedien wie Karten und Broschüren sowie die im Park vereinzelt aufgestellten Informationstafeln sind dazu nicht oder nur bedingt geeignet. Ein Großteil der Fragen, die bei den Besuchern entstehen, besitzt einen konkreten Raumbezug. Daher wird darüber hinaus die Anzeige der eigenen Position auf der Karte und die Navigation durch den Park als sehr nützlich angesehen. Des Weiteren wurde auch der Sicherheitsaspekt von den Befragten sehr hoch eingestuft. Die Möglichkeit, beispielsweise aktuelle Wetterinformationen abzufragen oder im Notfall automatische Hilferufe mit der eigenen Position absetzen zu können, ist sehr nützlich.

Dem gegenüber stehen die Bedürfnisse der Betreiber des Naturschutzgebietes oder des Parks. Dias u. a. (2004a) nennen hier in erster Linie das Anliegen, die Besucher auf den Wert der Natur in dem Schutzgebiet aufmerksam zu machen und somit ein umweltfreundliches und respektvolles Verhalten zu fördern. Dies erfordert eine umfassende Information der Wanderer. Die Betreiber haben für diesen Zweck das Potential moderner Dienste erkannt, die über klassische Medien hinaus gehen. Als ein weiterer wichtiger Aspekt wird die Kontrolle und Überwachung des Besucherverhaltens genannt. Das Wissen über die Verteilung der Besucher im Gelände wird als äußerst nützlich Instrument gesehen. Außerdem spielt auch bei den Betreibern der Sicherheitsaspekt eine nicht unwesentliche Rolle, zum Beispiel die Möglichkeit, Besucher im Notfall direkt kontaktieren zu können. Weitere Argumente werden auf der Internetseite des *WebPark*-Projekts (WebPark, 2010) genannt, beispielsweise der Abbau der Informationstafeln sowie die Erleichterung der Arbeit der Parkmitarbeiter.

Durch die Analysen werden noch existierende Lücken zwischen den hohen Informationsbedürf-

nissen der Besucher und den zum Zeitpunkt des Projekts vorhandenen Medien deutlich. Das *WebPark*-Projekt soll genau hier ansetzen und dem Nutzer reichhaltige, personalisierte und jederzeit verfügbare Informationen zur Verfügung stellen. Neben den Besuchern begrüßen aber auch die Betreiber das System, da auch sie vielfältigen Nutzen darin sehen.

Trotz der genannten Vorteile stellen Dias u. a. (2004a) klar, dass das *WebPark*-System nicht darauf abzielt, bewährte Informationsquellen vollständig zu ersetzen. Dies ist durch die beschränkten Ressourcen des kleinen Displays im Vergleich zu Papierkarten von hoher Qualität sowie die Gewohnheiten der Besucher auch nur schwer möglich. Vielmehr möchte das System innovative Dienste anbieten, die bislang noch gar nicht zur Verfügung standen, sowie bereits existierende Informationen kompakt und praktisch in einem Gerät vereinen, ständig bezogen auf die Nutzerposition.

2.3.2 System und Software

Um ein System für positionsbezogene Dienste zu entwickeln, muss zunächst eine Methode zur Positionierung des Nutzers gewählt werden. Im *WebPark*-Projekt wird dies durch einen im Gerät eingebauten oder externen GPS-Empfänger gelöst.

Im Rahmen des *WebPark*-Projekts hat die Firma Camineo (2010) eine Software entwickelt, die den *Location-Based Service* für die Besucher auf einem mobilen Gerät realisieren soll. Dias u. a. (2004b) beschreiben den Informationsgehalt der Software wie folgt: Als Basis sind zunächst alle bisherigen Informationen integriert, die über existierende klassischen Medien zur Verfügung stehen. Darüber hinaus fließen aber auch hochwertige aktuelle Forschungsdaten ein, die dem Nutzer beispielsweise Informationen über spezielle Pflanzen- oder Tierarten veranschaulichen. Weiterer Bestandteil ist die Idee einer kontinuierlichen Erweiterung der Informationen in der Softwaredatenbank, indem Besucher oder auch Förster des Parks bei ihren Wanderungen neue Informationen hinzufügen und der Allgemeinheit zur Verfügung stellen.

Auf der Webseite des Projekts (WebPark, 2010) werden die Geoinformationen, die die Software bereitstellt, generell in Hinter- und Vordergrundinformationen aufgeteilt. Erstere sind die zugrunde liegenden Kartendaten und damit elementar für das Funktionieren der Software. Vordergrundinformationen hingegen sind aktuelle und sich kontinuierlich ändernde Geodaten. Dazu gehören beispielsweise Tier- oder Pflanzenarten, Wetterbedingungen oder POIs wie Hotels oder Restaurants. Diese bereichern die Anwendung mit zusätzlichen Informationen für den Nutzer an.

Die Fülle an Informationen wird in der Anwendung individuell an den Besucher und seine Position angepasst. Burghardt u. a. (2003) nennen in diesem Kontext drei Grundeigenschaften der Software:

1. Die Anzeige des passenden Kartenausschnittes und weiterer Geoinformationen unter Berücksichtigung des aktuellen Nutzerstandpunktes.
2. Die Anfertigung eines individuellen Nutzerprofils, um die angezeigten Informationen entsprechend anzupassen.

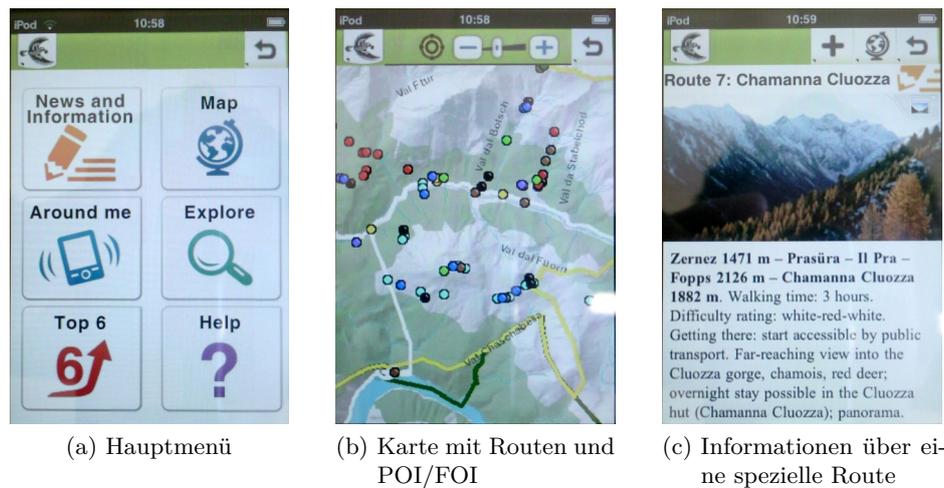


Abbildung 4: Ausschnitte aus der *iPhone*-Version der *WebPark*-Software.

3. Die Möglichkeit für den Nutzer, zusätzliche Informationen aus Datenbanken oder dem Internet abzufragen, beispielsweise durch eine Suchmaschine.

WebPark ist keine *Stand-Alone*-Software, sondern eine web-basierte Anwendung, die auf dem Gerät in einem Browser ausgeführt wird. Abbildung 4 zeigt drei exemplarische Ausschnitte. Durch die oben genannte Eigenschaft des permanenten Informationsaustausches steht sie in Kommunikation mit der online verfügbaren Datenbank. Die Abfrage der Daten wird laut Dias u. a. (2004b) mittels vom OGC⁴ standardisierten Webservices realisiert. Mit dem *Web Map Service (WMS)* kann die Software beliebige Kartenausschnitte und mit dem *Web Feature Service (WFS)* beliebige Vektordaten abfragen. Somit ist gesichert, dass immer die neuesten Daten verwendet und eigene Daten hinzugefügt werden können. Andererseits macht dies eine ständige Internetverbindung erforderlich, die mittels GPRS⁵ oder neuerer Übertragungstechniken realisiert wird. Dias u. a. (2004b) weisen in dem Zusammenhang auf das Problem der Netzabdeckung hin. Durch den überwiegenden Einsatz in ländlich geprägten Gegenden ist diese typischerweise schlecht und es ist mit instabilen Verbindungen zu rechnen. Um Ausfallsicherheit zu gewährleisten, ist das System daher so konzipiert, dass es auch ohne bestehende Internetverbindung in einer Art Basismodus funktioniert. In diesem sind immer noch die Grundfunktionen verfügbar, wie beispielsweise die Anzeige der eigenen Position auf der Karte.

Im Folgenden sind einige konkrete Grundfunktionen der entwickelten Software kurz aufgeführt, die Dias u. a. (2004b) und WebPark (2010) nennen:

Als Basisfunktion kann die Anzeige der eigenen Position, die durch das GPS-Modul ermittelt wird, auf der digitalen Karte bezeichnet werden. Neben der Karte selbst werden zusätzliche Vordergrundinformationen angezeigt. Diese beinhalten POIs wie beispielsweise Restaurants, Picknickplätze oder Fahrradverleihe. Außerdem existieren so genannte *Features of Interest (FOI)*, das sind zum Beispiel Gebiete mit bestimmten Tier- oder Pflanzenarten im Park. Über die Karten-

⁴Open Geospatial Consortium – <http://www.opengeospatial.org/>

⁵General Packet Radio Service

darstellung hinaus kann der Nutzer ebenfalls eine Profilsicht der verschiedenen Wanderrouten anzeigen lassen. Diese Funktion beinhaltet auch eine Anzeige der restlichen Routendistanz und voraussichtlichen Wanderzeit auf der momentan begangenen Route.

Eine weitere Funktion der Anwendung ist die Suche nach POIs/FOIs. Der Suchbereich kann dabei räumlich zwischen „um die eigene Position“, „vor mir auf der Route“ und „im ganzen Park“ festgelegt werden. Wenn gewünscht, kann der Besucher sich auch durch das Gerät benachrichtigen lassen, wenn er beispielsweise in die Nähe eines von ihm gewählten POI kommt oder ein Gebiet betritt, welches bestimmte Tier- oder Pflanzenarten beherbergt. Diese automatischen, positionsabhängigen Benachrichtigungen werden auch *Push-Services* genannt, im Gegensatz zu *Pull-Services*, welche explizit vom Nutzer angefordert werden müssen. Schließlich gibt es die Möglichkeit für den Wanderer, für ihn interessante Orte als *Geo-Bookmark* zu markieren und mit einem Kommentar oder Foto zu versehen. Er kann seine eigenen „interessanten Orte“ nach Wunsch auch veröffentlichen. In diesem Fall werden sie der Datenbank hinzugefügt und somit jedem anderen Besucher zur Verfügung gestellt.

Diese Basisfunktionen spiegeln die in Kapitel 2.1.2 erwähnte Unterstützung der fünf typischen Nutzeraktionen durch den LBS speziell für den Schweizerischen Nationalpark deutlich wider.

Seit Beendigung des Projektes wird die Software durch Camineo kontinuierlich weiterentwickelt, sodass mittlerweile noch weitaus mehr Funktionen zur Verfügung stehen. Im weiteren Verlauf der Arbeit wird noch detaillierter auf einzelne Funktionen eingegangen.

2.3.3 Praxiseinsatz

Das *WebPark*-System wurde bislang in zwei Naturschutzgebieten praktisch umgesetzt: Im Niederländischen Wattenmeer und im Schweizerischen Nationalpark. In dieser Arbeit ist jedoch nur das letztere Einsatzgebiet von Bedeutung, weshalb das erste nicht weiter berücksichtigt wird.

Das speziell auf den Schweizerischen Nationalpark abgestimmte System wird *WebPark^{SNP}* genannt und ist seit mehreren Jahren erfolgreicher Bestandteil des Parks.⁶ Der sich jährlich ändernde Funktionsumfang der Software kann auf der Webseite (SNP, 2010) nachgelesen werden. Für die Besucher stehen zur Zeit insgesamt 26 PDAs als digitale Wanderführer zur Ausleihe bei der Parkverwaltung zur Verfügung. Im Mai 2010 wurde darüber hinaus die erste Version der Software für das *iPhone* und den *iPod* von Apple veröffentlicht. Es wird damit versucht, die Popularität und Verbreitung des Wanderführers zusätzlich zu steigern. Somit können die immer größer werdende Zahl an Besuchern, die ein eigenes *Smartphone* besitzen, ebenfalls die *WebPark-App* auf ihr eigenes Gerät laden.

Auf den zur Ausleihe stehenden PDAs werden seit einigen Jahren teilweise die Bewegungsdaten anonym zu Forschungszwecken aufgezeichnet. Dafür ist das Einverständnis der Besucher erforderlich. Diese Daten bilden die Grundlage für die vorliegende Arbeit und werden in Kapitel 5.1 näher beschrieben. Zunächst erfolgt im folgenden Kapitel jedoch die Einordnung des Themas in den Forschungskontext.

⁶Im weiteren Verlauf ist mit *WebPark* stets *WebPark^{SNP}* gemeint.

3 Einordnung in den Forschungskontext

Dieses Kapitel dient der Einordnung des Themas in den Kontext der Forschung und beschreibt einige aktuelle Arbeiten, die einen Überblick über den Stand der Dinge und offene Fragen geben. Die Natur der Nationalparkdaten und die Arbeit mit diesen lässt sich nicht auf einen Forschungsbereich beschränken, sondern ist vielmehr verknüpft mit Ideen und Methoden verschiedener Felder:

Da das *WebPark*-System ein *Location-Based Service* ist, ist dieser Bereich grundlegend und dessen Forschungsstand im Folgenden als erstes beschrieben. Die Datengrundlage besteht ferner aus den aufgezeichneten, raumzeitlichen Bewegungen der Besucher. Die Vorverarbeitung wird durch Ideen aus dem *Map Matching* beeinflusst, welches anschließend kurz vorgestellt wird. Die Analyse der Bewegungsdaten steht darüber hinaus im Mittelpunkt. Sie wird einerseits in den rechnergestützten *Spatial Data Mining*-Teil sowie andererseits in den Teil der visuell gestützten Analyse und Geovisualisierung aufgeteilt. Abschließend werden kurz existierende Arbeiten beschrieben, die bereits mit den in dieser Arbeit verwendeten *WebPark*-Bewegungsdaten gemacht wurden.

3.1 Location-Based Services

Das Forschungsfeld *Location-Based Services* ist bedingt durch die Vielzahl verschiedener beitragender Technologien stark interdisziplinär ausgerichtet. Durch das enorme öffentliche Interesse und den großen Nutzen positionsbezogener Anwendungen auf mobilen Geräten wächst der Markt für LBS rasant. Dies führte laut Raper u. a. (2007, 1) in der Anfangszeit oft zur vorschnellen Veröffentlichung von neuen Produkten vieler Anbieter, welche bei den Nutzern zunächst wenig Akzeptanz fanden. Dies hat sich mittlerweile offenbar geändert und neue Anwendungen für *Smartphones* zeigen, dass sich LBS etabliert haben und permanent weiterentwickeln.

Anwendungen wie das *WebPark*-System und viele weitere zeigen, dass LBS nach heutigem Stand zwar noch nicht ausgereift, aber sicherlich einsatzfähig und massenkompatibel sind. Die Forschung in den verschiedenen mit LBS verknüpften Bereichen sowie der technologische Fortschritt führen zu einer ständigen Verbesserung der verfügbaren Anwendungen.

Raper u. a. (2007) geben einen guten Überblick über den Stand der Dinge in den verschiedenen für LBS wichtigen Bereichen: Wie in Abbildung 5 dargestellt, sind dies sowohl technische Disziplinen wie die Positionierung, Kartographie/GIS auf mobilen Geräten oder ubiquitäres Computing (d. h. Rechnerallgegenwart) (S. 3-22) als auch nutzerbezogene Fragen wie Geschäftsmodelle für LBS oder rechtliche, soziale und ethische Fragen bei der Nutzung (S. 22-26).

Im Folgenden wird der aktuelle Stand der Forschung in den drei Bereichen Positionierung, mobile Kartographie sowie rechtliche, soziale und ethische Fragen genauer dargelegt.

Die **Positionierung** bildet den unverzichtbaren Kern eines jeden LBS und ist daher auch der wichtigste Forschungszweig. Trotz der weiten Verbreitung und Akzeptanz der GPS-Technologie wird diese von Raper u. a. (2007, 4) nicht als die optimale Lösung gesehen. Gründe dafür sind

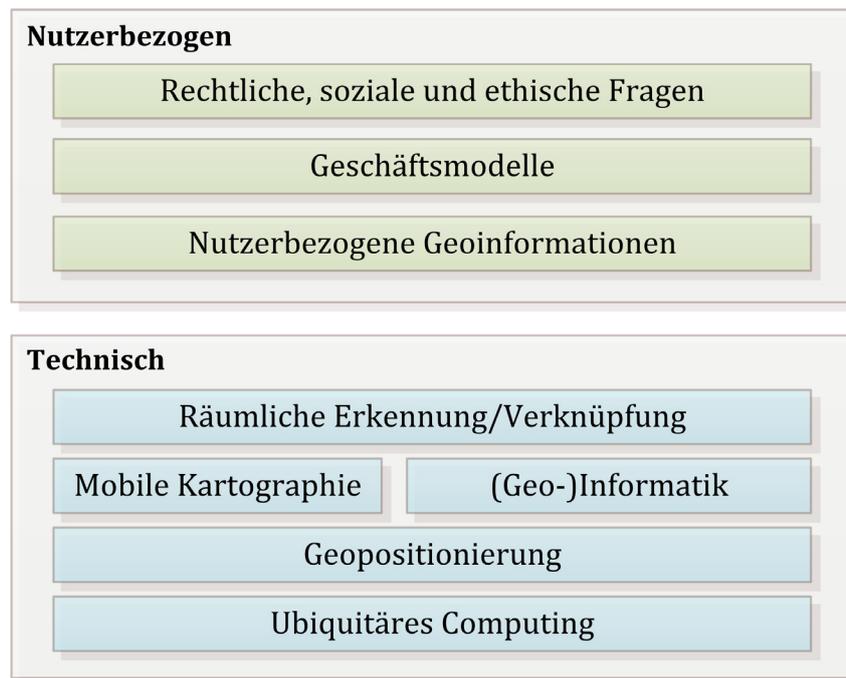


Abbildung 5: Verschiedene Bereiche der LBS (nach Raper u. a., 2007).

Ungenauigkeiten in stark abgeschatteten oder urbanen Gebieten und vor allem die nahezu fehlende Verfügbarkeit innerhalb von Gebäuden. Zum einen wird in der LBS-Forschung auf den Start bzw. operationellen Einsatz weiterer globaler Satellitennavigationssysteme (GNSS) wie GALILEO⁷ oder GLONASS⁸ gewartet, um die Zuverlässigkeit und Verfügbarkeit zu erhöhen. Zum anderen wird ständig nach Alternativen gesucht, besonders im Bereich der Ortung basierend auf terrestrischen Signalen. Hier bieten sich speziell die GSM-Signale der Mobiltelefone selbst an, da sie in jedem Telefon vorhanden sind und keine zusätzliche Infrastruktur benötigen. Verschiedene GSM-basierte Ansätze erzielen Genauigkeitssteigerungen (vgl. Chen u. a., 2006, Curran und Hubrich, 2009, Ibrahim und Youssef, 2010), dennoch reichen die Genauigkeiten im Außenbereich noch nicht an die des GPS heran. Prinzipiell wird die einfache Zellortung über Mobilfunksignale aber bereits in den meisten heutigen Geräten als unterstützende Komponente zum GPS genutzt (*Assisted GPS, A-GPS*). Andere Techniken nutzen WLAN-Signale, um eine möglichst genaue Positionierung sowohl in einer räumlich beschränkten Indoorumgebung (vgl. Di Flora und Hermersdorf, 2008) als auch über größere Gebiete wie eine ganze Stadt (vgl. Cheng u. a., 2005) zu ermöglichen.

Eine weitere Technik ist die Verwendung inertialer Messsysteme, die autonom Beschleunigungs- und Richtungsänderungen bestimmen und damit eine relative Positionierung erreichen können. In den letzten Jahren fand eine starke Miniaturisierung statt und die so genannten *MEMS*-Sensoren⁹ werden immer öfter in Mobiltelefonen verbaut. Verschiedene Arbeiten untersuchen die Möglichkeit dieser Technologie zur Positionierung von Mobilgeräten (vgl. Ofstad u. a., 2008,

⁷Europäisches Satellitennavigationssystem – <http://www.esa.int/esaNA/galileo.html>

⁸Russisches Satellitennavigationssystem – <http://www.glonass-ianc.rsa.ru>

⁹Micro-Electro-Mechanical Systems

Constandache u. a., 2010).

Raper u. a. (2007) gehen davon aus, dass GNSS auch in Zukunft keine zuverlässige Indoor-Lösung bieten wird und daher die Positionierungstechnik hin zu einem integrierten Multisensorsystem gehen wird. Dieses soll Satellitennavigation, terrestrische Signale sowie Inertialsensoren miteinander kombinieren.

Das *WebPark*-System stellt bezüglich der Positionierung eine besondere Situation dar, da eine Lösung mittels GSM- oder WLAN-Techniken aufgrund der schlechten Abdeckung in ländlichem Gebiet nicht sinnvoll ist. Aus diesem Grund werden ausschließlich Satellitennavigationsverfahren genutzt. In alpinen Regionen wie der des Nationalparks kann sich jedoch auch hier die Genauigkeit speziell durch starke Abschattungen in Tälern oder Wäldern um einige Meter verschlechtern. Eine Genauigkeitssteigerung des GPS wäre für bessere Ergebnisse hilfreich, aber nicht zwingend erforderlich, da die benötigte Genauigkeit nicht beliebig hoch ist. Die Indoor-Positionierung ist bei *WebPark* offensichtlich von keiner großen Bedeutung.

Ein weiterer für LBS bedeutender Forschungsbereich ist die **mobile Kartographie** bzw. mobile GIS. Im Zeitalter der allgegenwärtigen Kartographie möchte der Nutzer immer und überall Zugang zu aktuellen Karten haben, um räumliche Fragen zu beantworten (vgl. Morita, 2007). Mobile GIS unterscheiden sich von herkömmlichen *Desktop-GIS* durch beschränkte Bildschirmgröße, weniger verfügbare Ressourcen und der veränderten Nutzung. Laut Dilleuth (2005) geht die Forschung in diesem Bereich über die reine kartographische Visualisierung der Daten hinaus. Sie versucht, das Gesamtsystem der Vermittlung räumlicher Informationen und das veränderte Verhalten der Nutzer im mobilen Kontext zu verstehen.

Immer mehr frei verfügbare Geodaten und die Flut an neuen LBS-Anwendungen und Anbietern führen zu teils sehr heterogenen Datenbanken und Darstellungsformen. Ein bislang noch zu wenig verfolgter Aspekt ist die Standardisierung von Diensten und Datenformaten, beispielsweise basierend auf den Normen des OGC. Außerdem ist die Interoperabilität zwischen Daten verschiedener Anbieter meist nicht gegeben (vgl. Gordillo u. a., 2008, Karam u. a., 2010).

Bezüglich der Darstellung auf dem Display basieren die heutigen LBS-Anwendungen zum Großteil auf 2D-Kartendarstellungen, jedoch gibt es Entwicklungen in Richtung *Augmented* und/oder *Virtual Reality* 3D-Darstellungen (vgl. Liarakapis u. a., 2006). Erste Anwendungen dieser Art sind bereits auf dem Markt (vgl. Wikitude, 2010, Layar, 2010) und bieten einen Ausblick in mögliche LBS der nächsten Generation.

Der letzte Aspekt, der an dieser Stelle genauer erläutert werden soll, sind **rechtliche, soziale und ethische Fragen** im Zusammenhang mit LBS. Die Verbreitung von LBS geht einher mit einer generellen Veränderung im Umgang mit Geodaten im Internet, speziell auf mobilen Geräten. Die Verortung jeglicher Information ist allgegenwärtig und wird von mehr und mehr Menschen genutzt. Es existieren darüber hinaus verschiedene Phänomene wie die Verortung von Fotografien (*Geotagging*), das Veröffentlichen selbst erstellter 3D-Gebäude in *Google Earth* oder *Sociallight* (2010), welches die Erstellung eigener LBS erlaubt. Diese zeigen die Bereitschaft immer mehr interessierter Internetnutzer, aktiv an der Bereitstellung und Bearbeitung von Geodaten teilzunehmen. Das generell verstärkte stattfindende *Crowdsourcing* im Internet, das bedeutet die

Durchführung von Aufgaben durch eine große Menge an freiwilligen Nutzern, wird ebenfalls im Bereich der Geodaten immer bedeutender. Fritz u. a. (2009) nutzen dies beispielsweise zur Verbesserung der Qualität der Fernerkundungsdaten *Global Land Cover*. Im Schweizerischen Nationalpark erweitern die Besucher selbst die Informationsdatenbank, indem sie mit ihren digitalen Wanderführern verortete Fotos machen und *Geo-Bookmarks* speichern. Nachteil dieser Methodik ist die fehlende Qualitätssicherung, da im Prinzip jeder teilnehmen kann und nicht klar ist, wie oft Missbräuche auftreten und unbrauchbare Daten produziert werden.

Die generell positiv erscheinende Entwicklung zu mehr „räumlichem Denken“ in den Köpfen der Menschen birgt allerdings auch Risiken und Gefahren. Denn durch die Nutzung von LBS wird die eigene Position permanent dem Dienstleister preisgegeben. Neue Anwendungen wie *Google Latitude* (2010) präsentieren dem Nutzer per Mausklick den derzeitigen Aufenthaltsort seiner Freunde. Obwohl dies (noch) nur durch das Einverständnis des Nutzers möglich ist, so liegen die Daten den großen Anbietern vor und der Missbrauch selbiger ist möglich. Das Empfangen von ortsbezogenen Werbe-SMS auf dem Mobiltelefon und *Location-Based Spam* sind teilweise schon Realität. Der eigene Aufenthaltsort ist jedoch eine sehr persönliche Eigenschaft, dessen Schutz erstrebenswert ist. In bestehenden LBS-Anwendungen kommt dies jedoch deutlich zu kurz. Darüber hinaus gibt es nur wenige bis gar keine rechtlichen Bestimmungen, die Opfer im Falle eines ernsthaften Datenmissbrauchs absichern können.

Duckham u. a. (2007) weisen darauf hin, dass der Schutz der persönlichen Geodaten in LBS viel stärker berücksichtigt werden muss. Obwohl aus anderen Bereichen bereits allgemeine technische Methoden zum Schutz persönlicher Daten bekannt sind, erfordert dieser Aspekt der *location privacy* dennoch die grundlegende Erarbeitung neuer Ansätze, die speziell auf dieses Problem zugeschnitten sind. Dies bezieht sich nicht nur auf den technischen, sondern vor allem auch auf den sozialen Aspekt des Datenschutzes, da dieser durch existierende Gesetze im Falle von Geodaten noch nicht vollständig abgedeckt ist. Außerdem kann die Grenze, welche Daten schützenswert sind, vom individuellen Nutzer sehr unterschiedlich gesehen werden. Untersuchungen von Tsai u. a. (2010) zeigen, dass je nach Anwendungsszenario die Bereitschaft, persönliche Positionsdaten bereitzustellen, variiert. Im Optimalfall hat der Nutzer selbst die Kontrolle darüber, an wen, in welcher Situation und Detailstufe er seine Positionsinformationen weitergeben möchte (vgl. Küpper und Treu, 2010).

Im konkreten Fall des *WebPark*-Systems wird das Datenschutz-Problem durch die komplette Anonymität umgangen. Es werden zwar Positionsdaten sowie Kommentare und Fotos der Besucher gespeichert, jedoch keine persönlichen Details gesammelt. In der Regel werden die Geräte außerdem vom Park bereitgestellt und ausgeliehen, sodass selbst ein möglicher Zugriff auf im Gerät gespeicherte Informationen nutzlos ist, da diese nicht den Nutzern gehören und keine persönlichen Daten enthalten.

Die genannten LBS-Forschungsbereiche sind, wie zu Beginn angedeutet, nur einige der wichtigsten. Darüber hinaus gibt es weitere mehr oder minder bedeutsame Entwicklungen, die für die LBS der nächsten Generation relevant sind. Raper u. a. (2007, 26-27) bezeichnen unter anderem die Weiterentwicklung mobiler Suchtechniken oder neuer Interaktionsdesigns für LBS als dringend notwendig. Khurri und Luukkainen (2009) untersuchen Voraussetzungen und Bedingungen

für die erfolgreiche Entwicklung von LBS am heutigen Markt.

3.2 Map Matching

Die Vorverarbeitung der Daten in dieser Arbeit ähnelt dem Grundproblem der Kartenstützung (*Map Matching*). Das bedeutet allgemein die Einpassung gemessener Punkte in ein existierendes Wegenetz für die Lokalisierung und Fahrzeugnavigation. Bernstein und Kornhauser (1996) beschreibt die Ausgangssituation wie folgt: Ein Objekt bewegt sich auf den Wegen eines abgeschlossenen Netzes und in regelmäßigen Abständen wird seine Position geschätzt, z. B. mittels GPS-Messungen. Das Ziel des *Map Matching* besteht darin, zu jedem Messzeitpunkt mit möglichst hoher Genauigkeit den Teil des Wegenetzes zu bestimmen, auf dem sich das Objekt befindet. Die gemessene Position befindet sich aufgrund von Messfehlern in der Regel nicht genau auf dem Weg, jedoch weiß man, dass sich das Objekt nur genau dort befinden kann. Dieses Wissen wird beim *Map Matching* ausgenutzt, um den Bezug zu dem aktuell benutzten Weg des Objekts herzustellen. Bei hohen Messfehlern oder komplizierten Wegenetzen ist diese Aufgabe keinesfalls trivial, weshalb es viele komplexe Ansätze zur automatischen Lösung dieses Problems gibt.

Die meisten Forschungsarbeiten im Bereich des *Map Matching* stammen aus der Fahrzeugnavigation. Dort ist es unbedingt notwendig, die Position des Autos auf die aktuell befahrene Straße zu beziehen, nicht nur, um dem Fahrer die korrekte Position auf dem Display darzustellen, sondern auch für das korrekte Funktionieren der Routingalgorithmen.

Bernstein und Kornhauser (1996) beschreiben einige grundlegende Ansätze: Diese beruhen beispielsweise darauf, die gemessenen Positionen auf den nächstgelegenen Knoten bzw. das nächstgelegene Segment im Wegenetz abzubilden. Der Nachteil bei diesen „naiven“ Verfahren liegt unter anderem darin, dass sie nur geometrische Informationen benutzen und sowohl topologische Informationen aus dem Wegenetz als auch die Bewegungshistorie des Objekts, das heißt die vorangegangenen Positionen, nicht einbeziehen. Dies führt bei mehrdeutigen Situationen beispielsweise im Bereich von Kreuzungen oft zu falschen Ergebnissen. Diese können durch die Einbeziehung der vernachlässigten Informationen verbessert werden, sodass falsche Zuordnungen dadurch vermieden werden können, dass sie topologisch gar nicht möglich oder extrem unwahrscheinlich sind. Bernstein und Kornhauser (1996) unterstreichen die Schwierigkeit des *Map Matchings* und empfehlen die Verwendung möglichst vieler topologischer Informationen für die Entwicklung von Algorithmen.

Durch die rasante Entwicklung gerade im Bereich der Fahrzeugnavigation wurden in den vergangenen Jahren viele neue Verfahren entwickelt. Quddus u. a. (2007) stellen eine Übersicht zusammen, in der sie die existierenden Ansätze kategorisieren. Dabei wird zunächst ebenfalls die Unterteilung in rein geometrische Algorithmen und solche, die topologische Informationen einbeziehen, vorgenommen. Die Beschränkung der ersteren ist wie oben angesprochen offensichtlich, sodass dort nur wenige Ansätze entwickelt wurden. Auch einige vorgestellte topologische Verfahren (z. B. Greenfeld, 2002, Meng, 2006), können das Problem nicht zufriedenstellend lösen, sodass immer kompliziertere Ansätze erdacht wurden.

Die dritte genannte Kategorie behandelt wahrscheinlichkeitstheoretische Algorithmen. Diese verwenden den durch Genauigkeitsangaben der Messung schätzbaren elliptischen oder rechteckigen Konfidenzbereich um den gemessenen Punkt. Der Bereich wird dann mit dem Straßennetzwerk überlagert, um den wahrscheinlichsten Aufenthaltsort zu bestimmen (vgl. Zhao, 1997, Ochieng u. a., 2004).

Schließlich stellen Quddus u. a. (2007) in der vierten Kategorie weitere komplexe Ansätze vor, die teilweise die Techniken der ersten drei Kategorien vereinen und darüber hinaus komplizierte Algorithmen vorstellen. Diese arbeiten beispielsweise mit Kalman-Filtern (Obradovic u. a., 2006), Partikelfiltern (Gustafsson u. a., 2002) oder Fuzzy-Logik Ansätzen (Quddus u. a., 2006), sollen aber an dieser Stelle nicht im Detail erläutert werden.

Quddus u. a. (2007) vergleichen weiterhin die Ergebnisse der existierenden Methoden und formulieren einige Probleme, die als beschränkende Faktoren auftreten können: Zum Beispiel unzureichend genaue Straßendaten, ungenaue Ergebnisse bei komplizierten Straßenanordnungen oder fehlende Zuverlässigkeitsabschätzungen der Algorithmen. Daraus leiten sie einige Verbesserungsvorschläge für zukünftige Verfahren ab.

Im Kontext dieser Arbeit spielt in der Vorverarbeitung nicht die Fahrzeugnavigation, sondern das Einpassen der Positionsmessungen von Fußgängern auf die Wanderpfade die entscheidende Rolle (siehe Kapitel 5.3.3). Die Techniken des *Map Matchings* können ohne Weiteres auch für diesen Zweck angewendet werden, da die Problemstellung identisch ist. Viele Forschungsarbeiten im Bereich der Fußgängernavigation verwenden ebenfalls solche Techniken (vgl. Gilliéron und Merminod, 2003, Wasinger u. a., 2003, Spassov u. a., 2006).

3.3 Analyse raumzeitlicher Bewegungsdaten

Die bereits oben beschriebene Entwicklung zu immer mehr positionierungsfähigen Geräten und positionsbezogenen Diensten führt dazu, dass auch der Bestand an raumzeitlichen Daten in unserer Gesellschaft stetig wächst. Diese Geodaten bieten vielfältige Analysemöglichkeiten. Sie sind laut Andrienko u. a. (2008) und Rinzivillo u. a. (2008b) sehr speziell, da sie facettenreich sind und viele unterschiedliche Phänomene in Raum und Zeit beschreiben können. Für die Analyse bedarf es neuer, darauf zugeschnittener Methoden und Ansätze. Klassische Methoden der Datenanalyse sind schon allein aus dem Grund in der Regel nicht angemessen, da sie auf der Annahme stochastisch unabhängiger Daten basieren. Eine elementare Eigenschaft von Geodaten ist aber, dass sie laut Toblers erstem Gesetz der Geographie (vgl. Tobler, 1970) stark voneinander abhängig und hoch autokorreliert sind.

Oft beschreiben die auszuwertenden geographischen Daten Objekte, die sich durch den Raum bewegen und durch die Positionsmessung Trajektorien erzeugen. Dies ist auch bei den in dieser Arbeit betrachteten Daten im Schweizerischen Nationalpark ähnlich. Im Folgenden werden einige aktuelle Forschungsarbeiten speziell im Zusammenhang mit der Analyse solcher Bewegungsdaten beschrieben. Zunächst werden dazu einige Ansätze aus dem Gebiet des *Spatial Data Mining*

vorgestellt. Im zweiten Teil wird auf die durch Visualisierungsmethoden gestützte Analyse eingegangen, die gerade bei geographischen Daten eine hohe Bedeutung besitzt.

3.3.1 Spatial Data Mining

Das Bedürfnis, aus großen Datensätzen Wissen in Form bestimmter Muster zu extrahieren, ist nicht erst im Kontext geographischer Daten entstanden. Schon vor dieser Entwicklung wurde dieser oft als *Data Mining* bezeichnete Prozess mit jeglicher Form von Daten betrieben. Dabei werden oft basierend auf Regeln oder Entscheidungsbäumen intelligente Algorithmen entwickelt, die Struktur in die komplexen, teilweise hochdimensionalen Daten bringen und für den Anwender interessante Muster aufdecken sollen. Typische Aufgaben sind laut Han und Kamber (2001, 5-26) beispielsweise die Klassifikation der Daten, das Finden von *Clustern*, die Detektion von Ausreißern oder das Formulieren so genannter assoziativer Regeln, die Abhängigkeiten gewisser Attribute und Werte und somit bestimmte Muster in den Daten beschreiben. *Data Mining* ist ein Werkzeug, um Daten zu erklären und die implizit vorhandenen, reichhaltigen Informationen möglichst automatisch zu extrahieren sowie anhand der Ergebnisse Vorhersagen für zukünftige Daten machen zu können.

Spatial Data Mining bezeichnet speziell die Gewinnung von Wissen aus räumlichen Datenbanken. Das bedeutet, dass alle Daten in irgendeiner Weise einen Raumbezug besitzen. Es versucht ebenfalls, Muster und Beziehungen in den Daten zu extrahieren. Im Spezialfall von Bewegungsdaten kommt außerdem die Zeitkomponente mit ins Spiel, sodass von der Analyse raumzeitlicher Daten gesprochen werden kann, dem *Spatio-temporal Data Mining*. Dabei ist meist die automatische Erkennung von Bewegungsmustern in den Trajektorien einzelner oder mehrerer Objekte das Ziel.

Im folgenden sind beispielhaft einige aktuelle Arbeiten aufgeführt, die verschiedene Methoden zur Analyse raumzeitlicher Bewegungsdaten vorstellen.

Ashbrook und Starner (2003) haben ein Verfahren entwickelt, welches basierend auf GPS-Daten automatisch signifikante Aufenthaltsorte einer Person detektiert und Vorhersagen über ihre Bewegung trifft. Für diesen Zweck wurden Testdaten über mehrere Monate gesammelt. Aus den Rohdaten extrahiert ein *Clustering*-Algorithmus für jede Testperson die bedeutenden Orte wie beispielsweise die Wohnung, den Arbeitsplatz oder ein regelmäßig besuchtes Café. Es ergibt sich somit eine chronologische Abfolge besuchter Orte. Für die Vorhersage, wo eine Person von einem bestimmten Ort aus als nächstes hinreist, wird ein einfaches Markov-Modell verwendet. Dieses berechnet basierend auf relativen Häufigkeiten aller auftretenden Bewegungen die Wahrscheinlichkeiten.

Als Anwendungsbereich nennen die Autoren tragbare Assistenzsysteme, die einzelne Nutzer beispielsweise bei zu erledigenden Aufgaben an den signifikanten Orten unterstützen sollen. In einem vernetzten *Multi-User*-System ist darüber hinaus die Koordination der Personen untereinander durch das Wissen über den aktuellen und wahrscheinlich nächsten Aufenthaltsort möglich, was zu diversen denkbaren Anwendungen führt.

In zwei verschiedenen Tests hat das Verfahren gute, übereinstimmende Ergebnisse erzielt. Dennoch besitzt es einige Nachteile: Durch den wahrscheinlichkeitstheoretischen Ansatz funktioniert eine sichere Vorhersage erst dann, wenn genügend Daten gesammelt wurden und die signifikanten Orte einer Person über mehrere Monate konstant bleiben. Außerdem ist der Nutzen einer Aussage, dass eine bestimmte Person mit 80%-iger Wahrscheinlichkeit zum Kiosk und dann zur Arbeit fährt, begrenzt. Der Wert der Vorhersagen bleibt zweifelhaft, während das vorangegangene *Clustering* signifikanter Orte einen guten Ansatz darstellt, der auch bei der Analyse der Daten in dieser Arbeit in ähnlicher Weise verwendet wird (siehe Kapitel 5.4.3).

Giannotti u. a. (2007) extrahieren aus den Rohdaten ebenfalls Orte von Interesse (*Regions of Interest*), basierend auf der Dichteverteilung der aufgezeichneten Positionen. Anstatt auf die Vorhersage möglicher Bewegungen legen die Autoren mehr Wert auf die automatische Extraktion so genannter *trajectory patterns*. Dies sind Bewegungsmuster, bei denen mehrere individuelle Trajektorien die gleiche Abfolge von signifikanten Orten innerhalb ähnlicher „Reisezeiten“ besuchen. Die genau verwendete Route wird dabei der benötigten Zeit von A nach B untergeordnet. Auf diese Weise können Gruppen von Trajektorien ähnlicher Bewegungen entlang der *Regions of Interest* detektiert werden. Die Autoren stellen drei verschiedene Algorithmen zur Findung dieser formal definierten Muster vor.

Tests mit real gesammelten GPS-Daten ergeben gute Ergebnisse und die Autoren sehen ihre *trajectory patterns* als einen grundlegenden Teil raumzeitlichen *Data Minings*, der um weitere Konzepte erweitert werden kann. Die Schwachstelle ist zweifelsohne in der bloßen Berücksichtigung der Reisezeit zwischen den Regionen, nicht aber der genauen Route, zu sehen. Es werden dadurch nicht alle in den Daten enthaltenen räumlichen Informationen ausgenutzt.

Verhein und Chawla (2006) übertragen das klassische *Data Mining*-Konzept der assoziativen Regeln zur Findung von Mustern erstmals auf raumzeitliche Daten. Sie liefern eine formale Definition der *Spatio-Temporal Association Rules (STARs)*. Der Fokus liegt auf dem Verhalten sich durch einzelne Regionen bewegender Objektgruppen. Mittels effizienter Algorithmen können die Regionen den folgenden vier Typen zugeordnet werden: Der Bereich ist *stationär*, das heißt Objektgruppen halten sich länger in dieser Region auf. Er ist eine *Quelle*, wenn viele Objekte diesen Bereich verlassen oder eine *Senke*, wenn viele hineinströmen. Als *Durchgang* werden solche Regionen bezeichnet, die von einer großen Anzahl an Objekten passiert werden.

Das STAR-Konzept wurde zunächst nur in einer Simulationsumgebung getestet und hatte den Nachteil, dass die STARs jeweils lediglich kurze Abschnitte der Bewegung modellieren und keine längeren Sequenzen aufdecken konnten. Das Konzept wird daher von Verhein (2009) mit dem *k-STAR* Prinzip auf die Sequenzanalyse erweitert. In diesem besteht Robustheit gegenüber zeitlichen und räumlichen Lücken in den Daten, die Analyse der Bewegungsmuster wird somit wesentlich einfacher. Darüber hinaus wurde das erweiterte Konzept raumzeitlicher Assoziationsregeln mit realen, verrauschten Daten erfolgreich getestet.

Alle drei vorgestellten Ansätze haben eines gemeinsam: Die Betrachtung der Objektbewegungen geschieht zwischen und innerhalb gewisser räumlicher *Regionen* statt der gemessenen Position. Dies zeigt das von Nanni u. a. (2008) angesprochene Problem der Positionsunsicherheit, welches

in der Analyse raumzeitlicher Daten allgegenwärtig ist. Eine höhere Präzision durch bessere Positionierungsverfahren wäre, wie in Kapitel 3.1 angesprochen, wünschenswert. Dennoch bedeutet die Unsicherheit keine große Einschränkung für die Analyse, da in vielen Fällen eine sehr hohe Genauigkeit für eine umfassende Untersuchung nicht zwingend notwendig ist. Dies zeigen auch die Ergebnisse der vorgestellten Arbeiten.

Alternativ zu genaueren technischen Verfahren kann das Einbringen von Hintergrundwissen über den Erfassungsraum der Trajektorien die Genauigkeit und den Nutzen der Daten signifikant erhöhen. Dies können beispielsweise ein festes Straßennetzwerk oder vordefinierte signifikante Regionen sein (vgl. Giannotti u. a., 2007, Nanni u. a., 2008). Beides wird auch in dieser Arbeit bei der Vorverarbeitung der Daten des Schweizerischen Nationalparks ausgenutzt.

Die Forschungsarbeiten im Bereich raumzeitlichen *Data Minings* und der Erkennung von Bewegungsmustern zeigen viele verschiedene Ansätze, Konzepte des herkömmlichen *Data Minings* auf raumzeitliche Phänomene zu übertragen und stellen diverse formale Analysemethoden und eigens definierte Bewegungsmuster vor. Nanni u. a. (2008) zeigen auf, dass es bislang allerdings an einer Art Standardisierung, einem theoretischem Rahmen fehlt. Es existieren keine Definitionen generischer Bewegungsmuster und Algorithmen, die auf *jegliche* Art raumzeitlicher Daten anwendbar sind. Dies ist aufgrund der sehr breiten Vielfalt der Daten schwierig zu realisieren. Dodge u. a. (2008) erkennen die Wichtigkeit einer gemeinsamen Basis im *spatio-temporal Data Mining* und machen einen ersten Schritt in diese Richtung einer „Taxonomie“ für Bewegungsmuster. Sie stellen unter anderem eine einfach zugängliche Internetplattform bereit (vgl. Dodge, 2010), die zur Entwicklung von Standards für die Definition und Analyse raumzeitlicher Bewegungsmuster dienen soll.

3.3.2 Visuell gestützte Analyse und Geovisualisierung

Die Entwicklung und Anwendung von *Data Mining*-Algorithmen auf eine große Menge raumzeitlicher Daten stellt nur einen Teil der Analyse dar. Ein weiterer Teil, der nicht bloß die Präsentation der Ergebnisse, sondern einen integrierten Bestandteil der gründlichen Analyse darstellt, ist die *Visualisierung* und die visuelle Interpretation der Daten bzw. der Ergebnisse der Mustererkennung.

Andrienko und Andrienko (2006, 166-171) beschreiben den hohen Wert, den die graphische Darstellung als Analysewerkzeug besitzt. Sie ist notwendig, um die komplexen Daten selbst sowie mögliche computergestützt extrahierte Muster für den Betrachter wahrnehmbar zu machen. Ohne eine sinnvolle Visualisierung sind Ergebnisse einer algorithmischen Datenanalyse meist nicht viel wert. Der Mensch kann komplizierte Sachverhalte deutlich einfacher wahrnehmen und verstehen, wenn sie graphisch dargestellt werden. Neben dem reinen Verständnis der Daten kann er darüber hinaus in Graphiken Muster wesentlich leichter erkennen, als wenn sie in tabellierter Form vorliegen. Er kann intuitiv Dinge aus der Darstellung entnehmen, die der Computer vorher gar nicht explizit berechnet hat oder nicht berechnen konnte. Das Ziel der Visualisierung ist somit nicht nur die reine Darstellung von Ergebnissen, sondern das Ausnutzen der menschli-

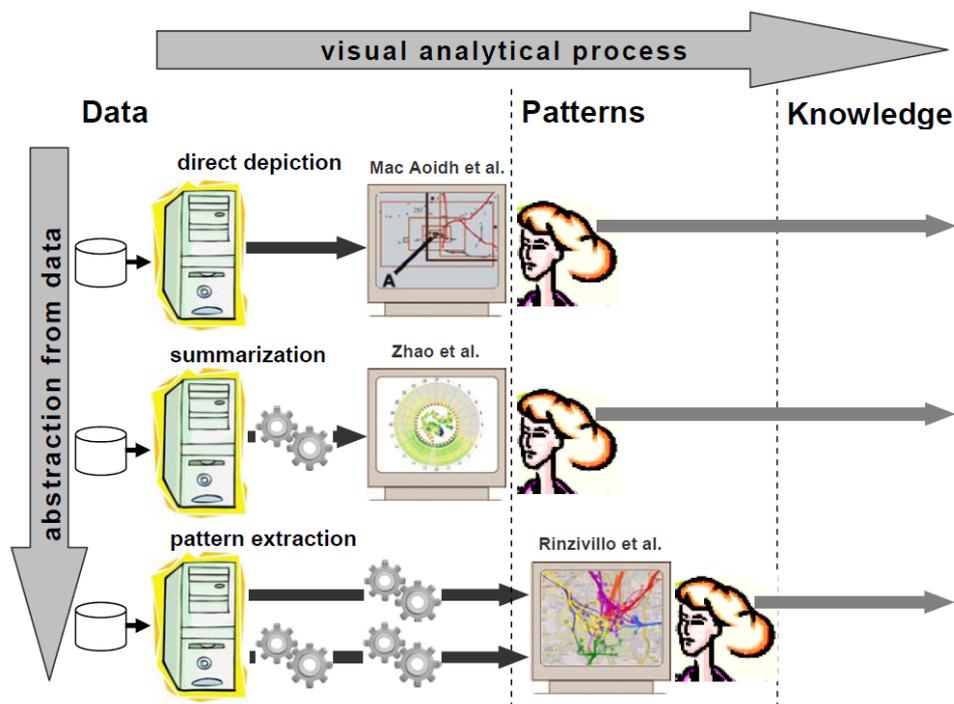


Abbildung 6: Drei verschiedene Ansätze zur visuell gestützten Analyse von Geodaten (aus Andrienko u. a., 2008).

chen Wahrnehmungsfähigkeit, selbst Muster zu erkennen und somit aktiv zur Analyse der Daten beizutragen (vgl. Gahegan u. a., 2001). Die interaktive Einbindung visueller Datenexploration im gesamten Analyseprozess und die Synthese von rechnergestütztem *Spatial Data Mining* und visueller Methoden erhöht den Wert und die Effektivität jeder Datenanalyse deutlich (vgl. Dykes und Mountain, 2003, Keim u. a., 2005, Andrienko u. a., 2007).

Die Wichtigkeit der visuellen Unterstützung prägte den Begriff eigener Forschungsbereiche wie *Visual Analytics* oder *Exploratory/Visual Data Mining*. Im Spezialfall geographischer Daten fällt dies in das Feld der Geovisualisierung, auf welches im Folgenden der Fokus gelegt wird.

Durch die Vielfalt geographischer Daten kommt die Geovisualisierung in vielen verschiedenen Anwendungsbereichen zum Einsatz. Wie oben erläutert, sollte die Visualisierung Teil eines jeden Analyseprozesses sein. Eine Schlüsselfrage ist, welche Teile in diesem Prozess durch die *Data Mining* Techniken computergestützt und welche visuell durch den Betrachter übernommen werden sollten. Andrienko u. a. (2008) nennen drei verschiedene Ansätze, die sich in der Forschung herauskristallisiert haben (vgl. Abbildung 6):

Der erste legt den Schwerpunkt auf die reine visuelle Analyse und überlässt dem Computer lediglich die Darstellung der Rohdaten. Es werden keine Mustererkennungsalgorithmen oder Ähnliches ausgeführt, allein der Betrachter versucht, aus der Graphik bestimmte Muster und Regelmäßigkeiten zu erkennen. Dies ist allerdings problematisch, wenn der Datensatz sehr groß und hochdimensional ist. Trotz der Entwicklung neuer Methoden (vgl. Carvalho u. a., 2008, Mac Aoidh u. a., 2008) ist dieser Ansatz zumindest bei komplexen Datensätzen meist nicht praktikabel.

Dies führt zum zweiten Ansatz, bei dem die umfangreichen Daten für den Betrachter zunächst aufbereitet werden. Es werden dabei zwar keine direkten Muster extrahiert, aber Abstrahierungen, Aggregationen und Generalisierungen durchgeführt. Durch teils innovative Visualisierungstechniken wie *Treemaps* (Slingsby u. a., 2008, 2010), *Flow Trees* (Wood u. a., 2009) oder *Growth Ring Maps* (Bak u. a., 2009) soll es darüber hinaus dem Betrachter einfacher ermöglicht werden, Muster zu erkennen. Durch die Komplexität der Visualisierungstechniken kann es zu dem Problem kommen, dass die Darstellung für den ungeübten Betrachter, der sich nicht mit den Daten auskennt, zu komplex und unübersichtlich und die beabsichtigte einfache Extraktion von Mustern nicht erreicht wird.

Bei der dritten Methode kommt dem *Data Mining* schließlich die höchste Bedeutung zu. Algorithmische Verfahren wie die aus Kapitel 3.3.1 werden angewendet. Die anschließende Visualisierung stellt die extrahierten Muster lediglich dar, besitzt jedoch trotzdem einen wichtigen Zweck: Der Betrachter kann zusätzliche Interpretationen vornehmen oder die angewendeten Methoden visuell evaluieren. Ein Beispiel einer solchen visuell gestützten Datenanalyse wird von Rinzivillo u. a. (2008a) gegeben.

Im Bereich interaktiver Geovisualisierung von Bewegungsdaten existieren viele Werkzeuge und Techniken, um diese effektiv zu gestalten. Frühe Ansätze basieren auf einfachen Pfeil- und Liniendarstellungen auf Karten, so genannten *Flow Maps* (Tobler, 1987, Vasiliev, 1997). Durch Pfeile verschiedener Stärke werden die Häufigkeiten bestimmter Bewegungsmuster dargestellt. Dieses Verfahren wird auch heute noch verwendet, z. B. von Andrienko u. a. (2007). Der Nachteil besteht darin, dass die zeitliche Komponente der Daten nicht direkt berücksichtigt wird. Dies kann aber durch den Einsatz eines zeitlichen Filters geschehen, sodass nur Bewegungen in bestimmten Zeitintervallen dargestellt werden.

Ein weiterer traditioneller Ansatz zur Visualisierung raumzeitlicher Daten ist der von Hägerstrand (1970) vorgestellte *Time Geography Cube*. Dieser Würfel stellt zweidimensionale Bewegungen über die Zeit dar. Viele Arbeiten basieren darauf und erweitern das Konzept (vgl. u.a. Kraak, 2003, Mountain, 2005, Yu, 2006). Diese Methode ist allerdings für periodisch auftretende Bewegungsmuster nur bedingt geeignet und kann darüber hinaus bei großen Datenmengen unübersichtlich und schwierig zu interpretieren sein.

In den letzten Jahren gewinnen interaktive, animierte Darstellungen an Bedeutung. Die meisten der neuen Visualisierungswerkzeuge sind im zweiten der drei oben genannten Bereiche einzuordnen. Das heißt, dass die komplexen Datenmengen in irgendeiner Weise aggregiert und dem Betrachter durch verschiedene Graphiken vermittelt werden. Beispiele sind das umfangreiche *GeoViz Toolkit* (Hardisty, 2005, GeoVISTA, 2010) oder der von Dykes und Mountain (2003) vorgestellte *Location Trends Extractor*. Die Gefahr besteht laut Andrienko u. a. (2007) hier darin, durch die Aggregation möglicherweise wertvolle Informationen der Originaldaten zu verlieren oder zu vernachlässigen.

Eine wichtige Rolle spielt auch in dieser Arbeit der Spezialfall der effektiven Visualisierung multivariater, komplexer Bewegungsdaten auf linienhaften Strukturen. Im Vergleich zur Visualisierung sich frei im Raum bewegendere Objekte ist dies dem Betrachter oft schwieriger in einer übersichtlichen Art und Weise vermittelbar. Farbliche Kodierungen sind häufig aufgrund der

linienhaften Strukturen ungeeignet. Ähnliche den *Flow Maps* stellt Huffman (2009) einen neuen Ansatz basierend auf der Linienstärke vor, der nicht nur die Visualisierung von Bewegungen, sondern beliebiger graphischer Variablen allgemein erlaubt. Die Darstellung ist für die menschliche Wahrnehmung intuitiver als durch Farbwerte oder –intensitäten. Das Konzept wird teilweise auch für die Visualisierung der Daten in dieser Arbeit verwendet.

Die genannten Methoden sind nur einige der existierenden. In der Geovisualisierung allgemein werden laufend neue Visualisierungstools entworfen. Dies wird meist entweder durch die individuell unterschiedliche Geodatengrundlage motiviert, die einen eigenen Ansatz erfordert, oder durch die von dem Programm bereitgestellten neuen, zuvor noch nicht existierenden Methoden. Andrienko u. a. (2005) nennen verschiedene Faktoren, die die Entwicklung immer neuer Tools erklären. Jedes dieser Werkzeuge ermöglicht zwar die Analyse des zugrunde liegenden Datensatzes, ist aber in der Regel nicht auf andere Daten anwendbar. Die bereits oben angedeutete Vielfalt geographischer Daten stellt das derzeitige Hauptproblem der Geovisualisierung dar: die Interoperabilität und Wiederverwendbarkeit der Visualisierungswerkzeuge. Die Forschung beschäftigt sich daher zunehmend damit, die Rolle des Visualisierungswerkzeuges im gesamten Analyseprozess auf einer abstrakteren, mehr konzeptuellen Ebene zu beschreiben, die nicht mehr so eng mit den spezifischen Daten verknüpft ist (vgl. Gahegan, 2005, Andrienko u. a., 2005). Dies soll den Weg für mehr generische und interoperable Werkzeuge ebnen.

3.4 Existierende Arbeiten mit den WebPark-Daten

Dieses Unterkapitel gibt einen kurzen Überblick über existierende Arbeiten, die bereits mit den Daten des Schweizerischen Nationalparks oder sehr ähnlichen Bewegungsdaten gemacht wurden. Die Daten der digitalen Wanderführer wurden im Jahr 2007 zum ersten Mal über eine komplette Saison von Juni bis Oktober aufgezeichnet. Die folgenden drei Arbeiten führen erste grobe Analysen mit den Daten durch:

Haller und Eisenhut (2008) stellen unter anderem eine Statistik über die Häufigkeit der von den Besuchern genutzten Programmfunktionen auf. Weiterhin werten sie die räumliche Verteilung der Abfragen aus, indem sie die Dichte der Aktionen betrachten. Dabei stellen sie beispielsweise fest, dass das Gerät auf Rastplätzen oder ähnlichen Gebieten, in denen die Wanderer pausieren, deutlich öfter genutzt wird als bei der Wanderung selbst. Außerdem nimmt die Anzahl der Nutzungen gegen Ende einer gewanderten Route im Vergleich zum Beginn ab.

Zu ähnlichen Ergebnissen kommen auch Eisenhut u. a. (2008), die darüber hinaus eine etwas weiter gehende Analyse durchgeführt haben. Gegenstand dieser ist die Untersuchung der Abhängigkeit der Programminteraktionen von verschiedenen Faktoren wie Topographie, Steigung des Pfades, Perspektive des Wanderers oder Vegetation. Erste Zusammenhänge lassen sich in den Ergebnissen erkennen, beispielsweise die erhöhte Nutzung des Gerätes in flachen oder sehr steilem Gelände, weniger aber bei moderater Steigung. Neben der erwähnten Anhäufung von Nutzungen im Bereich der Rastplätze, Aussichtspunkte oder Almhütten, wurde von den Autoren auch der abgefragte Inhalt in diesen untersucht. Dabei zeigte sich ebenfalls ein eindeutiges Ergebnis:

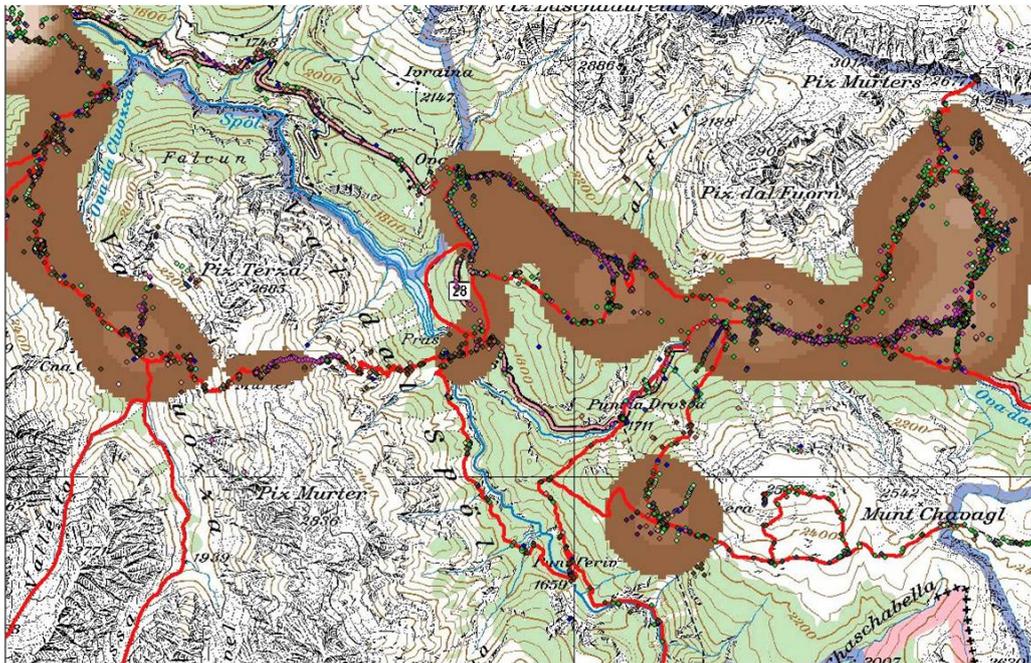


Abbildung 7: Vorarbeiten mit den *WebPark*-Daten beinhalten unter anderem eine Berechnung der Dichteverteilung aller Abfragen (aus Raper, 2008).

Bei der Rast werden verglichen zu den Wanderpfaden selbst weniger Programmfunktionen zur Navigation/Kartenanzeige verwendet.

Als drittes ist schließlich die Arbeit von Raper (2008) zu nennen. Er leitet unter anderem eine Dichteverteilung aller Abfragen ab (vgl. Abbildung 7). Des Weiteren identifiziert er anhand manueller Analysen verschiedene Nutzertypen, basierend auf den am häufigsten verwendeten Funktionen des Wanderführers. Diese Idee wird innerhalb der vorliegenden Arbeit weiterentwickelt.

Diese drei Arbeiten unterziehen die komplexen Daten ersten einfachen Analysen, bei denen bereits interessante Muster zu erkennen sind. In allen Artikeln wird deutlich, dass das Ziel eine tiefer gehende Analyse ist, bei der das Verhalten der Besucher im Park noch stärker untersucht wird.

Snozzi (2009) beschäftigt sich als erstes ausführlicher mit den *WebPark*-Daten und liefert für die vorliegende Arbeit einige wichtige Aspekte. Das Ziel des Autors ist es, ebenfalls basierend auf den Daten des Jahres 2007 eine quantitative Analyse im Hinblick auf verschiedene Kontextvariablen durchzuführen. Es wird dabei untersucht und quantifiziert, inwiefern die Interaktion der Besucher mit dem digitalen Wanderführer abhängig ist

- vom Aufenthaltsort, das heißt entweder auf einem Wanderpfad oder innerhalb einer der Picknickplätze,
- von den Umweltbedingungen, das heißt Topographie (Richtung und Steigung des Geländes), Vegetation und Wetter am Ort und zum Zeitpunkt der Gerätenutzung sowie
- von der absoluten Uhrzeit und der relativen Wanderzeit entlang eines Pfades.

Die Relevanz dieser einzelnen Kontextvariablen bezüglich der Struktur der getätigten Softwareinteraktionen wird umfassend statistisch ausgewertet. Die Picknickplätze spielen dabei eine besondere Rolle: Basierend auf der Aufenthaltsdauer der Besucher in diesen wird eine Klassifizierung in verschiedene Nutzergruppen vorgenommen. Die erste Gruppe besteht aus Wanderern, die ihr Gerät nicht in der Nähe des Rastplatzes verwendet haben und somit nicht klassifizierbar sind. Die übrigen Gruppen werden anhand der Aufenthaltsdauer der Besucher auf dem Picknickplatz definiert.

Vor der eigentlichen statistischen Analyse werden die Daten wie auch in dieser Arbeit einer Vorverarbeitung unterzogen, die grundsätzlich ähnliche Ansätze wie die in Kapitel 5 vorgestellte verfolgt. Es werden jedoch lediglich die vier meist genutzten Wanderpfade berücksichtigt, um eine räumliche Überlappung der Einträge zu vermeiden.

Für jede der Kontextvariablen stellt der Autor teils einfache, teils komplexe Modelle vor. Die jeweilige Relevanz der verschiedenen Informationen, die von den Nutzern abgefragt werden, wird mit diesen statistisch ausgewertet. Haupterkennnis ist, dass der Aufenthaltsort sowie die relative Wanderzeit entlang des Pfades die einflussreichsten Kontextvariablen sind. Die Auswertung der weiteren Variablen liefert zwar ebenfalls zum Teil bestimmte Abhängigkeiten, jedoch sind die Ergebnisse mit einer großen Unsicherheit behaftet, welche auf verschiedene Art und Weise begründet wird: Ein großer Unsicherheitsfaktor sind die Daten selbst, aufgrund der stark um die Pfade streuenden GPS-Positionen. Weitere Faktoren sind in der Modellierung bzw. dem Prozessierungsablauf zu finden. Einige Kontextvariablen unterliegen einer komplizierten Berechnung, die fehleranfällig ist, beispielsweise die Steigung des Weges. Andere basieren auf Modellannahmen, die ebenfalls mit einer hohen Unsicherheit behaftet sind, z. B. das Wetter. Schließlich werden Klassifizierungen auf Grundlage fester Klassengrenzen gemacht, wie beispielsweise die Einteilung der verschiedenen Nutzergruppen abhängig von einer festen Aufenthaltszeit im Bereich der Picknickplätze. Anders gewählte Grenzen führen schnell zu anderen Ergebnissen. Außerdem ist die zugrunde liegende Stichprobenmenge für die Analyse in einigen Bereichen noch zu klein, sodass die statistische Signifikanz nicht ausreichend ist.

Als Verbesserung schlägt der Autor unter anderem eine Fuzzy-Klassifizierung vor, die nicht auf festen Klassengrenzen basiert. Trotzdem stellt er ebenso fest, dass die bestehenden Unsicherheiten allein dadurch nicht gänzlich behoben werden können. Dennoch zeigen seine Ergebnisse das Interaktionsverhalten der Besucher mit der Software im Detail und sind für eine Weiterentwicklung des Wanderführers hilfreich.

Als letzte Arbeit soll an dieser Stelle der Artikel von Dias u. a. (2009) erwähnt werden, obwohl sich dieser nicht direkt mit den Daten des Schweizerischen Nationalparks, sondern eines Nationalparks in den Niederlanden beschäftigt.¹⁰ Die Autoren untersuchen in einem Experiment das unterschiedliche Verhalten drei verschiedener Nutzergruppen, die entweder mit keinerlei Informationen, mit Papierkarten und Broschüren oder digitalen Wanderführern ausgestattet sind. Trotz unterschiedlicher Datengrundlage und -erfassung ist der Ansatz für diese Arbeit relevant, da die Autoren eine Methode vorschlagen, die die Unsicherheiten in den GPS-Daten erheblich

¹⁰Es handelt sich um den *Duinen van Texel* Nationalpark auf der gleichnamigen Insel, bei dem ebenfalls ein LBS von der Firma Camineo zum Einsatz kommt

reduziert und die Datenverarbeitung vereinfacht. Sie nutzen für die Vorverarbeitung vorhandene Hintergrundinformationen in Form der festliegenden Pfadgeometrien im Nationalpark aus, um die GPS-Punkte auf diese zu projizieren. Sie wenden die in Kapitel 4.1.3 beschriebene Datenstruktur der linearen Bezugssysteme an. Dieses Verfahren entspricht im Prinzip einem einfachen *Map Matching*-Ansatz. Es ermöglicht eine Verringerung der Unsicherheiten, welche von Snozzi (2009) als Schwachpunkt bei der Analyse ausgemacht wurden, und wird in dieser Arbeit ebenfalls angewendet.

Basierend auf den beschriebenen existierenden Arbeiten entstand die Vorgehensweise dieser Arbeit. Sie knüpft nicht unmittelbar an die Ergebnisse von Snozzi (2009) an. Es wird keine Entwicklung einer Fuzzy-Klassifizierung oder Ähnlichem durchgeführt. Stattdessen wird ein anderer Ansatz verfolgt, welcher von den beschriebenen Erkenntnissen in mehrerer Hinsicht motiviert und inspiriert ist:

- Um die Fortpflanzung der Unsicherheiten von den Rohdaten durch die einzelnen Prozessierungsschritte bis hin zu den Ergebnissen zu vermeiden, wird sehr hoher Wert auf die Vorverarbeitung und dort auf den Einsatz linearer Bezugssysteme gelegt. Damit soll eine möglichst sichere Basis für die darauf folgenden Analysen geschaffen werden.
- Der Fokus wird hauptsächlich auf die Kontextvariablen Raum und Zeit gelegt, da diese offensichtlich die höchste Relevanz besitzen sowie statistisch am Sichersten extrahierbar sind.
- Diese Variablen werden in einer umfangreicheren Art und Weise auf eine größere Datengrundlage erweitert, um detaillierte Erkenntnisse zu gewinnen, wie es bislang noch nicht geschehen ist. Das bedeutet konkret, dass mehr Wanderpfade sowie neuere Datensätze hinzugezogen werden.
- Einfache, aber aussagekräftige Statistiken, wie sie bisher bereits einige nützliche Erkenntnisse geliefert haben, sollen dadurch ebenso auf mehr Daten erweitert und auf flexible Art und Weise in einer interaktiven Benutzeroberfläche zur visuell gestützten Analyse zusammengestellt werden.
- Mithilfe dieser Analysesoftware soll das Verhalten der Besucher und speziell der Zusammenhang zur Nutzung des Wanderführers untersucht werden.

4 Technische Grundlagen

Bevor die Vorverarbeitung der Daten und die Entwicklung der Analysesoftware beschrieben werden, beschäftigt sich dieses Kapitel zunächst mit den Grundlagen verschiedener Handwerkszeuge, derer sich im weiteren Verlauf der Arbeit bedient wird. Bei der Vorverarbeitung wird *ArcGIS* und die darin mögliche Geoprozessierung mit der Skriptsprache *Python* eingesetzt. Darüber hinaus spielen Datenbanken, im Speziellen *PostgreSQL* und *PostGIS* eine wichtige Rolle. Die Benutzeroberfläche SNP Analytics wurde in der Programmiersprache *Java* entwickelt.

4.1 Geoprozessierung in ArcGIS

In diesem Kapitel werden Methoden der Geoprozessierung in der kommerziellen GIS-Software *ArcGIS* von der Firma ESRI beschrieben. Im Anschluss wird die dafür eingesetzte Programmiersprache *Python* kurz vorgestellt. Der letzte Abschnitt beschäftigt sich mit linearen Bezugssystemen, einer speziellen Datenstruktur, welche während der Vorverarbeitung der Daten eine tragende Rolle spielt.

4.1.1 ArcGIS

ArcGIS ist der Oberbegriff für verschiedene von der Firma ESRI entwickelte GIS-Softwaretools, die zusammen ein umfangreiches Konzept zur Verarbeitung, Analyse und Präsentation von Geodaten bereitstellen (vgl. ESRI, 2010b). Es ist in der GIS-Branche weit verbreitet und wird auch vom Schweizerischen Nationalpark eingesetzt. Zur Vorverarbeitung der Daten in dieser Arbeit wird ausschließlich die *ArcGIS Desktop*¹¹ Komponente mit den Anwendungen

- **ArcCatalog** zur Verwaltung der *ArcGIS*-spezifischen Dateien wie beispielsweise *Shapefiles*, *Feature Classes* und *Geodatabases* sowie
- **ArcMap** zur Visualisierung und Verarbeitung der Geodaten

benutzt.

Ein elementarer und häufig verwendeter Bestandteil eines geographischen Informationssystems und somit auch von *ArcGIS* ist die *Geoprozessierung* (vgl. McCoy, 2004). Dies ist ein Oberbegriff für die Verarbeitung von Geodaten. Durch die Anwendung bestimmter Geoprozessierungsoperationen werden aus bestehenden neue Geodaten erzeugt. Einige typische Beispiele sind die Verschneidung von Flächen, die Berechnung eines Puffers um Punkte/Linien/Flächen oder Koordinatentransformationen.

ArcGIS stellt in der *Toolbox* eine Fülle an Werkzeugen zur Geoprozessierung bereit. Diese können vom Nutzer in Form eines Dialogfensters aufgerufen und auf die entsprechenden Geodaten angewendet werden. Häufig werden jedoch mehrere aufeinander folgende Werkzeuge benötigt

¹¹Im weiteren Verlauf ist mit *ArcGIS* stets *ArcGIS Desktop* gemeint.

und es sind große Datenmengen involviert. Daher sollten die Abläufe automatisiert werden, um Zeit und Aufwand zu sparen. *ArcGIS* bietet für diesen Zweck mehrere Möglichkeiten:

1. **Batch-Prozessierung:** Ermöglicht die automatische, aufeinander folgende Ausführung eines Tools auf unterschiedliche Daten.
2. **Model Builder:** Eine interaktive Benutzeroberfläche, die das Erstellen eines Modells für den kompletten Prozessierungsablauf, inklusive verschiedener Werkzeuge, Daten und Parameter ermöglicht (vgl. McCoy, 2004). Dieses kann gespeichert und wiederverwendet werden und ist mit einem Klick ausführbar.
3. **Scripting:** Das mächtigste Werkzeug, um die Geoprozessierung zu automatisieren (vgl. Tucker, 2004). Es können eigene Skripte für bestimmte Aufgaben geschrieben werden, mit denen verfügbare Werkzeuge und Daten beliebig verwendet werden können. Darüber hinaus können in der Skriptsprache selbst Berechnungen und Operationen durchgeführt werden.

Somit werden dem Benutzer einige Methoden zur Automatisierung bereitgestellt, wobei das Scripting die umfangreichste darstellt. *ArcGIS* unterstützt einige verbreitete Skriptsprachen wie *Python*, *VBScript*, *JScript* und *Perl*, wobei *Python* von ESRI empfohlen wird (vgl. Tucker, 2004).

4.1.2 Python

Die Programmiersprache *Python* ist eine objektorientierte Skriptsprache, die sich laut Bayer (2007) unter anderem durch eine übersichtliche Notation, vielseitige Anwendungsmöglichkeiten und leichte Erlernbarkeit auszeichnet. Als Skriptsprache ist sie zum einen für die schnelle Lösung kleiner Probleme mit möglichst wenig und einfachem Code (*Prototyping*) und für die Steuerung spezieller Anwendungen, wie z. B. *ArcGIS*, geeignet. Zum anderen unterstützt *Python* jedoch ebenso Konzepte der Objektorientierung wie Klassen, Kapselung und Vererbung. Im Gegensatz zu reinen Skriptsprachen ist es somit auch als vollständige Programmiersprache verwendbar.

Das folgende Skript zeigt beispielhaft den Einsatz von *Python* zur Geoprozessierung in *ArcGIS*:

```
1 # Skriptdatei "GeoprocessingTest.py"
2 import sys, arcgisscripting          # Module einbinden
3
4 if len(sys.argv) != 2:
5     sys.exit()                       # Beenden, falls falsche Anzahl an Argumenten übergeben wurde
6
7 # ArcGIS-Geoprocessor erzeugen, Workspace und Toolbox festlegen
8 gp = arcgisscripting.create()
9 gp.workspace = "c:/temp"
10 gp.toolbox = "analysis"
11
12 # Lade Fluss Feature Class aus dem übergebenen Pfad und führe Buffer-Tool aus
13 fluss = sys.argv[1]
```

```
14 if gp.Exists(fluss) == True:
15     print "Feature Class " + str(fluss) + " geladen."
16     gp.buffer(fluss, "fluss_buffer100.shp", 100)
17     print "Puffer erzeugt und in Datei fluss_buffer100.shp geschrieben."
```

Anhand dieses Beispiels sollen zunächst einige besondere Eigenschaften der *Python*-Sprache erklärt werden:

- (2): *Python* ist modularisiert. Funktionen anderer Module oder Skripte können durch `import` eingebunden werden.
- (5), (15)–(17): Um den Code lesbar und kurz zu halten, verzichtet *Python* fast vollständig auf Semikola und Klammern. Einzelne Codeblöcke werden durch Einrücken gekennzeichnet. Falsch eingerückte Blöcke führen zu Fehlern bei der Kompilierung
- (8), (13): *Python* verzichtet auf die explizite Definition von Variablen sowie die Angabe des Datentyps einer Variable. Dieser wird *dynamisch* zum Ausführungszeitpunkt erkannt. Trotzdem ist die Sprache streng typisiert, das heißt jede Variable besitzt einen Datentyp und eine Umwandlung ist nur durch explizites Casting möglich (vgl. Pilgrim, 2004).

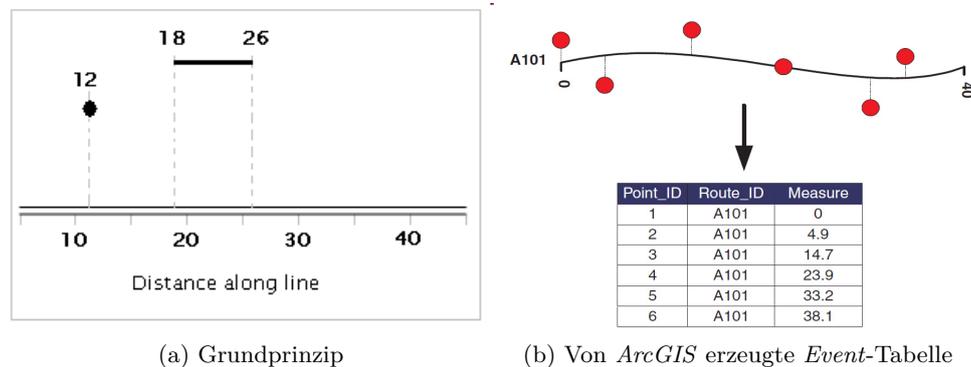
Des Weiteren sind im Beispielskript einige *ArcGIS*-spezifische Komponenten zur Geoprozessierung zu finden. Die Einbindung des `arcgisscripting`-Moduls in Zeile (8) ermöglicht unter anderem die Verwendung aller *ArcGIS*-Funktionen und Werkzeuge aus der *Toolbox*. Im obigen Skript wird exemplarisch das *Buffer*-Werkzeug aus der *Analysis-Toolbox* genutzt. Dieses erstellt um eine *ArcGIS Feature Class* einen Puffer. Der Speicherort dieser Datei ist der einzige Parameter des Skripts (Zeile (13)) und muss bei Aufruf mit übergeben werden, z. B. `GeoprocessingTest.py C:/Data/river.shp`. Die Argumente der `gp.buffer`-Funktion in Zeile (16) sind dieselben Parameter, welche bei einem Aufruf des Tools in *ArcGIS* über die Benutzeroberfläche festgelegt werden.

Eigene *Python*-Skripte mit entsprechenden Parametern können in *ArcGIS* importiert und ausgeführt werden. Die Eingabe der Parameter erfolgt dann genau wie bei den integrierten Werkzeugen in einem Dialogfenster.

4.1.3 Lineare Bezugssysteme

Lineare Bezugssysteme (*Linear Referencing*) beschreiben beliebige Geometrien (Punkte/Linien/Flächen) nicht anhand ihrer absoluten Koordinate im Objektsystem, sondern entlang einer Linie durch die Entfernungsangabe zum Startpunkt (vgl. Brinkhoff, 2008, 96-98). Abbildung 8a verdeutlicht dieses Prinzip. Die Punkte der Geometrien können allein durch den so genannten Messwert (*measure value*) lokalisiert werden.

In vielen Anwendungsbereichen, wo lineare Geometrien von Bedeutung sind, ist auf diese Art und Weise eine effiziente und intuitive Speicherung der Daten möglich. Beispiele sind Straßen-,



(a) Grundprinzip

(b) Von ArcGIS erzeugte Event-Tabelle

Abbildung 8: Das Prinzip linearer Bezugssysteme.

Schienen- und Gewässernetze oder Leitungssysteme (Strom/Gas/Wasser), die mit einer Kilometrierung versehen sind. Relative Angaben wie *Unfall bei Streckenkilometer 157* oder *Stau zwischen Streckenkilometer 321 und 328* sind wesentlich einfacher als die genaue Positionsangabe mittels beliebiger Koordinaten in einem übergeordneten Bezugssystem (vgl. Brennan und Harlow, 2004).

Es gibt darüber hinaus Ereignisse, von denen man weiß, dass sie genau auf der Strecke stattfinden, z. B. Unfälle. Durch lineare Bezugssysteme ist selbst bei Querabweichungen der Position durch Messfehler gesichert, dass diese Ereignisse auch wirklich auf der Linie gespeichert werden und in der Karte nicht neben selbiger erscheinen.

ArcGIS verfügt über eine Reihe von *Linear Referencing*-Werkzeugen in einer eigenen *Toolbox*. Die im Zusammenhang dieser Arbeit wichtigste Funktion ist *Locate Features Along Routes*. Dabei werden beliebige Geometrien, die ähnlich zu Abbildung 8a auch abseits der Polylinie liegen können, wie bei einem einfachen *Map Matching*-Ansatz senkrecht auf die Linie projiziert und deren Messwert berechnet. Voraussetzung für die Anwendung dieses Werkzeuges in *ArcGIS* ist, dass die Linie im speziellen *Route*-Format vorliegt. Denn nur dann besitzt sie die benötigte Kilometrierung für die Berechnung der Messwerte. Um dies zu erreichen, müssen unter anderem Startpunkt und Richtung der Polylinie definiert werden.

Ergebnis der Berechnung ist eine so genannte *Event*-Tabelle (vgl. Abbildung 8b), welche für jede projizierte Geometrie den korrespondierenden *measure value* sowie die orthogonale Entfernung (in der Abbildung nicht dargestellt) zur Polylinie enthält.

4.2 Datenbanken und (Postgre)SQL

In diesem Kapitel werden einige Grundlagen der Datenbanken und der Abfragesprache SQL behandelt sowie das in dieser Arbeit verwendete Datenbankmanagementsystem *PostgreSQL* vorgestellt.

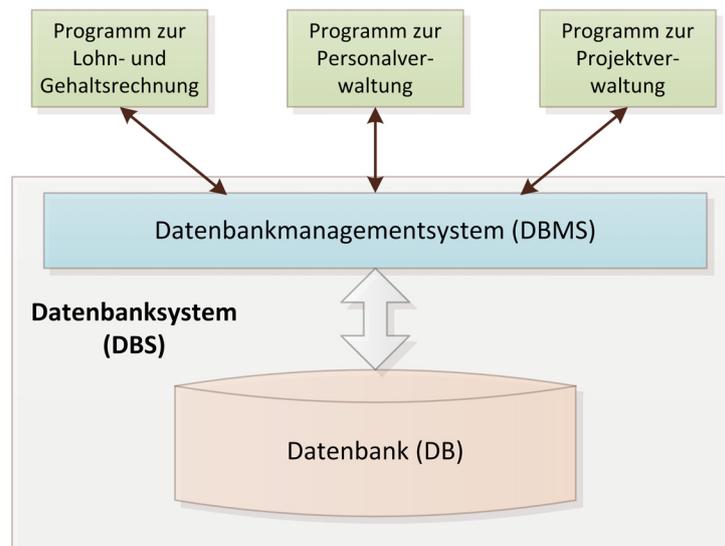


Abbildung 9: Aufbau eines Datenbanksystems. Das DBMS arbeitet als Schnittstelle zwischen Anwendungen und Datenbank (nach RRZN, 2004).

4.2.1 Allgemeines

Häufig muss eine Software auf möglichst effiziente Art und Weise mit großen, komplexen Datenmengen interagieren. Typischerweise kann die herkömmliche Datenspeicherung mittels Dateisystemen dies aus den folgenden Gründen nicht oder nur unzureichend erfüllen (vgl. RRZN, 2004): Gleiche Daten werden teilweise an mehreren Orten **redundant** abgespeichert. Das erschwert Änderungen und erhöht die Fehleranfälligkeit. Bei gleichzeitigem Zugriff mehrerer Nutzer kann es außerdem zu **Inkonsistenzen** kommen, da es keine Instanz gibt, die sich um einen synchronisierten Zugriff kümmert. Das bedeutet, dass beispielsweise zwei Nutzer dieselbe Datei zeitgleich bearbeiten können, und nicht einer nach dem anderen. Darüber hinaus existiert keine **unabhängige Datenverwaltung**. Jede Anwendung, die auf die Daten zugreift, muss diese selbst bereitstellen. Wird die Struktur der Daten geändert, müssen *alle* Anwendungen angepasst werden. Schließlich kann es bei einfachen Dateisystemen zu **Datenschutzproblemen** kommen, da teilweise uneingeschränkte Lesezugriffe existieren. Dieses Problem kann allerdings auch ohne Datenbanken durch Lese- bzw. Schreibschutz und Verschlüsselungsmechanismen gelöst werden.

Die genannten Nachteile werden durch den Einsatz eines *Datenbanksystems (DBS)* behoben, welches die Verwaltung der Daten und den Zugriff auf selbige völlig von der Anwendungsebene abkapselt. Abbildung 9 zeigt den typischen Aufbau eines DBS. Die Daten sind auf unabhängige Art und Weise, in einem nicht auf eine explizite Anwendung beschränkten Format, in einer oder mehreren *Datenbanken (DB)* abgespeichert. Jede Datenbank beinhaltet logisch zusammengehörige Daten eines bestimmten Gebietes. Die Anwendungen greifen nicht direkt auf diese Daten zu, sondern stellen Anfragen an den „Verwalter“ des DBS, das *Datenbankmanagementsystem (DBMS)*. Dieses fungiert als Schnittstelle zwischen Anwendungen und Rohdaten. Es nimmt die Anfragen der Anwendungen entgegen, greift auf die Datenbank(en) zu und liefert das gewünschte Ergebnis zurück. Des Weiteren kümmert es sich unter anderem um die Synchronisation mehrerer

paralleler Anfragen, den Datenschutz, Berechtigungen und Zugriffsregelungen.

Es gibt verschiedene Datenbankmodelle, in denen die Daten jeweils unterschiedlich repräsentiert werden. Beispiele sind hierarchische Datenbanken, Netzwerkdatenbanken oder objektorientierte Datenbanken. Weit verbreitet sind *relationale Datenbanken* (vgl. RRZN, 2004), in welchen die Daten in Tabellenform (Relationen) dargestellt werden. Eine Datenbank besteht aus beliebig vielen Tabellen. Jeder Eintrag (*Tupel*) kann mit denen anderer Relationen verknüpft werden, dabei wird zwischen 1:1, 1:n und n:m Beziehungen unterschieden.

Wenn relationale Datenbanken um objektorientierte Konzepte erweitert werden, wie beispielsweise die Unterstützung umfangreicherer Datentypen, spricht man auch von *objektrelationalen Datenbanken*. Diese vereinen die Vorteile relationaler und objektorientierter Datenbanken.

Die physische Architektur eines Datenbanksystems kann laut RRZN (2004) ebenfalls auf verschiedene Arten erfolgen. Am weitesten verbreitet und auch in dieser Arbeit angewendet ist das *Client-Server-DBS*. Der Server stellt das Datenbanksystem bereit und wartet auf Anfragen der Clients. Der Ablauf erfolgt analog zu Netzwerksystemen mittels der *Request-Response*-Methode. Das heißt, dass die Initiative vom Client ausgeht, der zunächst eine bestimmte Anfrage (*Request*) an den Server schickt. Der bearbeitet diese und teilt das Ergebnis in seiner Antwort (*Response*) mit. Client und Server müssen sich nicht zwangsläufig auf getrennten Rechnern befinden. Generell muss der Client jedoch immer als ersten Schritt eine Verbindung zum Server herstellen, um anschließend die Anfragen absenden zu können.

Die formale Gestalt der Datenbankabfragen wird ebenfalls durch das DBS festgelegt. Bei (objekt-)relationalen Datenbanken hat sich die Abfragesprache SQL durchgesetzt.

4.2.2 SQL

Die Datenbankabfragesprache SQL (*Structured Query Language*) wurde von IBM entwickelt und ist seit 1986 standardisiert. Ihre einfache Struktur soll es auch unerfahrenen Anwendern ermöglichen, mit Datenbanken zu arbeiten. Eine Abfragesprache für relationale Datenbanken soll laut RRZN (2004) das Erstellen, Verändern und Löschen von Relationen ermöglichen. Außerdem sollten neue Relationen aus vorhandenen Tabellen mit bestimmten Attributen erzeugt werden können. Dabei ist die Auswahl (*Retrieval*) und Anzeige der gewünschten Daten die Hauptaufgabe. SQL ist eine sehr umfangreiche und mächtige Sprache zur Arbeit mit Datenbanken. Im Folgenden werden lediglich einige grundlegende Sprachkonstrukte vorgestellt.

Der wichtigste und am häufigsten benötigte Teil von SQL ist die **SELECT**-Anweisung. Mithilfe dieses Konstruktes lassen sich ausgewählte Datenfelder aus einer oder mehreren Tabellen auswählen. Durch optionale Attribute können komplexe Bedingungen formuliert werden. Die Syntax ist immer wie folgt aufgebaut, optionale Befehle sind in eckige Klammern gesetzt (vgl. RRZN, 2004):

```
1 SELECT [DISTINCT] * | Datenfelder FROM Tabellenname
2     [WHERE Bedingung]
```

```

3      [GROUP BY Datenfelder [HAVING Bedingung] ]
4      [ORDER BY Datenfelder [ASC | DESC] ]
5      [LIMIT [Start,] Anzahl ];

```

Hinter dem `SELECT`-Schlüsselwort können die anzuzeigenden Spalten im Ergebnis ausgewählt werden, wobei `*` für *alle* steht. Nach dem `FROM`-Schlüsselwort folgen die Namen einer oder mehrerer Tabellen, aus denen die Daten abgefragt werden sollen. Ein wichtiger Teil folgt nach der `WHERE`-Klausel: Hier können Bedingungen an die Daten gestellt werden, die mittels logischer Operatoren verknüpft werden. Die Befehle in den letzten drei Zeilen werden zur Formatierung der ausgegebenen Daten verwendet.

Dieses grundlegende Sprachelement soll anhand zweier einfacher Beispiele verdeutlicht werden. Gegeben sei Tabelle 1 mit dem Namen `staedte`:

ID	Name	Breite	Laenge	Einwohner	Flaeche	Bundesland
001	Hannover	52,377	9,741	520966	204,14	Niedersachsen
002	Frankfurt am Main	50,107	8,664	671927	248,31	Hessen
003	München	48,140	11,560	1330440	310,43	Bayern
004	Freiburg	47,989	7,836	221924	153,07	Baden-Württemberg
005	Köln	50,943	6,958	1000298	405,16	Nordrhein-Westfalen
006	Hamburg	53,553	10,000	1774224	755,26	Hamburg
007	Berlin	52,521	13,410	3442194	891,85	Berlin

Tabelle 1: Ausgangstabelle des SQL-Beispiels.

Beispiel 1: Zeige alle Städte, die entweder südlich des 50. Breitengrades liegen oder deren Einwohnerzahl über eine Million beträgt.

```

1 SELECT * FROM staedte
2 WHERE Breite < 50.0 OR Einwohner > 1000000

```

ID	Name	Breite	Laenge	Einwohner	Flaeche	Bundesland
003	München	48,140	11,560	1330440	310,43	Bayern
004	Freiburg	47,989	7,836	221924	153,07	Baden-Württemberg
005	Köln	50,943	6,958	1000298	405,16	Nordrhein-Westfalen
006	Hamburg	53,553	10,000	1774224	755,26	Hamburg
007	Berlin	52,521	13,410	3442194	891,85	Berlin

Tabelle 2: Ergebnisrelation des ersten SQL-Beispiels.

Beispiel 2: Zeige die beiden flächengrößten Städte westlich des zehnten Längengrades. Die Ausgabe soll nur die Spalten `Name`, `Laenge` und `Flaeche` enthalten.

```

1 SELECT Name, Laenge, Flaeche FROM staedte
2 WHERE Laenge < 10.0
3 ORDER BY Flaeche DESC
4 LIMIT 2

```

Name	Laenge	Flaeche
Köln	6,958	405,16
Frankfurt am Main	8,664	248,31

Tabelle 3: Ergebnisrelation des zweiten SQL-Beispiels.

Wie oben angesprochen, ist neben diesem vorgestellten Kernstück der Datenabfrage das Anlegen neuer Relationen, sowie das Verändern und Löschen selbiger ein elementarer Teil jeder Abfragesprache. SQL stellt dafür die folgenden Befehle bereit:

- **CREATE DATABASE | TABLE Name:**
Ermöglicht das Erstellen von neuen Datenbanken oder Tabellen.
- **INSERT INTO Tabellenname(Feld1,...,FeldN) VALUES (Wert1,...,WertN):**
Wird zum Einfügen neuer Datensätze in existierende Relationen verwendet.
- **UPDATE Tabellenname SET Feld1 = Wert1,...,FeldN = WertN [WHERE Bedingung]:**
Kann für die Veränderung bestimmter Daten in einer Tabelle benutzt werden.
- **DELETE FROM Tabellenname [WHERE Bedingung]:**
Löscht bestimmte Einträge einer Relation.

Neben den vorgestellten Grundelementen von SQL gibt es eine Fülle weiterer Instrumente und Methoden, die an dieser Stelle jedoch nicht weiter behandelt werden sollen. Für die Datenbankinteraktion der Analysesoftware in dieser Arbeit reichen meist einfache Abfragen aus.

Die Verarbeitung von SQL-Anfragen und die Darstellung der Ergebnisse erfolgt durch das DBMS. Nicht selten stellt selbiges eine graphische Benutzeroberfläche zur Verfügung, die dem Nutzer die Interaktion mit der Datenbank, das Abfragen, Anlegen, Verändern und Löschen von Daten im Vergleich zur direkten Eingabe von SQL vereinfacht.

4.2.3 PostgreSQL und PostGIS

Es gibt viele DBMS auf dem Markt, die zwar im Kern dieselbe Aufgabe erfüllen, jedoch in ihrer Umsetzung stark variieren. Bekannte verfügbare Systeme sind beispielsweise *Microsoft Access*, *Oracle DB*, *MySQL* oder *MaxDB*, um nur einige Vertreter zu nennen. Ein weiteres verbreitetes System, welches in dieser Arbeit verwendet wird, ist das objektrelationale DBMS *PostgreSQL*. Dabei handelt es sich laut PostgreSQL (2009) aufgrund der umfangreichen Funktionalität um eines der fortschrittlichsten und mächtigsten *OpenSource*-Systeme.

Eine graphische Benutzeroberfläche wird von *PostgreSQL* selbst nicht angeboten. Die Software *pgAdmin* (2010) ist jedoch ein weitläufig genutztes Verwaltungstool für diesen Zweck. Abbildung 10 zeigt einen Ausschnitt aus der Programmoberfläche.

PostgreSQL ist plattformunabhängig und kann durch eigene Datentypen, Funktionen usw. beliebig erweitert und modifiziert werden. *PostGIS* (2010) ist eine dieser Erweiterungen, welche neue Funktionen und Datentypen bereitstellt, um *räumliche* Datenbanken zu erstellen und zu

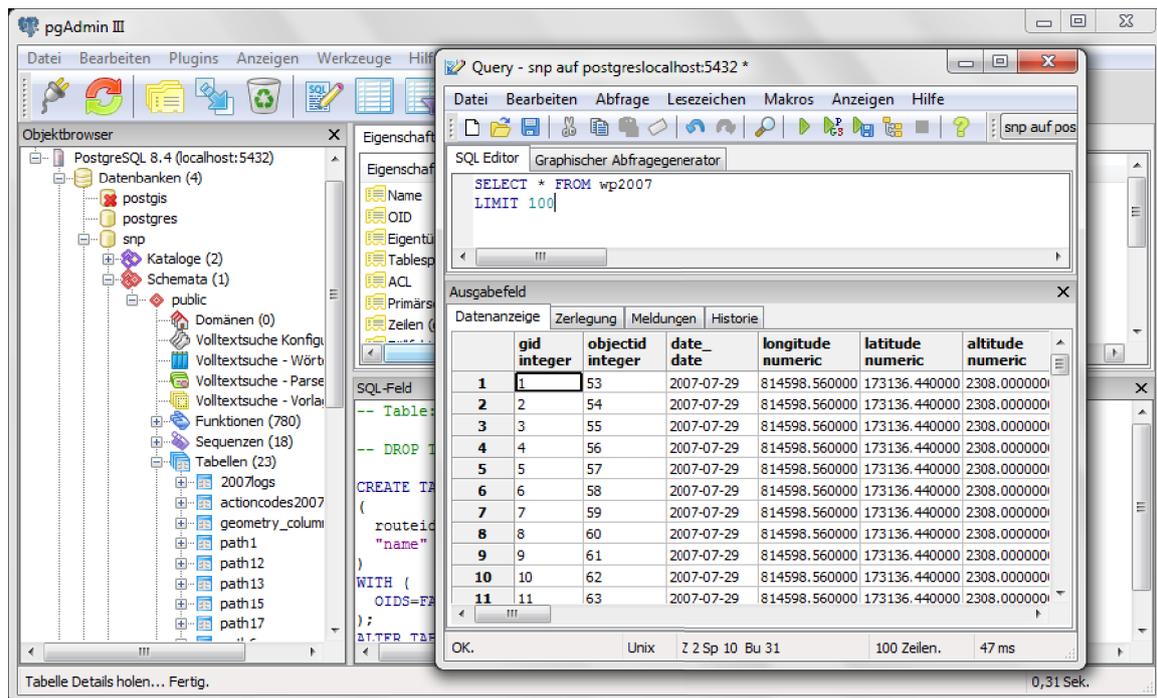


Abbildung 10: Oberfläche von *pgAdmin*, einem umfangreichen Verwaltungstool für *PostgreSQL*-Datenbanken.

verwalten. Dies beinhaltet die Bereitstellung räumlicher Datenstrukturen und geometrischer Primitive wie Punkte, Linien und Flächen. Das Geometriemodell von *PostGIS* genügt der *Simple Feature Specification* des OGC (1999) und bietet eine Vielzahl an Funktionen zur Prozessierung von Geodaten wie beispielsweise Koordinatentransformationen, Schnitt- oder Abstandsberechnungen. Dabei übernimmt *PostGIS* Konzepte und Funktionen vom ebenfalls standardisierten *SQL/MM Spatial*, einer Geodatenerweiterung von SQL (vgl. Stolze, 2003, Brinkhoff, 2008, 87-92).

4.3 GUI-Programmierung mit Java und Swing

Dieses Kapitel gibt einen kurzen Überblick über die Programmierung von graphischen Benutzeroberflächen mit *Java*. Die Analysesoftware wird auf diese Art und Weise erstellt und es gibt einige wichtige Eigenschaften der GUI-Programmierung, die hier erklärt werden. Besondere Berücksichtigung findet der Aspekt der Nebenläufigkeit mehrerer „Programmfüden“ (*Threads*).

4.3.1 Java und GUIs

Java ist eine vollständig objektorientierte Sprache. Bis auf wenige Ausnahmen sind alle Komponenten der Sprache so genannte *Objekte*. Die Programmstruktur setzt sich ausschließlich aus solchen Objekten und Methoden zusammen, welche in Klassen zusammengefasst sind (vgl. Kröckertskoth, 2007, 15).

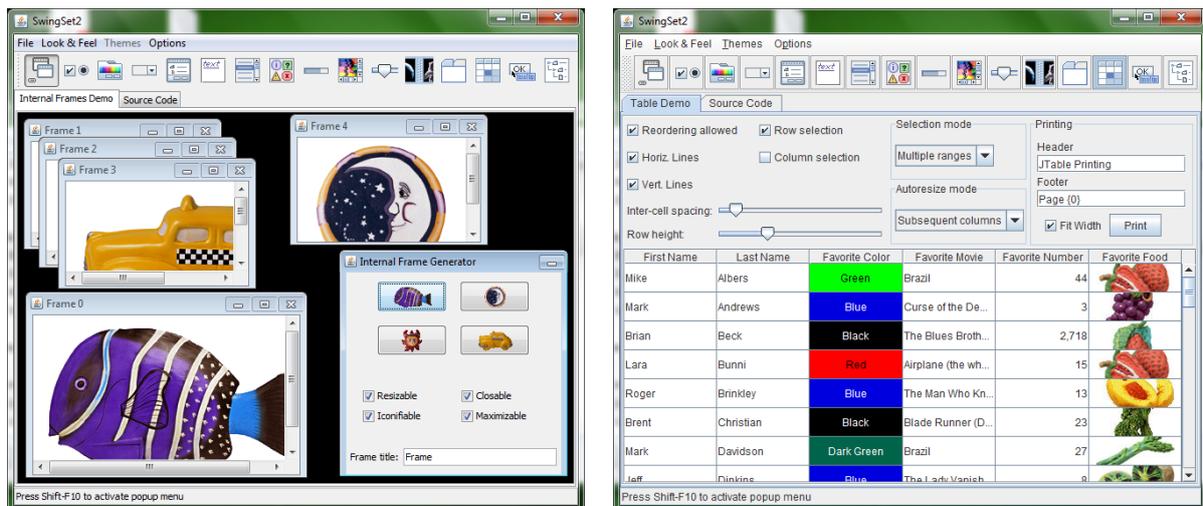
(a) JInternalFrames im *Windows Look & Feel*(b) JTable im *Java Look & Feel*

Abbildung 11: Die bei jeder Java-Installation mitgelieferte *Swing Demo* gibt einen Überblick über alle Swing-Komponenten.

Java stellt zur Programmierung graphischer Benutzeroberflächen (*Graphical User Interfaces – GUI*) das **Abstract Window Toolkit (AWT)** zur Verfügung. Dabei handelt es sich um eine Bibliothek, die Klassen mit entsprechenden Methoden enthält, mit denen GUI-Komponenten wie Fenster, Dialoge, Schaltflächen, Textfelder, Graphiken und vieles mehr erstellt werden können. Die AWT-Komponenten werden mittels der internen GUI-Funktionalität der jeweiligen Zielplattform dargestellt, was den Vorteil hat, dass sich die Oberfläche an das Aussehen des jeweiligen Betriebssystems anpasst. Da AWT, wie alle *Java*-Komponenten, plattformunabhängig ist, sind somit jedoch die darstellbaren Elemente auf die Schnittmenge der intern verfügbaren GUI-Komponenten aller unterstützten Plattformen beschränkt (vgl. Ullenboom, 2009). Daher bietet AWT eine nicht allzu große Auswahl an *Widgets*.

Aufgrund dieser Beschränkung von AWT wurden die *Java Foundation Classes (JFC)* mit dem **Swing**-Konzept entwickelt. Im Unterschied zu AWT bietet Swing wesentlich mehr und flexiblere GUI-Komponenten an, welche nicht mehr die plattformspezifischen GUI-Funktionalitäten nutzen, sondern selbst fast vollständig in *Java* geschrieben sind (vgl. Ullenboom, 2009). Sie werden als *lightweight components* bezeichnet. Somit ist die optische Darstellung nicht mehr von der jeweils ausführenden Plattform abhängig. Vielmehr gibt es die Möglichkeit, zwischen verschiedenen Darstellungsformen zu wählen und diese sogar zur Laufzeit zu ändern. Dieses Konzept wird als *pluggable Look & Feel* bezeichnet. Abbildung 11 zeigt eine Auswahl verschiedener Swing-Komponenten in zwei unterschiedlichen *Look & Feels*.

DOS-basierte Anwendungen sind durch einen linearen Programmablauf gekennzeichnet. Im Gegensatz dazu ist dieser bei graphischen Benutzeroberflächen stark nichtlinear und von Benutzereingaben abhängig. Durch Mausklicks oder Tastatureingaben werden laufend so genannte *Events* ausgelöst. Die Behandlung dieser Ereignisse wird nicht von den Swing-Komponenten übernommen, sondern ist bereits im AWT implementiert. Es gibt verschiedene *EventListener*,

die alle von der abstrakten Klasse *AWTEvent* abgeleitet sind. Der Programmierer kann mithilfe der *Listener* festlegen, welche Funktionen beim Eintreten bestimmter Ereignisse ausgeführt werden sollen.

4.3.2 Aspekt der Nebenläufigkeit

Die parallele Ausführung mehrerer Prozesse ist in jedem heutigen Computer selbstverständlich. Echte Nebenläufigkeit kann nur durch Rechner mit mehr als einem Prozessorkern erreicht werden. Dennoch funktioniert dieses Prinzip gleichwohl auf Einprozessorrechnern, indem das System ständig mit hoher Geschwindigkeit zwischen allen laufenden Prozessen wechselt und somit eine Pseudoparallelität suggeriert (vgl. Ullenboom, 2009).

Bei umfangreichen *Java*-Anwendungen mit vielen verschiedenen Programmsträngen und Aufgaben ist es ebenfalls sinnvoll, mehrere Prozesse parallel ablaufen zu lassen. Dies ist durch die Erstellung beliebig vieler *Threads* prinzipiell einfach realisierbar.

Bei der GUI-Programmierung spielt die Nebenläufigkeit mehrerer *Threads* laut Goetz u. a. (2006) eine besondere Rolle. Die meisten Swing-Komponenten sind im Gegensatz zum AWT nicht „*Thread*-sicher“. Das hat zur Folge, dass alle GUI-bezogenen Prozesse, wie die Anzeige der Komponenten und die Ereignisbehandlung, alle aus demselben *Thread* aufgerufen werden müssen, um keine Fehler zu produzieren. Das Swing-Konzept kann demnach wie viele andere GUIs als „Ein-*Thread*-System“ bezeichnet werden. Dieser eine *Thread* trägt den Namen *Event Dispatch Thread* und ist für die Darstellung und Interaktion mit der Benutzeroberfläche verantwortlich. Längere Prozesse sollten *nicht* in ihm ausgeführt werden, da dies dazu führt, dass die Oberfläche und die Ereignisbehandlung nicht mehr laufend aktualisiert werden. Die Folge ist das als „Einfrieren“ bekannte verzögerte oder fehlende Reagieren auf Benutzereingaben. Solch ein Verhalten muss zweifellos vermieden werden. Daher sollte für lang andauernde Berechnungen, beispielsweise eine komplexe Datenbankabfrage, ein separater *Thread* gestartet werden. Dieser läuft im Hintergrund ab und beeinflusst die Prozesse des *Event Dispatch Threads* nicht.

5 Vorverarbeitung der Daten

Dieses Kapitel beschreibt mit der Vorverarbeitung der Daten einen wichtigen Teil der Arbeit. Zunächst werden die Daten selbst beschrieben und die Notwendigkeit der Vorverarbeitung als Grundlage für das weitere Vorgehen dargelegt. Anschließend sind die einzelnen Schritte erklärt, die zur Verarbeitung der Daten jeweils eines ganzen Jahres durchgeführt werden müssen. Es beginnt mit der Konvertierung der Rohdaten, gefolgt von mehreren Verarbeitungsschritten in *ArcGIS*. Am Ende werden die Daten in einer *PostGIS*-Datenbank abgelegt, um sie im Anschluss in der Analysesoftware verwenden zu können.

5.1 Datenbeschreibung und Vorgehen

Der digitale Wanderführer mit der *WebPark*-Software ist seit mehreren Jahren im Schweizerischen Nationalpark erfolgreich im Einsatz. Wie zu Beginn der Arbeit erwähnt, werden zu Forschungszwecken einige Daten der zur Ausleihe verfügbaren Geräte des Nationalparks anonym aufgezeichnet. Dies bedeutet konkret, dass *jedliche* Interaktion des Nutzers mit dem Programm, jeder Aufruf einer bestimmten Funktion, abgespeichert wird. Neben Informationen über die Art der ausgeführten Aktion wird ebenfalls die GPS-Position, falls verfügbar, und die Uhrzeit zum Zeitpunkt des Aufrufs gespeichert. Das Ergebnis sind reichhaltige Datensätze, die für jede Wandersaison der vergangenen Jahre von den Geräten vorliegen. Mit diesen kann die essentielle Verbindung zwischen der räumlichen Position der Besucher und der getätigten Aktion mit dem digitalen Guide hergestellt werden.

Abbildung 12 zeigt einen Ausschnitt der Daten aus dem Jahr 2007. Bei der ersten visuellen Inspektion des Datensatzes fällt auf, dass sich die aufgezeichneten Positionen hauptsächlich um die festen Wanderrouen im Park bewegen. Darüber hinaus sind sie aber auch an vielen anderen Orten in und um den Nationalpark zu finden. Dies ist unter anderem durch die Nutzung des Wanderführers im Auto oder Zug zu erklären, da innerhalb bzw. in der nahen Umgebung des Nationalparkgebietes sowohl einige Straßen als auch Zugstrecken verlaufen. Außerdem haben Parkmitarbeiter das Gerät möglicherweise abseits der Wanderwege benutzt, was den Besuchern nicht gestattet ist (vgl. Kapitel 2.2).

Weiterhin können schnell einige Ballungsräume mit erhöhter Punktdichte ausgemacht werden. Dies sind hauptsächlich das Nationalparkzentrum in Zernez, welches Ausgangspunkt der meisten Touren ist, das Hotel *Il Fuorn* sowie innerhalb des Parkgebietes verteilte Picknickplätze oder Almhütten, zum Beispiel die *Cluozza*-Hütte. Die stellenweise sehr hohe Streuung der Positionen fällt gerade im Bereich dieser Regionen mit erhöhter Nutzung ins Auge. Unter den vielen vorhandenen Aufzeichnungen scheinen die GPS-Positionen teilweise stark fehlerbehaftet zu sein. Die in Kapitel 3.4 geschilderten Vorarbeiten mit diesen Daten zeigen, dass die Unsicherheit der Nutzerposition ein großes Problem darstellt. Die zum Teil sehr hohe Streuung bedeutet eine unsichere Basis für jede auf den Daten aufbauende Analyse, erschwert diese und resultiert letztendlich in ebenso unsicheren Ergebnissen und Aussagen. Die Addressierung dieses Problems ist

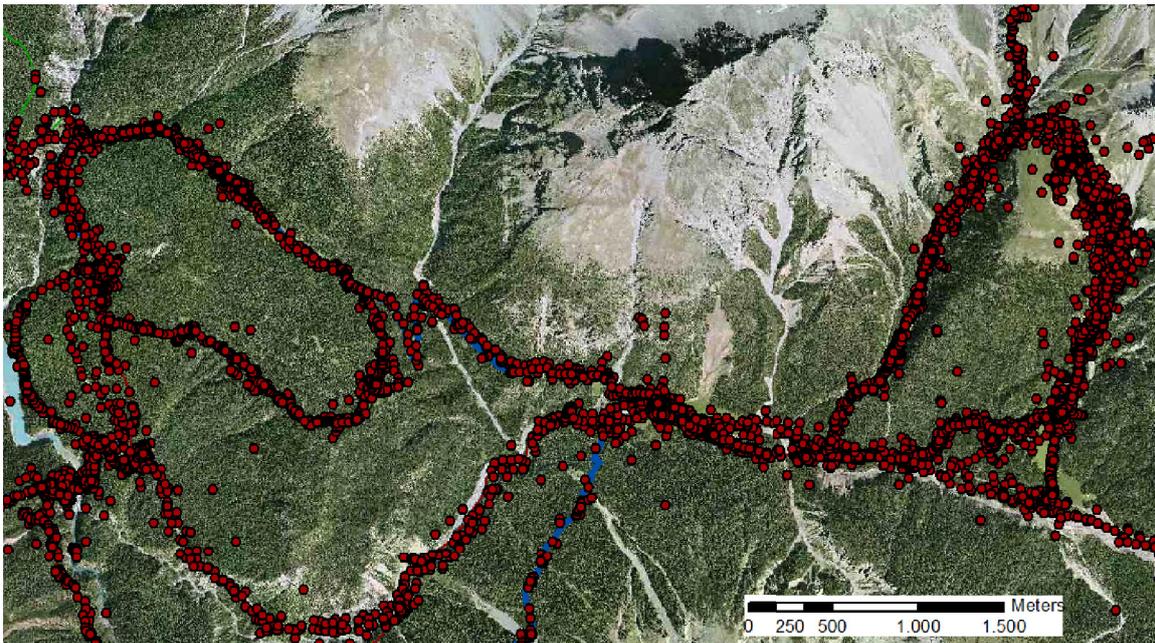


Abbildung 12: Ausschnitt aus den Daten von 2007. Jeder Punkt steht für die Interaktion des Besuchers mit dem digitalen Wanderführer an dieser Position. Einzelne Wanderrouen und Verdichtungsgebiete sind deutlich zu erkennen.

das Ziel und die Rechtfertigung für die umfangreiche Vorverarbeitung. Es wird damit eine stabile Grundlage für die darauf aufbauende Datenanalyse geschaffen. Im Folgenden wird die Strategie erläutert, die dies erreichen soll.

Die Ungenauigkeiten sind Teil der Daten und naturgemäß durch die GPS-Positionierung entstanden. Durch teils starke Abschattungen wird oft nicht die bestmögliche Genauigkeit erreicht. Hinzu kommt, dass die Breite der Wanderrouen im Bereich von Picknickplätzen im Vergleich zum normalen Rouenverlauf erhöht ist und dort somit ohnehin stärker verteilte Punkte vorzufinden sind. Dennoch gehören auch diese Punkte zu der Roue, auf dem sich der Picknickplatz befindet.

Es gibt im Nachhinein keine Möglichkeit, die durch GPS-Fehler entstandenen Streuungen gänzlich zu eliminieren bzw. die Positionen zu verbessern. Von vornherein ist es somit offensichtlich, dass auf Analysen verzichtet werden muss, die eine sehr hohe Positionsgenauigkeit erfordern. Dies liegt generell in der Natur raumzeitlicher Daten (vgl. Kapitel 3.3.1), solange keine präziseren Positionierungstechniken genutzt werden können bzw. existieren.

Um die Daten dennoch in gewissem Maße zu verbessern, wurde eine Strategie ausgearbeitet, die externes Hintergrundwissen über die Daten und ihre Herkunft einbringt, welches in den Rohdaten nicht vorhanden ist. Es wird die Tatsache ausgenutzt, dass die Wanderer sich *nur auf den festgelegten Wanderrouen* durch den Park bewegen und diese nicht verlassen dürfen (vgl. Kapitel 2.2). Somit kann jeder der um die Wanderpfade streuenden Einträge konkret dem Pfad zugeordnet werden, abgesehen von solchen, die von Parkmitarbeitern oder sich illegal verhaltenden Besuchern gemacht wurden. Da die Geometrie der Wanderpfade im GIS des Nationalparks mit hoher Genauigkeit vorliegt, bildet dieser Bezug auf die Wanderrouen das Herzstück der

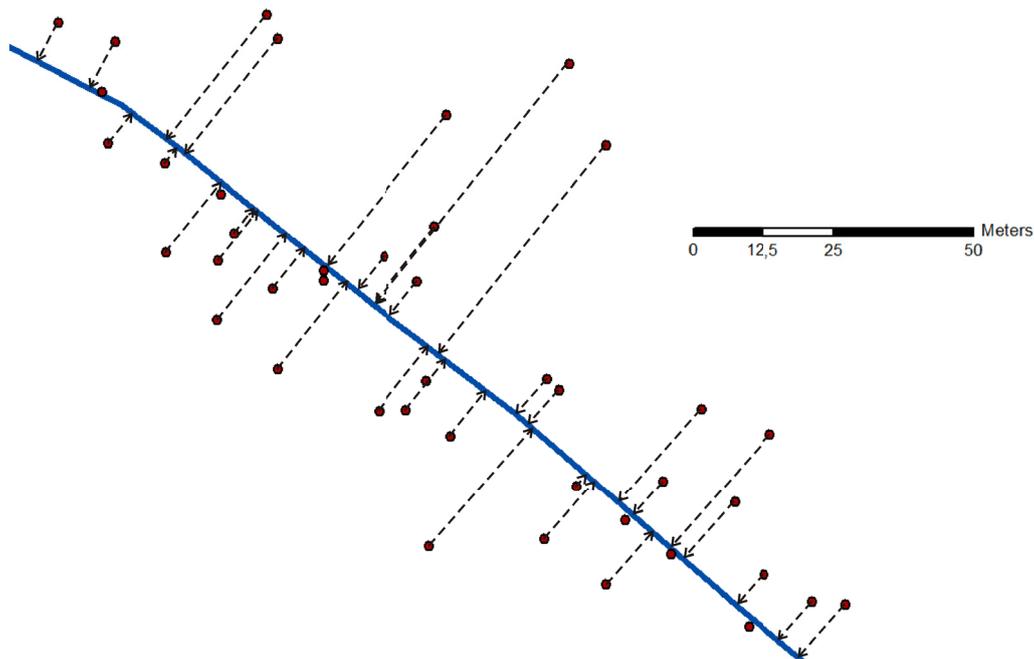


Abbildung 13: Alle Punkte werden gemäß linearer Bezugssysteme senkrecht auf den nächstgelegenen Wanderpfad projiziert.

Vorverarbeitung:

Die ursprüngliche Beschreibung der Besuchertrajektorien durch die stark streuenden GPS-Positionen wird zu Gunsten einer semantisch reichhaltigeren Beschreibung, die direkt auf die präzisen Wanderpfade bezogen ist, aufgegeben.

Dieser Bezug der gemessenen Nutzerpositionen auf die Pfadgeometrien ist ein *Map Matching*-Problem (vgl. Kapitel 3.2). In dieser Arbeit wird mit linearen Bezugssystemen ein relativ einfacher Lösungsansatz eingesetzt, der lediglich geometrische Aspekte berücksichtigt: Jeder Punkt in einem gewissen maximalen Abstand wird senkrecht auf den nächst gelegenen Wanderpfad *projiziert*. Topologische Eigenschaften werden vorerst nicht berücksichtigt. Abbildung 13 illustriert das Grundprinzip, in Kapitel 5.3.3 wird die technische Umsetzung genauer erklärt.

Es kann allerdings nur ein Teil aller Positionen auf die Wanderrouen projiziert werden. Viele Einträge liegen zu weit weg und können nicht mit einer konkreten Route in Verbindung gebracht werden. Aufgrund der starken Streuung der Daten ist es schwierig, einen festen Schwellwert für die maximale Entfernung eines Punktes zu einer Route festzulegen. Der Lösungsansatz hierfür wird ebenfalls in Kapitel 5.3.3 erläutert.

Diese Projektion der Punkte auf die Wanderpfade bildet das Kernstück der Vorverarbeitung und die zentrale Idee, ist aber nicht der einzige durchzuführende Schritt. Es müssen vielmehr noch weitere Stationen durchlaufen werden, um von den Rohdaten zu der angestrebten Datenbank zu

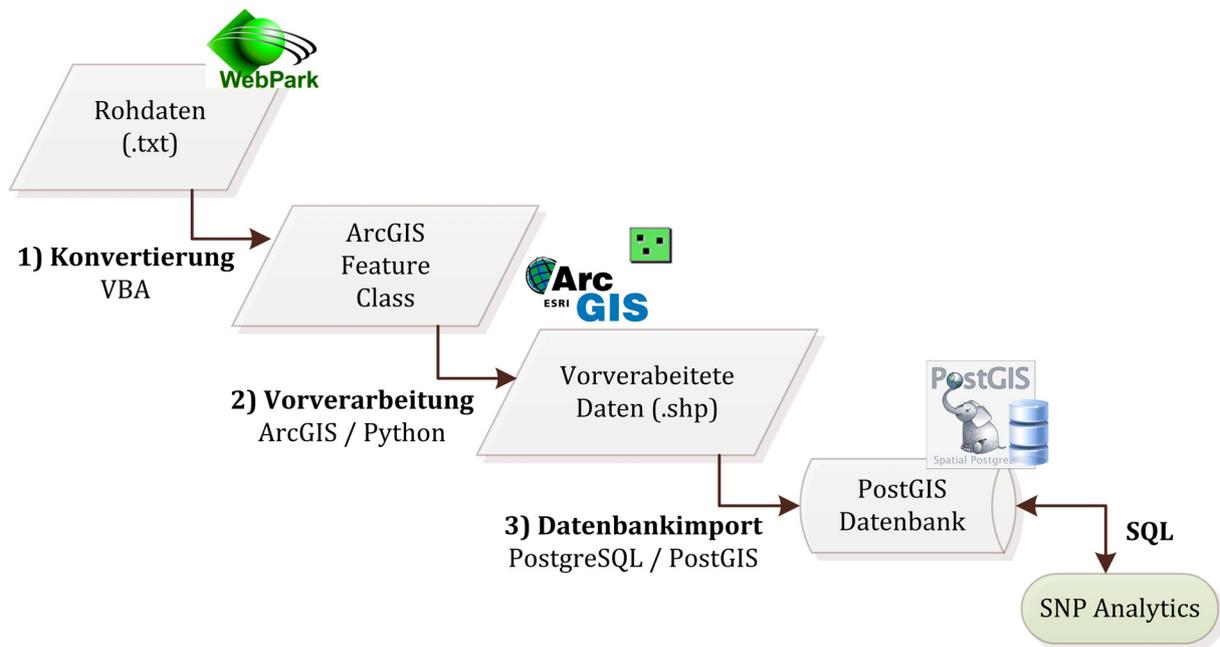


Abbildung 14: Ablauf der Vorverarbeitung.

gelangen. Die folgenden Erläuterungen dazu sind in drei Abschnitte gegliedert, die Abbildung 14 zeigt.

5.2 Konvertierung der Rohdaten

Die Bewegungsdaten liegen nach ihrer Aufzeichnung keineswegs in einem Format vor, welches wie in Abbildung 12 direkt in *ArcGIS* visualisiert werden kann. Die Geräte speichern die Daten zunächst in einfachen Textdateien. Daher besteht der erste Verarbeitungsschritt darin, die Daten eines Jahres von allen Geräten auszulesen und in einer von *ArcGIS* lesbaren, georeferenzierten *Feature Class* zusammenzustellen, die alle Einträge einer Saison enthält. Die Konvertierung der Rohdaten basiert auf dem Tabellenkalkulationsprogramm *Microsoft Excel* und der zugehörigen Skriptsprache *Visual Basic for Applications (VBA)*. Dieser Verarbeitungsschritt ist nicht Teil der vorliegenden Arbeit, sondern das entsprechende Skript zur übersichtlichen Zusammenstellung der Daten wird von Jonathan Raper, einem der Betreuer, erstellt. Da die *WebPark*-Software kontinuierlich weiterentwickelt wird, sehen die Rohdaten in jedem Jahr anders aus und das VBA-Skript muss stetig angepasst werden.

Das Skript stellt die für die weitere Bearbeitung in *ArcGIS* verwendbare *Feature Class* zusammen. Es extrahiert die GPS-Positionen und entschlüsselt unter anderem die in den teilweise kryptischen Zeichenketten der Rohdaten vorhandenen Informationen. Die daraus entstehende Tabelle stellt diese übersichtlich zusammen und enthält im Einzelnen folgende Spalten:¹²

- **ObjectID:** Die eindeutige ID eines Eintrags, wird von *ArcGIS* automatisch erzeugt.

¹²Die genaue Benennung der einzelnen Spalten basiert auf den Daten des Jahres 2007, kann aber, je nach Veränderung des Konvertierungsskripts, bei anderen Datensätzen variieren.

- **Date_**: Datum und – falls vorhanden – Uhrzeit des Eintrags. Da die Uhrzeit aus dem GPS-Empfänger stammt, ist sie nur dann verfügbar, wenn eine GPS-Positionslösung vorliegt.
- **Longitude, Latitude**: X- bzw. Y-Koordinate der Position in Metern im alten Lagebezugssystem der Schweizerischen Landesvermessung *CH1903* (vgl. SwissTopo, 2010).
- **Altitude**: Z-Koordinate der Position in Metern im alten Höhenbezugssystem der Schweizerischen Landesvermessung *LV03 / LN02* (vgl. SwissTopo, 2010).
- **Accuracy**: Lagegenauigkeit der GPS-Positionierung in Form des HDOPs (*Horizontal Dilution of Precision*).
- **Action_**: Die vom Benutzer ausgeführte Programmfunktion in ihrer nahezu unveränderten Form. Dies ist aufgrund der webbasierten Software in der Regel eine URL, die den Pfad der zu öffnenden HTML-Datei angibt, z. B. `baer/main.de.htm`. Im Gegensatz zu diesem einfachen Beispiel ist sie oft lang und nicht unmittelbar verständlich.
- **Code**: Eines der wichtigsten Felder der Tabelle, in dem jeder der im *WebPark*-Programm vorhandenen Funktionen eine Nummer zugewiesen wird. Dabei werden ähnliche Aktionen, deren unbearbeitete *Action_*-Zeichenketten leicht variieren können, zu einem *Action Code* zusammengefasst. Beispielsweise bedeutet der Code 101 den Aufruf der Hauptseite (*Main-Page*), oder der Code 302 den Aufruf des Profils einer bestimmten Route (*Route vertical profile*). Die Bedeutung aller Codes ist im Anhang dokumentiert. In der Analysesoftware wird später fast ausschließlich mit diesen Nummern anstelle des *Action_*-Feldes gearbeitet.
- **Route, Content_ur, IFOI, Around_me, Apps, Photo_bo, Search_ter, Language, Map_overla, Push, Sound**: Diese Spalten sind dazu da, um die in der *Action_*-Zeichenkette enthaltenen Informationen noch einmal explizit zu extrahieren und einfacher zugänglich zu machen. Sie enthalten Attribute der einzelnen Programmfunktionen, wie beispielsweise die abgefragte Wanderoute oder den eingegebenen Suchbegriff.
- **Device**: Die eindeutige ID des Gerätes, von dem die Daten stammen.

Die Vorverarbeitungsmethodik, die in dieser Arbeit verwendet wird, knüpft an diese *Feature Class* an und beginnt mit der Weiterverarbeitung in *ArcGIS*.

5.3 Verarbeitung in ArcGIS

Die georeferenzierten Daten jeweils eines Jahres liegen nach der Konvertierung in einer einzigen Tabelle vor, die alle Informationen über die Nutzeraktionen aus den Geräten auf kompakte Weise vereint. Im zweiten Verarbeitungsschritt wird die Geoprozessierung in *ArcGIS* genutzt (vgl. Kapitel 4.1).

Der Ablauf setzt sich zusammen aus der Ausführung verschiedener eigens für diesen Zweck programmierter *Python*-Skripte und bereits existierender Werkzeuge aus der *ArcGIS Toolbox*.

5.3.1 Vorbereitung

Zu Beginn wird in *ArcCatalog* eine neue *File Geodatabase* angelegt. Dies ist ein proprietäres *ArcGIS*-Datenformat, welches von den Herstellern empfohlen wird (vgl. ESRI, 2010a). Sie dient als Container für alle weiteren erzeugten Tabellen und Features. Falls die Datentabelle noch nicht in diesem Format vorliegt, kann sie auf einfache Weise umgewandelt werden.

Anschließend wird das erste Skript *PrepareData.py* ausgeführt, welches sich prinzipiell in zwei Abschnitte gliedert:

Der erste ist die Bereinigung der Original-Daten. Dies ist ein typischer Schritt bei der Datenverarbeitung, da die Rohdaten oft fehlerbehaftet sind (*dirty data*). Viele der Tabelleneinträge sind aus verschiedenen Gründen für die weitere Verarbeitung nutzlos: Zum einen treten ungültige Einträge auf, die unsinnige Werte beinhalten. Oder mehrere aufeinander folgende Zeilen haben den exakt gleichen Inhalt, sind unbrauchbare Duplikate. Dies beides ist auf Fehler in der *WebPark*-Software, z. B. Probleme beim Abspeichern der Informationen zurückzuführen. Einen nicht unerheblichen Teil aller Einträge, die in diesem Skript gelöscht werden, sind darüber hinaus solche, die zwar korrekte Werte bezüglich der getätigten Programmfunktion beinhalten, jedoch keine gültige GPS-Position besitzen. Daher sind sowohl die Uhrzeit als auch die Position nicht bekannt. Eine Projektion dieser Einträge ist somit unmöglich. Der hierdurch notwendigerweise entstehende Informationsverlust ist Teil der Diskussion in Kapitel 8.1.

Nach der Bereinigung wird die Tabelle im zweiten Teil des Skripts durch einige neue Spalten erweitert, welche im weiteren Verlauf sowohl der Vorverarbeitung als auch der Analyse genutzt werden. Deren Inhalt und Bedeutung wird im Folgenden detailliert erläutert:

Time Dieses Feld beinhaltet keinerlei neue Informationen, sondern extrahiert lediglich die Uhrzeit aus der *Date_*-Spalte. Der Grund dafür liegt darin, dass für die Erstellung der Datenbank gezwungenermaßen ein *Shapefile* erzeugt werden muss (siehe Kapitel 5.4.2). Dieses spezielle *ArcGIS*-Format für räumliche Daten erlaubt innerhalb einer Spalte lediglich die Speicherung eines Datums, nicht aber von Datum *und* Uhrzeit. Daher wird die Zeit hier noch einmal explizit in einem eigenen Feld als einfache Zeichenkette abgespeichert.

Session und Subsession Die umfangreichen Daten werden in einzelne Abschnitte unterteilt, das heißt *klassifiziert*. Dies geschieht durch die Erzeugung von *Sessions* mit einer eindeutigen Nummer. Jede *Session* steht dabei für einen Besucher und enthält die Daten eines bestimmten Gerätes an einem Tag des Jahres. Anschließend wird anhand der Uhrzeiten eine weitere Differenzierung in *Subsessions* vorgenommen. Immer wenn das Gerät innerhalb einer *Session* für eine bestimmte Zeitspanne nicht benutzt wurde, wird eine neue *Subsession* erstellt. Sie kennzeichnen somit die Abschnitte der kontinuierlichen Nutzung des Gerätes innerhalb eines Tages. Dieses Zeitintervall ist diskutabel und kann daher bei Aufruf des Skripts als Parameter vom Nutzer festgelegt werden. Werte zwischen 15 und 60 Minuten sind sinnvoll, 30 Minuten wurde für die im Rahmen dieser Arbeit getätigte Prozessierung der Jahre 2007 und 2008 gewählt.

Diese beiden neuen Felder ermöglichen sowohl in *ArcGIS* als auch in der späteren Analysesoftware die schnelle und einfache Selektion einzelner Besucher bzw. Nutzungssequenzen. Außerdem werden sie als beschränkender Faktor für die Berechnung der im Folgenden erklärten Größen verwendet, um dabei nicht die Werte über mehrere (*Sub-*)*Sessions* hinweg zu verwenden.

V_kmh, Acc_ms2 und Head_deg Durch die vorhandenen GPS-Positionen und Uhrzeiten können problemlos die Geschwindigkeit, Beschleunigung und Orientierung des Nutzers abgeleitet werden. Obwohl diese Kenngrößen bedingt durch die teilweise hohen Ausreißer in den GPS-Positionen nur eine begrenzte Aussagekraft besitzen, erweitern sie die Daten mit nützlichen Informationen über die Nutzerbewegung. Durch die unten beschriebene Glättung können trotzdem aussagekräftige Werte erhalten werden. Die Gleichungen 1 bis 3 zeigen die auf einfachen Grundformeln der Mechanik basierende Berechnung:

$$V_kmh = \frac{\Delta s}{\Delta t} = 3,6 \cdot \frac{\sqrt{(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2}}{t_i - t_{i-1}} \quad (1)$$

$$Acc_ms2 = \frac{\Delta v}{\Delta t} = \frac{v_i - v_{i-1}}{t_i - t_{i-1}} \quad (2)$$

$$Head_deg = 90^\circ - atan2(X_i - X_{i-1}, Y_i - Y_{i-1})_{[^\circ]} \quad (3)$$

Die Einträge innerhalb der *Sessions* sind chronologisch geordnet. Daher können für die Ermittlung des Wertes einfach die Positions- und Zeitwerte der jeweils aktuellen Tabellenzeile i sowie der unmittelbar vorhergehenden Zeile $i-1$ verwendet werden, sofern beide Teil derselben *Session* und *Subsession* sind.

TrimAvg Diese Spalte glättet die Geschwindigkeitswerte, um den Einfluss der Ausreißer bestmöglich zu vermindern und einen realistischen Wert der aktuellen mittleren Geschwindigkeit zu liefern.

Der erste Berechnungsansatz besteht in der Bestimmung eines gleitenden arithmetischen Mittels über fünf Zeilen. Diese Methode ist jedoch sehr empfindlich gegenüber den teils großen Ausreißern. Daher wird der gleitende Mittelwert zusätzlich getrimmt, um den Einfluss der Ausreißer zu verringern. Das Vorgehen besteht aus den folgenden Schritten:

1. Berechnung des gleitenden arithmetischen Mittels aus fünf Geschwindigkeitswerten: denen aus den vorhergehenden zwei, der aktuellen und den nachfolgenden zwei Zeilen. Falls nicht alle Werte Teil derselben *Session* und *Subsession* sind, verringert sich die Anzahl der verwendeten Werte entsprechend. Mindestens vier Werte sind allerdings für sinnvolle Ergebnisse nötig, ansonsten wird aufgrund der zu niedrigen Grundgesamtheit nicht getrimmt und das normale arithmetische Mittel verwendet.

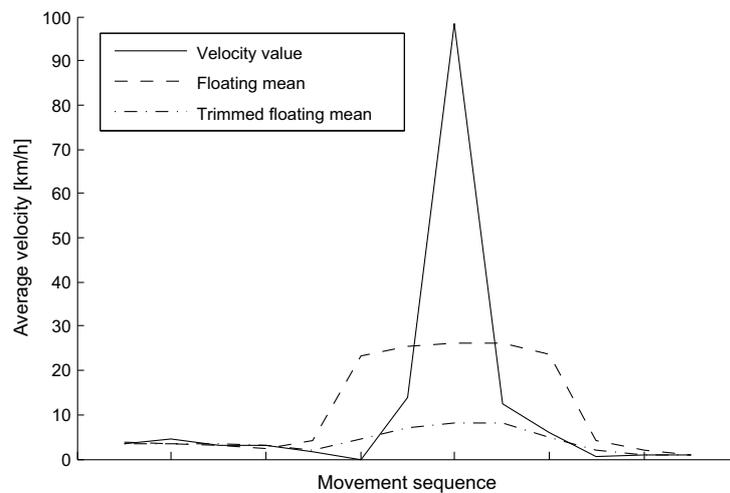


Abbildung 15: Ausreißerglättung durch getrimmtes, gleitendes Mittel. Während das ungetrimmte Mittel über einen größeren Bereich stark vom Ausreißer beeinflusst ist, zeigt dieser bei der getrimmten Variante nur noch eine sehr kleine Auswirkung.

2. Entfernung des Geschwindigkeitswertes mit der höchsten absoluten Differenz zum Mittelwert.
3. Berechnung des arithmetischen Mittels aus den verbliebenen Werten.

Dies führt im Vergleich mit anderen ähnlichen Ansätzen zu den besten Ergebnissen. Abbildung 15 zeigt exemplarisch die Geschwindigkeitswerte einer Teilsequenz, die von einem starken Ausreißer geprägt ist. Der Glättungseffekt des getrimmten Mittels ist deutlich besser als der des normalen, gleitenden Mittels. Dieses Verfahren ermöglicht eine realistische Aussage über die durchschnittlichen Geschwindigkeiten der Wanderer.

Nachdem die Tabelle der *Feature Class* durch die beschriebenen Größen erweitert wurde, werden die Daten einem neuen *ArcMap*-Projekt hinzugefügt und die dreidimensionalen Geometrien der Wanderpfade importiert. Diese müssen, wie in Kapitel 4.1.3 beschrieben, im speziellen *Route*-Format vorliegen und sowohl X/Y/Z-Koordinaten als auch *measure values* besitzen. Solche Pfadgeometrien sind a-priori nicht in der vom Nationalpark betriebenen GIS-Datenbank vorhanden und müssen daher selbst erstellt werden. Dieser Prozess ist nicht Teil der Arbeit sondern wird wiederum vom Betreuer Raper durchgeführt. Einige der in Abbildung 3 dargestellten 21 Wanderpfade verlaufen außerhalb des Parkgebietes. Da dort kein hochauflösendes digitales Geländemodell vorliegt, ist es bei diesen Routen problematisch, die Z-Koordinate präzise zu digitalisieren, sodass der Prozess insgesamt viel Zeit in Anspruch nimmt. Letztendlich wurden nicht alle 21 Pfade berücksichtigt, da einige zum einen sehr kurz und vernachlässigbar sind und zum anderen teilweise nur wenige bis gar keine Besucheraktionen aufweisen. Tabelle 4 listet die Pfade auf, welche Bestandteil der durchgeführten Verarbeitung und Analyse sind (vgl. SNP, 2010):

Nummer	Name	Länge [m]
01high	Alp Trupchun	6192,43
01low	Alp Trupchun	5664,66
02	Fuorcla Trupchun	3408,45
03	Fuorcla Val Sassa	10527,83
06	Murtaröl	7988,99
07	Chamanna Cluozza	6830,40
08	Murtersattel	6590,39
09	Margun Grimmels	2926,44
10	Spöltal	6463,88
12	P1 – Grimmels	3877,22
13	Ova Spin – Il Fuorn	4896,29
15	Munt la Schera	11944,99
17	Margunet (Naturlehrpfad)	6245,15
18	Fuorcla Val dal Botsch	10378,37

Tabelle 4: Liste aller 13 Wanderrouten und ihrer Länge.

Die Geometrie der Pfade entspricht zum Teil nicht der offiziellen Definition und wurde angepasst, um Überlappungen von Pfaden zu vermeiden. Dies würde dem Grundprinzip, jeden Eintrag auf *genau eine* Route zu beziehen, widersprechen.

5.3.2 Herausfiltern hoher Geschwindigkeiten

Bevor mit der Projektion der Punkte das Kernstück der Vorverarbeitung gestartet werden kann, wird ein weiterer Optimierungsschritt eingeschoben, der sich die berechneten Durchschnittsgeschwindigkeiten zu Nutze macht. Wie bereits oben erwähnt, gibt es eine Reihe von Einträgen, die im Auto oder Zug gemacht wurden. Diese befinden sich meist nicht in der Nähe einer Wanderroute und werden daher aufgrund der hohen Entfernung ohnehin gar nicht erst projiziert. Dennoch kommt es an manchen Stellen vor, dass sich Einträge von Wanderern mit solchen von Auto- oder Zugfahrern überlappen. Letztere sind in den Daten unerwünscht und sollen herausgefiltert werden.

ArcGIS bietet die Möglichkeit, Teile der *Feature Class* basierend auf bestimmten Bedingungen, welche durch SQL-ähnliche Syntax festgelegt werden können, zu selektieren und zu löschen. Die visuelle Überprüfung verschiedener Parameter hat gezeigt, dass das Beibehalten aller Einträge mit einer Durchschnittsgeschwindigkeit (Spalte *TrimAvg*) von unter 15 km/h die Auto- und Zugeinträge relativ klar herausfiltert, da diese offensichtlich eine höhere Geschwindigkeit besitzen. Ein kleinerer Wert, der näher an der typischen Schrittgeschwindigkeit von 4–7 km/h liegt und vermeintlich bessere Ergebnisse liefern sollte, erwies sich als *zu* gering. Es würden dann zu viele Einträge herausgefiltert, die eindeutig in Sequenzen von Wanderern liegen und somit bewahrt werden sollen. Die gleitende Durchschnittsgeschwindigkeit solcher Sequenzen wird oft von sprunghaften Ausreißern in der GPS-Position etwas in die Höhe gezogen (vgl. Abbildung 15). Ein Ausschnitt der Filterungsergebnisse ist in Abbildung 16 dargestellt. Es ist deutlich zu sehen, dass die Abfragen entlang von Straßen und Zugstrecken in und um den Park herum zu einem Großteil eliminiert werden, während nur wenige falsche Filterungen auf den Wanderrouten

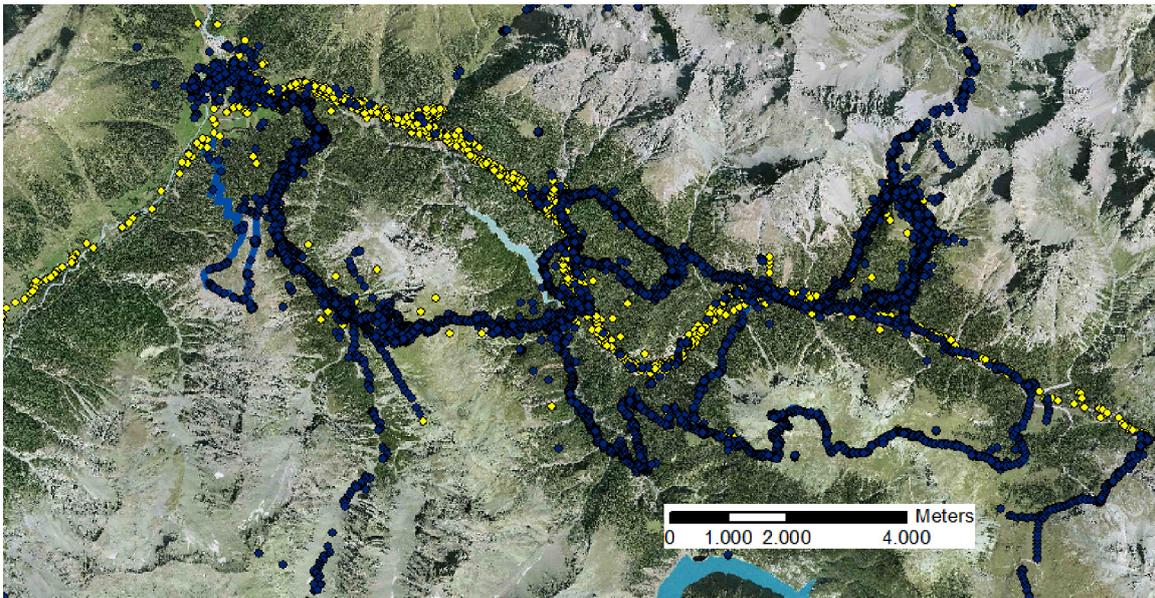


Abbildung 16: Herausfiltern von Einträgen mit einer Durchschnittsgeschwindigkeit über 15 km/h. Die eliminierten Einträge (gelb) befinden sich deutlich entlang der Straßen bzw. Zugstrecken, die beibehaltenen Abfragen (blau) entlang der Wanderpfade. Es treten nur wenige falsche Zuordnungen bedingt durch Ausreißer auf.

erfolgen.

5.3.3 Projektion mittels linearer Bezugssysteme

Alle Einträge des neu erzeugten Layers sollen nun auf den Wanderpfad projiziert werden, auf dem sie auch ausgeführt wurden. Diese Beschreibung der Punkte bezogen auf die linearen Geometrien der Pfade ist prädestiniert für den Einsatz von linearen Bezugssystemen (vgl. Kapitel 4.1.3). Die *Linear Referencing Toolbox* in *ArcGIS* beinhaltet das Werkzeug *Locate Features Along Routes*, welches eine solche Projektion ausführt und für jede Route separat durchlaufen werden muss. Dafür empfiehlt sich die Nutzung der Batch-Funktion.

Neben dem Layer mit den Punkten und den Wanderwegen benötigt das Tool einige andere Parameter, von denen besonders einer von essentieller Bedeutung ist: Die maximal tolerierte Entfernung der Punkte von der Route. Es muss die Distanz angegeben werden, bis zu welcher ein Punkt noch projiziert werden soll. Dies erweist sich als nichttrivialer Schritt, da die Streuungen je nach Route und Ort im Park verschieden stark ausfallen. Die Lösung dieses Problems ist im Folgenden dargelegt.

Es soll ein Wert für jede Route gefunden werden, der nicht bloß durch „genaues Hinschauen“ definiert wird. Vielmehr soll dieser der Charakteristik der Punktstreuungen für jede Route einzeln Rechnung tragen und überdies automatisch ermittelt werden. Abbildung 13 zeigte bereits den beispielhaften Ausschnitt eines Routenteils, welcher demonstriert, wie unterschiedlich die Punkte um die Linie streuen können. Die Analyse von Histogrammen der Punktentfernung zu den Routen hat ergeben, dass offensichtlich annähernd eine Normalverteilung vorliegt, das heißt

$\bar{d} \sim N(\mu, \sigma^2)$. Das bedeutet, dass die meisten Punkte sich nahe des Weges befinden und mit größer werdender Distanz die Anzahl abnimmt.

Der Lösungsansatz besteht darin, nur diejenigen Punkte zu projizieren, deren Distanz zur Route in dem 95%-Konfidenzintervall liegt, welches durch den Mittelwert und die Standardabweichung der Entfernungen zur Route sowie der Annahme einer Normalverteilung definiert ist. Alle übrigen Punkte werden dann mit der typischen Irrtumswahrscheinlichkeit von $\alpha = 5\%$ als zu weit weg eingeschätzt und können nicht mehr als zum Pfad zugehörig eingestuft werden. Dieses Vorgehen bringt den gewünschten Effekt, dass der Verteilungscharakteristik jeder Route im Einzelnen Rechnung getragen wird und kein global einheitlicher Schwellwert verwendet wird.

Zunächst werden mit dem arithmetischen Mittel \bar{d} und der Standardabweichung s_d aus allen Entfernungen die für das Konfidenzintervall benötigten Parameter berechnet:

$$\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i \quad (4)$$

$$s_d = \sigma_d = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (d_i - \bar{d})^2} \quad (5)$$

Mit den folgenden Gleichungen gelingt schließlich die Berechnung des 95%-Konfidenzintervalls K_{95} (vgl. Niemeier, 2008, 92-95):

$$K_{95} = [a; b] = [\bar{d} - \sigma_d \cdot y_{1-\alpha/2}; \bar{d} + \sigma_d \cdot y_{1-\alpha/2}], \quad (6)$$

sodass

$$P\{a \leq \mu_d \leq b\} = 1 - \alpha. \quad (7)$$

Einzigste noch unbekannte Größe ist das Quantil $y_{1-\alpha/2}$ der zugehörigen symmetrischen Wahrscheinlichkeitsverteilung. Aufgrund der oben erwähnten Beobachtungen wird dieses aus der Normalverteilung berechnet: $y_{1-\alpha/2} = y_{0,975} = 1,96$. Statistisch streng formuliert ist dann jeder der später getesteten Entfernungswerte der oben auftretende „Mittelwert“ μ_d , dessen Zugehörigkeit zum Konfidenzintervall mit der Irrtumswahrscheinlichkeit von 5% getestet wird.

Praktisch wird dieses Vorgehen iterativ in zwei Schritten gelöst. Die Konfidenzintervalle für jede Route können nicht direkt ermittelt werden, da die Distanzen der Punkte zu den Routen und somit die nötigen Berechnungsparameter a-priori nicht vorliegen. Außerdem ist nicht klar, welche Punktmenge im Einzugsbereich eines Pfades überhaupt zur Berechnung herangezogen werden soll. Daher wird im *Locate Features Along Routes*-Tool für jede Route zunächst ein Anfangswert von 150 Meter für die maximale Entfernung gewählt, der die Punktmenge auf recht großzügige Art und Weise einschränkt. Dieser Wert ist nicht willkürlich entstanden, sondern zum einen durch visuelle Inspektion, zum anderen durch die Betrachtung des Histogrammes der

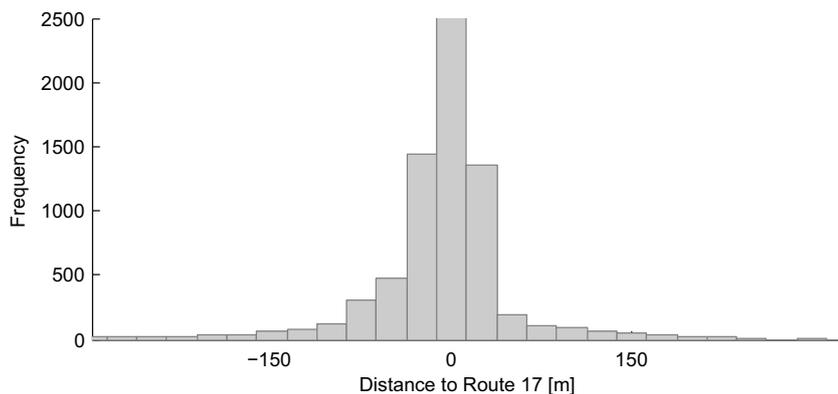


Abbildung 17: Histogramm der Punktentfernungen zu Route 17. Alle Punkte innerhalb von 150 m werden im ersten Iterationsschritt zur Berechnung des Konfidenzintervalls verwendet.

Entfernungen aller Punkte zu jeweils einer Route. Ein solches ist beispielhaft für Route 17 in Abbildung 17 dargestellt. Punkte mit einer noch größeren Entfernung können nicht mehr mit ausreichender Sicherheit der Route zugeordnet werden.

Nach der Batch-Prozessierung liegen die *Event*-Tabellen für jede Route vor. Diese enthalten die ID jedes projizierten Punktes, seine Position entlang des jeweiligen Pfades sowie die für die Konfidenzintervallberechnung benötigte Distanz senkrecht zur Route innerhalb von maximal 150 Meter. Im zweiten Schritt wird nun auf jede der Tabellen das Skript *RejectOutliers.py* angewendet. In diesem wird mit den oben aufgeführten Formeln zunächst das Konfidenzintervall berechnet. Anschließend werden alle Zeilen der *Event*-Tabelle entfernt, deren Distanzwert außerhalb dieses Intervalls liegt. Die entsprechenden Einträge in der ursprünglichen *Feature Class* werden allerdings nicht gelöscht. Sie können später noch für allgemeine Nutzungsanalysen, die nicht auf die einzelnen Pfade bezogen sind, genutzt werden. Durch den fehlenden Eintrag in der *Event*-Tabelle werden Zeilen nicht projizierter Punkte bei der Zusammenführung der Tabellen lediglich keine zugeordnete Route erhalten.

5.3.4 Lösen mehrdeutiger Projektionen

Nach diesem Vorgehen existiert für jede Route eine entsprechende *Event*-Tabelle, die die Position entlang des Pfades (*measure value*) für jeden der projizierten Punkte enthält, welche sich innerhalb des 95%-Konfidenzintervalls befinden. Da die Projektion für jede Route nacheinander ausgeführt wurde, gibt es nicht wenige Einträge, die in mehreren dieser Tabellen auftauchen, da sie im „Einzugsbereich“ mehr als einer Route liegen. Dies ist, wie beim *Map Matching* typisch, vor allem im Bereich von Kreuzungen der Fall, wie Abbildung 18 zeigt. Da jeder Punkt entweder gar nicht oder auf *genau eine* Route projiziert werden muss, sind auftretende Mehrdeutigkeiten zu lösen.

Um die Punkte mit solchen Mehrdeutigkeiten zunächst auffindig zu machen, werden alle *Event*-Tabellen aus dem vorhergehenden Schritt in einer einzigen Tabelle zusammengefasst. Dies gelingt

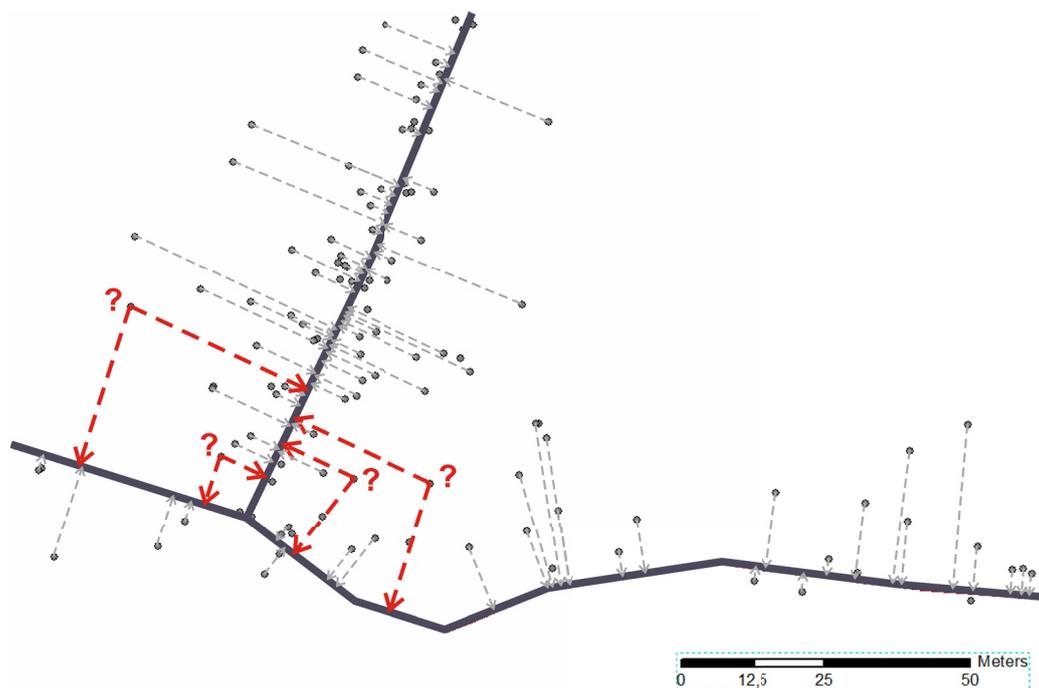


Abbildung 18: Mehrdeutigkeiten im Kreuzungsbereich zweier Routen. Für einige Einträge ist die korrekte Zugehörigkeit unklar.

mit dem *Append*-Werkzeug aus der *Toolbox Data Management/General*. Auf die Tabelle wird dann das nächste Skript, *SolveAmbiguities.py* angewendet, welches Punkte mit mehr als einer Projektion mittels der identischen Objekt-ID findet. Von diesen behält es gemäß eines einfachen geometrischen *Map Matching*-Ansatzes nur den Eintrag bei, der den kürzesten Distanzwert zur Route aufweist. Alle anderen werden aus der Tabelle gelöscht und somit die Projektion verworfen. Anstatt dieses einfachen, geometrischen Ansatzes sind auch kompliziertere Methoden denkbar, die in Kapitel 8.1 diskutiert werden.

5.3.5 Tabellenverknüpfung und Shapefile-Export

Nach dem vorhergehenden Schritt liegt nun mit der bearbeiteten *Event*-Tabelle die Information über alle Punktprojektionen vor. Zu jedem Punkt steht die entsprechende Route-ID, der *measure value* sowie die Entfernung zum Pfad zur Verfügung. Diese Tabelle wird nun mit der ursprünglichen, bereinigten *Feature Class* verknüpft. Das geschieht durch einen *Join* basierend auf den in beiden Tabellen vorhandenen IDs der Punkte. Die *ObjectID* der ursprünglichen Punkte ist in der *Event*-Tabelle in der mit *InputOID* bezeichneten Spalte vorhanden. Die Verknüpfung führt dazu, dass die Datentabelle durch die Spalten der *Event*-Tabelle erweitert wird. Diese werden mit den korrespondierenden Werten gefüllt, sofern für die jeweilige Punkt-ID ein zugehöriger Eintrag in der *Event*-Tabelle vorhanden ist, sprich der Punkt auf einen Pfad projiziert wurde. Es kommen somit die folgenden Spalten zur *Feature Class* hinzu:

- **RID:** Die Nummer der Wanderoute, auf die der Eintrag projiziert wurde bzw. leer, falls

der Punkt auf keine Route bezogen werden konnte.

- **Meas:** Die Position des Punktes entlang der Route (*measure value*).
- **Distance:** Der Abstand des Originalpunktes zur Route.
- **InputOID:** Die ID des Punktes aus der Event-Tabelle. Identisch zum Feld *ObjectID*.

Somit wurde der zu Beginn dieses Kapitels beschriebene Übergang von der absoluten Punktposition hin zur Beschreibung der Einträge entlang der Routen umgesetzt. Dieses neue Bezugssystem wird von der Analysesoftware an vielen Stellen genutzt. Der letzte Schritt innerhalb von *ArcGIS* besteht darin, die Tabelle als *Shapefile* zu exportieren, um sie später auf einfache Art und Weise in die Datenbank importieren zu können. Dies kann durch Rechtsklick auf den Layer und den entsprechenden Eintrag im Kontextmenü einfach vollzogen werden.

5.4 Erstellen der PostGIS-Datenbank

5.4.1 Begründung

Nach der umfangreichen Verarbeitung in *ArcGIS* liegen die vorverarbeiteten und durch diverse Informationen angereicherten Daten jeweils eines Jahres in einem *Shapefile* vor. Für die Analyse der Daten wird im Verlauf der Arbeit eine Software entwickelt, der diese Daten auf flexible Art und Weise zur Verfügung gestellt werden sollen. Aufgrund der großen Datenmengen und anderer Vorteile (vgl. Kapitel 4.2.1) ist die Verwendung einer Datenbank prinzipiell der Speicherung im Dateisystem vorzuziehen. Vor dem oben geschilderten *Shapefile*-Export liegen die Daten im *File Geodatabase*-Format von *ArcGIS* vor. Der Aufbau einer Verbindung zwischen dem *Java*-Programm und einer solchen proprietären Geodatenbank ist prinzipiell möglich, aber sehr umständlich und langsam. Es gibt einige Gründe, warum die Daten an dieser Stelle von *ArcGIS* in *PostgreSQL/PostGIS* und damit in eine *OpenSource*-Geodatenbank konvertiert werden (vgl. Brinkhoff, 2008, 5):

- Mit *PostgreSQL* wird ein „herkömmliches“ Datenbankmanagementsystem genutzt, welches eine einfache Verbindung mit der *Java*-Software erlaubt und zudem als schneller gilt als die *ArcGIS*-Geodatenbank. Außerdem ist es durch seine Herkunft aus der Datenbankbranche näher am aktuellen Datenbankstandard als die von ESRI unabhängig entwickelten Geodatenbanken aus der GIS-Branche, die hinsichtlich Geschwindigkeit und Funktionsumfang Defizite haben.
- Der früher bestehende Nachteil fehlender Geodatenverarbeitung in regulären Datenbanksystemen ist mittlerweile durch diverse Erweiterungen wie *PostGIS*, *SQL/MM Spatial* oder *Oracle Spatial* behoben. Diese Systeme bieten eine Vielzahl an Möglichkeiten zur Geoprocessing, die mindestens ebenso viele Funktionen besitzen wie eine umfangreiche GIS-Software.

- *PostGIS* genügt den OGC-Standards und steht für die Entwicklung hin zu offenen, interoperablen GIS-Lösungen, die problemlos mit anderen Anwendungsprogrammen verknüpft werden können.
- Die weitere Verarbeitung und Analyse der Daten ist nicht mehr von einem Systemanbieter bzw. einer GIS-Software abhängig, sondern kann durch die Speicherung in einer herkömmlichen Datenbank flexibler eingesetzt werden.

Das Erstellen der Datenbank beginnt mit dem Import sowohl der in *ArcGIS* vorbereiteten Daten als auch der einzelnen Pfadgeometrien. Anschließend werden zusätzliche Tabellen erstellt, die für die Analyse benötigt werden. Dort fließen weitere Vorverarbeitungsschritte ein, unter anderem das zentrale Konzept der *Regions of Interest (RoI)*.

5.4.2 Import der ArcGIS-Geodaten

Bevor die *ArcGIS*-Daten einer *PostGIS*-Datenbank hinzugefügt werden können, muss zunächst eine solche erstellt werden. Mit der in Kapitel 4.2.3 erwähnten Benutzeroberfläche *pgAdmin* ist dies auf einfache Art und Weise möglich. Es muss lediglich darauf geachtet werden, dass die *PostGIS*-Vorlage verwendet wird. Nur dann wird die entsprechende Erweiterung eingebunden und alle räumlichen Funktionen sind in der Datenbank nutzbar.

PostGIS enthält das Tool *shp2pgsql*, welches *Shapefiles* in eine *PostgreSQL*-Datenbank importieren kann (vgl. *PostGIS*, 2010, 32-33). Es ist in *pgAdmin* als Plugin verfügbar. Dies ist der Grund, warum der obige Export in ein *Shapefile* vorgenommen wurde. Solch ein Werkzeug existiert nicht für *File Geodatabases*. Dieses Tool wandelt die zugrunde liegende Datentabelle des *Shapefiles* in eine neue Datenbankrelation um. Außerdem wird die darin intern abgespeicherte Geometrie der Features (Punkte, Linien oder Flächen) in einen *PostGIS*- und damit OGC-konformen Datentyp umgewandelt und in der Spalte *the_geom* zur Verfügung gestellt. Dort liegt die Geometrie im Binärformat vor, kann aber durch entsprechende Funktionen auch als Klartext angezeigt werden. Mit dieser Geometrie kann nun in *PostGIS* gearbeitet und durch Anwendung diverser Funktionen beispielsweise einzelne Punkte aus einer Linie extrahiert oder kompliziertere Geoprozessierungen vorgenommen werden.

Die Extraktion der internen Geometrie aus dem *Shapefile* ist jedoch offensichtlich bei der Datentabelle bedeutungslos, da schließlich die ursprünglichen Koordinaten zu Gunsten des linearen Bezugssystems aufgegeben wurden.

Der wichtigste Teil der Daten liegt somit als Datenbanktabelle vor. Die Tabellen jeweils eines Jahres heißen **wp**[Jahr]. Diese Namensgebung muss auch bei neueren Jahren eingehalten werden, damit SNP Analytics sie lesen kann.

Neben den Besucherdaten werden ebenfalls einmalig alle vorhandenen Pfadgeometrien in die Relation **routes** importiert, da diese in der Software ebenfalls benötigt werden. Hierbei ist die Extraktion der internen Geometrie im Gegensatz zu den obigen Besucherabfragen essentiell, um an die dreidimensionalen Koordinaten der Route zu gelangen.

5.4.3 Weitere Relationen und das Konzept der Regions of Interest

Schließlich werden noch einige andere Tabellen der Datenbank hinzugefügt, die weitere nützliche Informationen enthalten. Dies geschieht manuell, da keine entsprechenden *Shapefiles* vorliegen.

In Kapitel 5.2 wurde die Konvertierung der Rohdaten und die damit einhergehende Erstellung der *Action Codes* erläutert. Die Bedeutung dieser, das heißt die dahinter stehende Programmfunktion, wird in den Tabellen **actioncodes**[Jahr] abgelegt. In der Analysesoftware muss sich der Nutzer somit nicht oder nur wenig mit den ursprünglichen Codes auseinandersetzen. Die Tabellen werden nicht einmalig, sondern für die Daten jedes Jahres erstellt. Sie unterscheiden sich je nach Nutzungsjahr, da die *WebPark*-Software auf den Geräten laufend weiterentwickelt und demzufolge ebenfalls das Konvertierungsskript in VBA angepasst wird. Im Falle neu hinzukommender Programmfunktionen oder Änderungen in der Struktur sind auch die *Action Codes* betroffen.

Ein weiterer Abstrahierungsschritt stellt die Tabelle **actiongroups** dar. Oft sind nicht alle *Action Codes* im einzelnen von Interesse. Snozzi (2009, 38) führte aus diesem Grund eine Gruppierung ein, die ähnliche Programmfunktionen zu den folgenden Themen zusammenfasst:

- **General:** Allgemeine Funktionen wie das Aufrufen der Hauptseite oder einer Übersichtsseite über die beliebtesten Funktionen des Wanderführers.
- **Info Around:** Jegliche Informationen darüber, was sich in der Umgebung des Besuchers befindet, beispielsweise die in Kapitel 2.3.2 erwähnten FOIs.
- **Info on Route:** Karten, Profile sowie weitere Hintergrundinformationen über die Wanderrouten.
- **Orientation:** Funktionen zur Navigation, z. B. die Anzeige einer Karte mit der eigenen Position.
- **Content:** Informationsseiten über verschiedene Dinge wie beispielsweise Tierarten, Pflanzen, POIs und FOIs.
- **Special Content:** Zusatzfunktionen des Wanderführers: Textsuche, Lesezeichen (*Geo-Bookmarks*), Kamera.
- **Info on Device:** Geräteinformationen, z. B. Warnung über niedrigen Batteriestand oder der Aufruf der Hilfe.

Die zu jeder Gruppe gehörenden Codes sind für die Jahre 2007 und 2008 im Anhang aufgeführt. In den *actioncodes*[Jahr]-Tabellen ist die zugehörige Gruppe für alle Codes in einer zusätzlichen Spalte angegeben.

Die letzte Relation, die zur Vervollständigung der SNP-Datenbank erstellt wird, heißt **roi**. Sie beinhaltet alle so genannten *Regions of Interest (RoI)* im Parkgebiet. Das sind Orte im Park, an denen die Besucher ihren Wanderführer typischerweise häufig benutzen, sich länger aufhalten

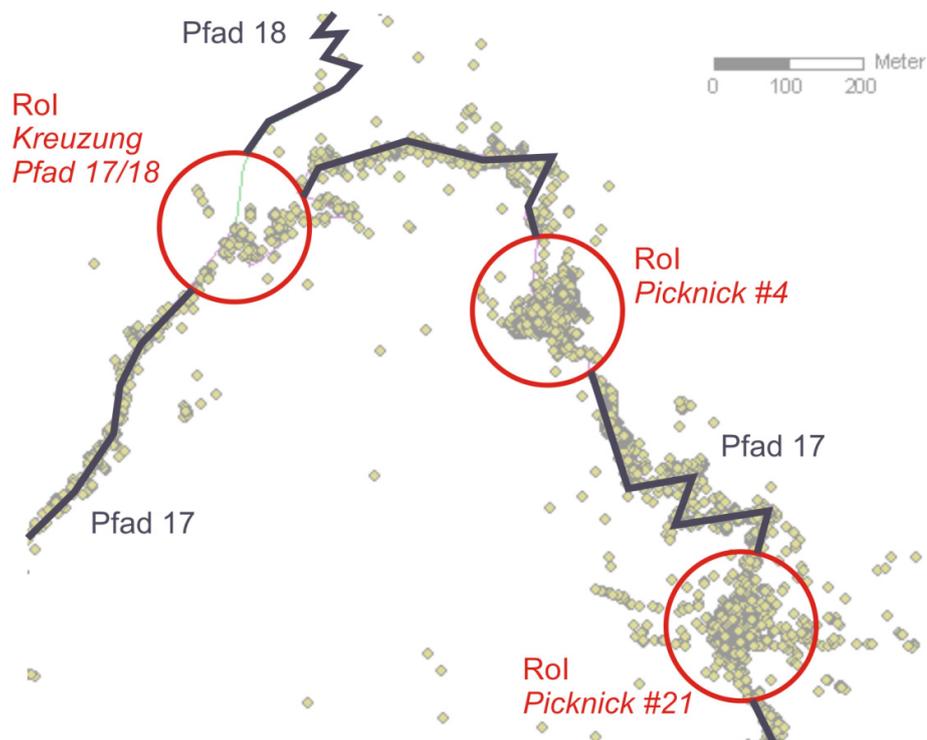


Abbildung 19: Das Konzept der *Regions of Interest*. Interessante Orte im Park wie Picknickplätze oder Kreuzungen können in der Analyse separat betrachtet werden.

oder die eine Veränderung der Bewegungsrichtung zufolge haben können. Es gibt insgesamt 35 solcher Regionen, ein Großteil davon sind feste Picknickplätze, teilweise aber auch Hütten/Hotels, alle Kreuzungen von Wanderpfaden sowie Start- und Endpunkte (vgl. Tabelle und Karte im Anhang). In all diesen Bereichen ist die genaue Position entlang der Route weniger interessant als die Tatsache, dass sich der Besucher generell in dieser groben Region befindet. Darüber hinaus ist, wie bereits oben in Abbildung 18 gezeigt, die Projektion auf eine Route gerade in Kreuzungsbereichen ohnehin sehr unsicher. Dieses Konzept der *Regions of Interest* wird ebenfalls in früheren Arbeiten zur Verarbeitung und Analyse raumzeitlicher Bewegungsdaten verwendet (vgl. Kapitel 3.3.1) und stellt neben der Ausnutzung der festen Pfadgeometrien die Einführung weiteren Hintergrundwissens dar. Abbildung 19 zeigt einen repräsentativen Ausschnitt aus den Originaldaten mit überlagerten Routen- und *RoI*-Informationen.

Das räumliche Ausmaß jeder *RoI* wurde nach visueller Inspektion der Daten auf einen Umkreis von 100 Metern ausgehend von den Kreuzungspunkten bzw. den Mittelpunkten der Picknickplätze festgelegt. Letztere sowie die Start- und Endpunkte der *RoI* sind in der Datenbanktabelle im linearen Bezugssystem der Routen hinterlegt. Die Regionen decken somit in der Regel einen Bereich von 200 Metern entlang einer Route ab, jedoch kann es bei Start- und Endpunkten und einigen anderen Ausnahmefällen vorkommen, dass lediglich in eine Richtung ein Wanderpfad existiert und somit die Länge nur 100 Meter beträgt.

Neben der Position des Punktes entlang der Route wird somit eine weitere räumliche Information für jeden Eintrag der Datentabelle gewonnen: die Zugehörigkeit zu der „interessanten“ Region

des Parks, innerhalb der er sich befindet. Eine entsprechende Spalte mit dieser Information wird der *wp[Jahr]*-Tabelle später durch SNP Analytics automatisch hinzugefügt. In der Analyse spielt das Konzept der *RoI* eine wichtige Rolle und wird unter anderem dazu genutzt, die Positionssequenzen jedes Nutzers durch eine semantisch reichhaltigere Abfolge von hintereinander besuchten *Regions of Interest* auszudrücken. Dies ermöglicht wiederum weitere interessante Analysen, die in Kapitel 7.2 beschrieben werden.

Nach diesem letzten Schritt ist die Prozesskette der Vorverarbeitung vollständig durchlaufen. Die Datenbank mit allen notwendigen Informationen wurde erstellt und die Daten sind nun in einem Format, in dem sie von der Analysesoftware mittels SQL beliebig abgefragt und verarbeitet werden können.

6 Das Programm SNP Analytics

Die nun erfolgte Vorverarbeitung der Daten ist zugeschnitten auf die darauf aufbauende visuell gestützte Analyse mit SNP Analytics, dem Hauptteil dieser Arbeit. Dieses Kapitel beinhaltet die detaillierte Vorstellung der entstandenen Benutzeroberfläche. Begonnen wird mit den Zielen und dem generellen Aufbau des Programmes, bevor die Herstellung der Datenbankverbindung erläutert wird. Anschließend wird auf die einzelnen Komponenten, die zur Analyse der Daten zur Verfügung stehen, eingegangen. Dies sind mit dem Datenfilter zuallererst der wichtigste Bestandteil. Anschließend werden die so genannten *Views* sowie einige weitere Funktionen der Software thematisiert.

6.1 Ziele und Aufbau des Programmes

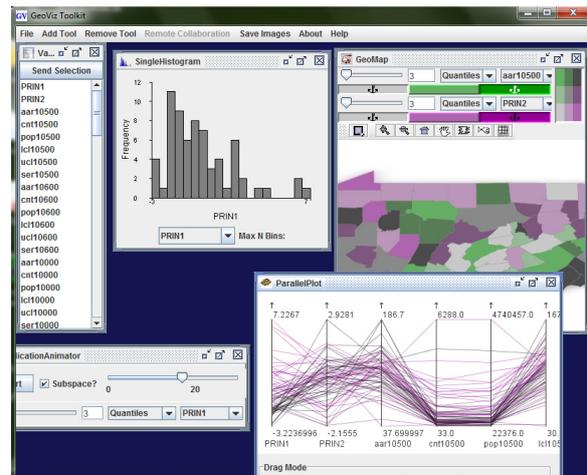
Zu Beginn der Arbeit wurden in Kapitel 1.2 anhand der vorliegenden Daten einige Forschungsfragen aufgelistet, die im Vorfeld bezüglich der Datenanalyse als relevant festgestellt wurden. Dort wurde ebenfalls die Methodik der Analyse erwähnt. Diese soll nicht ausschließlich aus der Abarbeitung der Fragen bestehen, deren Beantwortung auch durch diverse „manuelle“ Datenanalysetechniken versucht werden könnte. Im Einklang mit dem Titel dieser Arbeit und der in Kapitel 3.3.2 beschriebenen *Visual Analytics*-Techniken soll vielmehr ein flexibles, generisches Werkzeug entstehen, welches die visuelle Exploration der umfangreichen Daten durch den Nutzer ermöglicht. Mit diesem können im Anschluss ebenso die gestellten Forschungsfragen untersucht, darüber hinaus aber noch viele weitere interessante Zusammenhänge aus den Daten extrahiert werden. Denn während des Prozesses der visuellen Exploration durch den Nutzer werden sicherlich Muster und Zusammenhänge in den Daten entdeckt, die zu Beginn bei der Formulierung der Fragen noch gänzlich unbekannt waren. Überdies wird das Programm flexibel auf die Datensätze mehrerer Jahre anwendbar sein. Somit bietet es für die Betreiber des Schweizerischen Nationalparks möglicherweise auch in Zukunft ein nützliches Werkzeug zur umfassenden Analyse der *WebPark*-Daten.

Das Ziel von SNP Analytics ist es, dem Nutzer als Grundlage zunächst ein breites Spektrum an einfachen, so genannten *low-level* Statistiken anzubieten. Diese sollen ein Gefühl für die Daten vermitteln und die Nutzung der *WebPark*-Geräte innerhalb eines Jahres übersichtlich zusammenstellen. Darüber hinaus sollen einige kompliziertere, *high-level* Analysen ermöglicht werden, welche spezielle Sachverhalte thematisieren, die sich hauptsächlich mit den Inhalten der in Kapitel 1.2 aufgeführten Fragen beschäftigen. Bei der Bedienung des Programmes sollen dem Nutzer möglichst viele Interaktionsmöglichkeiten mit den Daten gegeben werden. Er soll frei entscheiden können, welche Statistiken, Karten etc. er anzeigen lassen möchte und sich beliebige Konstellationen zusammenstellen. Es wird somit keine feste Analysemethode vorgegeben, der Nutzer kann die gewünschten Bestandteile der Oberfläche vielmehr selbst bestimmen und sie frei positionieren.

Dieses Konzept ist hauptsächlich von zwei existierenden Tools zur visuellen Analyse inspiriert,



(a) Google Analytics (2010)



(b) GeoViz Toolkit (GeoVISTA, 2010)

Abbildung 20: Screenshots der beiden Programme, die Vorbild für SNP Analytics sind.

die einen solchen Ansatz wählen: Zum einen die webbasierte Software *Google Analytics* (2010), die verschiedene Besucherstatistiken einer eigenen Webseite anbietet und zum anderen das bereits in Kapitel 3.3.2 erwähnte *GeoViz Toolkit* (Hardisty, 2005, GeoVISTA, 2010), welches vielfältige Methoden zur Analyse raumzeitlicher Daten bereitstellt. Abbildung 20 zeigt Ausschnitte beider Programme. Die angesprochene freie Positionierung verschiedener statistischer Ansichten (*Views*) wird bei beiden durch ein so genanntes *Dashboard* angeboten. Dies ist eine Art Armaturenbrett, dessen Gestalt vom Nutzer bestimmt werden kann.

SNP Analytics ist in der Programmiersprache *Java* geschrieben. Als Entwicklungsumgebung wurde die frei verfügbare Software *Eclipse IDE*¹³ genutzt. Die selbst erstellten Klassen sind in insgesamt 63 *.java*-Dateien gespeichert. In *Java* ist es üblich, diese Dateien in Paketen (*packages*) zu ordnen. Dies erlaubt eine klare Strukturierung und gute Lesbarkeit. SNP Analytics besteht aus den folgenden 15 Paketen:

- Hauptklassen des Programmes: **org.gicentre.uden.snp.main**
- Nützliche Klassen z. B. zur Fehlerbehandlung, zum Erscheinungsbild oder mit eigenen mathematischen Funktionen: **org.gicentre.uden.snp.convenience**
- Klassen, die die Verbindung zur Datenbank regeln: **org.gicentre.uden.snp.db**
- Klassen für die wichtige Komponente des Datenfilters: **org.gicentre.uden.snp.filter**
- Definition verschiedener generischer Diagramme (z. B. Säulendiagramm, Kreisdiagramm), die in den einzelnen Statistiken verwendet werden: **org.gicentre.uden.snp.charts**
- Klassen für die einzelnen verfügbaren *Views*:
 - **org.gicentre.uden.snp.actionchains**
 - **org.gicentre.uden.snp.codedistribution**

¹³<http://www.eclipse.org>

- `org.gicentre.uden.snp.interactivemap`
- `org.gicentre.uden.snp.notepad`
- `org.gicentre.uden.snp.roi`
- `org.gicentre.uden.snp.route2dmap`
- `org.gicentre.uden.snp.routeprofile`
- `org.gicentre.uden.snp.tagcloud`
- `org.gicentre.uden.snp.totalusage`
- Klassen für die Programmhilfe: `org.gicentre.uden.snp.help`

Für alle Details der in diesen Klassen bereitgestellten Methoden wird auf die aus den Kommentaren im Quellcode automatisch erzeugte Dokumentation *Javadoc* verwiesen.¹⁴

Neben diesen selbst programmierten Klassen und Methoden können natürlich die aus der *Java*-Standardklassenbibliothek verwendet werden. Darüber hinaus werden vier weitere externe Bibliotheken anderer Autoren eingebunden, die in *Java*-eigenen Archiven (*.jar*) komprimiert sind. Diese beinhalten Klassen für folgende zusätzliche Funktionalitäten:

- Nötige JDBC¹⁵-Treiber, die als Schnittstelle zur *PostgreSQL/PostGIS*-Datenbank dienen:
 - `postgresql-8.4-701.jdbc4.jar`
 - `postgis-1.5.0.jar`
- Interaktive Formen mit einfacher Mausinteraktion von Wood (2008): `woodshapes.jar`
- Die frei verfügbare *OpenCloud API* (2010), die zur einfachen Generierung einer *tag cloud* für einen der *Views* verwendet wird: `opencloud.jar`

Beim Starten der Software wird die Hauptklasse *SNPMain* aufgerufen, welche das Hauptfenster der Anwendung (*MainWindow*) öffnet. Abbildung 21 zeigt beispielhaft ein mögliches Aussehen der Oberfläche nach Aufbau der Datenbankverbindung und Hinzufügen einiger *Views*. An dieser Stelle soll zunächst das Layout des Hauptfensters erläutert werden: Wie oben angedeutet, besteht der Hauptteil der Oberfläche aus dem zentralen *Dashboard*, auf dem die einzelnen Ansichten erzeugt und beliebig angeordnet werden können. Daneben besitzt der Filter als einflussreichste Komponente eine Sonderstellung. Er ist als eigenes Bedienungsfeld am linken Rand des Hauptfensters verankert, kann aber bei Inaktivität auch ausgeblendet werden. Die Steuerung des Programmes erfolgt über die Menüleiste am oberen Rand, die aus folgenden vier Menüs besteht:

- **File:** Herstellen oder Trennen der Verbindung zur Datenbank, Speichern und Laden von Projekten, Beenden des Programmes.
- **View:** Ein-/Ausblenden des Filters, Anlegen neuer *Views* auf dem *Dashboard*.
- **Extras:** Automatisches Anordnen der Ansichten, speichern von *Screenshots*.

¹⁴Diese befindet sich im Unterordner *doc* des Programmverzeichnis

¹⁵Java Database Connectivity – <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136101.html>

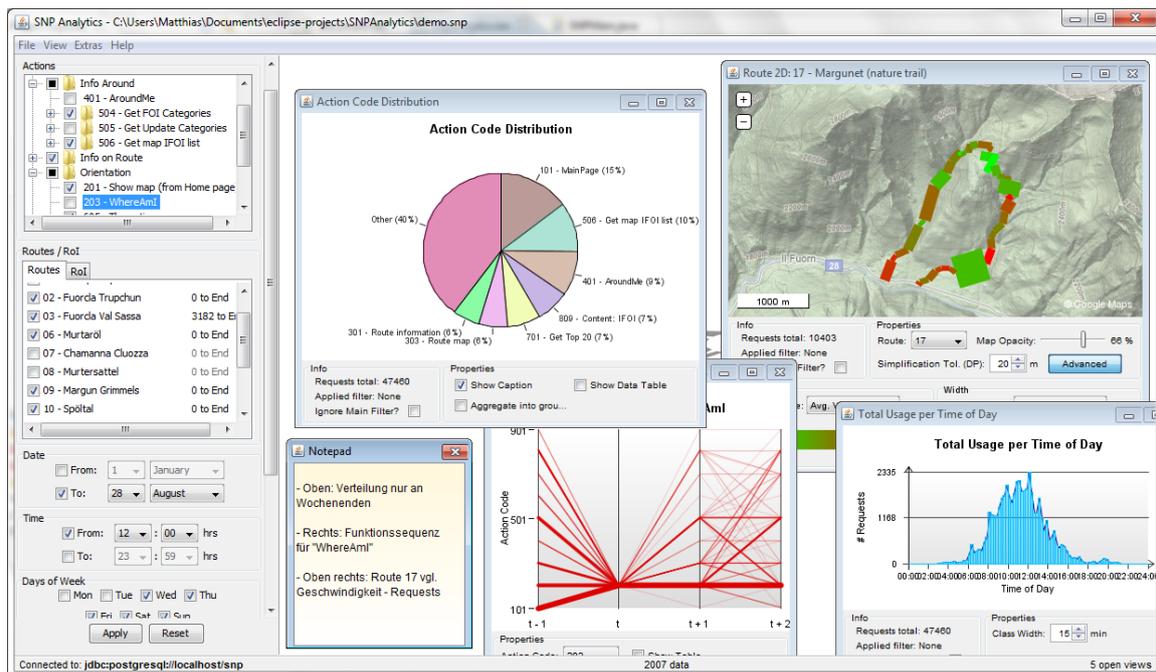


Abbildung 21: Die Oberfläche von SNP Analytics.

- **Help:** Aufrufen der Programmhilfe bzw. Informationen über die Software.

Schließlich rundet die Statusleiste am unteren Rand das Hauptfenster ab. Diese informiert den Nutzer über die Datenbankverbindung sowie die Anzahl zur Zeit geöffneter *Views*.

Bevor die einzelnen Ansichten zur Analyse erzeugt werden können, muss zunächst die Verbindung zur SNP Datenbank hergestellt werden.

6.2 Herstellen der Datenbankverbindung

Um auf die Tabellen in der SNP Datenbank zugreifen zu können, muss eine Verbindung zwischen SNP Analytics als *Client* mit dem Datenbankserver hergestellt werden. Mittels JDBC ist dies recht einfach möglich und wird in der Klasse *DBConnect* durchgeführt. Die Schnittstelle benötigt die folgenden vier Verbindungsparameter: Den **Treiber** für die entsprechende Datenbank, für *PostgreSQL* lautet dieser `org.postgresql.Driver` und ist Bestandteil des oben erwähnten extern hinzugefügten Archives. Die **Datenquelle**, das heißt der Name des Servers und der Datenbank, sowie **Benutzername** und **Passwort**. Sowohl die Namen von Server, Datenbank und Benutzer als auch das Passwort werden bei der Installation von *PostgreSQL/PostGIS* gewählt. Das Aufsetzen der Datenbank selbst wird an dieser Stelle nicht erläutert, kann aber Schritt für Schritt in der Programmhilfe nachvollzogen werden. Der Nutzer wird bei Auswahl des Menüpunktes *File* → *Connect...* zur Eingabe der Parameter aufgefordert.

Während des Verbindungsprozesses muss der Nutzer das Jahr auswählen, aus dem er die Daten analysieren möchte. SNP Analytics erlaubt im Allgemeinen immer nur die Analyse der Daten

eines Jahres, nicht mehrerer Jahre gleichzeitig. Anschließend werden über die eigentliche Verbindung hinaus einige weitere Funktionen ausgeführt, die in der Klasse *Connector* zu finden sind: Alle benötigten Tabellen der Datenbank werden gelesen und unter anderem die Pfadgeometrien, die Anzahl aller Einträge und der Gesamtnutzungszeitraum intern abgespeichert. Die Tabellen werden jedoch nicht komplett in den Speicher gelesen, sondern die benötigten Daten werden später mittels SQL weiterhin direkt von der Datenbank abgefragt. Die Begründung für dieses Vorgehen ist Teil der in Kapitel 8.2 zu findenden Diskussion.

Falls das Programm zum ersten Mal eine Verbindung zu der Datentabelle eines bestimmten Jahres herstellt, dauert der Prozess deutlich länger, da in dem Fall zusätzlich zwei Dinge getan werden müssen: Erstens wird der *wp[Jahr]*-Tabelle eine weitere Spalte **roi** hinzugefügt, in der für jeden Eintrag getestet wird, ob er innerhalb einer der *Regions of Interest* liegt und falls ja, die entsprechende ID eingefügt wird. Und zweitens wird im Anschluss für jede *Session*, das heißt für jeden Besucher, die chronologische Abfolge der passierten *RoI* ermittelt und einmalig für die Beschleunigung zukünftiger Verbindungen zu diesen Daten in einer separaten Datei namens *sessions[Jahr].ses* gespeichert. Diese Beschreibung des Besucherverhaltens in Form passierter *RoI* ist notwendig für einige der unten beschriebenen *Views*.

Der gesamte Verbindungsprozess nimmt somit aufgrund der Datenbankinteraktion und aufwändiger Berechnungen einige Rechenzeit in Anspruch. Wie auch im weiteren Verlauf im Falle längerer Berechnungsprozesse wird dann, wie in Kapitel 4.3.2 erläutert, dafür gesorgt, dass die Oberfläche währenddessen nicht „einfriert“, sondern dem Nutzer signalisiert, dass Ladeprozesse im Hintergrund ablaufen. Dies wird dadurch erreicht, dass die entsprechenden Berechnungen nicht im *Event Dispatch Thread*, sondern in einem separaten Hintergrundthread abgearbeitet werden. Die Oberfläche reagiert weiterhin auf Nutzereingaben und signalisiert ihm den laufenden Ladevorgang, sodass dieser weiß, dass das Programm augenblicklich mit Hintergrundarbeiten beschäftigt ist.

Ist der Verbindungsprozess erfolgreich abgeschlossen, steht eine Instanz der *DBConnect*-Klasse zur Verfügung, deren *performQuery*- bzw. *updateQuery*-Methoden es erlauben, beliebige SQL-Anfragen an die Datenbank zu stellen. Falls diese ein Ergebnis in Form einer Tabelle zurückliefern, wie beispielsweise in der Regel bei einer **SELECT**-Anfrage, wird diese als so genanntes *ResultSet* zurückgegeben und kann im Anschluss durch die Software verarbeitet werden. Kapitel 6.3 enthält einige Beispiele in Zusammenhang mit der Filterung der Daten.

Die Statusleiste informiert den Nutzer nun über die bestehende Datenbankverbindung und die Menüs *View* und *Extras* werden aktiviert, sodass mit der Analyse begonnen werden kann. Die Trennung von der Datenbank ist jederzeit über den Menüpunkt *File* → *Disconnect* möglich.

6.3 Der Filter

Bevor im weiteren Verlauf dieses Kapitels auf die verschiedenen Visualisierungsmethoden (*Views*) der Daten eingegangen wird, soll zunächst die Filterfunktion des Programmes erläutert werden. Der Filter stellt die mächtigste Funktion von SNP Analytics dar. Mit ihm können die zugrunde

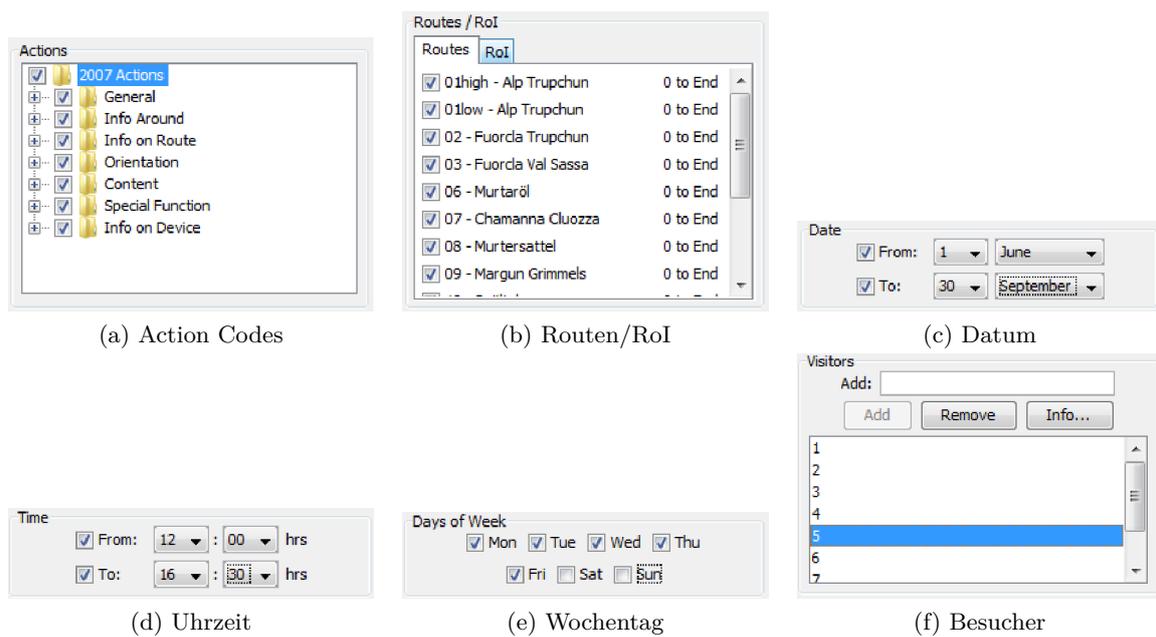


Abbildung 22: Die sechs Komponenten des Filters.

liegenden Daten in umfangreicher Art und Weise beeinflusst werden. Jeder der *Views* visualisiert die Daten in Form einer bestimmten Statistik. Bei Inaktivität des Filters nutzen diese prinzipiell alle verfügbaren Daten aus der Datenbanktabelle. Durch Aktivieren kann der Nutzer in der Seitenleiste am linken Fensterrand bestimmen, welche Daten er herauslassen möchte und welche nicht. Durch diese Selektion, basierend auf den im Folgenden erklärten Bereichen, beeinflusst er somit die Datengrundlage der Ansichten.

Der erste Bereich des Filters (vgl. Abbildung 22a) erlaubt die Auswahl der zu berücksichtigen *Action Codes*, das heißt im Endeffekt der einzelnen Funktionen der *WebPark*-Software. Diese werden aus der Datenbanktabelle *actioncodes*[*Jahr*] gelesen und entsprechend der in der Tabelle *actiongroups* hinterlegten Informationen gruppiert. In der Standardeinstellung werden alle *Codes* einbezogen. Die Hierarchie ist in Form eines Baumes umgesetzt, in dem einzelne Einträge mittels Kontrollkästchen an- oder abwählbar sind. Durch die Gruppierungen können auch bequem ganze Blöcke von *Codes* gewählt werden. *Java* bietet in seiner Standardbibliothek zwar eine Datenstruktur zur Erstellung eines solchen Baumes an, nicht jedoch die erweiterte Variante mit Kontrollkästchen. Daher muss diese selbst konstruiert werden, was mithilfe frei verfügbarer Codebeispiele von Timothy Wall (vgl. Wall, 2006) geschieht.

Eine Besonderheit in diesem Baum stellen die *Action Codes* 301, 302, 303, 504, 505 und 506¹⁶ dar. Bei diesen existiert eine weitere Ebene, in der zusätzliche Attribute dieser Programmfunktion spezifiziert werden können. Die 300er-*Codes* stehen alle für Programmfunktionen zur Abfrage von Informationen über eine der Wanderrouten. Hier können in der weiteren Baumebene die Routen, über die der Besucher Informationen angefordert hat, spezifiziert werden. Bei den 500er-*Codes* hingegen ist es die Information, in welchem Umkreis der Wanderer (*around me*) nach bestimmten

¹⁶Bezogen auf die Benennung im Jahr 2007, im Jahr 2008 sind diese nur noch teilweise vorhanden, siehe Kapitel 8.3.

POIs/FOIs das Programm hat suchen lassen. Diese Detailstufe erlaubt somit für diese beiden Fälle eine noch genauere Auswahl der Daten als nur basierend auf den *Action Codes*. Der nächste Bereich des Filters erlaubt die Einschränkung der zu benutzenden Daten im *räumlichen* Kontext (vgl. Abbildung 22b). Das bedeutet, dass entweder nur solche Einträge, die auf eine der Routen projiziert wurden *oder* sich darüber hinaus zusätzlich innerhalb der Grenzen bestimmter *Regions of Interest* befinden, berücksichtigt werden sollen. Die Spezifikation von beidem gleichzeitig ist nicht möglich, da dies zu inkonsistenten Abfragen führen könnte, beispielsweise wenn eine der ausgewählten *RoI* auf einer Route liegt, die abgewählt ist. Die Standardeinstellung bezieht die Einträge aller Routen ein und sieht keine spezielle Einschränkung bezüglich der *RoI* vor. Solche Einträge der Tabelle, die auf keine Route projiziert werden konnten, werden im Allgemeinen vernachlässigt. Der Grund, warum diese nicht von vornherein ganz aus der Tabelle gelöscht wurden, liegt darin, dass manche der *Views* den Filter ausnahmsweise nicht berücksichtigen und auch nicht projizierte Punkte in ihre Statistik einbeziehen (siehe unten).

Im Falle der Routen bietet der Filter nicht nur die Möglichkeit, diese komplett an- oder abzuschalten, sondern auch, nur einen bestimmten Abschnitt der Route festzulegen. Hier kommt zum ersten Mal der Vorteil der linearen Bezugssysteme mit den abgespeicherten Messwerten (*measure values*) zum Tragen. Durch dieses Konzept können einzelne Abschnitte schnell festgelegt und Einträge aus der Datentabelle ohne Weiteres sehr effizient auf die Zugehörigkeit zu diesen Abschnitten getestet werden. Dafür ist lediglich zu prüfen, ob der Messwert eines Eintrages innerhalb des Intervalls des entsprechenden Abschnittes liegt.

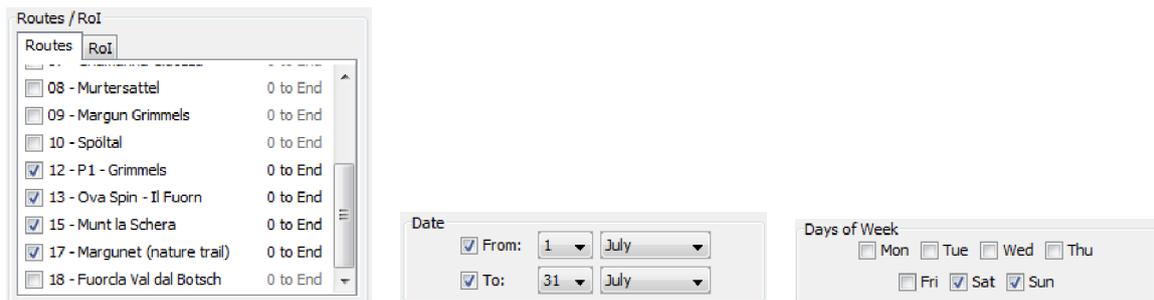
Die nächsten drei Bereiche des Filters (vgl. Abbildungen 22c, d und e) beschäftigen sich mit der zeitlichen Einschränkung der Daten. Es können sowohl der „erlaubte“ Zeitraum im Jahr, die Uhrzeit am Tag sowie der Wochentag des Eintrags spezifiziert werden. Die Standardeinstellung beinhaltet keinerlei zeitliche Einschränkungen.

Der letzte Teil des Filters schließlich bietet die Möglichkeit, nur die Daten einzelner Besucher zu berücksichtigen (vgl. Abbildung 22f). Diese sind den *Sessions* entsprechend mit Nummern versehen, welche der Nutzer im Textfeld eingeben kann, um sie zur Liste „erlaubter“ Wanderer hinzuzufügen. Das Programm wird nur solche Nummern akzeptieren, die auch tatsächlich in den Daten vorhanden sind. Da der Nutzer a-priori in der Regel nicht die Nummern der einzelnen Besucher kennt, erscheint diese Funktion zunächst nicht sehr nützlich. Im Zusammenhang einiger *Views* ermöglicht diese aber interessante Analysen, wie in Kapitel 7.3 gezeigt wird. Die Standardeinstellung sieht keine Einschränkungen hinsichtlich der Benutzer vor. Bei Doppelklick auf einen Besucher in der Liste oder Auswahl der *Info. . .*-Schaltfläche erscheint eine kleine Informationsbox, die mehr Details über diesen enthält. Sie beinhaltet den Tag der Nutzung, die Nummer des ausgeliehenen Gerätes, die Nutzungsdauer sowie die Gesamtanzahl getätigter Aktionen.

Am unteren Rand der Seitenleiste besteht die Möglichkeit, die Filtereinstellungen zurückzusetzen (*Reset*) oder auf die Daten anzuwenden (*Apply*). Letzteres bedeutet, dass die Einstellungen für den Filter übernommen werden und die Datengrundlage aller zu dem Zeitpunkt geöffneter *Views* auf dem Armaturenbrett aktualisiert wird. Dies kann allerdings für einzelne Ansichten verhindert werden, indem sie den Filter ignorieren (siehe Kapitel 6.4). Im Klartext heißt das,

dass die per GUI-Elemente interaktiv vorgenommenen Einstellungen des Nutzers in eine SQL-Anfrage umgewandelt und die Daten aus der Datenbank neu gelesen werden. Die Klasse *Filter* stellt unter anderem Methoden für diese Konvertierung bereit. Zwei einfache Beispiele sollen dies verdeutlichen:

Beispiel 1: Es sollen nur solche Einträge berücksichtigt werden, die auf Route 12, 13, 15 oder 17 gemacht wurden und zwar an allen Wochenenden im Juli:



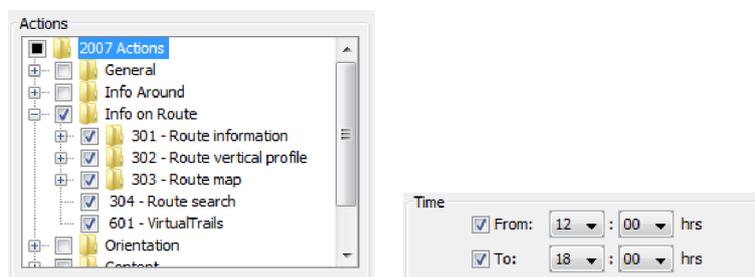
```

1 SELECT * FROM wp2007
2     WHERE rid IN ('12','13','15','17')
3     AND date_ BETWEEN '2007-07-01' AND '2007-07-31'
4     AND EXTRACT(dow FROM Date_) IN (0,6)

```

Als Beispieletabelle wurde hier *wp2007* verwendet. In der SQL-Anfrage werden die Einstellungen des Filters in entsprechende Bedingungen für die einzelnen Spalten umgewandelt. In Zeile (4) werden SQL-interne Funktionen verwendet, um den Wochentag aus dem Datum zu extrahieren. Die Nummerierung beginnt in SQL bei 0 am Sonntag als ersten Tag der Woche.

Beispiel 2: Es sollen nur solche Einträge berücksichtigt werden, bei denen der Besucher Funktionen aus der Gruppe *Info on Route* genutzt hat und dies lediglich zwischen 12 und 18 Uhr:



```

1 SELECT * FROM wp2007
2     WHERE code IN (301,302,303,304,601)
3     AND rid IN ('01high','01low','02','03','06','07','08','09','10','12',
4                '13','15','17','18')
5     AND time BETWEEN '12:00:00' AND '18:00:00'

```

Da die *Code*-Gruppen nicht explizit in der Datentabelle vorhanden sind, verwendet das Programm stets die einzelnen *Action Codes*, wie in Zeile (2) zu erkennen ist. In Zeile (3) wird eine

Einschränkung der Routen vorgenommen, obwohl diese nicht explizit vom Nutzer festgelegt wurde. Dies entspricht aber lediglich der Standardeinstellung, welche nur die Einträge akzeptiert, die auf eine Route projiziert werden konnten. Ohne diese Zeile würden hingegen auch solche ohne jeglichen Routenbezug erfasst.

Der Filter bietet über diese Beispiele hinaus noch vielfältigere Möglichkeiten, die Daten zu beeinflussen. In Kapitel 7 wird dies anhand diverser Beispiele gezeigt.

6.4 Die Views

Die *Views* sind das zentrale Element von SNP Analytics. Sie füllen das *Dashboard* mit Leben. Jede dieser Ansichten wird in einem eigenen Fenster angezeigt, das in Position und Größe frei angepasst werden kann. Diese Fenster werden in *Java* durch die Klasse *JInternalFrame* realisiert. Die eigene Klasse *MyInternalFrame* ist von dieser abgeleitet und legt einige spezielle Optionen fest, die für alle *Views* gelten. Letztere sind dann wiederum Subklassen von *MyInternalFrame*, in denen ihr genauer Inhalt festgelegt wird. Jede Ansicht stellt eine bestimmte Art von Statistik dar, dessen Datengrundlage, wie oben erklärt, vom Filter beeinflusst wird.

Es gibt jedoch einige Ausnahmen, bei denen der Datenfilter keine Auswirkung zeigt. Das ist dann der Fall, wenn eine Einschränkung der Datengrundlage nicht sinnvoll ist. Bei der Vorstellung der einzelnen *Views* weiter unten wird dieser Aspekt gegebenenfalls näher erläutert. Die übrigen Ansichten sind im Allgemeinen vom Filter beeinflussbar, haben aber die Option gemein, dass sie neu zugewiesene Filter aus der Seitenleiste ignorieren können. Ermöglicht wird das durch ein Kontrollkästchen innerhalb des Informationsbereiches unten links einer jeden solchen Ansicht. Das Auswählen dieser Option wird im Programmablauf dazu führen, dass nicht alle *Views* stets denselben Filter als Grundlage besitzen. Daher wird im Infobereich ebenfalls angezeigt, ob aktuell entweder **kein** Filter, der in der Seitenleiste eingestellte **Hauptfilter**, oder aber ein anderer, der Ansicht **eigener** Filter angewendet wird. Durch Klicken auf diesen Text kann der Nutzer darüber hinaus den Filter individuell für den jeweiligen *View* anpassen. Dritter Bestandteil des Infobereiches ist die Anzeige der Gesamtanzahl der unter Berücksichtigung des aktuellen Filters verwendeten Tabelleneinträge, das heißt die Grundgesamtheit der dargestellten Statistik.

Durch den hohen Interaktionsgrad des Programmes treten häufig Situationen auf, in denen der Nutzer durch Mausklicks aus einer Statistik heraus eine neue Ansicht öffnet. Hier ist anzumerken, dass die Filtereinstellung des aufrufenden *Views* auf den neuen „vererbt“ wird. Dies ist sinnvoll, damit die Datengrundlage der neuen Statistik mit der der aufrufenden übereinstimmt und der Nutzer nicht verwirrt wird. Diese Verknüpfung bleibt allerdings *nicht* bestehen, wenn der Filter des aufrufenden *Views* im Anschluss verändert wird. In dem Fall wird der Filter des neuen *Views* nicht entsprechend angepasst. Dies könnte andernfalls zu ungewollten Änderungen mehrerer *Views* führen, die den Nutzer verwirren.

Im weiteren Verlauf dieses Unterkapitels wird der Inhalt jeder einzelnen Ansicht vorgestellt:

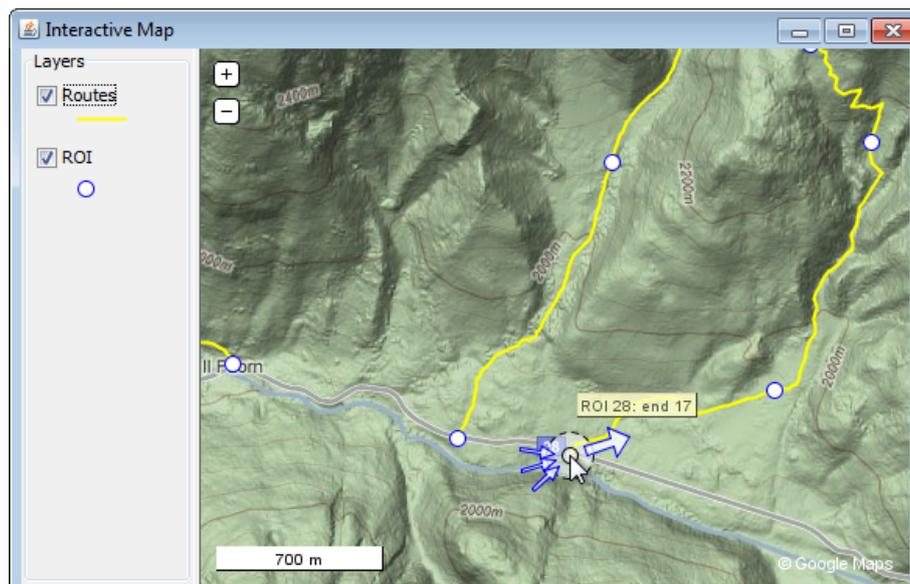


Abbildung 23: Die interaktive Karte beim Bewegen des Mauszeigers über eine *RoI*. Die Pfeile symbolisieren die Hauptbewegungsrichtung der Besucher, die die Region durchqueren. In diesem Beispiel kommen sie von verschiedenen Orten und bewegen sich mehrheitlich zur nordöstlich gelegenen *RoI*.

Interaktive Karte (*View* → *Interactive Map*) Die interaktive Karte stellt gleich zu Beginn unter allen *Views* eine Besonderheit dar: Von ihr ist nur eine Instanz erlaubt, während von den anderen beliebig viele Instanzen dem *Dashboard* hinzugefügt werden dürfen. Der Grund dafür liegt zum einen darin, dass die Karte recht viele Ressourcen benötigt und das System bei mehreren Instanzen unnötig verlangsamen würde. Zum anderen ist der Inhalt der Karte nur wenig anpassbar und mehr als eine Instanz schlicht nicht notwendig.

Die Karte dient in erster Linie als Übersicht des Parks sowie der existierenden Routen und *RoI*. Die einzige Möglichkeit, die Karte anzupassen, besteht in der Seitenleiste am linken Fensterrand, in der diese beiden Ebenen an- und ausgeschaltet werden können. Durch Ziehen des Mauszeigers kann die Karte verschoben, durch Betätigung des Mausrades oder der oben links angezeigten Schaltflächen die Zoomstufe angepasst werden. Dieser *View* ist einer derjenigen, die nicht durch den Filter beeinflusst werden.

Die Koordinaten der Routen liegen, wie auch die ursprünglichen Positionen der Besucheraktionen, in Schweizerischen Landeskoordinaten vor. Da die Hintergrundkarte eine Mercator-Projektion geographischer WGS84-Koordinaten¹⁷ darstellt, muss für die korrekte Georeferenzierung zunächst eine Koordinatentransformation erfolgen. Für die Umrechnung der Schweizerischen Koordinaten $(y, x)_{[m]}$ in das WGS84-System $(\lambda, \varphi)_{[^\circ]}$ werden folgende Formeln angewendet, die eine direkte Näherungslösung mit ausreichender Genauigkeit anbieten (vgl. SwissTopo, 2005):

¹⁷World Geodetic System 1984

$$\begin{aligned} y' &= (y - 600000m)/1000000 \\ x' &= (x - 200000m)/1000000 \end{aligned} \quad (8)$$

$$\begin{aligned} \varphi' &= 16,9023892 + 3,238272x' - 0,270978y'^2 - 0,002528x'^2 - 0,0447y'^2x' - 0,0140x'^3 \\ \lambda' &= 2,6779094 + 4,728982y' + 0,791484y'x' + 0,1306y'x'^2 - 0,0436y'^3 \end{aligned} \quad (9)$$

$$\begin{aligned} \lambda &= \lambda' \cdot \frac{100}{36} \\ \varphi &= \varphi' \cdot \frac{100}{36} \end{aligned} \quad (10)$$

Mithilfe der gegebenen Eckpunktkoordinaten der Kartenkacheln im WGS84-System wird anschließend durch einfache lineare Interpolation die Kartenposition ermittelt werden. Bei der Implementierung hat sich ergeben, dass die Mercator-Projektion aufgrund des kleinen Ausdehnungsbereiches des Gebietes und infolgedessen nur sehr kleinen Abbildungsfehlers vernachlässigt werden kann. Die Methoden für diese Umrechnung werden in der Klasse *SNPMap* bereitgestellt. Für die korrekte Georeferenzierung der *Regions of Interest* ist hingegen keine solche Umrechnung nötig, da sie, wie bereits in Kapitel 5.4 beschrieben, im linearen Bezugssystem der Routen vorliegen. Sind die Routen somit georeferenziert, können auch die Positionen der *RoI* entlang dieser ermittelt werden.

Bei Bewegung des Mauszeigers über eine Route wird diese rot eingefärbt und es erscheint ein *Tooltip* mit Nummer und Name. Durch einen Klick mit der linken Maustaste wird eine neuer *Route2D-View*, durch Rechtsklick das Profil der entsprechenden Route erzeugt (siehe unten). Bewegt der Nutzer die Maus über eine *Region of Interest*, so werden auch hier Nummer und Name angezeigt sowie die *RoI* optisch hervorgehoben. Darüber hinaus erscheinen ein oder mehrere Pfeile, die auf die *RoI* zu bzw. von ihr weg weisen (vgl. Abbildung 23). Diese greifen bereits das Hauptergebnis der *RoI*-Analyse vor, die sich erst bei Mausklick auf den *RoI* im Detail öffnet (siehe unten). Basierend auf den in Kapitel 6.2 beschriebenen Sequenzen nacheinander besuchter *RoI* eines jeden Wanderers, wird dabei das räumliche Verhalten extrahiert. Dies bedeutet konkret, dass bei allen Sequenzen, welche den aktuellen *RoI* enthalten, die Häufigkeiten unmittelbar vorher und nachher passierter Regionen untersucht werden. Wenn eine davon mehr als 50% aller gefundener Vorgänger bzw. Nachfolger ausmacht, wird dies als „Hauptbewegungsrichtung“ gekennzeichnet. In diesem Fall wird ein einzelner Pfeil aus bzw. in Richtung der entsprechenden *RoI* gezeigt. Wenn allerdings keine Mehrheit existiert, bedeutet dies in der Regel, dass viele verschiedene Regionen vorher bzw. nachher besucht werden und kein klares Verhalten auszumachen ist. Dies wird dann durch die Anzeige dreier kleiner, in leicht unterschiedliche Richtungen weisender Pfeile verdeutlicht. Außerdem kann auch gar kein Pfeil angezeigt werden, falls keine Daten vorhanden sind, das heißt keine Sequenz gefunden wurde, die Vorgänger bzw. Nachfolger der aktuellen *RoI* enthält. Durch diese Visualisierung wird der Nutzer auf simple Art und Weise über die Hauptbewegungsrichtung der Besucher informiert. Für die Details aller Sequenzen und deren Häufigkeiten kann dann durch Mausklick die *RoI*-Analyse herangezogen werden.

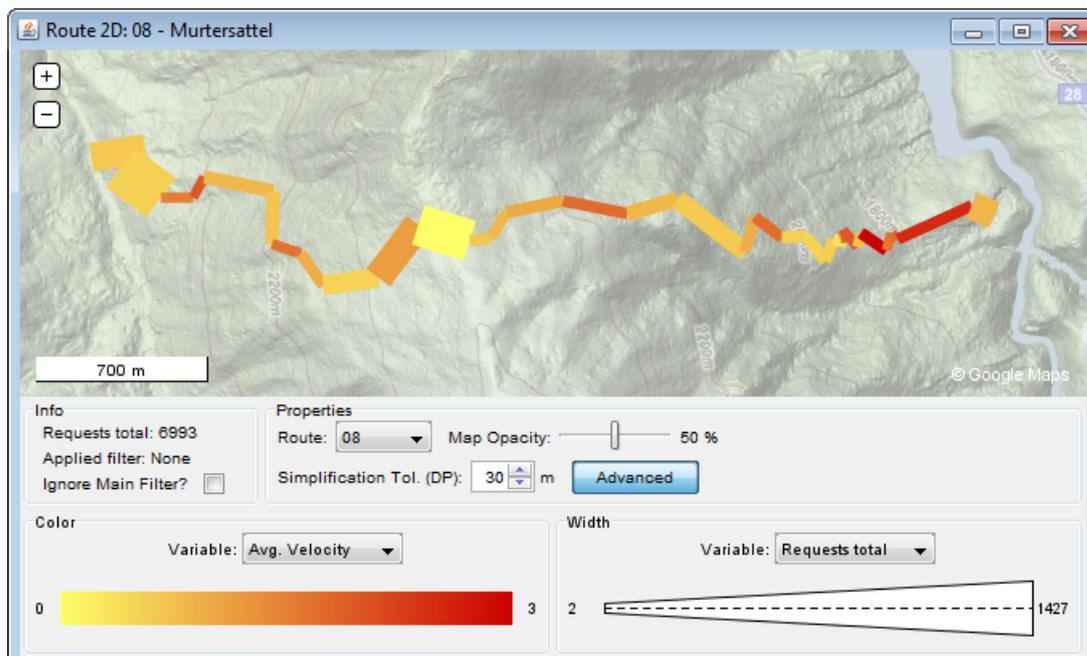


Abbildung 24: Der *Route2D-View*. Die beiden graphischen Variablen Farbton und Linienstärke können jeweils entweder die Anzahl der Abfragen oder die Geschwindigkeit pro Segment darstellen.

Route2D Karte (View → Route2D Map) Ähnlich des vorhergehenden *Views* wird hier ebenfalls eine Karte des Parkgebietes angezeigt. Allerdings ist nur jeweils eine der Routen dargestellt, auf der die Verteilung der Häufigkeiten getätigter Aktionen im Prinzip als eine Art zweidimensionales Histogramm dargestellt wird. Für jedes Segment der Polylinie können entweder die absolute Anzahl der Aktionen oder die relative Anzahl pro Meter visualisiert werden. Darüber hinaus besteht die Möglichkeit, die Durchschnittsgeschwindigkeit der Besucher pro Segment anzuzeigen. Für die kartographische Umsetzung werden im erweiterten Bedienungsfeld unterhalb der Karte zwei graphische Variablen angeboten: Farbe und Linienbreite. Beide können unabhängig voneinander mit verschiedenen Werten belegt werden. So wird beispielsweise in Abbildung 24 die Anzahl der Aktionen mit der Linienbreite und die Durchschnittsgeschwindigkeit per Farbgradient visualisiert. Sowohl Farbgebung als auch Linienbreite der maximalen und minimalen Werte können durch Doppelklick beliebig angepasst werden.

Die Routen sind bedingt durch eine präzise Digitalisierung meist aus sehr vielen einzelnen Segmenten zusammengesetzt. Speziell die graphische Variable der Linienbreite ist in solch einem Falle wenig aussagekräftig. Daher kann die Liniengeometrie durch den weit verbreiteten *Douglas-Peucker-Algorithmus* (vgl. Douglas und Peucker, 1973) vereinfacht und die Anzahl der Segmente somit verringert werden. Der Grad der Vereinfachung wird bei diesem Verfahren durch die Angabe des maximalen Abstandes der ursprünglichen Punkte von der neu erzeugten Polylinie gesteuert. Dieser Toleranzwert in Metern kann durch den Benutzer im Bedienungsfeld angepasst werden. Die Verwendung der höchsten Detailstufe ist in der Regel nie sinnvoll, die Standardeinstellung schlägt einen Schwellwert von 15 Metern vor.

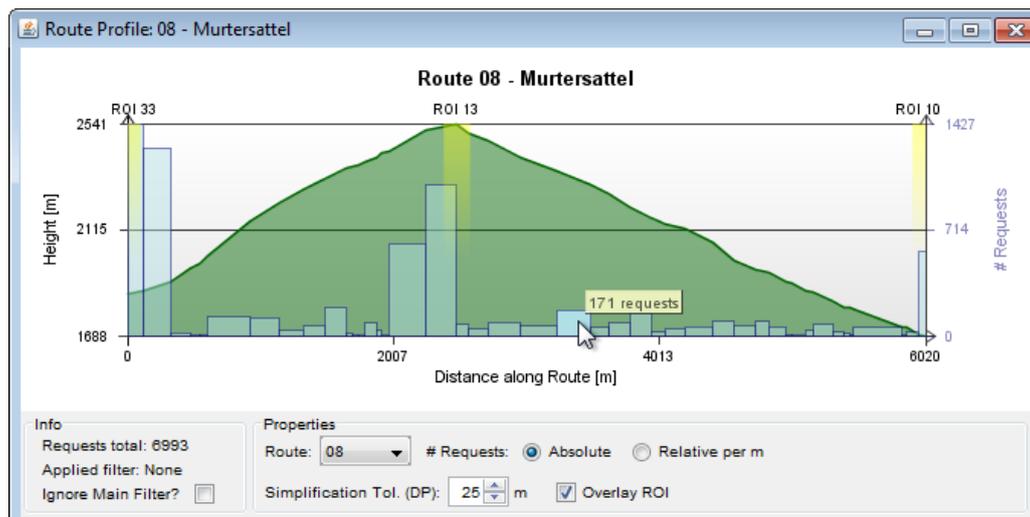


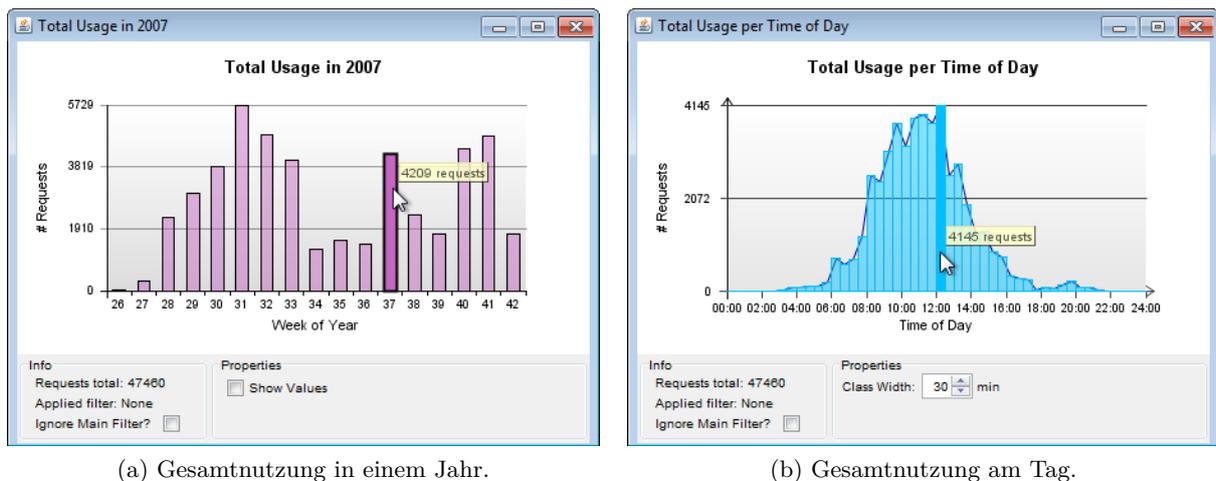
Abbildung 25: Die Routenprofil-Ansicht. Die Anzahl der getätigten Aktionen kann in Zusammenhang mit dem Höhenprofil und den *RoI* gebracht werden.

Routenprofil (View → Route Profile) Auch diese Ansicht stellt die Verteilung der Häufigkeiten entlang der Route dar. Im Gegensatz zur *Route2D*-Karte wird hier jedoch das Profil der Route, das heißt der Höhenverlauf gezeigt (vgl. Abbildung 25). Das Histogramm erscheint in der klassischen Balkenform und die Breite dieser wird wiederum durch die unregelmäßigen Segmente der Route definiert. Der Nutzer kann ähnlich dem vorhergehenden *View* zwischen der Anzeige der absoluten oder relativen Häufigkeit pro Meter wählen. Bei der letzteren Einstellung ist somit die Fläche der Balken proportional zur Häufigkeit, bei der ersteren nur die Höhe relevant. Ebenfalls gleich der *Route2D*-Ansicht ist die Vereinfachung der Liniengeometrie möglich. Darüber hinaus kann der Nutzer wahlweise die Bereiche der Route, in denen sich *RoI* befinden, ein- oder ausblenden.

Die Histogrammbalken sind interaktiv in den *View* eingebunden. Wird der Mauszeiger über diese bewegt, wird der Balken farblich hervorgehoben und der genaue Wert angezeigt. Bei Mausklick öffnet sich eine neue Statistik, welche die Verteilung der *Action Codes* innerhalb des entsprechenden Segments anzeigt (siehe unten).

Gesamtnutzungsstatistiken (View → Total Usage → ...) Unter diesem Menüpunkt sind insgesamt sechs verschiedene *Views* zusammengefasst, die alle einfache Statistiken zur Gesamtnutzung der Geräte anzeigen. Das bedeutet, dass jeweils Säulendiagramme berechnet werden, die die Daten nach verschiedenen Gesichtspunkten gruppieren und die Gesamtanzahl der getätigten Aktionen der Besucher darstellen.

Die Gesamtnutzung **pro Wanderroute** zeigt die Anzahl der Einträge auf jeder Route. Dabei kann zwischen der absoluten oder der relativen Anzahl pro Meter gewählt werden. Bei Bewegung des Mauszeigers über die Säulen wird, wie generell bei allen *Views* dieser Art, die Säule farblich hervorgehoben und der genaue Wert gezeigt. Durch Linksklick öffnet sich das Profil der entsprechenden Route in einem neuen Fenster, durch Rechtsklick eine neue Ansicht, welche die



(a) Gesamtnutzung in einem Jahr.

(b) Gesamtnutzung am Tag.

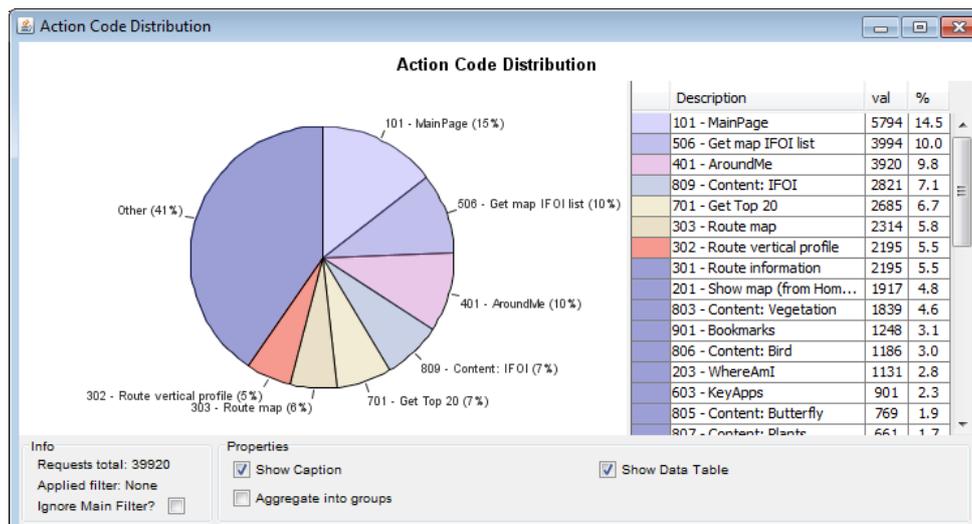
Abbildung 26: Zwei verschiedene Gesamtnutzungsstatistik-Views.

Verteilung der *Action Codes* nur auf der Route enthält (siehe unten).

Die nächste Möglichkeit besteht in der Gruppierung **pro RoI**. Dabei wird für jede Region die Anzahl der Aktionen dargestellt, die innerhalb ihrer Grenzen gemacht wurden. Bei Mausklick auf eine Säule wird auch hier die Verteilung der *Action Codes*, bezogen auf den entsprechenden *RoI*, geöffnet.

Die folgenden drei *Views* stellen alle die Gesamtnutzung gruppiert nach verschiedenen zeitlichen Variablen dar. Die Nutzung **pro Woche** schlüsselt zunächst die Anzahl der Aktionen nach den einzelnen Kalenderwochen des Jahres auf (vgl. Abbildung 26a). Durch Rechtsklick öffnet sich wiederum ein neuer *View*, der die Verteilung der *Action Codes* innerhalb der gewählten Woche zeigt. Durch Linksklick öffnet sich die Gesamtnutzung **pro Wochentag**. Dabei wird in diesem Fall konkret die jeweils angeklickte Woche dargestellt. Wählt man hingegen den Menüeintrag für diese Ansicht, so werden *alle* Aktionen, pro Wochentag kumuliert, angezeigt. Auch in diesem Fenster öffnet sich durch Rechtsklick die Verteilung der *Action Codes*, durch Linksklick die Gesamtnutzung **pro Tageszeit**, sodass der Nutzer immer weiter ins Detail gehen kann (vgl. Abbildung 26b). Dabei besteht wieder der Unterschied darin, ob der Aufruf aus einem *View* einer konkreten Kalenderwoche erfolgt oder durch den Menüeintrag. Im ersteren Fall wird die Häufigkeitsverteilung der Aktionen für diesen einen Tag im Jahr gezeigt, während im zweiten Fall alle Daten verwendet werden. Die Statistik der tageszeitlichen Nutzung unterscheidet sich außerdem etwas von den übrigen Gesamtnutzungsstatistiken, da bei ihr keine diskrete Abgrenzung der X-Variable vorliegt. Die Tageszeit ist kontinuierlich, daher müssen für die Erstellung des Histogrammes Klassen gewisser Breite gebildet werden, die die kumulierte Anzahl der getätigten Aktionen innerhalb dieses Zeitraumes enthalten. Die Standardeinstellung beträgt hier 15 Minuten, kann aber durch den Nutzer angepasst werden.

Als letzter Eintrag des Untermenüs kann schließlich noch die Gesamtnutzung **pro Gerät** aufgerufen werden. Dabei besteht für den Nutzer die Wahl zwischen zwei unterschiedlichen Berechnungsarten: entweder die absolute Anzahl pro Gerät oder die relative Anzahl der Aktionen pro Tag. Einige Geräte werden häufiger verliehen als andere und bei der letzteren Option wird der

Abbildung 27: Statistik zur Verteilung der *Action Codes*.

Einfluss dieser Tatsache auf die Gesamtstatistik eliminiert, sodass die durchschnittliche Anzahl der Aktionen pro Gerät pro Tag angezeigt wird. Ein Mausklick auf die Säulen öffnet auch hier einen neuen *View* mit der Verteilung der *Action Codes* auf dem jeweiligen Gerät (siehe folgender Abschnitt).

Verteilung der Action Codes (View → Action Code Distribution) Nachdem in den vorhergehenden Abschnitten schon des Öfteren von dieser Statistik die Rede war, soll hier nun geklärt werden, was diese genau darstellt: Wie Abbildung 27 zeigt, enthält die Ansicht ein Kreisdiagramm mit den meistgenutzten Programmfunktionen der *WebPark*-Software, das heißt den *Action Codes*. Bei Aufruf aus der Menüleiste bezieht sich die Statistik auf alle Daten unter Berücksichtigung eines möglichen Filters. Jedoch wurde bereits oben deutlich, dass dieser *View* auch mit einer Reihe unterschiedlicher Restriktionen, unabhängig vom Filter, aufgerufen wird. So werden beispielsweise nur die Verteilung der Aktionen auf bestimmten Routen oder innerhalb bestimmter Zeiträume dargestellt. Diese Einschränkungen sind bei den Ansichten dieser Art, welche aus anderen heraus aufgerufen werden, fest eingestellt und auch nicht durch den Filter änderbar.

Da es zu viele *Action Codes* gibt, um sie alle in einem übersichtlichen Diagramm darzustellen, werden immer nur die sieben meistgenutzten angezeigt und der Rest in einem achten Kreissegment („Andere“) zusammengefasst. Die vollständigen Daten können durch Einblendung der detaillierten Datentabelle eingesehen werden. Darüber hinaus besteht die Möglichkeit, das Diagramm nicht basierend auf den einzelnen *Action Codes*, sondern auf den bereits im Filter angewendeten Codegruppen zu berechnen. Dies ist in manchen Fällen aufschlussreicher.

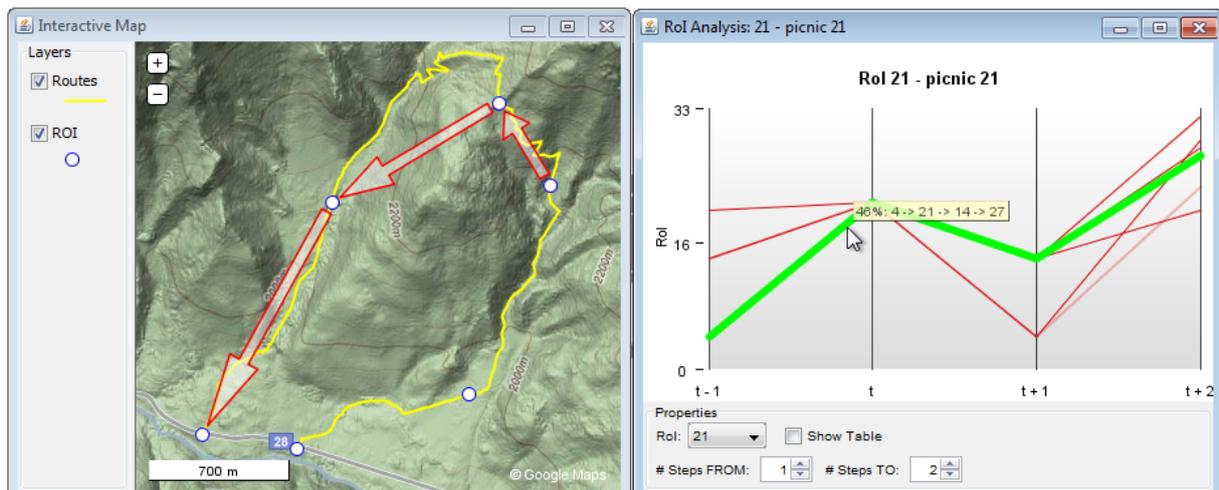
Rol-Analyse (View → Rol Analysis) Im Zuge der obigen Erklärung der interaktiven Karte kam bereits die Analyse der *RoI*-Sequenzen, das heißt der von den Wanderern nacheinander besuchten interessanten Regionen zur Sprache. Dort wurde dem Nutzer bei Positionierung des

Mauszeigers über einer *RoI* durch Pfeile die Bewegungsrichtung der Mehrzahl der Besucher angedeutet. Ein Mausklick oder die Wahl des entsprechenden Menüpunktes öffnet eine Ansicht mit der detaillierten Statistik der Sequenzen. Es wird darin jede Sequenz verwendet, in der die *RoI*, welche Gegenstand der Analyse ist, vorkommt. Bei all diesen Sequenzen werden dann nicht nur die unmittelbar vorhergehenden und nachfolgenden Regionen betrachtet, sondern wahlweise jeweils bis zu drei Schritte voraus bzw. zurück in der Sequenz. Die Visualisierung aller gefundenen Sequenzen erfolgt in einem so genannten Parallelkoordinatensystem. Bei diesem steht jede senkrechte Achse für eine *RoI* innerhalb der Sequenz. Die Achse t steht in Anlehnung an die zeitliche Abfolge der Wanderung für die aktuelle *RoI*, $t - x$ für die vorhergehenden und $t + x$ für die nachfolgend besuchten. Auf der Y-Achse wird der Wert der *RoI*, das heißt ihre eindeutige ID, aufgetragen. Dies führt zu einer kompakten Darstellungsweise der Bewegungssequenzen, welche die visuelle Musterextraktion fördert. Durch die Dicke der Linien im Diagramm sowie die Farbsättigung wird darüber hinaus jeweils die Anzahl der gefundenen Sequenzen des entsprechenden Typs gekennzeichnet.

Im Bedienungsfeld kann der Nutzer die Anzahl der Schritte vor und nach der *RoI* von null bis drei anpassen und damit auch die Achsenanzahl beeinflussen. Da immer nur solche Sequenzen berücksichtigt werden, die die komplette Anzahl an spezifizierten Schritten beinhalten, wird die Grundgesamtheit in der Regel bei höherer Dimension sehr gering oder sogar null. Der Grund dafür liegt darin, dass Sequenzen, welche beispielsweise die aktuelle *RoI* sowie die Höchstzahl von drei Vorgängern und Nachfolgern enthalten, in den Daten sehr selten oder gar nicht vorkommen. Denn eine *RoI* kann nur dann Bestandteil der Sequenz sein, wenn der Besucher innerhalb der Grenzen sein Gerät benutzt hat. Dieses Problem verhindert die vollständige Analyse der Besucherbewegung und wird in Kapitel 8.3 diskutiert.

Mithilfe dieser Analyse können somit nicht nur die hauptsächlich auftretenden Vorgänger und Nachfolger in einer Sequenz, sondern auch alle anderen Bewegungsmuster untersucht werden. Durch das Einblenden der Datentabelle können die Details aller Sequenzen abgerufen werden. Bei Bewegungen des Mauszeigers über eine Sequenz im Parallelkoordinatensystem zeigt ein *Tooltip* die textliche Beschreibung dieser sowie ihre Häufigkeit an. Nun sind die einzelnen IDs der *RoI* für den Nutzer nicht gerade intuitiv. Um aus diesem abstrakten Diagramm deshalb einfacher die Verbindung zur räumlichen Situation im Park herstellen zu können, existiert eine weitere Funktion, die allerdings nur dann funktioniert, wenn gleichzeitig die interaktive Karte geöffnet ist. In dieser wird dann die Sequenz, über der sich der Mauszeiger augenblicklich befindet, durch Pfeile angezeigt. Somit kann das Besucherverhalten einfacher nachvollzogen und mit den räumlichen Gegebenheiten in Verbindung gebracht werden. Abbildung 28 zeigt ein Beispiel.

Die *RoI*-Analyse ist eine der oben angesprochenen Ansichten, die nicht vom Filter beeinflusst wird. Ihre Datengrundlage bleibt somit immer identisch. Dies ist so gewählt, da die Bewegungssequenzen sowohl räumlich als auch zeitlich eine große Ausdehnung besitzen und die Filterung dieser erstens nicht sinnvoll ist und zweitens die ohnehin schon recht kleine Datengrundlage noch weiter verringern würde.

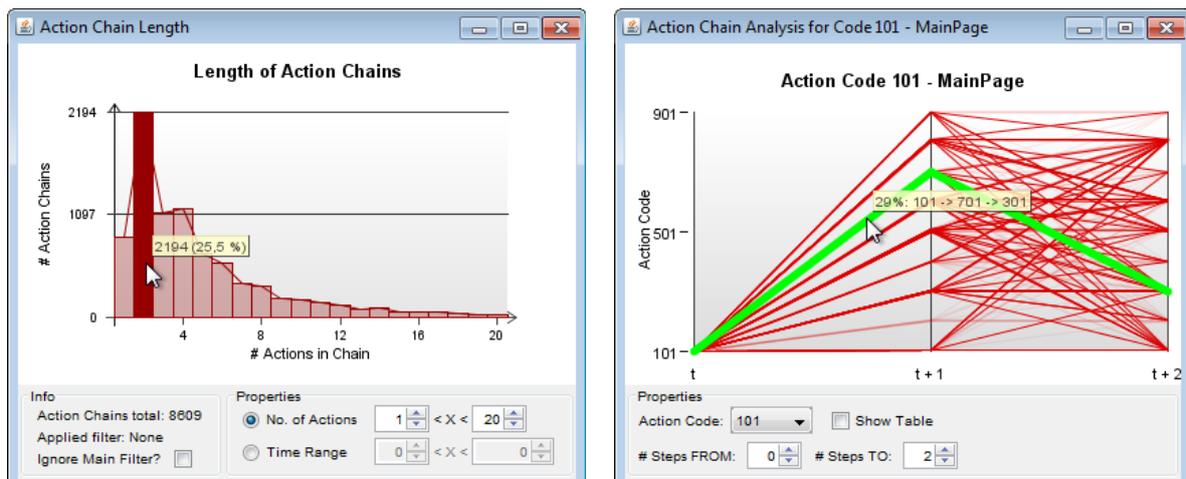
Abbildung 28: Detaillierte *ROI*-Analyse.

Aktivste Besucher (View → Top Visitors) Dieser *View* stellt ähnlich der Gesamtnutzungsstatistiken ein Säulendiagramm dar, welches sich in diesem Fall auf die einzelnen Besucher, das heißt die *Sessions*, bezieht. Es werden diejenigen Besucher dargestellt, welche ihr Gerät am häufigsten benutzt haben. Per Linksklick auf eine Säule kann die gleiche Informationsbox mit Angaben über den entsprechenden Wanderer geöffnet werden, wie schon im Filter (vgl. Kapitel 6.3). Ein Rechtsklick öffnet die Verteilung der *Action Codes* für den Besucher.

Der Nutzen dieser Statistik allein ist relativ gering. Interessant wird es erst dann, wenn die Datengrundlage der Statistik durch den Filter eingeschränkt wird. Wenn dies der Fall ist, kann der Benutzer im Bedienungsfeld auf eine zweite Ansicht umschalten, welche den prozentualen Anteil der getätigten Aktionen unter Berücksichtigung des aktuellen Filters von *allen* Aktionen anzeigt. Mit entsprechender Filtereinstellung kann beispielsweise herausgefunden werden, welche Benutzer gewisse Aktionen am häufigsten verwenden. In Kapitel 7.3 wird dies verdeutlicht.

Analyse der Aktionsketten (View → Action Chains → ...) Unter diesem Menüpunkt werden zwei weitere *Views* angeboten, die sich näher mit der Struktur der *Action Codes* auseinandersetzen. Genauer gesagt beschäftigen sie sich mit den so genannten *Action Chains*, das heißt Aktionsketten. Eine solche Aktionskette ist definiert als die chronologische Abfolge aller Programmfunktionen, die der Besucher auf seinem Gerät wählt. Sie beginnt und endet mit dem Aufruf der Hauptseite, das heißt dem *ActionCode 101 – MainPage*. Da die Einträge während der Vorverarbeitung zeitlich sortiert worden sind, können diese Ketten ohne Weiteres aus der Datentabelle extrahiert werden. Jedes mal, wenn der Code 101 auftaucht, wird eine neue Kette gestartet und ihr alle folgenden Aktionen hinzugefügt. Diese Ketten enthalten wertvolle Informationen, nicht über das räumliche oder zeitliche Benutzerverhalten, sondern über die Art und Weise, wie die Wanderer die *WebPark*-Software bedienen.

Der erste der beiden Untermenüpunkte analysiert die **Länge der Action Chains**. Es wird ein Histogramm dargestellt, welches die Anzahl der in den Daten vorhandenen Aktionsketten verschiedener Längen aufsummiert (vgl. Abbildung 29a). Die Länge kann entweder bezüglich der



(a) Histogramm zur Länge der Aktionsketten.

(b) Parallelkoordinaten zur detaillierten Aktionskettenanalyse.

Abbildung 29: Die beiden *Views* zur Analyse der Aktionsketten.

Anzahl der Aktionen in der Kette oder der Zeitspanne verwendet werden. Bei beiden Varianten kann der Nutzer den Definitionsbereich der Länge anpassen. Dies ist deshalb sinnvoll, da das Diagramm andernfalls durch einige wenige, außergewöhnlich lange Ketten in den Daten verzerrt wird. Bei Bewegungen des Mauszeigers wird ein *Tooltip* mit dem Wert der Histogrammklasse angezeigt, weitere Mausinteraktion ist hier nicht möglich.

Die zweite Ansicht bietet eine genauere **Analyse der Action Chains**. Sie ist ähnlich aufgebaut wie die oben beschriebene *RoI*-Analyse, beinhaltet jedoch Sequenzen der *Action Codes*, die von einem bestimmten *Code* ausgehen. Es werden ebenfalls parallele Koordinaten verwendet, um die Sequenzen der aufeinander folgenden Funktionsaufrufe darzustellen (vgl. Abbildung 29b). Auch hier wird der Filter, aus denselben Gründen wie oben, generell ignoriert und immer alle in den Daten vorhandenen Aktionsketten zur Berechnung herangezogen. Darüber hinaus verwendet diese Ansicht auch diejenigen Einträge der Datentabelle, welche nicht auf eine Route projiziert werden konnten, da dieser räumliche Kontext hier eine eher unwesentliche Rolle spielt und damit eine Vergrößerung der Grundgesamtheit erreicht werden kann. Die *Action Chain*-Analyse ermöglicht es dem Nutzer, das typische Verhalten der Besucher mit ihren Geräten zu untersuchen. Es steht ebenfalls die Möglichkeit der Einblendung der Datentabelle zur Verfügung.

Inhalt der Such- und Geo-Bookmark-Funktion (*View* → *Content of Search/Bookmark*)

Diese Ansicht hat, im Gegensatz zu den bisherigen, nichts mit der räumlichen, zeitlichen oder *Action Code*-bezogenen Analyse zu tun, sondern beschäftigt sich speziell mit zwei Funktionen der *WebPark*-Software: der freien Textsuche sowie dem Erstellen von *Geo-Bookmarks*. Bei beiden können die Besucher beliebigen Text eingeben. Im ersten Fall mit dem Ziel, nach bestimmten Informationen, z. B. über Tierarten, zu suchen und im zweiten Fall, um den von ihnen gesetzten *Geo-Bookmark* zu benennen und zu beschreiben. Dieser eingegebene Text wird während der Konvertierung der Daten (vgl. Kapitel 5.2) aus der ursprünglichen *Action_*-Zeichenkette extrahiert und in separaten Feldern der Tabelle gespeichert. Der *View* stellt alle von den Besuchern

verwendeten Wörter in einer so genannten *tag cloud* zusammen. In solch einer „Wolke“ werden die Wörter entsprechend ihrer Häufigkeit hervorgehoben, z. B. durch größere oder fettgedruckte Schrift. Somit wird dem Nutzer sowohl eine kompakte Übersicht der verwendeten Ausdrücke geboten als auch gleichzeitig die wichtigsten Wörter schnell erkennbar herausgestellt. Da viele Wörter wie Artikel, Konjunktionen, Präpositionen usw. für diesen Zweck uninteressant sind, durchlaufen alle Wörter einen Filter, der zum einen eine Mindestanzahl von drei Buchstaben voraussetzt und zum anderen eben diese genannten Wörter anhand einer manuell zusammengestellten Liste vernachlässigt. Im Bedienungsfeld unter der *tag cloud* wird dem Nutzer die Option angeboten, die Anzahl der dargestellten Wörter anzupassen.

Notizblock (*View* → *Notepad*) Der letzte aller verfügbaren *Views* besteht lediglich aus einem einfachen Notizblock, auf dem der Nutzer beliebige Anmerkungen, beispielsweise zu Mustern, die er in den Statistiken entdeckt hat, vermerken kann.

6.5 Weitere Funktionen

Mit dem Filter und den verschiedenen *Views* wurden nun die wichtigsten Komponenten von SNP Analytics detailliert beschrieben. Dieses Unterkapitel beinhaltet abschließend einige weitere Funktionen des Programmes:

Speichern und Laden von Projekten (*File* → *Save/Save As.../Load...*) Im *File*-Menü kann der Nutzer jederzeit sein aktuelles Projekt speichern sowie früher gesicherte Projekte laden. Dies ist eine sehr nützliche Funktion, da das Zusammenstellen der einzelnen Statistiken und Filtereinstellungen eine gewisse Zeit in Anspruch nehmen kann. Durch die Speicherung gehen die Einstellungen nicht verloren und können auch zu einem späteren Zeitpunkt wieder rekonstruiert werden.

Die Projektdatei besitzt die Endung *.snp*, ist aber in Wahrheit nichts anderes als unverschlüsseltes XML¹⁸ und kann mit den meisten Texteditoren eingesehen werden. Die einzelnen *tags* dieser Datei enthalten alle benötigten Informationen über die einzelnen *Views* und Filtereinstellungen. Neben der Position auf dem *Dashboard* und der Größe des Fensters müssen für jede Ansicht die Einstellungen gespeichert werden, die vom Nutzer über die Bedienungsfelder angepasst werden können. Daher schreibt die Basisklasse *MyInternalFrame* vor, dass jede abgeleitete *View*-Klasse Code zur Umwandlung ihrer spezifischen Einstellungen in XML-*tags* implementiert. Dieser Prozess kann als *Serialisierung* bezeichnet werden, da die Eigenschaften der Objekte, die zur Programmlaufzeit existieren, mittels lesbarem Text ausgedrückt werden. In der *MainWindow*-Klasse werden die Informationen aller Ansichten gesammelt sowie die aktuelle Einstellung des Hauptfilters serialisiert, um dann alle Informationen in die Datei zu schreiben. Beim Ladeprozess werden die Attribute der einzelnen *tags* wieder eingelesen und die zugehörigen *Views* mit den entsprechenden Einstellungen wiederhergestellt. Das Einlesen der XML-Datei wird durch

¹⁸Extensible Markup Language – <http://www.w3.org/XML>

den von *Java* bereitgestellten, so genannten *SAX Handler*¹⁹, erleichtert. Mit diesem können die einzelnen *tags* und ihre Eigenschaften bequem mit Hilfe ihrer Bezeichnungen eingelesen werden. Eine komplizierte Rekonstruktion der *tags*, wie sie sonst beim „normalen“ Einlesen aller Zeilen nacheinander vonnöten wäre, wird umgangen.

Views anordnen (*Extras* → *Arrange Views* → ...) Der erste Eintrag des *Extras*-Menüs erlaubt die automatische Anordnung der Ansichten auf dem *Dashboard*. Bei einer großen Anzahl kann die Situation schnell unübersichtlich werden. Daher bietet das Programm die Möglichkeit, alle *Views* entweder nebeneinander in Form eines regelmäßigen **Gitters**, oder **überlappend** anzuordnen.

Screenshots (*Extras* → *Take Screenshot...*) Eine weitere Zusatzfunktion im *Extras*-Menü ermöglicht es dem Nutzer, verschiedene *Screenshots* anzufertigen. Bei systemeigenen Funktionen zur Speicherung des Bildschirminhaltes ist es teilweise nur mühsam möglich, einzelne Ausschnitte der Anzeige abzuspeichern. Daher ermöglicht diese Funktion das unkomplizierte Anfertigen von *Screenshots*, die speziell auf die Struktur von SNP Analytics zugeschnitten sind. Bei Auswahl des Menüpunktes öffnet sich ein Dialogfenster, in dem der Nutzer zwischen drei Optionen wählen kann:

1. Speicherung des kompletten Bildschirminhaltes,
2. Speicherung nur des Bildschirmausschnittes, welcher vom Hauptfenster des Programmes eingenommen wird oder
3. Speicherung lediglich des momentan ausgewählten *Views*.

Darüber hinaus kann direkt ein Speicherort für die Bilddatei angegeben werden.

Zusätzlich zur oben erklärten Speicherfunktion der Software können somit auch individuelle *Screenshots* angefertigt werden, die beispielsweise für Präsentationszwecke nützlich sind.

Programmhilfe Zu guter Letzt beinhaltet SNP Analytics eine umfangreiche Programmhilfe, welche über das *Help*-Menü aufgerufen werden kann. In ihr sind die in diesem Kapitel beschriebenen einzelnen Komponenten des Programmes erklärt. Überdies enthält sie alle notwendigen Schritte der Datenvorverarbeitung aus Kapitel 5 sowie eine Anleitung zur korrekten Installation der erforderlichen Begleitsoftware *PostgreSQL/PostGIS* und das daran anschließende Erstellen der SNP Datenbank. Somit werden dem Nutzer zum einen alle notwendigen Schritte erläutert, die zur korrekten Funktionsweise von SNP Analytics benötigt werden, zum anderen Erklärungen zu den einzelnen Komponenten angeboten.

An dieser Stelle ist die Beschreibung der Analysesoftware abgeschlossen. Im nächsten Unterkapitel wird demonstriert, wie alle zur Verfügung stehenden Funktionen der Software genutzt werden können, um eine umfangreiche visuelle Analyse der Bewegungsdaten durchzuführen.

¹⁹Simple API for XML – <http://www.saxproject.org>

7 Ergebnisse der visuellen Analyse

Nach der Vorverarbeitung und der Entwicklung von SNP Analytics folgen in diesem Kapitel einige mit dem Programm durchgeführte Analysen der *WebPark*-Daten. Diese sind alle unter dem Gesichtspunkt der Hauptfrage aus Kapitel 1.2 zu sehen: der Erforschung des Zusammenhangs zwischen der raumzeitlichen Bewegung der Besucher und den getätigten Aktionen mit ihrem Wanderführer. Die Software bietet dazu viele mögliche Wege, welche an dieser Stelle nicht alle abgedeckt werden können und sollen. Vielmehr geht es in den Kapiteln 7.1 bis 7.5 zunächst darum, einige konkrete Beispiele von *high-level*-Analysen in SNP Analytics zur Beantwortung dieser und der übrigen Forschungsfragen vorzustellen. In Kapitel 7.6 werden verschiedene weitere Analyseergebnisse präsentiert. Für die meisten der folgenden Beispiele wurden, falls nicht anders gekennzeichnet, die Daten von 2007 verwendet.

7.1 Zusammenhang raumzeitliches Verhalten – Verwendung des Wanderführers

Bezüglich der konkreten Beantwortung der Hauptfrage wird als erstes die Anzahl der Programmfunktionen, die die Besucher innerhalb der Picknickplätze getätigt haben, mit der auf den sonstigen Bereichen der Routen gemachten verglichen. Auf diese Weise sollen Unterschiede im Verhalten auf den Rastplätzen gefunden werden. Mithilfe des *Action Code Distribution-Views* kann zum einen die Gesamtverteilung aller getätigten Aktionen abgerufen werden. Zum anderen ist durch Anwendung des Filters die Anzeige der Verteilung aller Aktionen möglich, die innerhalb der Picknickplätze gemacht wurden. Tabelle 5 zeigt die Ergebnisse:

Codegruppe	Nur Picknick		Rest	
	n	%	n	%
General	2278	22,89	9234	24,62
Info Around	2390	24,01	7621	20,32
Info on Route	1365	13,71	7210	19,22
Orientation	772	7,76	3070	8,19
Content	2302	23,13	7078	18,87
Special Function	500	5,02	1909	5,09
Info on Device	346	3,48	1385	3,69
Summe	9953	100,0	37507	100,0

Tabelle 5: Anzahl der Programmaktionen innerhalb der Picknickplätze und im Rest des Parks.

Die Besucher tätigen ein Fünftel aller Aktionen innerhalb der Rastplätze. Unter Berücksichtigung des Flächenanteils der Plätze ist das sehr viel, da dieser nicht annähernd einen solch großen Teil des Parkgebietes ausmacht. Sichtbar ist das ebenfalls in der Verteilung entlang der Routenprofile, da dort oft die hohen Säulen innerhalb der *RoI* auftreten. Diese Erkenntnis geht einher mit der schon früh festgestellten Nutzungsanhäufung in diesen Gebieten. Die Besucher haben dort, im Gegensatz zur Wanderung, mehr Zeit, sich mit ihrem Gerät zu beschäftigen. Sie nutzen diese offensichtlich, um die Funktionen ihres Gerätes genauer auszuprobieren. Bezüglich der Art der abgefragten Informationen fallen in einigen der aufgeführten Gruppen Unterschiede auf: Die

Wanderer benutzen häufiger Funktionen aus der *Info Around*-Gruppe. In dieser kann beispielsweise herausgefunden werden, was sich um sie herum an *Features of Interest (FOI)* befindet (*AroundMe*). Außerdem wird die *Content*-Gruppe verstärkt benutzt. In ihr finden sich detaillierte Beschreibungen bestimmter Spezies oder Pflanzen. Informationen über die Wanderrouuten (*Info on Route*) sowie allgemeine Kartenfunktionen (*Orientation*) werden hingegen seltener abgefragt. Sie sind offensichtlich während der Wanderung selbst interessanter.

Diese Erkenntnisse sind nicht neu und wurden unter anderem schon von Eisenhut u. a. (2008) gemacht. Mit SNP Analytics können die entsprechenden Statistiken allerdings auf sehr einfache und schnelle Art und Weise erzeugt werden. Außerdem besitzt das Programm über die hier gezeigten Beispiele hinaus durch die Filterung noch weitaus mehr Möglichkeiten. Unter anderem können die *Action Codes* und der räumliche Kontext beliebig gefiltert werden.

Eine tiefer gehende Analyse bezüglich der Hauptfrage stellt das folgende Beispiel dar: Es wird untersucht, wie die Interaktion mit dem Wanderführer die anschließende Bewegung der Besucher direkt beeinflusst. Das geschieht durch nähere Betrachtung des Inhaltes der *Info on Route*-Gruppe. Durch die dritte Bauebene bei den *Codes* 301, 302 und 303 kann herausgefunden werden, über *welchen* der Wanderpfade die Besucher Informationen abfragen. Dies wird in Verbindung mit der eigenen Position gesetzt. Dazu wird jeweils eine Gesamtnutzungsstatistik pro Route erstellt, welche die folgende Filtereinstellung besitzt: Die *Action Codes* werden auf *Route information/map/vertical profile* beschränkt. Darüber hinaus wird in der dritten Ebene als Funktionsparameter nur die Abfrage jeweils einer bestimmten Route zugelassen. So können lediglich solche Einträge berücksichtigt werden, welche eine der drei genannten Aktionen enthalten und zwar nur dann, wenn diese die Abfrage beispielsweise von Route 2 beinhalten. Für jede Route muss eine eigene Gesamtnutzungsstatistik erstellt werden. Das Ergebnis ist in Tabelle 6 zusammengefasst:

	Info über Route													Σ
	1	2	3	6	7	8	9	10	12	13	15	17	18	
1	867	11	13	0	6	1	1	1	3	1	0	8	0	912
2	227	43	6	0	0	0	0	0	0	0	0	4	0	280
3	1	1	47	0	0	0	0	0	0	0	0	0	0	49
6	0	0	0	42	36	0	0	0	0	0	0	0	0	78
7	47	20	19	87	1596	68	4	8	15	7	35	120	6	2032
8	7	1	33	22	154	882	4	41	2	0	0	1	0	1147
9	0	0	2	2	1	68	75	51	4	2	0	0	1	206
10	5	0	1	0	2	8	0	202	3	0	0	0	9	230
12	6	3	2	0	4	3	19	0	354	15	5	5	7	423
13	42	5	6	3	8	7	4	2	212	33	7	38	54	421
15	3	0	2	0	0	2	0	1	2	1	355	4	0	370
17	15	2	9	2	44	8	1	1	2	4	11	1197	57	1353
18	3	0	0	0	1	0	0	0	0	0	1	6	152	163
Σ	1223	86	140	158	1852	1047	108	307	597	63	414	1383	286	

Tabelle 6: Anzahl der Informationsabrufe über jede der Routen, abhängig von der eigenen Position.

In der letzten Spalte ist zunächst eine hohe Korrelation der Häufigkeiten nur dieser speziellen Funktion zur Gesamtzahl der Abfragen pro Route zu erkennen, da die Pfade 1, 7, 8 und 17

die höchsten Werte aufweisen. Weiterhin wird mit der räumlichen Korrelation ein interessanter Aspekt deutlich: Die Route, über welche Informationen angefordert werden, ist mehrheitlich die, auf der sich die Wanderer selbst befinden (Hauptdiagonale der Tabelle). Das leuchtet ein, da Informationen über beispielsweise die verbleibende Strecke oder Wanderzeit auf dem begangenen Pfad sehr nützlich sind. Die übrigen Werte zeigen, dass weitere abgefragte Routen zum Großteil solche sind, die sich in der Nähe des aktuellen Pfades befinden. Die Besucher wollen offensichtlich Informationen über angrenzende Wege abrufen, um zu entscheiden, ob sie an einer Begehung interessiert sind.

Dieser Aspekt ist besonders im Bereich von Kreuzungen interessant, um zu untersuchen, ob gerade dann die Informationsabfrage über die abzweigende Route verstärkt auftritt. Abbildung 30 bestätigt das: Die zentrale *RoI* in der Karte ist die Kreuzung zwischen Pfad 12 und 13 (*RoI* # 26). Das Profil sowie die *Route2D*-Karte stellen die Häufigkeitsverteilung auf Route 13 dar. Die Daten sind durch den Filter auf solche Aktionen eingeschränkt, in denen Informationen über Route 12 abgefragt werden. Aus der *RoI*-Analyse ist außerdem bekannt, dass sich die Besucher hauptsächlich von Westen nach Osten auf dem oberen Pfad (13) bewegen. Beide Häufigkeitsverteilungen zeigen deutlich, dass Informationen über Route 12 fast ausschließlich vor der Kreuzung abgerufen werden. Sobald diese passiert wurde, haben sich die Besucher für den einen oder anderen Weg entschieden und benötigen keine Informationen mehr.

Diese beiden Beispiele zeigen, wie die direkte Beeinflussung des Wanderführers auf das Besucherverhalten mit SNP Analytics untersucht werden kann. Auch die nächsten Unterkapitel haben alle mehr oder weniger mit diesem Aspekt zu tun, beschäftigen sich allerdings konkret mit den weiteren Fragen aus Kapitel 1.2.

7.2 (Un-)typisches räumliches Besucherverhalten

Die zweite Frage beschäftigt sich mit der alleinigen Untersuchung des räumlichen Verhaltens der Besucher, losgelöst von ihrer Interaktion mit dem Wanderführer. Speziell darauf zugeschnitten ist die *RoI*-Analyse. Diese untersucht die Sequenzen nacheinander besuchter Regionen und kann dadurch (un-)typische Bewegungen der Wanderer durch den Park extrahieren. Im Folgenden sind einige konkrete Beispiele aufgeführt, die Ergebnis dieser Analyse sind und das räumliche Besucherverhalten an verschiedenen Stellen des Parks erläutern.

Die beliebten Routen 7 und 8 werden vorzugsweise von Zernez startend in südöstlicher Richtung abgelaufen. Das ist nicht überraschend, da dort das Nationalparkzentrum steht, welches Ausgangspunkt der meisten Touren ist. Der ebenfalls stark frequentierte Pfad 17 wird von den meisten Besuchern gegen den Uhrzeigersinn durchlaufen. Pfad 15 wird im Jahr 2007 meist vom Startpunkt Il Fuorn in Richtung Südost gewandert, während die Statistik im Jahr 2008 ein relativ ausgeglichenes Verhältnis aufweist: Ebenso viele Wanderer fahren dort zunächst mit dem Auto zum *RoI* #29 und schlagen anschließend Weg 15 in südwestlicher Richtung ein.

Die meisten der auf diesen Wegen liegenden Regionen zeigen bereits in der interaktiven Karte dieses *Durchgangsmuster*. Beim Bewegen des Mauszeigers über sie werden zwei eindeutige Pfei-

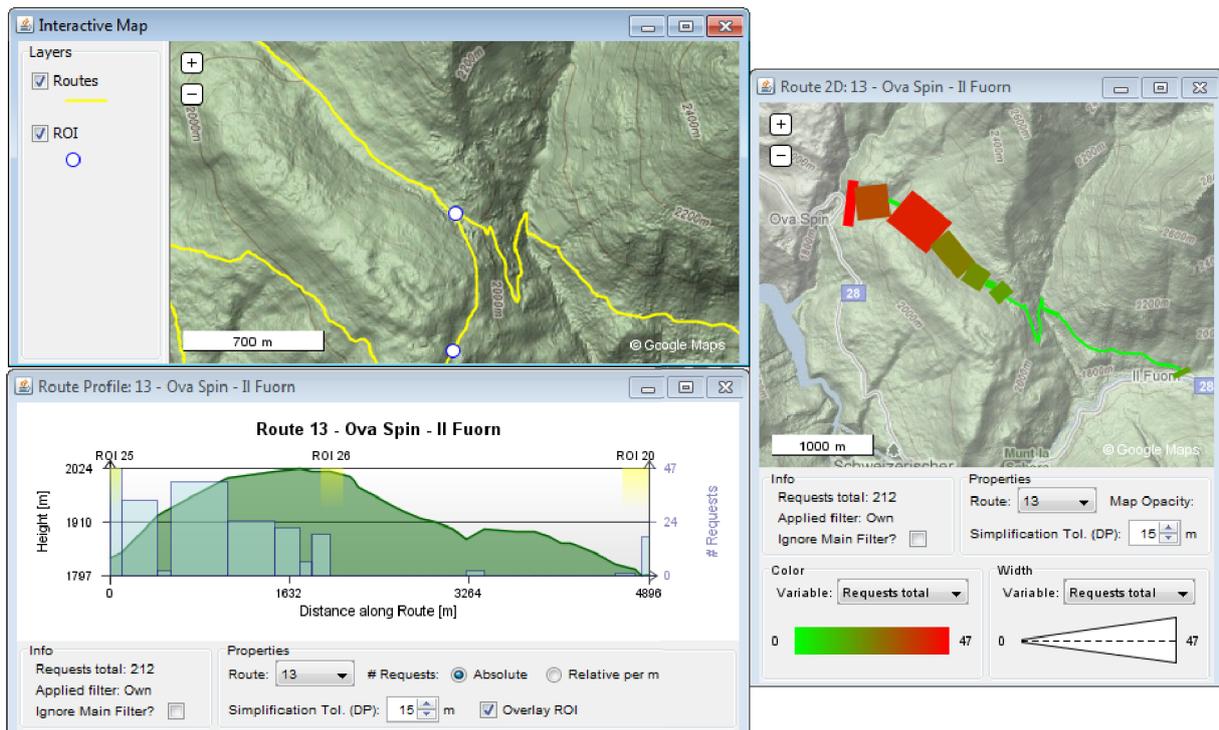
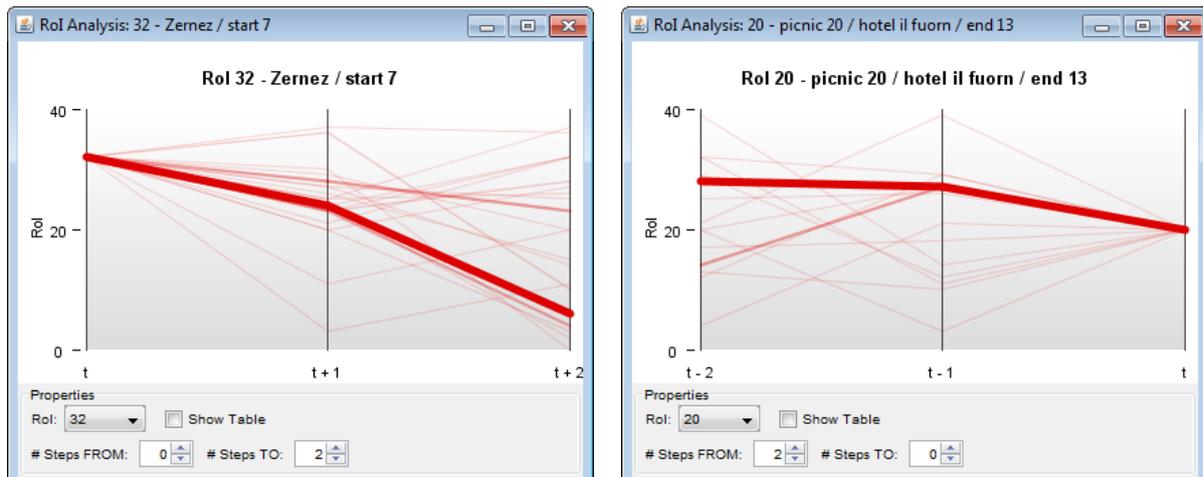


Abbildung 30: Beispiel zum Zusammenhang von räumlichem Verhalten und Gerätenutzung. Die Wanderer auf Pfad 13 rufen Informationen über Pfad 12 nur vor der Abzweigung desselbigen ab.

le angezeigt. Bei den *RoI* auf anderen Pfaden, beispielsweise 9 und 10, zeigt sich dieses klare Muster allerdings nicht. Zum einen liegt das daran, dass zu wenig Daten vorhanden sind (siehe Kapitel 8.3), zum anderen, dass tatsächlich kein bestimmtes Durchgangsmuster existiert und die Besucher diese Pfade nicht in einer bevorzugten Richtung durchlaufen.

Außer den Durchgangsmustern zeigen weiterhin einige Regionen eine Art *Endpunktmuster*. Dabei kehren die Besucher, da sie keine andere Option haben, am Ende des Pfades um. Dies ist beispielsweise bei den Routen 2 und 18 deutlich sichtbar. Der zweite Wanderweg führt stetig bergauf in Richtung der italienischen Grenze und ermöglicht keinen Rundgang. Das Endpunktmuster tritt hier sogar bereits mitten auf dem Pfad an *RoI* # 1 auf. Viele der Besucher kehren offensichtlich schon vor Ende des Weges wieder um.

Über diese Hauptflussrichtungen in der interaktiven Karte hinaus bietet die *RoI*-Analyse eine detailliertere Statistik. Diese zeigt beispielsweise den besonderen Charakter der *RoI* # 32 (Zernez) und # 20 (Hotel Il Fuorn): Im Anschluss an Zernez werden zwar hauptsächlich die *RoI* der siebten Route besucht, darüber hinaus jedoch auch sehr viele verschiedene andere Regionen (vgl. Abbildung 31a). Nach der Ausleihe des Gerätes probieren die Besucher dieses offensichtlich kurz aus und planen ihre Tour. Diese starten sie möglicherweise an einem beliebig entfernten Punkt des Parks und benutzen dort das Gerät irgendwann erneut. Über die quer durch den Nationalpark verlaufende Landstraße ist es beispielsweise möglich, Route 17 von Zernez aus anzufahren oder Route 1 per Zug zu erreichen. Die Wanderer starten somit zwar mehrheitlich, aber nicht ausschließlich auf Pfad 7. Das Hotel Il Fuorn besitzt als Ausgangspunkt von Tagestou-



(a) Zernez als Ausgangspunkt vieler verschiedener Touren. (b) Il Fuorn als Endpunkt vieler verschiedener Touren.

Abbildung 31: Die *RoI*-Analyse zeigt die Sonderstellung der Regionen Zernez und Il Fuorn.

ren ebenfalls viele verschiedene Regionen, die anschließend besucht werden. Noch ausgeprägter sind allerdings die *vorhergehenden RoI* (vgl. Abbildung 31b). Ebenso wie bei Zernez existiert zwar eine Mehrheit der nahe liegenden *RoI* von Pfad 17, darüber hinaus aber viele verschiedene weitere zuvor besuchte Regionen. Die Wanderer benutzen ihr Gerät vermutlich an einem beliebigen Punkt des Parks zuletzt, bevor sie sich zurück in ihr Hotel begeben. Dort angekommen wird der Wanderführer, zum Abschluss des Tages, noch einmal herausgeholt. Dieser mittels der *RoI*-Analyse postulierte Sonderstatus von Zernez als Ausgangspunkt und Il Fuorn als Endpunkt von Tagesstouren wird durch weitere Funktionen in SNP Analytics bestätigt: Die tageszeitliche Filterung der Daten ergibt, dass in Zernez 69 % aller Einträge vor 11:00 Uhr getätigt werden (im Durchschnitt 48 %). Beim Hotel Il Fuorn sind sogar 59 % aller Aktionen nach 14:00 Uhr verzeichnet (im Durchschnitt nur 12 %). Die Verteilung der *Action Codes* stützt diese These ebenfalls für beide Regionen: Zernez beinhaltet überdurchschnittlich viele allgemeine Programmfunktionen der Gruppe *General*, Il Fuorn hingegen sehr viele speziellere Aktionen der Gruppe *Content*. Der Umgang der Besucher mit ihrem Gerät ist somit zu Beginn noch sehr oberflächlich, während am Ende des Tages auch speziellere Funktionen aufgerufen werden.

Das (un-)typische Verhalten an den Wegekreuzungen ist ein weiterer interessanter Punkt und kann ebenfalls recht gut abgeleitet werden: So gibt es beispielsweise in den Daten des Jahres 2007 nur sehr wenige bzw. gar keine Wanderer, die von Route 1 oder 7 auf die alpine und schwer zu begehende Route 3 abbiegen. Diese enthält insgesamt nur sehr wenig Einträge. Von Zernez auf Pfad 7 kommend wählen in *RoI* # 24 außerdem mit 7 % im Jahr 2007 nur sehr wenige die Abbiegung auf Route 6. Bei der bereits oben thematisierten Kreuzung von Pfad 12 und 13 bleiben von Westen kommend 38 % auf Route 13, während 25 % auf Route 12 weiterwandern. Eine deutliche Präferenz zeigt sich bei Region 39 auf Pfad 17, in der nur 12 % im Jahr 2007 bzw. 5 % im Jahr 2008 die Abzweigung auf Route 18 nehmen.

Die Analyse der nacheinander besuchten Regionen wird durch das *RoI*-Konzept ermöglicht und

trägt so zu einer guten Analyse des (un-)typischen räumlichen Besucherverhaltens bei. Es existieren allerdings auch Problematiken bei dieser Methodik, die in Kapitel 8.3 diskutiert werden.

7.3 Unterschiedliche Benutzertypen

Eine weitere eingangs gestellte Frage thematisiert unterschiedliche Nutzungsarten der *WebPark*-Software. Mit SNP Analytics kann dieser Aspekt durch folgende Methodik untersucht werden: Das Finden von Besuchern, welche bestimmte Programmfunktionen überdurchschnittlich oft benutzt haben und die somit ein bestimmter Typ kennzeichnet. Dazu wird hauptsächlich die *Top Besucher*-Ansicht in Kombination mit dem Filter eingesetzt. Im folgenden werden die vier verschiedenen Nutzertypen *Navigator*, *Follower*, *Information seeker* und *Special function fanatic* definiert und jeweils die zugehörigen Wanderer aufgelistet.

Als erstes sollen Besucher gefunden werden, welche als Typ *Navigator* bezeichnet werden können, da sie vornehmlich Orientierungs- und Navigationsfunktionen des Wanderführers nutzen. Hierzu werden im Filter nur solche Einträge zugelassen, deren *Action Code* der Gruppe *Orientation* angehört. Diese vereint verschiedene Funktionen zur Anzeige der Umgebungskarte inklusive der eigenen Position. Anschließend wird der *Top Besucher-View* aufgerufen und der Modus auf *% of all* gestellt. Damit wird der Anteil der gefilterten Einträge an allen Aktionen angezeigt (vgl. Kapitel 6.4). Einige Säulen des Diagrammes verfälschen die Statistik durch eine sehr geringe Gesamtanzahl an getätigten Aktionen. Diese werden nicht berücksichtigt, da sie statistisch nicht signifikant sind. Tabelle 7a zeigt die drei Wanderer mit den meisten Einträgen aus der *Orientation*-Gruppe.

Als ein *Follower* werden hier die Besucher bezeichnet, welche häufig Funktionen der Gruppe *Info on Route* benutzen. Damit verfolgen sie vermutlich regelmäßig den Verlauf ihrer Wanderung auf der aktuellen Route. Tabelle 7b enthält die entsprechenden Ergebnisse.

Wanderer, welche überdurchschnittlich oft Funktionen der Gruppe *Info Around* aufrufen, werden als *Information seeker* bezeichnet, da sie sich permanent über POI bzw. FOI in ihrer unmittelbaren Umgebung informieren. Die Top 3 Besucher dieses Typs sind in Tabelle 7c aufgelistet.

Als vierte und letzte Gruppe wird an dieser Stelle der *Special function fanatic* vorgestellt. Dieser gehören Wanderer an, welche außergewöhnlich oft die Suchfunktion benutzten und/oder viele Geo-Bookmarks und Photos erstellen. Tabelle 7d zeigt die Ergebnisse für diesen Typ.

Neben diesen vier Typen sind beliebige weitere denkbar, die durch Einschränkung bestimmter *Action Codes* im Filter charakterisiert werden können. Die Ergebnisse lassen deutlich erkennen, dass es offensichtlich Besucher des Parks gibt, die ihr Gerät nahezu ausschließlich für bestimmte Funktionen verwenden und sich für die übrigen gar nicht interessieren. Die gefundenen Zahlen können mittels SNP Analytics auch räumlich eingeordnet werden, indem der Filter mit der entsprechenden Besuchernummer auf die *Route2D* oder Routenprofil-Ansicht angewendet wird. Abbildung 32 zeigt beispielsweise die räumliche Verteilung der Nutzung von Besucher 314, dem Top *Navigator* unter den Wanderern in 2007. Er ist fast ausschließlich auf Pfad 8 unterwegs und

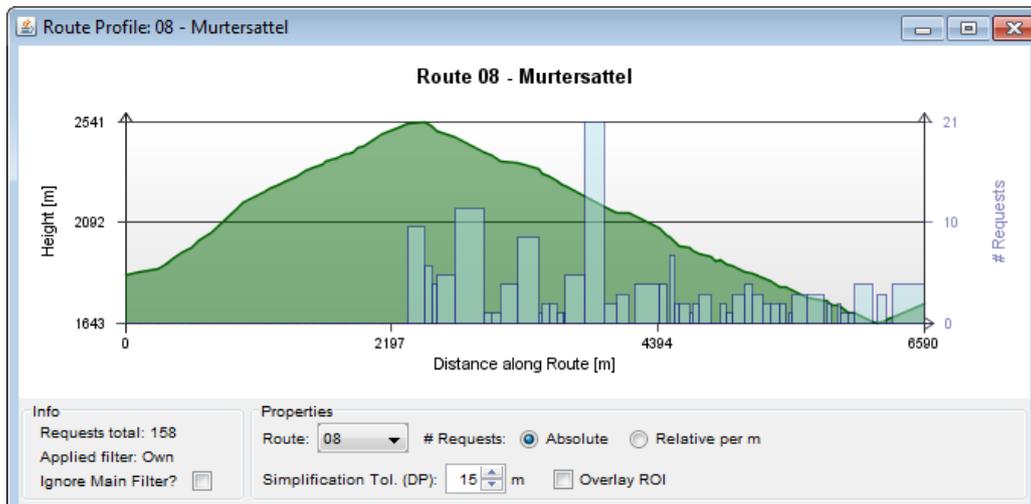


Abbildung 32: Die räumliche Verteilung der Abfragen von Besucher 314.

Besucher #	Anzahl	%
314	168 von 199	84,40
360	61 von 89	68,50
119	94 von 140	67,10
Alle	3842 von 47460	8,10

(a) „Navigator“

Besucher #	Anzahl	%
396	150 von 162	92,60
306	125 von 141	88,70
187	84 von 105	80,00
Alle	8575 von 47460	18,10

(b) „Follower“

Besucher #	Anzahl	%
373	186 von 194	95,90
106	104 von 110	94,50
59	136 von 144	94,40
Alle	10011 von 47460	21,10

(c) „Information seeker“

Besucher #	Anzahl	%
322	149 von 194	76,80
316	32 von 60	53,50
386	118 von 308	38,30
Alle	2409 von 47460	5,10

(d) „Special function fanatic“

Tabelle 7: Vier verschiedene Nutzertypen und ihre Top-Vertreter im Jahr 2007.

benutzt sein Gerät zur Orientierung auf einem Teil der Route gar nicht, auf dem anderen Teil hingegen mit großer Regelmäßigkeit.

7.4 Dauer und Abstand der Gerätenutzung

In diesem Unterkapitel steht die vierte Frage aus der Einleitung im Fokus, welche sich auf die temporalen Eigenschaften der Gerätenutzung bezieht. Zunächst soll hier geklärt werden, ob und inwieweit mit SNP Analytics herausgefunden werden kann, wie lange und in welchen Abständen die Besucher ihr Gerät benutzen. Für diesen Zweck wurde die Analyse der Aktionsketten entworfen, welche mit dem Aufruf der Hauptseite der *WebPark*-Software beginnen und enden. Die Betrachtung der Länge dieser Ketten liefert Informationen über die Nutzungsdauer. Der entsprechende *View* stellt die Häufigkeiten aller *Action Chains* dar, klassifiziert entweder nach der Anzahl der Aktionen in der Kette oder nach der Zeitspanne (vgl. Kapitel 6.4). Bei verschiedenen räumlichen und zeitlichen Einschränkungen des Filters sehen diese Graphiken immer nahezu identisch aus.

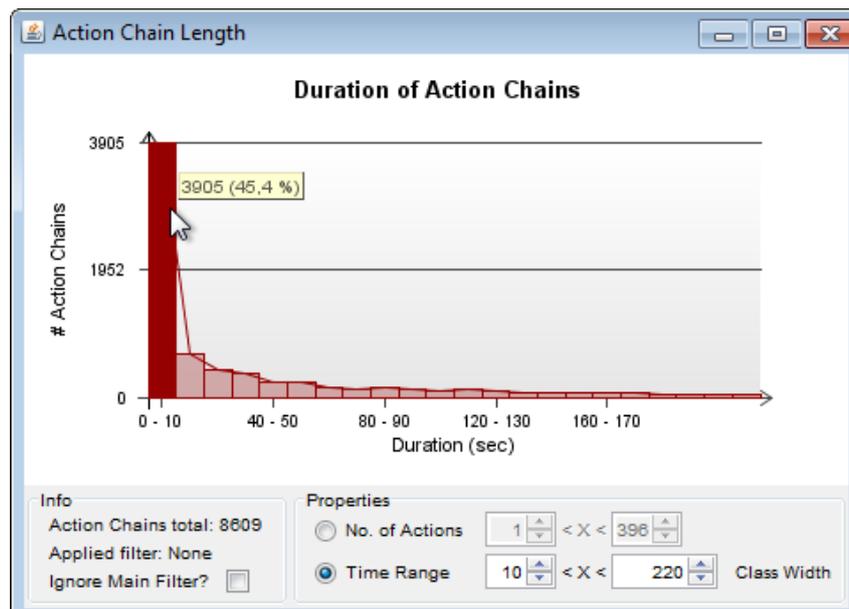


Abbildung 33: Statistik zur Dauer der Aktionsketten. Nahezu die Hälfte aller Gerätenutzungen dauert unter zehn Sekunden.

Abbildung 33 zeigt ein repräsentatives Beispiel, in dem der Großteil der Aktionsketten eine sehr kurze Zeitspanne besitzt und die Häufigkeiten mit längerer Dauer sehr schnell abnehmen und gegen null streben. Mit 45 % haben etwa die Hälfte aller Aktionsketten eine Länge von unter zehn Sekunden. Auch innerhalb der *RoI* ist keine längere Nutzungsdauer feststellbar. Eine entsprechende Filterung ergab keinen signifikanten Unterschied. Die eingangs gestellte Frage ist somit nur teilweise zu beantworten. Die Geräte werden in der Regel nur sehr kurz benutzt und die Anzahl der Nutzungen nimmt mit längerer Dauer stark ab. Dieses Ergebnis kann jedoch nicht näher spezifiziert werden. Die Abstände, in welchen die Besucher ihr Gerät nutzen, können mit SNP Analytics gar nicht erst dargestellt werden. Ebenso sieht die Verteilung bei unterschiedlichen räumlichen Restriktionen, zum Beispiel auf verschiedenen Routen, stets gleich aus. Dies führt zu dem Schluss, dass auch die Korrelation mit der Beschaffenheit der Umgebung nicht oder nur in geringem Maße vorhanden ist. Allerdings kann diese ohnehin schwierig mit der Analysesoftware extrahiert werden, da beispielsweise die Visualisierung der räumlichen Ausdehnung der Aktionsketten nicht möglich ist (siehe Kapitel 8.3).

7.5 Verwendung der Geo-Bookmark Funktion

Die Beantwortung der letzten in Kapitel 1.2 aufgeführten Frage soll in diesem Unterkapitel gefunden werden. Sie beschäftigt sich speziell mit der *Geo-Bookmark*-Funktion der Wanderführer. Diese ist deshalb so bedeutend, weil sie das Verhalten der Besucher in hohem Maße charakterisiert. Räumlicher Kontext und Nutzerinteraktion sind direkt miteinander verbunden. Durch die Texteingabe des Wanderers können Einblicke in die Thematiken gewonnen werden, welche er in Flora und Fauna erlebt und offenbar für so interessant hält, dass er sie abspeichert. Dies

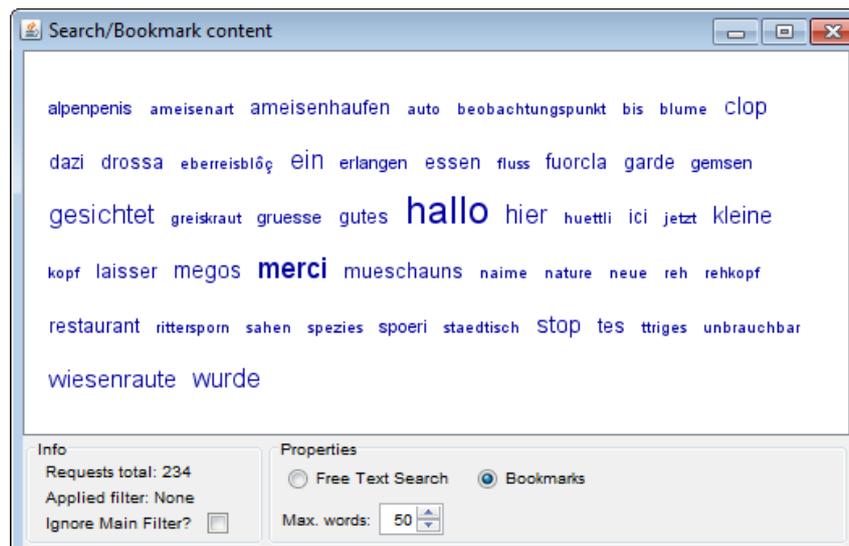


Abbildung 34: *Tag cloud* mit dem Inhalt der *Geo-Bookmarks*. Viele Besucher verwenden die Funktion offenbar nicht, um ernsthafte Informationen über besondere Beobachtungen abzuspeichern.

ermöglicht gemeinsam mit dem räumlichen Kontext eine Ableitung der Themen und Orte, die für den Besucher von besonderem Interesse sind.

Die naheliegendste Analyse besteht in dem Vergleich der Nutzung der *Geo-Bookmark*-Funktion pro Wanderroute. Die Häufigkeiten können mittels der Gesamtnutzungsstatistik pro Route und entsprechender Filtereinstellung schnell abgerufen werden. Die Ergebnisse zeigen nur geringfügige Unterschiede im prozentualen Anteil der *Geo-Bookmark*-Funktion an allen getätigten Aktionen. Der Höchstwert ist mit 5,15 % auf Route 10 zu finden, während Pfad 18 mit 0,77 % das Schlusslicht bildet. Die Speicherung von *Geo-Bookmarks* hängt möglicherweise mit der Anzahl an Aussichtspunkten oder der allgemeinen Gestalt der Route zusammen. Daher könnten die einzelnen Wanderwege in dieser Hinsicht eingestuft werden. Dagegen spricht allerdings der geringe Unterschied in den Ergebnissen und die Tatsache, dass *Geo-Bookmarks* auch in anderen Fällen, wie beispielsweise beim Sichten einer bestimmten Tier- oder Pflanzenart oder beim Überschreiten der Schneefallgrenze, gespeichert werden. Diese Faktoren sind nicht von der Route abhängig. Gleiches gilt ebenfalls für teilweise auftretende Einträge, in denen Besucher den Standort ihres Autos gespeichert haben.

Eine genauere räumliche Analyse lässt ebenfalls kein bestimmtes Muster erkennen. Die *Geo-Bookmark*-Funktion wird nicht verstärkt in *RoI* aufgerufen. Bei genauerer Betrachtung der Verteilung auf den einzelnen Routen kann ebenfalls nicht festgestellt werden, dass beispielsweise bestimmte Aussichtspunkte mit erhöhter Nutzung existieren. Die Aufrufe sind in den meisten Fällen mehr oder weniger gleich verteilt. Ebenso ergibt die unterschiedliche zeitliche Filterung keine sichtlichen Unterschiede. Dies führt zu dem Schluss, dass keine direkte raumzeitliche Verbindung zur Nutzung der *Geo-Bookmark*-Funktion vorhanden ist.

SNP Analytics ermöglicht es mit der Darstellung des Textinhaltes in Form einer *tag cloud* weiterhin, diesen hinsichtlich häufig auftretender Begriffe zu analysieren. Doch auch hier ist das

Ergebnis eher enttäuschend. Im Jahr 2007 treten einzig die beiden Wörter *Hallo* und *Merci* hervor (vgl. Abbildung 34). Die Besucher probieren die Funktion offenbar zum Großteil schlicht „aus Spaß“ aus, speichern aber nur wenig sinnvolle Informationen über ihren momentanen Ort. Mit geringerer Häufigkeit sind in der Wortwolke dennoch einige Tier- und Pflanzennamen zu finden, was zeigt, dass es zumindest einige Wanderer gibt, die ihre Beobachtungen mittels eines *Geo-Bookmarks* der Allgemeinheit zur Verfügung stellen wollen.

7.6 Weitere Ergebnisse

Die vorangegangenen Unterkapitel haben anhand einiger Beispiele gezeigt, dass sich die *high-level* Analysewerkzeuge der Software teils gut und teils weniger gut eignen, um die komplexen Daten hinsichtlich der eingangs gestellten Fragen zu analysieren. In Kapitel 6.1 wurde betont, dass SNP Analytics nicht allein auf die Beantwortung dieser zugeschnitten ist, sondern Möglichkeiten darüber hinaus anbietet. Daher sollen an dieser Stelle beispielhaft einige weitere Ergebnisse vorgestellt werden, welche bei eingehender Beschäftigung mit den einzelnen Komponenten des Programmes gefunden werden können.

Zunächst ist die breite Auswahl an *low level*-Statistiken zu erwähnen, z. B. zur Gesamtnutzung oder zur Verteilung der *Action Codes*. Unter Verwendung des Filters können viele derjenigen Erkenntnisse gewonnen werden, welche bereits in vorhergehenden Arbeiten über die *WebPark*-Daten gemacht wurden. Der Unterschied besteht darin, dass diese in SNP Analytics auf flexiblere Art und Weise zur Verfügung stehen und durch den Filter nach Belieben angepasst werden können: Hinsichtlich der tageszeitlichen Nutzung beispielsweise finden bereits Eisenhut u. a. (2008) heraus, dass die meisten Aktionen am späten Vormittag getätigt werden. In SNP Analytics ist dies durch die entsprechende Gesamtnutzungsstatistik ebenfalls sehr schnell klar. Darüber hinaus kann die tageszeitliche Verteilung unter bestimmten Restriktionen untersucht werden. Ein Ergebnis ist zum Beispiel, dass sich die Verteilungskurve je nach Wanderroute leicht verschiebt, da die Pfade teilweise bevorzugt vormittags oder nachmittags gelaufen werden. Auch die Gesamtverteilung der Häufigkeiten auf die einzelnen Routen ist einfach abrufbar. Sie ergibt gleich der Statistik von Eisenhut u. a. (2008), dass der Wanderführer auf den Pfaden 1, 7, 8 und 17 am häufigsten genutzt wird. Durch das *RoI*-Konzept können darüber hinaus die beliebtesten Plätze im Park abgeleitet werden: dies sind der Startpunkt in Zernez, das Hotel Il Fuorn sowie die Cluozza-Hütte. Die von Snozzi (2009, 58-68) und Haller und Eisenhut (2008, 3) erstellten Statistiken hinsichtlich der Verteilung der *Action Codes* auf den Wanderpfaden sind mit SNP Analytics mittels des entsprechenden *Views* einfach erstellbar und darüber hinaus auf mehr Routen und Daten anwendbar.

Ein weiterer zu untersuchender Aspekt ist die Gesamtnutzung pro Wochentag, welche bislang in diesem Kapitel noch nicht verwendet wurde. Diese liefert jedoch keine interpretierbaren Ergebnisse. Die meisten Aktionen wurden mittwochs, die wenigsten donnerstags gemacht. Die Vermutung einer erhöhten Nutzung am Wochenende bestätigt sich somit nicht.

Der *Route2D-View* bietet mit der Visualisierung der durchschnittlichen Besuchergeschwindigkeit

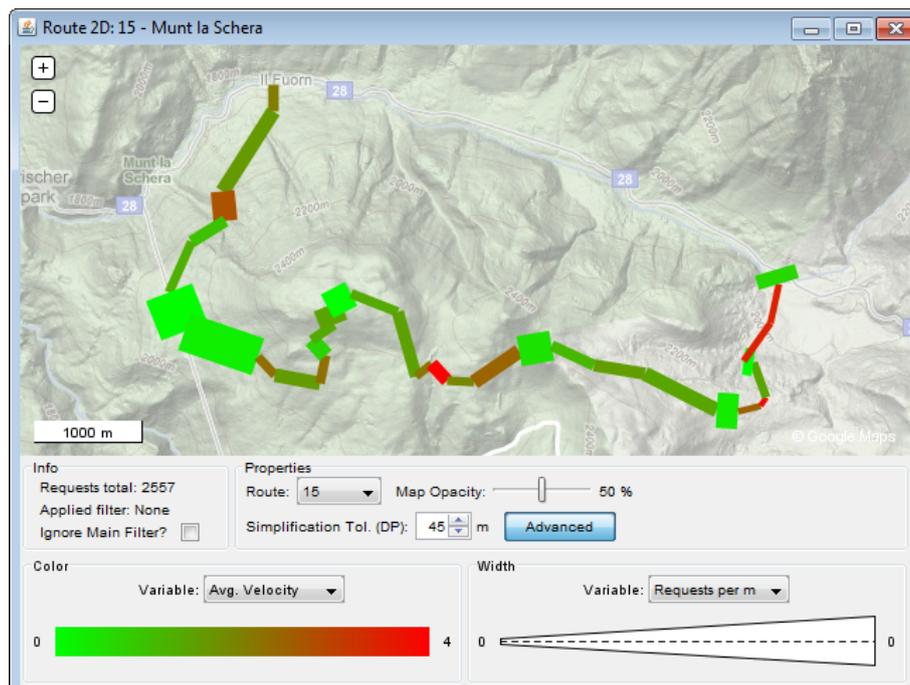


Abbildung 35: Beispiel zur negativen Korrelation von Geschwindigkeit (Farbgradient) und Anzahl der Abfragen (Lineinbreite) auf Route 15.

ten eine interessante Funktion. Durch die graphische Kombination mit der Häufigkeitsverteilung der getätigten Aktionen wird bei einigen Routen deutlich, dass die Geschwindigkeit mit der Anzahl der Aktionen negativ korreliert ist, das heißt der Häufigkeitswert mit höherer Geschwindigkeit abnimmt (vgl. Abbildung 35). Während der Benutzung ihres Wanderführers bewegen sich die Besucher demnach langsamer fort.

Mithilfe der hinterlegten Karte könnte anhand der *Route2D*-Ansicht ebenso der Zusammenhang zwischen Geländeneigung und Geschwindigkeit analysiert werden. Dies führt jedoch nicht zu brauchbaren Ergebnissen, da die Wanderer die Pfade in beide Richtungen ablaufen können. Schnelleres bergab gehen und langsames bergauf gehen gleichen sich folglich aus und es ist kein bestimmtes Muster erkennbar.

Die Aktionsketten in der *WebPark*-Software wurden bislang nur hinsichtlich ihrer Länge untersucht, jedoch nicht in der Abfolge der *Action Codes* an sich. Dies kann, wie in Kapitel 6.4 beschrieben, durch die *Action Chain Analysis*-Ansicht geschehen. Darin finden sich einige Muster der typischen Programmnutzung: Beispielsweise navigiert im Jahr 2007 jeder dritte Besucher von der Hauptseite des Programmes zunächst zu den beliebtesten Anwendungen (*Get Top 20*). Im Anschluss daran werden Informationen über eine der Wanderrouten aufgerufen. Die mit 15 % am zweitmeisten von der Startseite aus genutzte Funktion liefert Informationen über die aktuelle Umgebung (*AroundMe*). Viele der Aktionsketten enthalten überdies Sequenzen mit einem ständigen Wechsel zwischen *Code 701 (Get Top 20)* und *301 (Route information)*. Dies zeigt, dass die Nutzer offensichtlich Informationen verschiedener Routen nacheinander abfragen, wobei sie jeweils zuerst zurück zur vorherigen Seite (*Top 20*) navigieren müssen.

Die in diesem Kapitel vorgestellten Beispiele haben gezeigt, dass die Verbindung des raumzeitlichen Verhaltens der Besucher mit der Benutzung ihres Wanderführers durch SNP Analytics in vielerlei Hinsicht gut analysiert werden kann. Wie zu Beginn des Kapitels erwähnt, sind die Möglichkeiten an dieser Stelle nicht erschöpft und es bestehen vielfältige weitere Aspekte, die potentiell mit der Software untersucht werden können.

8 Diskussion

Der Hauptteil dieser Arbeit wurde in den vorangegangenen Kapiteln mit der Vorverarbeitung der Daten, der Entwicklung von SNP Analytics und schließlich der Vorstellung einiger Analyseergebnisse beschrieben. Dieses Kapitel enthält die kritische Betrachtung dieser drei Bestandteile. Es wird zunächst die Vorverarbeitungsmethodik diskutiert, bevor die Erörterung der Analysesoftware hinsichtlich ihrer Programmierung folgt. Der dritte Teil setzt sich mit der Art und Weise der visuell gestützten Datenanalyse sowie den gewonnenen Erkenntnissen auseinander.

8.1 Vorverarbeitung

Die Vorverarbeitung der Daten ist ein wichtiger Teil dieser Arbeit und ein essentieller Schritt bei jeder Datenanalyse. Sie ist zur Aufbereitung der Rohdaten für SNP Analytics notwendig. Die verwendete Methodik beinhaltet eine Reihe an Konzepten, welche im Folgenden erörtert werden sollen.

In Kapitel 5 wird eine rundum neue Methodik vorgestellt, die speziell auf die Ziele dieser Arbeit ausgerichtet ist. Sie bedient sich nur an wenigen Stellen existierender Ansätze. Dabei sind diese vorhanden: Bereits Snozzi (2009, 32-36) beschreibt eine Vorverarbeitungsmethodik für prinzipiell dieselben Daten, die auch dieser Arbeit zugrunde liegen. Es sind dort einige Ähnlichkeiten zu finden, beispielsweise die Konvertierung der Rohdaten zu einer *ArcGIS Feature Class*, das Löschen ungültiger und doppelter Einträge mittels *Python*-Skripten sowie das Erzeugen von *Sessions*. Dennoch kann dieser Ansatz aus mehreren Gründen nicht für die vorliegende Arbeit übernommen werden: Es werden lediglich die vier populärsten Pfade betrachtet, die räumlich klar voneinander abgegrenzt werden können. Dies ermöglicht die korrekte Zuordnung der Besucheraktionen. Sobald mehr als die vier Pfade berücksichtigt werden, welche im Park teilweise direkt aneinander grenzen, muss eine neue Methodik gefunden werden, um die Einträge den Routen eindeutig zuzuordnen.

In dieser Arbeit ist dies die Projektion der Punkte auf die Routen und die Kernidee der gesamten Vorverarbeitung. Überdies werden die Daten für die Verwendung in SNP Analytics mit zusätzlichen Größen erweitert, beispielsweise den *Subsessions*, der Durchschnittsgeschwindigkeit oder den *Regions of Interest*. Diese sind nicht Bestandteil des existierenden Ansatzes. Snozzi (2009) spricht die Problematik der Überlappungen einiger Wanderwege an, wodurch Einträge zu mehreren Pfaden gehören. Dieses Problem wird hier dadurch umgangen, dass die Geometrie der Pfade anders definiert wird, sodass Wegteile, welche zu zwei oder mehr Routen gehören, nicht existieren. Sofern dies dem Nutzer von SNP Analytics bewusst ist, stellt diese Veränderung kein Problem dar. Es werden keine Pfadsegmente gänzlich gelöscht.

Im weiteren Verlauf dieses Unterkapitels werden die einzelnen Schritte der Vorverarbeitung kritisch betrachtet, um Stärken und Schwächen auszumachen und mögliche Verbesserungen vorzuschlagen.

Der erste Teil, die Konvertierung der Rohdaten, ist zweifellos der größte Schwachpunkt der Verarbeitungskette. Diese ist nicht Teil der vorliegenden Arbeit (vgl. Kapitel 5.2) und somit nicht direkt in die Prozesskette integriert. Es existiert keine einheitliche Methodik zur Erstellung der benötigten *Feature Class* aus den Rohdaten. Das Vorgehen hängt vielmehr individuell von den Daten des jeweiligen Jahres ab und erfordert eine eingängige Beschäftigung mit diesen. Die Rohdaten haben durch die sich jährlich verändernde *WebPark*-Software immer eine andere Struktur. Die Datentabelle muss für die darauf folgenden Schritte jedoch ein einheitliches Format haben. Dies erschwert die Situation zusätzlich. Durch das von Raper entworfene VBA-Skript existiert zwar ein Schritt in Richtung der Automation, dieser ist jedoch nicht ausgereift und erfordert zusätzliche manuelle Bearbeitung der Daten. Falls die Betreiber des Schweizerischen Nationalparks andere als die bereits vorverarbeiteten Daten aus den Jahren 2007 und 2008 für SNP Analytics aufbereiten wollen, stellt diese unübersichtliche Konvertierung sicherlich eine recht große Hemmschwelle dar. Eine mögliche Verbesserung besteht in der Entwicklung eines eigenen Programmes, welches die Rohdaten einliest und als Produkt die fertige *Feature Class* liefert. Bei Unstimmigkeiten aufgrund von veränderter Strukturierung der Rohdaten oder Ähnlichem könnte der Nutzer eingreifen und die nötige Hilfestellung zur Interpretation der Daten geben.

Abgesehen von der Konvertierung existiert für den Rest der Vorverarbeitung ein einheitliches Vorgehen, welches auch für neue Daten kein Problem darstellt, da der Ablauf immer identisch ist. Das erste *Python*-Skript löscht zunächst ungültige und doppelte Einträge. Als ungültig gelten in diesem Fall nur solche Zeilen, die inkorrekt abgespeichert wurden und fehlerhafte Werte enthalten. Bei genauerer Inspektion der Daten fällt jedoch auf, dass weitere verfälschte Einträge existieren, die allerdings nicht bereinigt werden können: In der Datentabelle treten stellenweise große Blöcke von Einträgen auf, welche unterschiedliche Aktionen beinhalten, die alle zur gleichen Zeit am gleichen Ort getätigt wurden. Hier liegt die Vermutung nahe, dass in der *WebPark*-Software bei der Speicherung der Position und des Zeitstempels Fehler aufgetreten sein müssen. Denn ein Besucher kann unmöglich eine solche Vielzahl an Aktionen innerhalb einer Sekunde tätigen. Zum Teil wird somit auch die Qualität der bereinigten Daten immernoch durch *dirty data* negativ beeinflusst.

Darüber hinaus löscht das Skript alle diejenigen Einträge, welche keine Position besitzen und somit räumlich nicht zugeordnet werden können. Diese treten dann verstärkt auf, wenn die Besucher ihr Gerät einschalten und noch keine GPS-Positionslösung vorhanden ist oder sie sich innerhalb von Gebäuden befinden, zum Beispiel im Hotel Il Fuorn oder der Almhütte Cluozza. Der Anteil solcher Einträge ist nicht unerheblich. Insgesamt werden inklusive der ungültigen und doppelten Einträge im Jahr 2007 mit ca. 23 % der 78421 Einträge, 2008 sogar mit etwa 27 % der 101674 Einträge, ein Viertel der Daten aus der Originaltabelle entfernt. Dabei sind Zeilen ohne Position nicht gänzlich unbrauchbar, da sie zumindest noch Informationen über die getätigte Programmaktion enthalten. Für die lückenlose Analyse der *Action Chains* ist die Beibehaltung der Null-Einträge nötig. Geht durch das Löschen der nicht verorteten Einträge somit wertvolle Information verloren? In gewisser Weise ist dies der Fall. Allerdings müssten die meist in Blöcken auftretenden Null-Einträge für die sinnvolle Verwendung einer vorhergehenden oder nachfolgenden Sequenz „vollständiger“ Daten zugeordnet werden. Dies aber stellt sich als

unmöglich heraus. Der Grund dafür ist, dass der Zeitstempel von der GPS-Uhr stammt und folglich ebenfalls fehlt, wenn keine Position verfügbar ist. Ein Null-Block könnte beispielsweise am Ende einer Sequenz beim Eintritt des Besuchers in ein Gebäude auftreten, bei dem das GPS-Signal abreißt. Oder er kennzeichnet den Beginn einer neuen Sequenz, bei der während der ersten Aktionen noch keine Positionslösung vorlag. In den Daten ist dies nicht zu unterscheiden und der Informationsverlust bezüglich der Aktionsketten somit nicht zu vermeiden. Da aber die räumliche Verknüpfung der Einträge in SNP Analytics ohnehin eine wesentlich größere Rolle spielt als die *Action Chains*, ist dieser Verlust nicht allzu kritisch.

Die weitere Vorbereitung der Daten im Verlauf des ersten *Python*-Skripts erweitert die Datentabelle um die in Kapitel 5.3.1 beschriebenen Spalten. Das Erstellen der *Sessions* und somit die Nummerierung der einzelnen Besucher stellt eine nützliche Klassifizierung der großen Datenmenge dar. Durch Verwendung des getrimmten Mittelwertes wird eine realistische Durchschnittsgeschwindigkeit der Besucher berechnet, welche die häufig vorkommenden Ausreißer in den Daten in den meisten Fällen gut eliminiert. Dieser Geschwindigkeitswert ermöglicht das recht sichere Herausfiltern der in Fahrzeugen gemachten Einträge in der weiteren Vorverarbeitung. Außerdem wird er im *Route2D-View* der Analysesoftware verwendet.

An dieser Stelle wäre die Berechnung weitergehender Kennziffern möglich, basierend auf der Orientierung (*heading*) oder der Beschleunigung der Besucher. Eine sehr wertvolle Erweiterung bestünde beispielsweise in der Klassifizierung jedes Eintrages als „statisch“ oder „in Bewegung“. Es wurden Ansätze in Erwägung gezogen, die dafür die *Sinuosität* der Besuchertrajektorie heranziehen. Das ist der Quotient aus tatsächlich zurückgelegter und kürzester Distanz zwischen zwei Punkten. Dies könnte als Maßstab der Zielstrebigkeit oder Unsicherheit der Besucher in ihrer räumlichen Bewegung dienen. Da der Fokus der Arbeit aber nicht ausschließlich auf der Vorverarbeitung liegt, wurden diese Ideen nicht weiter verfolgt.

Die *Subsessions* werden basierend auf einem zeitlichen Mindestabstand aufeinander folgender Einträge definiert. Der Wert kann durch den Nutzer des Skripts festgelegt werden und sollte nicht zu klein sein. Es geht nicht darum, Nutzungsunterbrechungen von fünf oder zehn Minuten voneinander abzutrennen. Die *Subsessions* sollen vielmehr alle Einträge eines ganzen Tages in einige wenige Abschnitte längerer kontinuierlicher Nutzung unterteilen. Daher wird mit 30 Minuten ein etwas größerer Wert gewählt. Letztendlich werden die erzeugten *Subsessions* aber lediglich als Abgrenzung verwendet, um die Geschwindigkeitswerte nicht über einen zu langen Zeitraum hinweg zu mitteln. Auch hier bestehen Möglichkeiten, die *Subsessions* für weitere Berechnungen zu nutzen. Zum Beispiel könnte anhand der mittleren Geschwindigkeit und Sinuosität einer gesamten *Subsession* diese gänzlich als „innerhalb eines Fahrzeuges“ klassifiziert und aus den Daten gelöscht werden. Eine Tabelle mit den mittleren Geschwindigkeiten wird sogar als Nebenprodukt des Skriptes erzeugt. Bei näherer Betrachtung muss diese Idee allerdings verworfen werden. Es existieren viele gemischte *Subsessions*, welche offenbar sowohl Einträge in Fahrzeugen als auch solche zu Fuß enthalten und daher nicht gelöscht werden sollten. Für den Zweck dieser Filterung wird daher jeder Eintrag unabhängig betrachtet und der Schwellwert der maximalen Durchschnittsgeschwindigkeit von 15 km/h verwendet. Diese Methode führt zu guten Ergebnissen (vgl. Kapitel 5.3.2).

Die Projektion der Einträge auf die Wanderwege mittels linearer Bezugssysteme ist das Kernstück der Vorverarbeitung. Die Grundidee besteht darin, jeden Punkt nach Möglichkeit auf eine feste Route zu beziehen. Damit wird das Hintergrundwissen ausgenutzt, dass die Besucher sich, unabhängig von der Streuung ihrer Position, zwingend auf einer der Routen befinden müssen, mit Ausnahme derer, die den Pfad unerlaubt verlassen. Dieses Konzept ist essentiell für die Analyse und die Basis für viele Komponenten in SNP Analytics. Das Erstellen der Gesamtnutzungsstatistiken pro Route und pro *RoI*, das Ableiten der *RoI*-Sequenzen aus den Daten sowie die räumlichen Filterungsmöglichkeiten sind einige Beispiele, die ohne dieses Konzept gar nicht oder nur mit wesentlich höherem Rechenaufwand möglich wären. Der Ansatz beinhaltet jedoch im Wesentlichen zwei Problemfelder:

Eines davon ist die Festlegung des für eine Projektion maximal zulässigen Abstandes der Punkte zur Route. Wie in Kapitel 5.3.3 beschrieben, wird dies iterativ gelöst, indem zunächst alle Punkte mit einer maximalen Entfernung von 150 Meter projiziert werden. Im Anschluss wird eine genauere Auswahl durch Berechnung des Konfidenzintervalls vorgenommen. Obwohl Genauigkeitsangaben des GPS in der Tabellenspalte *Accuracy* in Form des HDOPs vorhanden sind, werden diese nicht in die Berechnung einbezogen. Der Grund dafür ist, dass keine Korrelation zwischen niedriger Genauigkeit und hoher Pfadabweichung festgestellt werden konnte. Es treten ohnehin im Bereich breiter Kreuzungen oder Picknickplätze auch Punkte hoher Genauigkeit weiter abseits des Pfades auf (vgl. Kapitel 5.1). Dennoch sollten diese auf die Route bezogen und somit projiziert werden. Mit dem Konfidenzintervall soll die Streuung der Einträge für jede Route individuell charakterisiert werden. Ist es aber auch tatsächlich für eine dem Pfad angepasste Selektion geeignet? Tabelle 8 zeigt die berechneten Konfidenzintervalle für alle Routen mit den Daten des Jahres 2007:

Route	n	$d[m]$	$s_d[m]$	$K_{95}[m]$
01high – Alp Trupchun	3665	-7, 15	91, 61	[-186, 70; 172, 40]
01low – Alp Trupchun	4251	-5, 39	67, 21	[-137, 13; 126, 34]
02 – Fuorcla Trupchun	2178	-3, 36	20, 58	[-43, 69; 36, 97]
03 – Fuorcla Val Sassa	897	70, 73	85, 71	[-97, 26; 238, 72]
06 – Murtaröl	1961	58, 49	57, 43	[-54, 07; 171, 06]
07 – Chamanna Cluozza	13329	-21, 00	53, 45	[-125, 76; 83, 75]
08 – Murtersattel	8334	1, 83	46, 29	[-88, 89; 92, 56]
09 – Margun Grimmels	1885	19, 75	43, 03	[-64, 59; 104, 10]
10 – Spöltal	2069	-5, 87	44, 04	[-92, 19; 80, 46]
12 – P1 – Grimmels	2814	-1, 41	39, 49	[-78, 81; 75, 98]
13 – Ova Spin – Il Fuorn	3653	13, 77	37, 02	[-58, 79; 86, 32]
15 – Munt la Schera	2923	-6, 61	38, 33	[-81, 74; 68, 52]
17 – Margunet (Naturlehrpfad)	11290	-3, 24	30, 41	[-62, 85; 56, 37]
18 – Fuorcla Val dal Botsch	1913	-20, 01	57, 67	[-133, 05; 93, 03]

Tabelle 8: Konfidenzintervalle der Punktabstände zu den Routen im Jahr 2007.

Das Vorzeichen der Distanzwerte kennzeichnet die Seite, auf der sich der Punkt relativ zum Pfad befindet. Einige der Intervalle schränken den Bereich in der Tat ein, verglichen zu den anfangs gewählten 150 Metern. Im Falle der Routen 2, 8, 10, 12, 15 und 17 liegt zum Beispiel der Mittelwert der Distanzen nahe an null. Außerdem ist die Standardabweichung und folglich

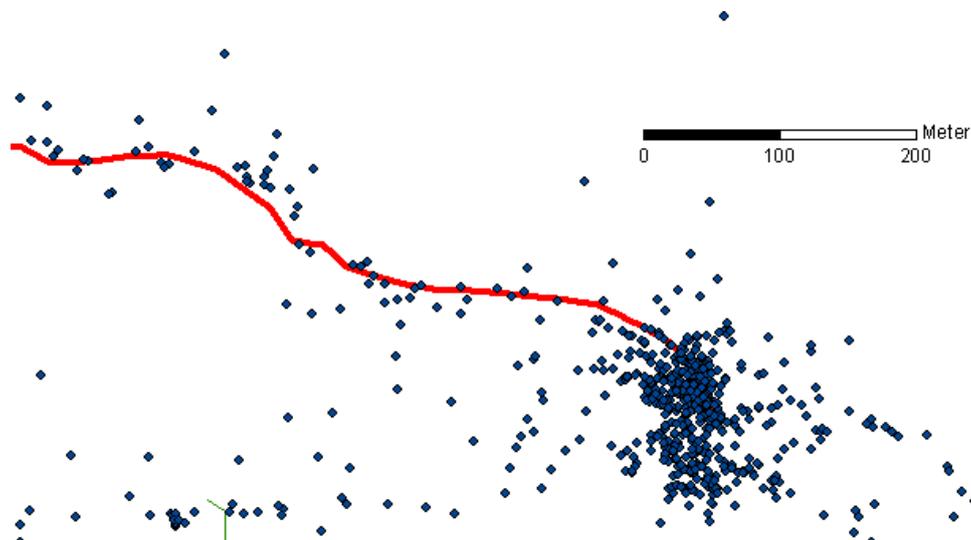


Abbildung 36: Eine abseits des Pfades gelegene Punktanhäufung hat großen Einfluss auf das Konfidenzintervall.

ebenfalls das Intervall relativ klein. Die Beschränkungen liegen dann in einem Bereich, der den bei GPS-Positionierungen erwarteten Abweichungen näher kommt als die sehr großzügig wirkenden 150 Meter. Die Charakteristik der Punktstreuungen wird für diese Pfade durch das Intervall offensichtlich gut ausgedrückt. Im Gegensatz dazu gibt es Routen, bei denen der Mittelwert nicht annähernd bei 0 liegt, die Standardabweichung und damit das berechnete Intervall sehr groß sind. So ergeben sich beispielsweise bei den Routen 3 und 6 Konfidenzbereiche, die sogar über die ursprünglichen 150 Meter hinaus gehen und sehr unrealistisch erscheinen. Eine mögliche Ursache dafür sind Teile der Daten, bei denen alle Einträge konstant um einige Meter zur Route versetzt sind, verursacht möglicherweise durch Mehrwegeeffekte des GPS. Jedoch sind die obigen Abweichungen zu groß, um allein dadurch verursacht worden zu sein. Bei näherer Betrachtung zeigt sich der eigentliche Grund: Die in Bereichen von Picknickplätzen oder Almhütten typischen Anhäufungen von Nutzeraktionen liegen stellenweise etwas abseits des Pfades (vgl. Abbildung 36). Durch die hohe Zahl an Einträgen wird die Statistik dadurch stark in eine Richtung beeinflusst. Solche *Cluster* stellen ohnehin einen Streitpunkt im Zusammenhang linearer Bezugssysteme dar: Diese Datenstruktur ist nicht primär für die Anhäufung vieler Punkte auf einer ähnlichen Position gedacht. Der Messwert entlang der Route ist nach der Projektion bei allen fast identisch und die räumliche Ausdehnung des *Clusters* geht verloren. Einerseits erscheint eine separate Betrachtung dieser Bereiche als punkt- oder kreisförmige Region, welche bei der Berechnung des Konfidenzintervalls aufgrund obiger Verfälschung nicht berücksichtigt werden sollte, sinnvoller. Dies würde andererseits das Grundprinzip der Vorverarbeitung, *jeden* Eintrag einem festen Wanderpfad zuzuordnen, verletzen. Aus diesem Grund wurde eine Art Mittelweg gewählt, welcher die Projektion der Punktanhäufungen zulässt und auch zur Berechnung des Konfidenzintervalls verwendet. Der vorgeschlagenen separaten Betrachtung der *Cluster*regionen wird anschließend durch das Konzept der *Regions of Interest* Rechnung getragen und somit in die Analyse einbezogen.

Das zweite Problemfeld besteht in der Lösung der mehrdeutigen Projektionen, die im Kreuzungsbereich mehrerer Pfade zwangsläufig auftreten. Entsprechend einfacher *Map Matching*-Ansätze wird in dieser Arbeit der Punkt auf die Route projiziert, welche diesem am nächsten liegt. Dies ist offensichtlich nicht immer die richtige Wahl und hat verschiedene Nachteile. Bereits Bernstein und Kornhauser (1996, 7) zeigen anhand von Beispielen, dass dieser Ansatz zu falschen Projektionen führen kann. Es sind kompliziertere Ansätze denkbar, die in erster Linie die individuelle Besuchersequenz bei der Projektion, das bedeutet die zeitlich vorangegangenen und nachfolgenden Einträge der *Session*, berücksichtigen. Abbildung 37 zeigt ein typisches Szenario, in dem der dritte Punkt der Sequenz aufgrund des geringeren Abstandes zur abzweigenden Route auf diese projiziert wird. Die Sequenz zeigt jedoch deutlich, dass der Besucher nicht abgelenkt ist. Auch topologische Restriktionen könnten verwendet werden, sodass unmögliche Projektionen vermieden werden. Ein Beispiel sind Punkte, die sich zwischen dem zwar nahe beieinander liegenden, aber nicht verbundenen hohen und niedrigen Teil des ersten Pfades befinden. Durch das Distanzverfahren „springen“ die Punkte einer Sequenz teilweise zwischen den beiden Teilpfaden. Das ist topologisch gesehen gar nicht möglich, da der Besucher innerhalb kürzester Zeit keinesfalls zwischen diesen beiden Pfaden wechseln kann und aufgrund der fehlenden Verbindung auch nicht darf. Auch hier werden Ideen zur Verbesserung nicht umgesetzt, da das Hauptaugenmerk auf der Entwicklung von SNP Analytics liegt und das angewendete, einfache Verfahren in vielen Fällen zufriedenstellende Ergebnisse liefert. Darüber hinaus ist anzumerken, dass es im Nahbereich von Kreuzungen ohnehin, auch in der Natur, schwierig ist, die Position des Besuchers einem der sich kreuzenden Pfade zuzuordnen, speziell bei unsicherer GPS-Position. Außerdem ist dies nur bis zu einem gewissen Maßstab sinnvoll. In Abbildung 37 geht es lediglich um einige wenige Meter. Eine Differenzierung auf dieser Genauigkeitsstufe ist selbst in der Realität nicht sinnvoll. Allerdings können falsche Projektionen auch in größerem Maßstab auftreten. Folglich bietet sich auch in diesem Fall die Verwendung von *Regions of Interest* an, welche den Kreuzungsbereich separat betrachten. Infolgedessen werden neben den oben beschriebenen Anhäufungen an Picknickplätzen auch die Kreuzungsbereiche als *RoI* definiert.

Am Ende der Vorverarbeitung steht die Datenbankerstellung in *PostgreSQL/PostGIS*, welche bereits in Kapitel 5.4 begründet wird. Für SNP Analytics ist die Verwendung dieser offenen Datenbank im Vergleich zu proprietären *ArcGIS*-Formaten einfacher und schneller. Kritikpunkt ist an dieser Stelle, dass nach *Microsoft Excel* und *ArcGIS* nun das dritte Programm zum Einsatz kommt und die Vorverarbeitung insgesamt sehr kompliziert wird. Dies ist unter dem Gesichtspunkt einer möglichen Durchführung durch die Betreiber des Schweizerischen Nationalparks unangemessen. Für diesen Zweck sollte der Ablauf möglichst einfach gehalten werden. Im Nachhinein ist die Verwendung von *ArcGIS* nicht begründet, da alle darin stattfindenden Prozesse ebenso gut vollständig automatisiert ablaufen könnten und kein Eingreifen des Nutzers erforderlich ist. Die Werkzeuge bezüglich linearer Bezugssysteme sind in *PostGIS* ebenso vorhanden. Es wäre dadurch eine Integration der gesamten Vorverarbeitung in SNP Analytics denkbar. Der einzige Vorteil der gewählten *ArcGIS*-Umgebung besteht darin, dass dem Nutzer jede Einzelheit des Prozesses klar und das Verständnis für die Daten erhöht wird. Außerdem sind die Betreiber des Nationalparks mit dem Umgang dieses Systems vertraut. Ein einziges Programm,

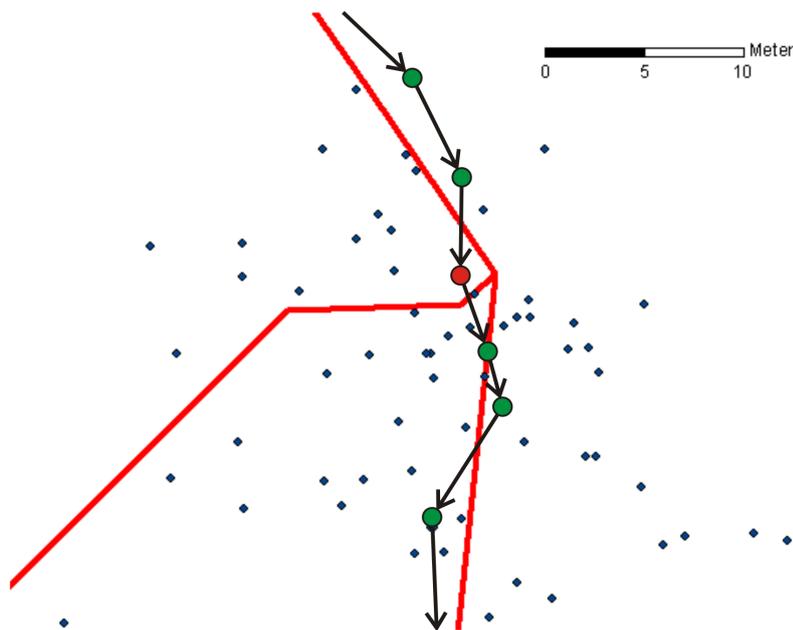


Abbildung 37: Im Kreuzungsbereich treten mehrdeutige Projektionen auf. Die Sequenz zeigt eine Bewegung des Benutzers ausschließlich auf dem rechten Pfad. Durch das Distanzverfahren wird der dritte Sequenzpunkt fälschlicherweise auf den abzweigenden Pfad projiziert.

welches mittels *PostGIS* die gesamte Vorverarbeitung übernimmt, würde hingegen einer *Black Box* gleichen, dessen Inhalt dem Nutzer verschlossen bliebe. Dieser hat aber als primäres Ziel die möglichst schnelle Aufbereitung der Daten zur anschließenden visuellen Exploration und Analyse mit SNP Analytics. Somit überwiegen die Vorteile eines ganzheitlichen, unkomplizierten und vollautomatischen Ansatzes, der in die Analysesoftware integriert werden sollte.

Abschließend stellt sich die Frage, inwieweit die ausführliche Vorverarbeitung der Daten gerechtfertigt ist. Wurde das Hauptziel, eine solide Grundlage für die Analyse zu schaffen, erreicht? Die vorgestellte Methodik geht über bereits existierende Ansätze im Zusammenhang mit den *WebPark*-Daten hinaus und erlaubt im Vergleich zu diesen die Verarbeitung von mehr Daten und die Einbeziehung von mehr Routen. Die zu Beginn teilweise fehlerbehafteten und auf die einzelnen Rohdateien verstreuten Daten werden in einer Tabelle zusammengefügt, welche durch diverse neue Spalten erweitert wird. Dabei kann aber nicht vermieden werden, dass Teile der Daten immernoch *dirty* sind und keine vollständige Bereinigung möglich ist. Dies ist aber bei solch komplexen Daten durchaus normal und eine negative Auswirkung auf die Analyse kann nicht festgestellt werden. Durch die Projektion der Punkte auf die festen Wanderpfade wird Hintergrundwissen eingebracht, welches die bereits in vorhergehenden Arbeiten als Hauptproblem ausgemachte Unsicherheit in den Daten zwar nicht eliminiert, aber dennoch verringert. Einige Glieder der Prozesskette sind noch ausbaufähig und die vorgestellte Strategie kann nicht als ausgereift angesehen werden. Außerdem ist der Ablauf insgesamt, speziell die uneinheitliche Konvertierung der Rohdaten, zu kompliziert für den Anwender, der neue Daten für SNP Analytics aufbereiten möchte. Ein integrierter, automatischer Ansatz, der alle Schritte in einem Programm

vereinigt, wäre eine bessere Lösung. Das Ziel einer soliden Basis wird durch die vorgestellte Methodik dennoch insgesamt erreicht. Viele Teile von SNP Analytics profitieren von dem Konzept der linearen Bezugssysteme sowie das Programm insgesamt von der Art der Datenspeicherung in einer offenen und einfach zugänglichen Datenbank.

8.2 Evaluation von SNP Analytics

Die Entwicklung der Benutzeroberfläche ist zentraler Teil dieser Arbeit. In diesem Unterkapitel wird SNP Analytics bezüglich der Programmierung und der *Usability* evaluiert.

Um das Programm einsatzbereit zu machen, muss zunächst die SNP-Datenbank erstellt werden. Alle nötigen Schritte sind in der Programmhilfe erklärt. Diese beinhaltet ebenfalls einen Abschnitt über die Vorverarbeitung der Daten. Sobald die Verbindung zur *PostgreSQL*-Datenbank erfolgreich hergestellt wurde, steht dem Betrachter eine einfach aufgebaute Oberfläche mit dem noch leeren *Dashboard* zur Verfügung. Alle Funktionen sind dann in der Menüleiste aufrufbar und beliebige *Views* können hinzugefügt werden. Der Nutzer kann auf unkomplizierte Art und Weise jede Ansicht ausprobieren, um herauszufinden, was deren Inhalt ist. Alternativ können in der Hilfe Informationen über die einzelnen *Views* nachgelesen werden.

Ein Ziel von SNP Analytics ist die hohe Interaktivität. Die Programmierung ist deshalb stark auf den Nutzer ausgerichtet und bietet in nahezu jeder Ansicht eine Form von Mausinteraktion. Es können fast immer Teile der Statistik angeklickt werden, die das Öffnen eines neuen *Views* zur Folge haben. Darüber hinaus werden meist Zusatzinformationen angezeigt, wenn der Mauszeiger über die Elemente der Graphik bewegt wird. Der Nutzer kann auf diese Weise interaktiv die einzelnen Funktionen des Programmes erkunden. Bis zu diesem Punkt ist die unkomplizierte Bedienung für jeden beliebigen Nutzer möglich. Jedoch wird sich die Bedeutung der meisten *Views* nur dann erschließen, wenn ein gewisses Verständnis für die *WebPark*-Daten mitgebracht wird. Gleiches gilt für die sinnvolle Nutzung des Filters. SNP Analytics bietet durch ihn die Möglichkeit, komplizierte Sachverhalte zu untersuchen. Allerdings muss der Nutzer dafür auch recht genau vor Augen haben, was er erreichen und in welcher Hinsicht er die Daten analysieren möchte. Einige Szenarien können nur durch recht umständliche Filtereinstellungen umgesetzt werden, wie die Ergebnisse aus Kapitel 7.1 und 7.3 zeigen. Dies zeigt den Charakter von SNP Analytics: Es wird wenig Konkretes vorgegeben, aber vieles ist durch die generische Ausrichtung prinzipiell machbar. Um dennoch den Nutzer besser durch den Analyseprozess zu leiten, wäre die Vorgabe verschiedener Beispielprojekte eine sinnvolle Erweiterung. Er müsste dann nicht „bei Null“ anfangen und der Funktionsumfang des Programmes würde dadurch auf anschauliche Weise demonstriert.

Im weiteren Verlauf werden einige wichtige Konzepte von SNP Analytics kritisch betrachtet. Erstens ist dies der Umgang mit dem Filter. Der Hauptfilter in der linken Seitenleiste ist dafür gedacht, die Daten für mehrere oder alle der existierenden *Views* gleichzeitig zu ändern. Dadurch, dass nahezu jede Ansicht aber diesen ignorieren und darüber hinaus ihren eigenen Filter definieren kann, wird die Situation schnell unübersichtlich und die Rolle des Hauptfilters unbedeutend.

Es sind Szenarien denkbar, in denen jeder *View* einen individuell zugeschnittenen Filter besitzt und eine Veränderung des Hauptfilters überhaupt keinen Einfluss mehr auf das Geschehen hat. Könnte dieser daher ganz abgeschafft und der Filter ausschließlich in den einzelnen Ansichten direkt festgelegt werden? Dann ginge allerdings die oben beschriebene Funktion verloren, die Daten mehrerer Ansichten gleichzeitig mit einem Klick zu ändern, falls dies erforderlich ist. Aus diesem Grund findet der Hauptfilter seine Berechtigung und bleibt Teil der Oberfläche. Zumindest zur Klärung der Unübersichtlichkeit sind jedoch Verbesserungen denkbar. Zum Beispiel eine Art „Filter-Manager“, der auf kompakte Art und Weise die verschiedenen, aktuell verwendeten Filter auflistet und deren Anpassung ermöglicht.

Der Vergleich der Daten unterschiedlicher Jahre ist in SNP Analytics nicht gleichzeitig möglich. Das Programm kann immer nur zu der Datentabelle eines Jahres, nicht aber zu mehreren parallel verbinden. Für einen solchen Vergleich muss zunächst die Verbindung und somit auch alle geöffneten Ansichten geschlossen werden. Erst danach kann eine neue Datengrundlage festgelegt werden. Dies ist sehr umständlich. Dabei ist gerade wenn noch mehr als die momentan bestehenden zwei Jahre zur Datenbank hinzugefügt werden sollten, eine solche Gegenüberstellung interessant. Ein Verbesserungsvorschlag besteht in der Möglichkeit, mehrere Projekte mit unterschiedlichen Datenbankverbindungen nebeneinander öffnen zu können. Das Oberflächendesign würde, ähnlich gängigen Internetbrowsern, mehrere *Tabs* für die Bearbeitung unabhängiger Projekte zur Verfügung stellen.

Die Verbindung zur *PostgreSQL*-Datenbank ist das Herzstück von SNP Analytics. Das Programm liest die Datentabelle nicht in den Speicher ein, sondern belässt sie in der Datenbank und holt sich die jeweils benötigten Teile per SQL-Abfrage. Die Anwendung dieser Methodik hat mehrere Gründe: Erstens ist dies die Datenmenge. Das vollständige Einlesen der großen Tabelle mit allen Feldern belastet den dynamischen Speicher und kann dadurch zur Verlangsamung des Programmes führen. Andererseits nehmen auch manche SQL-Abfragen viel Zeit in Anspruch, wenn diese durch entsprechende Filtereinstellungen sehr kompliziert werden. Dies ist gerade dann zu beobachten, wenn verschiedene Abfragen parallel ausgeführt werden, zum Beispiel beim zeitgleichen Aktualisieren mehrerer Ansichten. Bezüglich der Geschwindigkeit haben demnach beide Methoden Vor- und Nachteile. Der Hauptgrund, der für diesen Ansatz spricht, ist vielmehr die Flexibilität. Die gesamte Selektion und Sortierung der Daten wird durch die Datenbank vorgenommen. Bis auf die SQL-Syntax müssen keine eigenen Algorithmen zur Filterung der Datentabelle entwickelt werden. Diese wären vermutlich wesentlich langsamer in der Ausführung als SQL. Außerdem kann somit auf den großen Funktionsumfang von SQL selbst sowie *PostGIS* zurückgegriffen werden. Folglich stellt diese Variante die deutlich elegantere Lösung im Vergleich zur eigenen Implementierung dar. Ein repräsentatives Beispiel ist der in den Routenprofil- und *Route2D*-Ansichten verwendete *Douglas-Peucker*-Algorithmus. Eine eigene, effiziente Implementierung dieses komplexen Verfahrens zur Verarbeitung der Liniengeometrie ist zweifelsohne schwierig umzusetzen. Daher ist die Nutzung der zur Verfügung stehenden Datenbankfunktionen wesentlich einfacher. Mittels der *PostGIS*-Funktion *st_simplify* wird das Ergebnis sehr effizient berechnet.

In der interaktiven Karte werden dem Nutzer bei Bewegungen des Mauszeigers über die *RoI* die

Hauptflussrichtungen der Besucher durch Anzeige verschiedener Pfeile dargestellt. Diese Visualisierungstechnik erlaubt eine schnelle Erkennung des Besucherverhaltens in der entsprechenden Region. Durch die lediglich zwei verschiedenen Pfeilformen ist sie bewusst einfach gehalten, um den Nutzer nicht durch zu viele verschiedene Darstellungen zu verwirren. Dennoch ist zumindest an den drei kleinen Pfeilen Kritik zu äußern. Diese vermitteln den Eindruck, dass die Besucher des Parks aus vielen verschiedenen Richtungen kommen bzw. gehen. In manchen Fällen trifft dies auch zu, wie bei dem in Kapitel 7.2 gezeigten Beispiel der Region Zernez und Il Fuorn. Allerdings werden solche Pfeile beispielsweise auch dann gezeigt, wenn *zwei* verschiedene vorhergehende bzw. darauffolgende *RoI* mit jeweils 50 % in der Sequenz vertreten sind. Dann ist die gewählte Visualisierung unpassend und vermittelt ein falsches Bild. Daher muss dieses Konzept noch verbessert werden, zum Beispiel durch die Einführung von mehr als der existierenden Pfeilformen bei gleichzeitiger Bewahrung der Übersichtlichkeit.

Die Verwendung der Linienbreite als graphische Variable in der *Route2D*-Ansicht ist von der in Kapitel 3.3.2 erwähnten Arbeit von Huffman (2009) inspiriert. Sie stellt ein gutes Mittel dar, um Häufigkeitsverteilungen entlang von linienhaften Strukturen darzustellen. Im Gegensatz zu der erwähnten Arbeit bietet sich dem Nutzer von SNP Analytics allerdings ein recht eckiges, zerstückeltes Bild. Es ist in manchen Fällen nicht auf Anhieb zu erkennen, dass es die Linienbreite ist, welche variiert. Der Grund dafür ist, dass keine kontinuierliche Variable, wie beispielsweise die Steigung des Pfades, an jeder Stelle desselbigen existiert. Es wird ein Histogramm berechnet, welches immer die Bildung von Klassen erfordert. In diesem Fall sind diese durch die einzeln digitalisierten Segmente der Routengeometrie definiert. Alle auf einem Segment getätigten Aktionen werden jeweils aufsummiert und entsprechend die Linienbreite für den gesamten Abschnitt gesetzt. Je kürzer die Routensegmente, desto detaillierter wird das Bild und desto mehr ähnelt es der „glatten“ Darstellung von Huffman (2009). Allerdings geht damit auch eine Erhöhung der Rechenzeit sowie eine Verkleinerung der Histogrammklassen einher. Letztere führt bei zu kleiner Segmentlänge, wie allgemein bei Histogrammen, zu unbrauchbaren Ergebnissen. Daher ist eine glattere Darstellung in diesem Zusammenhang nicht möglich.

Ähnlich der *Route2D*-Karte basieren auch die Klassen des Histogrammes im Routenprofil-*View* auf dem Detailgrad der Digitalisierung. Die Segmentlängen variieren sehr stark und führen somit zu einem heterogenen Bild, welches die Interpretation für den Betrachter des Histogrammes erschwert. Diese Variante erhält gegenüber einer alternativen und allgemein üblichen Definition gleich großer Klassen den Vorzug, da so eine direkte Vergleichbarkeit mit der *Route2D*-Ansicht möglich ist. Bei letzterer hingegen ist die Berechnung eines gleichabständigen Histogrammes algorithmisch zu aufwändig. Somit ist die Visualisierungstechnik bei beiden *Views* noch als ausbaufähig anzusehen.

Ein Großteil der in Kapitel 7 gezeigten Ergebnisse geht nicht direkt aus den *Views* des Programmes hervor, sondern erfordert die manuelle Erstellung von Tabellen und Listen. Es kann argumentiert werden, dass SNP Analytics zwar die Daten in Form verschiedener Statistiken visualisiert, jedoch kaum numerische Ergebnisse anbietet. Die Datentabellen der Ansichten können nur bei einigen *Views* eingeblendet werden. Dies hat den Grund, dass die Software in erster Linie ein Werkzeug zur *visuellen* Analyse der Daten ist und keine direkten Ergebnisse berechnet. Der

Nutzer soll Muster vorrangig nicht durch Lesen von Tabellen, sondern durch das Betrachten der Graphiken selbst extrahieren. Für die spätere Präsentation der Ergebnisse können *Screenshots* genutzt werden. Nichtsdestotrotz wäre zumindest eine Export-Funktion für die Datentabellen der *Views* eine sinnvolle Erweiterung, um die numerische Auswertung zu vereinfachen.

Die *tag cloud* stellt den textlichen Inhalt der Such- und *Geo-Bookmark*-Funktionen dar. Diese Visualisierungsmethode erlaubt es dem Betrachter, häufig auftretende Begriffe schnell zu erkennen. Bei der Erstellung einer solchen Wolke besteht das Hauptproblem darin, unwichtige Wörter herauszufiltern. Es wird dafür ein Filter verwendet (vgl. Kapitel 6.4), der allerdings noch unzureichend ist. Er legt lediglich fest, dass die Wörter mindestens drei Buchstaben haben müssen. Außerdem wird eine manuell zusammengestellte Liste einiger weniger Wörter, welche nicht berücksichtigt werden sollen, spezifiziert. Dies sind Artikel, Konjunktionen, Präpositionen usw. Da die Besucher des Nationalparks aus verschiedenen Ländern kommen, werden verschiedene Sprachen verwendet, darunter Deutsch, Französisch, Englisch, Italienisch und in der Gegend verbreitete rätoromanische Sprachen. Die Liste der zu filternden Wörter ist daher in hohem Maße unvollständig und die *tag cloud* enthält viele unbrauchbare Begriffe.

Weiterhin existieren diverse Möglichkeiten, die Bedienung und die *Usability* von SNP Analytics im Allgemeinen zu verbessern. Dazu zählt beispielsweise die Bereitstellung einer zusätzlichen Schaltfläche im Filter, die das an- oder abwählen aller Routen/*RoI* erlaubt, da dies manuell recht mühsam ist. Außerdem wäre das Einbinden einer Direkthilfe für jeden *View* sinnvoll, damit der Nutzer nicht erst in der Programmhilfe zu dem entsprechenden Abschnitt navigieren muss. Der Umgang mit den Ansichten auf dem *Dashboard* könnte durch die Bereitstellung von Tastenkombinationen zum einfachen Erstellen, Löschen und Duplizieren von Fenstern verbessert werden.

Überdies gibt es Ideen für weitere Statistiken, welche in Zukunft problemlos eingebunden werden könnten. Dafür ist lediglich das Erstellen einer neuen Subklasse von *MyInternalFrame* erforderlich, die die entsprechende Graphik bereitstellt. Durch dieses modulare Konzept ist die Erweiterbarkeit des Programmes somit gegeben.

Zusammenfassend ist festzustellen, dass SNP Analytics sich größtenteils sehr benutzerfreundlich präsentiert und die Bedienung prinzipiell keine speziellen Vorkenntnisse erfordert. Der Nutzer kann in hohem Maße mit dem Programm interagieren und durch Ausprobieren die einzelnen Komponenten erforschen. Der Filter bietet umfangreiche Möglichkeiten, die Daten in flexibler Art und Weise zu untersuchen. Zur sinnvollen Verwendung wird allerdings eine gute Kenntnis der *WebPark*-Daten benötigt. Ferner ist die Bedienung in manchen Fällen umständlich und kann noch verbessert werden.

8.3 Methodik und Ergebnisse der Analyse

Zu Beginn der Arbeit in Kapitel 1.2 wurde das Ziel ausgegeben, das Besucherverhalten im Nationalpark visuell gestützt zu analysieren. Nachdem die Analysesoftware entwickelt und einige Ergebnisse mit ihr erzielt wurden, soll im Folgenden diskutiert werden, inwieweit dies gelungen

ist. Kann SNP Analytics tatsächlich das Verständnis für die Daten erhöhen und die Extraktion von Mustern unterstützen? Ermöglicht dieses Werkzeug die eingehende visuelle Analyse der Daten? Was ist noch verbesserungswürdig? Wie sind die Ergebnisse aus Kapitel 7 einzustufen? Was für Erkenntnisse wurden hinsichtlich der Daten gewonnen?

Bei der Erläuterung der Grundlagen von *Visual Analytics* in Kapitel 3.3.2 wurden die Visualisierungstools in drei typische Klassen eingeteilt. Bei Typ 1 ist die Interpretation allein dem Betrachter überlassen und der Computer führt keine algorithmische Analyse durch. Typ 3 hingegen kennzeichnet eine umgekehrte Gewichtung, bei welcher dem Betrachter lediglich die vom Rechner extrahierten Muster dargestellt werden. Es kam oben ebenfalls zur Sprache, dass die meisten der *Visual Analytics*-Programme zu Typ 2 gehören und damit zwischen diesen beiden Extremen angesiedelt sind. Dies trifft auch auf den Großteil von SNP Analytics zu. Die meisten *Views* enthalten keine bestimmten Mustererkennungsalgorithmen. Die Daten werden vielmehr in Form verschiedener Statistiken und Graphiken aggregiert und hinsichtlich unterschiedlicher Parameter klassifiziert. Die Extraktion der Muster bleibt dem Betrachter überlassen, der allerdings durch den Filter die Möglichkeit besitzt, die Datengrundlage zu beeinflussen und damit sehr verschiedene Sachverhalte zu analysieren. Die Pfeildarstellung der Besucherbewegungen in der interaktiven Karte ist die einzige Komponente, die als Typ 3 bezeichnet werden kann. Denn bei dieser extrahiert der Computer aus allen Einträgen zunächst die *RoI*-Sequenzen und sucht dann nach einer mehrheitlichen Bewegung der Wanderer. Sofern eine solche vorhanden ist, wird dies dem Nutzer durch den breiten Pfeil gekennzeichnet, andernfalls die drei kleinen Pfeile angezeigt. Der Betrachter bekommt folglich das Muster bereits vorgesetzt und muss selbst keine Analyse mehr vornehmen. Visualisierungen vom ersten Typ hingegen kommen gar nicht vor, obwohl sie an manchen Stellen eine sinnvolle Erweiterung darstellen würden. So wäre zum Beispiel die Darstellung der Sequenzen einzelner Besucher auf der Karte eine hilfreiche Ergänzung. Damit könnten im Einzelfall individuelle Besucherbewegungen einfacher nachvollzogen werden.

Eines der Ziele von SNP Analytics ist es, ein breites Spektrum an einfachen *low-level* Statistiken anzubieten. Dieses Ziel kann als erfüllt angesehen werden. Mittels der Gesamtnutzungsstatistiken und der Aufteilung der *Action Codes* kann sich der Nutzer eine umfassende Übersicht über die Verteilung der Daten verschaffen und ein besseres Verständnis für diese erlangen. Außerdem können, wie in Kapitel 7.6 gezeigt, einige der bereits in vorherigen Arbeiten gemachten Erkenntnisse schnell nachvollzogen werden. Darüber hinaus ermöglichen die Routenprofil- und *Route2D*-Ansichten zusammen mit den Filterungsmöglichkeiten eine umfangreiche räumliche Verknüpfung der Daten.

Eine weitere Zielsetzung ist die Bereitstellung einiger spezieller *high-level* Analyseaufgaben, um unter anderem die eingangs gestellten Fragen näher zu untersuchen. Als erstes ist in diesem Zusammenhang festzuhalten, dass bereits die einfach erscheinenden *low-level*-Statistiken durch die umfangreichen Filterungsmöglichkeiten ebenfalls als Werkzeuge für *high-level*-Analysen fungieren können. Ein typisches Beispiel ist der *Top Besucher-View*. Er stellt anfangs ein einfaches Säulendiagramm dar, welches die Nutzer mit der höchsten Anzahl an Einträgen beinhaltet. Kapitel 7.3 demonstriert, wie anschließend durch geschickte Filterung die verschiedenen Nutzertypen extrahiert werden können. SNP Analytics ermöglicht nicht nur die numerische Beschreibung

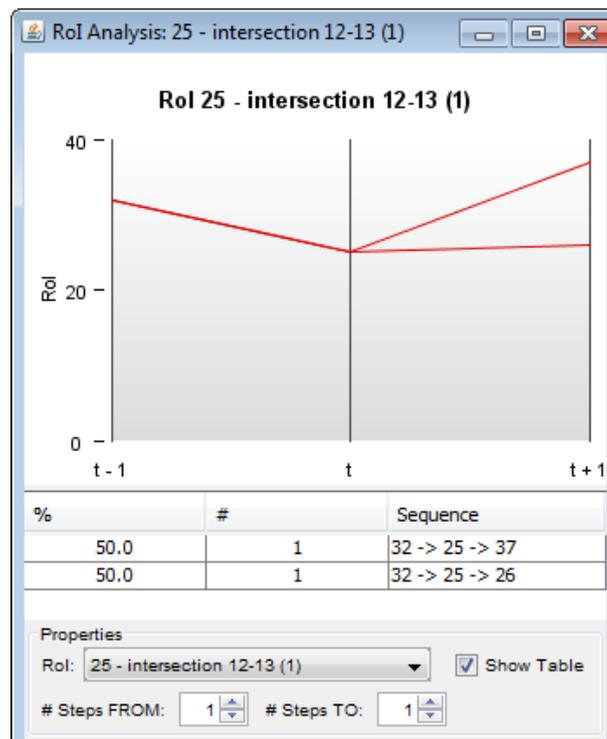


Abbildung 38: Problem der geringen Grundgesamtheit bei der Analyse der *RoI*-Sequenzen.

dieser Typen, sondern mithilfe der entsprechenden *Views* auch, die räumliche Verbindung zu diesen Nutzungsarten herzustellen. Ein anderes Beispiel ist die *Route2D*-Karte, welche durch die Verwendung der beiden graphischen Variable Linienbreite und Farbe sowohl einfache als auch komplexe Sachverhalte darstellen kann und dem Nutzer dadurch die Extraktion von interessanten Mustern ermöglicht.

Für weitere *high-level*-Analysen stehen darüber hinaus einige *Views* zur Verfügung, die speziell auf eine Aufgabe zugeschnitten sind. Als erstes ist die *RoI*-Analyse zu nennen. Durch das *RoI*-Konzept im Allgemeinen wird Hintergrundwissen über die interessanten Orte im Park sinnvoll ausgenutzt. Die ursprüngliche Abfolge von Koordinaten im Schweizer Landeskoordinatensystem wird durch eine Sequenz nacheinander besuchter Regionen ersetzt. Das ermöglicht auf relativ einfachem Wege die Analyse von (un-)typischem Besucherverhalten. Die Ergebnisse aus Kapitel 7.2 zeigen, dass die von den Wanderern hauptsächlich gewählten Pfade an Kreuzungen oder die bevorzugte Richtung, die Routen zu durchlaufen, gut erkannt werden können.

Sie täuschen allerdings über ein existierendes Problem in diesem Zusammenhang hinweg: Bei näherer Betrachtung der Datentabellen in der *RoI*-Analyse wird deutlich, dass die Grundgesamtheit oft sehr gering ist. Abbildung 38 zeigt ein repräsentatives Beispiel einer der vielen im Park vorhandenen Regionen, bei denen nur wenige bis gar keine Sequenzen gefunden werden können, vor allem wenn die Länge der Sequenz durch den Nutzer erhöht wird. Trotz insgesamt über 40000 Einträgen und mehr als 400 verschiedener Besucher (in 2007) treten in den gefundenen Sequenzen sehr geringe Häufigkeiten auf. Das ist insofern kritisch, da es bei dieser Analyse gerade darum geht, das Verhalten der Mehrheit der Besucher zu extrahieren. Bei Grundgesamtheiten

von beispielsweise nur fünf oder zehn Sequenzen ist die statistische Signifikanz nicht annähernd gegeben. Was ist die Ursache dieses Problems? Der Grund liegt in einer allgemeinen Charakteristik der Daten, die eine bessere raumzeitliche Analyse, nicht nur in diesem Zusammenhang, verbietet: Die Positionen der Wanderer werden nicht kontinuierlich in festen Zeitabständen aufgezeichnet, sondern lediglich dann, wenn der Besucher sein Gerät benutzt. In den seltensten Fällen ergibt sich dadurch eine regelmäßige Trajektorie. Es werden offensichtlich nicht wenige *RoI* vom Wanderer durchlaufen, in denen er sein Gerät nicht benutzt. Somit gibt es in der Realität zweifelsohne mehr als die wenigen gefundenen Sequenzen. Jedoch können sie nicht nachgewiesen werden, da die meisten Daten in dieser Hinsicht nicht *vollständig* sind.

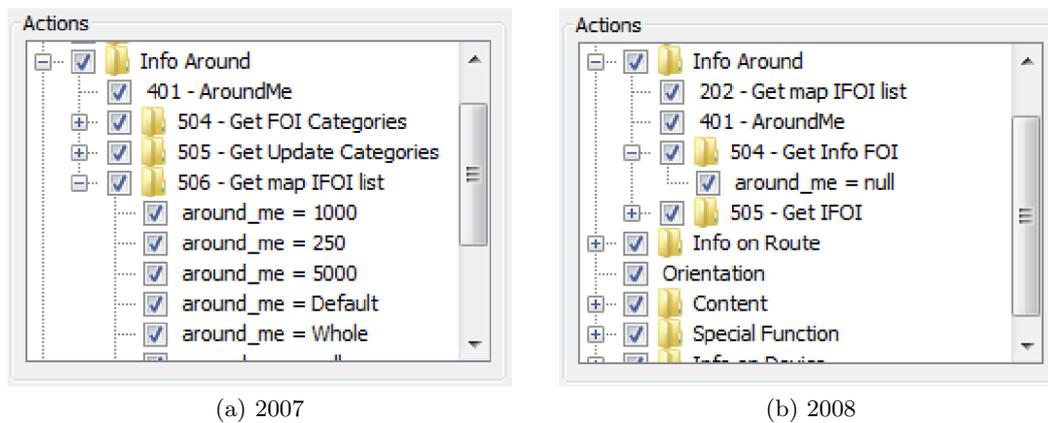
In dieser Arbeit werden zur Lösung des Problems räumliche Zwangsbedingungen eingeführt und damit versucht, die Datengrundlage künstlich zu vergrößern. Das bedeutet, dass in den untersuchten Sequenzen gegebenenfalls *RoI* ergänzt werden, welche der Besucher zwangsläufig passiert haben *muss*. Ein Beispiel ist die Sequenz 6–13, welche durch SNP Analytics durch 6–33–13 ersetzt wird. Der Wanderer muss auf dem Weg von Picknickplatz # 6 zu # 13 die *RoI* der Cluozza-Hütte passiert haben, auch wenn er sein Gerät dort offensichtlich nicht verwendet hat. Dies verbessert die Statistik allerdings nur unwesentlich. Außerdem können diese Zwangsbedingungen nicht immer angebracht werden, da an manchen Stellen des Parks nicht mit Bestimmtheit gesagt werden kann, wie sich der Besucher genau bewegt hat. Ein Beispiel ist die Teilsequenz 0–30. Es ist nicht klar, über welchen Teil des ersten Pfades der Besucher gewandert ist, solange er sich nicht durch die Benutzung des Gerätes innerhalb der Region 34 „verräät“. Hier schließt sich ein Aspekt an, der die mangelnde Ausnutzung der vorhandenen Informationen *zwischen* den *RoI* kritisiert. Es werden im Rahmen der *RoI*-Analyse lediglich die Sequenzen der besuchten Regionen betrachtet, nicht aber die Bewegungen der Besucher zwischen ihnen. Beispielsweise könnte das zuletzt genannte Beispiel einfach gelöst werden, indem untersucht wird, auf welchem der beiden Teilpfade der Wanderer sich in der Zwischenzeit befindet. Die *RoI*-Analyse ermöglicht somit zwar die genaue Untersuchung des (un-)typischen räumlichen Besucherverhaltens. Bei näherem Hinsehen stellt man aber fest, dass sich die Daten zum einen nur bedingt dafür eignen und zum anderen die Methodik noch verbessert werden könnte.

Die Beantwortung der vierten Frage aus der Einleitung bezüglich der Aktionsketten kann mit SNP Analytics nur teilweise erfolgen (vgl. Kapitel 7.4). Allerdings zeigen die Statistiken zur Länge der *Action Chains* in der Software auch, dass diesbezüglich grundsätzlich nur wenig Variation in den Daten vorhanden ist. Die detaillierte Analyse der Aktionsketten lässt häufig verwendete Abfolgen von Programmaufrufen der Besucher gut erkennen. Die Grundgesamtheit ist hier, anders als bei der *RoI*-Analyse, groß genug. Auffällig ist das häufige Auftreten von Teilsequenzen, welche ausschließlich aus dem wiederholten Aufruf desselben *Codes* bestehen. Die Entstehung dieser ist auf zwei Ursachen zurückzuführen: Erstens durch *dirty data*, das heißt der fälschlicherweise wiederholten Abspeicherung derselben Aktion, obwohl sie nur einmal getätigt wurde. Zweitens durch die in Kapitel 2.3.2 erwähnten *Push*-Dienste der *WebPark*-Software. Diese rufen Funktionen unter bestimmten Bedingungen ohne direkte Aufforderung des Nutzers automatisch auf, zum Beispiel die ständige Aktualisierung der Umgebungskarte bei Fortbewegung des Besuchers. Die räumliche Verknüpfung der *Action Chains* fehlt völlig. Die Untersuchung der Ausdehnung

der Aktionsketten oder des räumlichen Nutzungsabstandes ist folglich nicht möglich. Die Ursache dafür liegt wiederum darin, dass die Daten es nicht erlauben. Folgendes Beispiel verdeutlicht das Problem: Angenommen, das erste Glied einer Aktionskette findet 45 Minuten später an einem ganz anderen Ort des Parks statt als das letzte der vorhergehenden Kette. Der Wanderer könnte das Gerät nun entweder unmittelbar nach der letzten aufgezeichneten Aktion ausgeschaltet und es eine Dreiviertelstunde nicht benutzt haben. In diesem Fall wäre die Aktionskette mit der letzten Aktion beendet. Oder er hat die aufgerufene Seite als dauerhafte Information für 45 Minuten geöffnet und setzt den Wanderführer dann auf die Startseite zurück. Die Aktionskette kann erst zu dem späteren Zeitpunkt als beendet angesehen werden. Im letzteren Fall wird das Gerät somit noch länger benutzt, aber keine neue Aktion damit ausgeführt, sodass kein neuer Eintrag in den Daten zu finden ist. Welche Situation vorliegt, kann aus den Daten nicht extrahiert werden.

Dieser Mangel an Informationen in den Daten ist ebenso der Grund dafür, warum SNP Analytics die zeitliche Komponente allgemein nur sehr wenig berücksichtigt. Schließlich ist dies Bestandteil der Hauptfrage aus Kapitel 1.2, die Verbindung der Programmaktionen mit dem *raumzeitlichen* Verhalten der Wanderer. Die Zeit kann lediglich in größerem Maßstab durch den Filter berücksichtigt werden, beispielsweise durch Einschränkungen auf bestimmte Wochen oder Tageszeiten. Genauere zeitliche Analysen der Besucherbewegung sind nicht durchführbar: Zum Beispiel die benötigte Zeit zwischen zwei *RoI* oder die Aufenthaltsdauer auf einem Picknickplatz. Diese Analysen sind zum Teil auch in existierenden Arbeiten zu finden, aber streng genommen mit den Daten nur in Ausnahmefällen möglich. Nur wenn eine kontinuierliche Nutzungssequenz eines Besuchers vorliegt, kann die Aufenthaltszeit auf einem Picknickplatz berechnet werden. Der Wanderer muss das Gerät vor seiner Rast bis in die *RoI* hinein benutzen und es nach der Pause ebenso direkt weiter verwenden. Dann ergibt die Berechnung der Zeitspanne der innerhalb der *RoI* getätigten Aktionen wirklich die wahre Aufenthaltszeit. Dieses Szenario ist jedoch äußerst selten. Das typische Verhalten ist eher dadurch gekennzeichnet, dass ein Wanderer beispielsweise vor und während der Rast sein Gerät kontinuierlich benutzt hat, es danach aber erst später irgendwo auf dem Wanderpfad reaktiviert. Die Bestimmung der Aufenthaltszeit ist unmöglich. Gleiches gilt für die Reisezeit zwischen zwei *RoI*. Durch diese Einschränkung sind zeitliche Analysen allgemein nicht oder nur sehr unzuverlässig möglich.

Der Großteil der Ergebnisse aus Kapitel 7 wurde mit den Daten von 2007 gemacht, obwohl die Datenmenge in 2008 größer ist. Der Grund dafür liegt erstens darin, dass die Konvertierung der Rohdaten des Jahres 2008 erst zu einem späten Zeitpunkt der Bearbeitungszeit abgeschlossen war. Zweitens wird ein Unterschied in den vorhandenen *Action Codes* deutlich (vgl. Tabellen im Anhang). Die Struktur der *WebPark*-Software wurde vom Entwickler Camineo von 2007 auf 2008 offensichtlich deutlich verändert, sodass sich bei der Verteilung der *Action Codes* ein gänzlich verändertes Bild ergibt. Die *Codes* sind zum Teil anders benannt, es wurden ganze Funktionen gestrichen und andere hinzugefügt. Die Gruppe *Orientation* enthält sogar gar keine *Codes* mehr. Es wird im Jahr 2008 viel höherer Wert auf das Konzept der *Features of Interest (FOI)* gelegt. Die Funktion *Get Info FOI* macht mehr als ein Drittel aller getätigten Aktionen aus, wohingegen dies im Jahr 2007 nur etwa 10 % waren. Im Jahr 2008 fehlen scheinbar wichtige Aktionen wie *WhereAmI* oder *Route information*. Es kann jedoch davon ausgegangen werden,



(a) 2007

(b) 2008

Abbildung 39: Unterschiede in der Struktur der *WebPark*-Software und somit auch in den *Action Codes*.

dass diese lediglich anders strukturiert und in anderen *Codes* „versteckt“ sind. Dies führt zu dem Schluss, dass SNP Analytics zwar ohne Probleme mit den 2008er Daten umgehen kann, einzelne Komponenten des Programmes aber auf die Daten von 2007 abgestimmt sind. Ein Beispiel ist das Konzept der dritten Bauebene im Filter: In 2007 kann zum einen die Route, über die Informationen angefordert wurden ($route = \dots$), zum anderen der Radius, in dem der Besucher um sich herum nach *FOI* sucht ($around_me = \dots$), festgelegt werden. Ersteres ist auch in 2008 möglich. Die Spezifikation des Suchradiuses mit *AroundMe* hingegen ist offensichtlich nicht mehr Teil des Programmes oder in einer anderen Funktion untergebracht (vgl. Abbildung 39). SNP Analytics passt sich an dieser Stelle nicht ausreichend an die veränderte Software an, das Verständnis für die internen Abläufe in den *WebPark*-Anwendungen der verschiedenen Jahre ist zu wenig vorhanden. Für das Konzept der *FOI* bietet SNP Analytics allgemein mangelnde Unterstützung. Denn die *Features of Interest* besitzen eine eindeutige Nummer, welche die genaue Bedeutung entschlüsselt. Die Beschreibung für jede dieser IDs ist von Camineo dokumentiert. Demnach könnte näher spezifiziert werden, welches *Feature of Interest* der Besucher abfragt und was dieses bedeutet. Ähnlich der Routeninformation und des *AroundMe*-Radiuses wäre eine dritte Bauebene auch für die Art des *FOI* eine sinnvolle Erweiterung.

Viele der in Kapitel 3.3.2 angedeuteten Typ 2-Programme verwenden irgendeine Art innovativer und gleichzeitig sehr komplexer Visualisierungsmethoden wie beispielsweise *Treemaps* oder *Growth Ring Maps*. SNP Analytics arbeitet hingegen zu einem Großteil mit einfachen Statistiken. Eine Ausnahme bilden die Parallelkoordinatensysteme in der *RoI*- und *Action Chain*-Analyse sowie die Nutzung der Linienbreite in der *Route2D*-Ansicht. Die Besonderheit bei der entwickelten Analysesoftware liegt darin, dass die einfachen Statistiken für sehr unterschiedliche Zwecke eingesetzt und durch den Datenfilter auf äußerst flexible Art und Weise angepasst werden können. Somit sind auch komplizierte Muster aus ihnen extrahierbar.

Trotz der genannten Verbesserungsmöglichkeiten kann der Schluss gezogen werden, dass mit SNP Analytics ein Werkzeug geschaffen wurde, welches die *WebPark*-Daten in einer Art und Weise auswertet, wie es bislang noch nicht geschehen ist. Die Ziele aus Kapitel 6.1 wurden größtenteils erfüllt und es ist ein äußerst flexibles Programm entstanden, welches von einfachen Nutzungssta-

tistiken bis hin zu komplizierten, detailreichen Analysen viele verschiedene Möglichkeiten bietet. Dies wird in erster Linie durch die umfangreiche Datenfilterung erreicht, mit der nahezu beliebige Teile der Daten ausgewählt werden können. Die letzte der in Kapitel 1.2 gestellten Fragen bezüglich dem Nutzen eines solchen Visualisierungswerkzeuges, kann bejaht werden. Der Umgang mit SNP Analytics erhöht durchaus das Verständnis für die Natur der *WebPark*-Daten und ermöglicht die Extraktion von Verhaltensmustern.

Weiterhin wurde in der Einleitung die Frage gestellt, inwieweit die Analysesoftware generisch ist. Am Ende von Kapitel 3.3.2 wurde im Zuge dieses Aspektes erwähnt, dass die mangelnde Wiederverwendbarkeit und Übertragbarkeit auf andere Datensätze in der Regel das Hauptproblem der derzeitigen Visualisierungstools ist. Hinsichtlich dieser im Bereich von Geodaten sehr schwierigen Thematik kann auch SNP Analytics keine wirklich neuen Erkenntnisse liefern. Das Tool ist nur im Kontext der *WebPark*-Daten insofern als generisch zu bezeichnen, dass es die Daten verschiedener Jahre analysieren kann und dabei keine feste Methode vorschreibt. Auf andere Datensätze ist es selbstverständlich nicht anwendbar. Wie bei jedem anderen Werkzeug dieser Art ist die Ursache dafür die folgende: Geodaten können sehr verschieden sein und jeder Datensatz erfordert eine individuell angepassten Ansatz. Dennoch werden zumindest auf der Ebene der verwendeten Konzepte aus anderen Arbeiten existierende Methoden eingearbeitet: So zum Beispiel das von Ashbrook und Starner (2003) und Giannotti u. a. (2007) inspirierte Konzept der *Regions of Interest*, welches bei jenen Arbeiten in einem anderen Kontext zum Einsatz kam. Im Verlaufe der Arbeit wurden auch Erkenntnisse über die Daten an sich gewonnen. So reichhaltig und einzigartig diese auch sind, sind sie zumindest für eine präzise räumliche und zeitliche Analyse teilweise ungeeignet, da keine kontinuierlichen Trajektorien vorliegen. Der Hauptaspekt der Arbeit, das raumzeitliche Verhalten mit den Abfragen der Besucher in Verbindung zu bringen, wird durch SNP Analytics insgesamt eingängig behandelt und es konnten einige interessante Ergebnisse gefunden werden. Dabei werden jedoch weniger die individuellen Besuchersequenzen betrachtet, als vielmehr die raumzeitliche Verknüpfung der getätigten Aktionen allgemein. Die Ergebnisse aus Kapitel 7.1 haben aber gezeigt, dass mittels geschickter Filtereinstellungen auch die Verhaltensweisen einzelner Besucher untersucht werden können. Hinsichtlich der direkten Beeinflussung des Bewegungsverhaltens durch den Wanderführer sind dennoch weitere Analysen denkbar, die von der Software mehr Funktionen in Richtung individueller Sequenzen verlangt.

9 Zusammenfassung und Ausblick

In dieser Arbeit wurde eine neue Methode zur Untersuchung des Besucherverhaltens im Schweizerischen Nationalpark entwickelt. Die oberflächenbasierte Software SNP Analytics ermöglicht die visuell gestützte Analyse der *WebPark*-Daten. Das raumzeitliche Verhalten der Wanderer kann damit in einem Umfang erforscht werden, der über die bisherigen Arbeiten in diesem Kontext hinaus geht.

Ein wichtiger Teil der Arbeit besteht in der Vorverarbeitung der Daten. Es wird eine komplexe Strategie vorgestellt, wie die Rohdaten aufbereitet, verbessert und zusammengefasst werden können. Das Kernstück bildet dabei der räumliche Bezug jeder getätigten Besucheraktion auf eine der festen Wanderrouen. Die einzelnen durch GPS ermittelten Positionen streuen teilweise stark um die Pfade. Anstatt der ursprünglichen Beschreibung durch Koordinaten im Schweizerischen Landessystem werden lineare Bezugssysteme eingesetzt. Jeder Punkt wird senkrecht auf die nächstgelegene Route projiziert und seine Position mittels der Entfernung entlang des Pfades ausgedrückt. Mit dieser Methodik konnten die auftretenden Positionsunsicherheiten verringert werden. Die Daten jeweils eines Jahres werden auf kompakte Art und Weise in einer offenen *PostGIS*-Datenbank gespeichert. Es wird somit eine solide Basis für die Analyse geschaffen, welche an vielen Stellen von ihr profitiert.

Die anschließend entwickelte Analysesoftware SNP Analytics zeichnet sich durch einen hohen Grad an Interaktivität und Flexibilität aus. Sie stellt verschiedene Statistiken zur vielseitigen Analyse bereit, die der Nutzer frei wählen und auf der Oberfläche zusammenstellen kann. Diese beinhalten eine breite Auswahl an einfachen Graphiken zur Gesamtnutzung des Wanderführers. Überdies stehen einige spezielle Statistiken zur Verfügung, mit denen die Untersuchung des raumzeitlichen Verhaltens der Besucher genauer möglich ist. Das Programm beruht auf einer permanenten Verbindung zur Datenbank und greift auf diese mittels SQL-Abfragen zu. Der umfangreiche Filter erlaubt es dem Nutzer, die zugrunde liegenden Daten räumlich, zeitlich und bezüglich der von den Besuchern getätigten Aktionen mit der *WebPark*-Software beliebig einzuschränken. Durch das modulare Konzept ist die Möglichkeit der Erweiterung des Programmes mit neuen Statistiken gegeben. Das Ziel der Arbeit war es, ein Werkzeug zur Visualisierung zu schaffen, welches das Verständnis für die komplexen Daten erhöht und die visuelle Extraktion von Verhaltensmustern ermöglicht. Dies ist mit SNP Analytics gelungen.

Es wurden im weiteren Verlauf der Arbeit einige beispielhafte Analysen durchgeführt. Mit diesen konnten hinsichtlich der Verknüpfung des raumzeitlichen Verhaltens der Besucher mit der Verwendung ihres digitalen Wanderführers und der übrigen eingangs gestellten Forschungsfragen einige interessante Muster gefunden werden. Das Konzept der *Regions of Interest* ermöglicht eine detailreiche Analyse von (un-)typischem räumlichen Verhalten der Besucher. Beispielsweise konnte der Sonderstatus der Region Zernez als Ausgangspunkt vieler Touren und des Hotels Il Fuorn als Endpunkt vieler Wanderungen gefunden werden. Darüber hinaus wurden anhand häufig genutzter Programmfunktionen verschiedene Besuchertypen identifiziert, welche ihr Gerät auf sehr unterschiedliche Art und Weise benutzen. Weiterhin wurde deutlich, dass die Verwendung des Wanderführers mehrheitlich durch eine sehr kurze Nutzungsdauer von nur einigen

Sekunden geprägt ist. Bezüglich der *Geo-Bookmark*-Funktion konnten keine räumlichen Präferenzen festgestellt werden. Der Inhalt dieser zeigt, dass viele Besucher die Funktion nicht für die Speicherung interessanter Beobachtungen, sondern lediglich aus Neugierde verwenden. Mit der Analysesoftware wurden auch weitere, über die zu Beginn der Arbeit gestellten Fragen hinausgehende Verhaltensmuster gefunden. Beispielsweise wird auf einigen Routen deutlich, dass die Geschwindigkeit der Besucher mit der Anzahl der getätigten Aktionen mit ihrem Wanderführer negativ korreliert.

Neben den genannten Stärken können jedoch sowohl bezüglich der Vorverarbeitung als auch der Analysesoftware gleichwohl einige Kritikpunkte geäußert werden, bei denen diese Arbeit an ihre Grenzen stößt.

Die in der Vorverarbeitung angewendete Projektion auf die Routen ist nicht in allen Bereichen des Nationalparks trivial. So müssen Strategien für die Lösung mehrdeutiger Projektionen an Wegkreuzungen oder die separate Betrachtung von *Clustern* entwickelt werden. Dies wird durch recht einfache Ansätze erreicht, welche nicht immer optimale Ergebnisse liefern und durchaus noch Verbesserungspotential besitzen. Des Weiteren ist der Prozess der Vorverarbeitung sehr kompliziert und erfordert viele verschiedene Schritte. Speziell die Konvertierung der Rohdaten ist nicht ausgereift und bietet kein durchgehend automatisiertes Verfahren. Ein ganzheitlicher, automatischer Ansatz ist hier erstrebenswert. Wenn die Betreiber des Nationalparks in Zukunft eigene Datensätze verarbeiten, ist eine einfache Prozessierung besonders wichtig.

SNP Analytics kann im Zusammenhang mit den *WebPark*-Daten als generisch bezeichnet werden, da es keinen festen Analyseweg vorgibt und durch den Filter sehr flexibel ist. Es ist aber klar auf die Datensätze aus dem Schweizerischen Nationalpark begrenzt und kann nicht auf andere angewendet werden. Die Nachteile der flexiblen Natur des Programmes liegen darin, dass der unerfahrene Benutzer wenig durch den Analyseprozess geleitet wird. Es ist zwar sehr vieles mit dem Programm möglich, doch erfordert die sinnvolle Benutzung ein gewisses Grundverständnis für die *WebPark*-Daten. Teilweise ist die Analyse bestimmter Sachverhalte mit dem Filter nur auf recht umständliche Art und Weise realisierbar. Überdies sind einige Statistiken noch unausgereift und liefern lediglich einen kleinen Beitrag zur Analyse der Daten. Ferner können die Daten mehrerer Jahre mit SNP Analytics nicht zeitgleich analysiert und verglichen werden. Die zeitliche Komponente der Besucherbewegung wird durch die Analysesoftware ebenfalls zu wenig berücksichtigt. Die Ursache dafür liegt jedoch auch in den Daten selbst. Sie sind für diesen Zweck nur bedingt geeignet, da sie keine vollständige Bewegungstrajektorie der Besucher enthalten. Es werden schließlich nur dann Positionen aufgezeichnet, wenn die Wanderer ihr Gerät verwenden. Eines der Hauptprobleme des entwickelten Programms ist außerdem die sich jährlich verändernde Struktur der *WebPark*-Software, die mit einem anderen Funktionsumfang einhergeht. Beispielsweise sind für SNP Analytics wichtige Konzepte aus dem Jahr 2007 in 2008 nicht mehr enthalten. Die Analysesoftware passt sich den Veränderungen zu wenig an. Unter anderem wird der wichtige Bestandteil der *Features of Interest* unzureichend behandelt.

Im Zuge dieser Arbeit konnten nicht alle der 21 Wanderpfade in die Analyse einbezogen werden. In Zukunft ist daher in erster Linie die Integration der bislang vernachlässigten Wanderpfade

anzustreben, um das Analysewerkzeug zu vervollständigen. Die Einbindung aller Vorverarbeitungsschritte in ein eigenes Modul in SNP Analytics würde darüber hinaus die Verarbeitung neuer Daten deutlich erleichtern.

Für eine Erweiterung der Analysesoftware existieren vielfältige Ideen. Sowohl Konzepte zur besseren Bedienbarkeit als auch für weitere Visualisierungstechniken sind vorhanden. Die Darstellung einzelner Besuchersequenzen könnte integriert und andere Kontextvariablen wie zum Beispiel das Gelände stärker berücksichtigt werden. Nützlich wären speziell weitere Ansätze zur Untersuchung des unmittelbaren Einflusses, den der Wanderführer auf die raumzeitliche Bewegung der Besucher im Park hat. Diese würden eine noch intensivere Analyse dieses Aspektes ermöglichen. Das Programm muss darüber hinaus besser auf die Struktur der *WebPark*-Software abgestimmt werden und die Unterstützung der wichtigen *Features of Interest* anbieten. Weiterhin ist es denkbar, den Anwendungsbereich von SNP Analytics über den Schweizerischen Nationalpark hinaus zu erweitern. Der Hersteller der *WebPark*-Software betreibt ähnliche Systeme in weiteren Naturschutzgebieten. Die dort gewonnenen Daten haben möglicherweise eine ähnliche Struktur, sodass sie mit geringem Aufwand für die Analysesoftware aufbereitet werden könnten. Somit würde ein generisches Werkzeug für die Untersuchung des Besucherverhaltens in mehr als nur diesem einen Nationalpark geschaffen werden.

Abschließend lässt sich sagen, dass in dieser Arbeit ein Werkzeug entwickelt wurde, mit dem die Betreiber des Schweizerischen Nationalparks in Zukunft die Daten des *WebPark*-Systems sehr detailliert untersuchen können. Das entwickelte Programm schließt die Lücke der bislang nicht einheitlich vorhandenen Nachbereitung der Daten. Es besitzt das Potential, die Rolle eines dauerhaften Begleitwerkzeuges zur Auswertung zu übernehmen.

Literatur

- [Abderhalden u. a. 2002] ABDERHALDEN, W. ; DIAS, E. ; HALLER, R. ; KRUG, K. ; MOUNTAIN, D.: *Analysis and Definition of User Needs. Web Park Project delivery D2.2.1, EC Project Number IST 200031041*, 2002
- [Andrienko u. a. 2008] ANDRIENKO, G. ; ANDRIENKO, N. ; DYKES, J. ; FABRIKANT, S. I. ; WACHOWICZ, M.: Geovisualization of Dynamics, Movement and Change: Key Issues and Developing Approaches in Visualization Research. In: *Information Visualization* 7 (3) (2008), S. 173–180
- [Andrienko u. a. 2005] ANDRIENKO, G. ; ANDRIENKO, N. ; DYKES, J. ; MOUNTAIN, D. ; NOY, P. ; GAHEGAN, M. ; ROBERTS, J. C. ; RODGERS, P. ; THEUS, M.: Creating Instruments for Ideation: Software Approaches to Geovisualization. In: DYKES, J. (Hrsg.) ; MACEACHREN, A. M. (Hrsg.) ; KRAAK, M.-J. (Hrsg.): *Exploring Geovisualization*. Elsevier, 2005, Kap. 5, S. 103–125
- [Andrienko u. a. 2007] ANDRIENKO, G. ; ANDRIENKO, N. ; WROBEL, S.: Visual Analytics Tools for Analysis of Movement Data. In: *ACM SIGKDD Explorations* 9 (2) (2007), S. 38–46
- [Andrienko und Andrienko 2006] ANDRIENKO, N. ; ANDRIENKO, G.: *Exploratory Analysis of Spatial and Temporal Data*. Springer, Berlin, 2006
- [Ashbrook und Starner 2003] ASHBROOK, D. ; STARNER, T.: Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users. In: *Personal and Ubiquitous Computing* 7 (5) (2003), S. 275–286
- [Bak u. a. 2009] BAK, P. ; MANSMANN, F. ; JANETZKO, H. ; KEIM, D. A.: Spatiotemporal Analysis of Sensor Logs using Growth Ring Maps. In: *IEEE Transactions on Visualization and Computer Graphics* 15 (6) (2009), S. 913–920
- [Bayer 2007] BAYER, S.: Python. In: HENNING, P. A. (Hrsg.) ; VOGELSSANG, H. (Hrsg.): *Handbuch Programmiersprachen. Softwareentwicklung zum Lernen und Nachschlagen*. Carl Hanser Verlag, München, 2007, Kap. 16, S. 539–566
- [Bernstein und Kornhauser 1996] BERNSTEIN, D. ; KORNHAUSER, A.: An Introduction to Map Matching for Personal Navigation Assistants / New Jersey TIDE Center. 1996. – Forschungsbericht
- [Brennan und Harlow 2004] BRENNAN, P. ; HARLOW, M.: *Linear Referencing in ArcGIS*. ESRI, Redlands, California, 2004
- [Brimicombe 2002] BRIMICOMBE, A. J.: GIS. Where are the frontiers now? In: *Proceedings for GIS 2002*, 2002, S. 33–45

- [Brinkhoff 2008] BRINKHOFF, T.: *Geodatenbanksysteme in Theorie und Praxis. Einführung in objektrelationale Geodatenbanken unter besonderer Berücksichtigung von Oracle Spatial*. 2., überarbeitete und erweiterte Auflage. Wichmann, 2008
- [Burghardt u. a. 2003] BURGHARDT, D. ; EDUARDES, A. ; WEIBEL, R.: WebPark. Entwicklung eines mobilen Informationssystems für Erholungs- und Naturschutzgebiete. In: *Kartographische Nachrichten* 2/2003 (2003), S. 58–64
- [Camineo 2010] CAMINEO: *Webseite der Firma Camineo*. 2010. – URL <http://www.camineo.com>. – Stand: 27.10.2010
- [Carvalho u. a. 2008] CARVALHO, A. ; AUGUSTO DE SOUSA, A. ; RIBEIRO, C.: A Temporal Focus + Context Visualization Model for Handling Valid Time Spatial Information. In: *Information Visualization* 7 (3) (2008), S. 265–274
- [Chen u. a. 2006] CHEN, M. Y. ; SOHN, T. ; CHMELEV, D. ; HAEHNEL, D. ; HIGHTOWER, J. ; HUGHES, J. ; LAMARCA, A. ; POTTER, F. ; SMITH, I. ; VARSHAVSKY, A.: Practical Metropolitan-Scale Positioning for GSM Phones. In: *Proceedings of UbiComp*, 2006, S. 225–242
- [Cheng u. a. 2005] CHENG, Y.-C. ; CHAWATHE, Y. ; LAMARCA, A. ; KRUMM, J.: Accuracy Characterization for Metropolitan-scale Wi-Fi Localization. In: *Proceedings of the 3rd international conference on Mobile systems, applications, and services (MobiSys)*, 2005, S. 233–245
- [Constandache u. a. 2010] CONSTANDACHE, I. ; CHOUDHURY, R. R. ; RHEE, I.: Towards Mobile Phone Localization without War-Driving. In: *Proceedings of IEEE Infocom*, 2010
- [Curran und Hubrich 2009] CURRAN, K. ; HUBRICH, S.: Optimising mobile phone self-location estimates by introducing beacon characteristics to the algorithm. In: *Journal of Location-Based Services* 3 (1) (2009), S. 55–73
- [DataCenter 2010] DATACENTER: *Webseite des SNP-DataCenters*. 2010. – URL <http://www.parcs.ch/mmds/index.php>. – Stand: 27.10.2010
- [Dias u. a. 2004a] DIAS, E. ; BEINAT, E. ; RHIN, C. ; SCHOLTEN, H.: Location Aware ICT in Addressing Protected Areas Goals. In: PRASTACOS, P. (Hrsg.) ; MURILLO, M. (Hrsg.): *Research on Computing Science. Special Edition on e-Environment* Bd. 11, 2004, S. 273–289
- [Dias u. a. 2004b] DIAS, E. ; RHIN, C. ; HALLER, R. ; SCHOLTEN, H.: Adding Value and Improving Processes Using Location-Based Services in Protected Areas: The WebPark Experience. In: PRASTACOS, P. (Hrsg.) ; CORTÉS, U. (Hrsg.) ; LEÓN, J. Díaz de (Hrsg.) ; MURILLO, M. (Hrsg.): *e-Environment: Progress and Challenge. Special Edition on e-Environment*. Instituto Politécnico Nacional Mexico, Mexico City, 2004, S. 291–302
- [Dias u. a. 2009] DIAS, E. S. ; EDUARDES, A. J. ; PURVES, R. S.: Analysing and aggregating visitor tracks in a protected area. In: STEIN, A. (Hrsg.) ; SHI, W. (Hrsg.) ; BIJKER, W. (Hrsg.): *Quality Aspects in Spatial Data Mining*. CRC Press, 2009, Kap. 20, S. 265–281

- [Dillemuth 2005] DILLEMUTH, J.: Map Design Evaluation for Mobile Display. In: *Cartography and Geographic Information Science* 32 (4) (2005), S. 285–301
- [Dodge 2010] DODGE, S.: *Patterns of Movement Wiki*. 2010. – URL <http://movementpatterns.pbworks.com/>. – Stand: 27.10.2010
- [Dodge u. a. 2008] DODGE, S. ; WEIBEL, R. ; LAUTENSCHÜTZ, A.-K.: Towards a Taxonomy of Movement Patterns. In: *Information Visualization* 7 (3) (2008), S. 240–252
- [Douglas und Peucker 1973] DOUGLAS, D. H. ; PEUCKER, T. K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. In: *Cartographica: The International Journal for Geographic Information and Geovisualization* 10 (2) (1973), S. 112–122
- [Duckham u. a. 2007] DUCKHAM, M. ; MOKBEL, M. ; NITTEL, S.: Editorial. Special issue on privacy aware and location-based mobile services. In: *Journal of Location-Based Services* 1 (3) (2007), S. 161–164
- [Dykes und Mountain 2003] DYKES, J. A. ; MOUNTAIN, D.: Seeking structure in records of spatio-temporal behaviour: visualization issues, efforts and applications. In: *Computational Statistics & Data Analysis* 43 (2003), S. 581–603
- [Eisenhut u. a. 2008] EISENHUT, A. ; HALLER, R. ; RAPER, J.: *How does topography influence the use of the mobile guide WebParkSNP in the Swiss National Park?* The Fourth International Conference on Monitoring and Management of Visitor Flows in Recreational and Protected Areas, Montecatini. 2008
- [ESRI 2010a] ESRI: *ArcGIS Desktop Help 9.2. Webseite*. 2010. – URL http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=Types_of_geodatabases. – Stand: 27.10.2010
- [ESRI 2010b] ESRI: *Homepage von ESRI Deutschland GmbH, Abschnitt Produkte, ArcGIS*. 2010. – URL <http://www.esri-germany.de/products/arcgis/index.html>. – Stand: 27.10.2010
- [di Flora und Hermersdorf 2008] FLORA, C. di ; HERMERSDORF, M.: A practical implementation of indoor location-based services using simple WiFi positioning. In: *Journal of Location-Based Services* 2 (2) (2008), S. 87–111
- [Fritz u. a. 2009] FRITZ, S. ; MCCALLUM, I. ; SCHILL, C. ; PERGER, C. ; GRILLMAYER, R. ; ACHARD, F. ; KRAXNER, F. ; OBERSTEINER, M.: Geo-Wiki.Org: The Use of Crowdsourcing to Improve Global Land Cover. In: *Remote Sensing* 1 (3) (2009), S. 345–354
- [Gahegan 2005] GAHEGAN, M.: Beyond Tools: Visual Support for the Entire Process of GIScience. In: DYKES, J. (Hrsg.) ; MACEACHREN, A. M. (Hrsg.) ; KRAAK, M.-J. (Hrsg.): *Exploring Geovisualization*. Elsevier, 2005, Kap. 4, S. 83–99

- [Gahegan u. a. 2001] GAHEGAN, M. ; WACHOWICZ, M. ; HARROWER, M. ; RHYNE, T.-M.: The Integration of Geographic Visualization with Knowledge Discovery in Databases and Geocomputation. In: *Cartography and Geographic Information Systems (special issue on the ICA research agenda)* 28 (1) (2001), S. 29–44
- [GeoVISTA 2010] GEOVISTA: *Webseite des GeoViz Toolkit*. 2010. – URL <http://www.geovista.psu.edu/geoviztoolkit/index.html>. – Stand: 27.10.2010
- [Giannotti u. a. 2007] GIANNOTTI, F. ; NANNI, M. ; PEDRESCHI, D. ; PINELLI, F.: Trajectory Pattern Mining. In: *Proceedings of the 13th International Conference on Knowledge Discovery and Data Mining*, 2007
- [Gilliéron und Merminod 2003] GILLIÉRON, P.-Y. ; MERMINOD, B.: Personal Navigation System for Indoor Application. In: *Proceedings of the 11th IAIN World Congress on Smart Navigation, Systems and Services*, 2003
- [Goetz u. a. 2006] GOETZ, B. ; PEIERLS, T. ; BLOCH, J. ; BOWBEER, J. ; HOLMES, D. ; LEA, D.: *Java Concurrency in Practice*. Addison-Wesley, 2006
- [Google Analytics 2010] GOOGLE ANALYTICS: *Webseite von Google Analytics*. 2010. – URL <http://www.google.com/intl/de/analytics/>. – Stand: 27.10.2010
- [Google Latitude 2010] GOOGLE LATITUDE: *Webseite von Google Latitude*. 2010. – URL <http://www.google.com/latitude>. – Stand: 27.10.2010
- [Gordillo u. a. 2008] GORDILLO, S. ; LAURINI, R. ; MOSTACCIO, C. ; RAFFORT, F. ; SERVIGNE, S.: Towards Multi-provider LBS Visual Portals. In: *The 14th International Conference on Distributed Multimedia Systems*, 2008, S. 208–213
- [Greenfeld 2002] GREENFELD, J. S.: Matching GPS observations to locations on a digital map. In: *Proceedings of the 81st Annual Meeting of the Transportation Research Board*, 2002
- [Gustafsson u. a. 2002] GUSTAFSSON, F. ; GUNNARSSON, F. ; BERGMAN, N. ; FORSELL, U. ; JANSSON, J. ; KARLSSON, R. ; NORDLUND, P.-J.: Particle filters for positioning, navigation, and tracking. In: *IEEE Transactions on Signal Processing* 50 (2) (2002), S. 425–437
- [Haller und Eisenhut 2008] HALLER, R. ; EISENHUT, A.: Was fragen Wanderer den digitalen Wanderführer im Webpark? Neue Medien in der Informationsvermittlung. In: EDER, R. (Hrsg.) ; ARNBERGER, A. (Hrsg.): *Auf den Pfaden von Natur und Kultur - Wodurch werden Lehrpfade, Themen- und Erlebniswege zu attraktiven Destinationen?* Institut für Landschaftsentwicklung, Erholungs- und Naturschutzplanung. Universität für Bodenkultur, Wien, 2008
- [Han und Kamber 2001] HAN, J. ; KAMBER, M.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco, 2001
- [Hardisty 2005] HARDISTY, F.: *The GeoViz Toolkit: Making Geographic Visualization Accessible*. Auto-Carto Conference. 2005. – Presentation

- [Hägerstrand 1970] HÄGERSTRAND, T.: What about people in regional science? In: *Papers of the Regional Science Association* 24 (1970), S. 7–21
- [Huffman 2009] HUFFMAN, D.: *New ideas in terrain mapping for cyclists.* Axismaps Internet Blog. 2009. – URL <http://www.axismaps.com/blog/2009/05/new-ideas-in-terrain-mapping-for-cyclists/>. – Stand: 27.10.2010
- [Ibrahim und Youssef 2010] IBRAHIM, M. ; YOUSSEF, M.: *CellSense: A Probabilistic RSSI-based GSM Positioning System.* unveröffentlicht. 2010. – URL <http://arxiv.org/abs/1004.3178>. – Stand: 27.10.2010
- [Karam u. a. 2010] KARAM, R. ; FAVETTA, F. ; KILANY, R. ; LAURINI, R.: Integration of Similar Location Based Services Proposed by Several Providers. In: *NDT 2010: Network and Digital Technologies, Lecture Notes on Computer Science (LNCS)*. Springer, Berlin (im Druck), 2010
- [Keim u. a. 2005] KEIM, D. A. ; PANSE, C. ; SIPS, M.: Information Visualization: Scope, Techniques and Opportunities for Geovisualization. In: DYKES, J. (Hrsg.) ; MACEACHREN, A. M. (Hrsg.) ; KRAAK, M.-J. (Hrsg.): *Exploring Geovisualization*. Elsevier, 2005, Kap. 2, S. 23–52
- [Khurri und Luukkainen 2009] KHURRI, A. ; LUUKKAINEN, S.: Identification of preconditions for an emerging mobile LBS market. In: *Journal of Location-Based Services* 3 (3) (2009), S. 188–209
- [Küpper und Treu 2010] KÜPPER, A. ; TREU, G.: Next Generation Location-based Services: Merging Positioning and Web 2.0. In: YANG, Laurence T. (Hrsg.) ; WALUYO, Agustinus B. (Hrsg.) ; MA, Jianhua (Hrsg.) ; TAN, Ling (Hrsg.) ; SRINIVASAN, Bala (Hrsg.): *Mobile Intelligence*. Wiley, Hoboken, 2010, Kap. 10, S. 213–236
- [Kraak 2003] KRAAK, M.-J.: The space-time cube revisited from a geovisualization perspective. In: *Proceedings of the 21st International Cartographic Conference (ICC), Durban, 2003*, S. 1988–1995
- [Kröckertskothén 2007] KRÖCKERTSKOTHÉN, T.: *Java 2. Grundlagen und Einführung.* 6., unveränderte Auflage. RRZN, Universität Hannover, 2007
- [Layar 2010] LAYAR: *Webseite des Layar Augmented Reality Browsers.* 2010. – URL <http://www.layar.com/>. – Stand: 27.10.2010
- [Liarokapis u. a. 2006] LIAROKAPIS, F. ; MOUNTAIN, D. ; PAPAKONSTANTINOY, S. ; BRUJIC-OKRETIC, V. ; RAPER, J.: Mixed Reality for Exploring Urban Environments. In: BRAZ, José (Hrsg.) ; JORGE, Joaquim A. (Hrsg.) ; DIAS, Miguel (Hrsg.) ; MARCOS, Adérito (Hrsg.): *GRAPP 2006 - Proceedings of the First International Conference on Computer Graphics Theory and Applications*, 2006, S. 208–215

- [Mabrouk 2005] MABROUK, M. (Hrsg.): *OpenGIS Location Services (OpenLS): Core Services*. Open Geospatial Consortium Inc., 2005
- [Mac Aoidh u. a. 2008] MAC AOIDH, E. ; BERTOLOTTO, M. ; WILSON, D. C.: Understanding Geospatial Interests by Visualising Map Interaction Behaviour. In: *Information Visualization* 7 (3) (2008), S. 275–286
- [McCoy 2004] MCCOY, J.: *Geoprocessing in ArcGIS*. ESRI, Redlands, California, 2004
- [Meng 2006] MENG, Y.: *Improved Positioning of Land Vehicle in ITS Using Digital Map and Other Accessory Information*, Department of Land Surveying and Geoinformatics, Hong Kong Polytechnic University, Dissertation, 2006
- [Morita 2007] MORITA, T.: Theory and development of research in ubiquitous mapping. In: *Location Based Services and TeleCartography*. Springer, Berlin : Gartner, G. and Cartwright, W. and Peterson, M. P., 2007 (Lecture Notes in Geoinformation and Cartography), S. 89–106
- [Mountain 2005] MOUNTAIN, D.: Visualizing, Querying and Summarizing Individual Spatio-Temporal Behaviour. In: DYKES, J. (Hrsg.) ; MACEACHREN, A. M. (Hrsg.) ; KRAAK, M.-J. (Hrsg.): *Exploring Geovisualization*. Elsevier, 2005, Kap. 9, S. 181–200
- [Mountain und MacFarlane 2007] MOUNTAIN, D. ; MACFARLANE, A.: Geographic information retrieval in a mobile environment: evaluating the needs of mobile individuals. In: *Journal of Information Science* 33 (5) (2007), S. 515–530
- [Nanni u. a. 2008] NANNI, M. ; KUIJPERS, B. ; KÖRNER, C. ; MAY, M. ; PEDRESCHI, D.: Spatiotemporal Data Mining. In: *Mobility, Data Mining and Privacy. Geographic Knowledge Discovery*. Springer, Berlin, 2008, Kap. 10, S. 267–296
- [Niemeier 2008] NIEMEIER, W.: *Ausgleichsrechnung. Statistische Auswertemethoden*. 2. Auflage. Walter de Gruyter, 2008
- [Nivala und Sarjakoski 2003] NIVALA, A.-M. ; SARJAKOSKI, T.: An Approach to Intelligent Maps: Context Awareness. In: *Proceedings of the workshop HCI in Mobile Guides held at the 5th International Symposium on Human Computer Interaction with Mobile Devices and Services (Mobile HCI)*. Udine, 2003, S. 45–50
- [Obradovic u. a. 2006] OBRADOVIC, D. ; LENZ, H. ; SCHUPFNER, M.: Fusion of Map and Sensor Data in a Modern Car Navigation System. In: *Journal of VLSI Signal Processing* 45 (1-2) (2006), S. 111–122
- [Ochieng u. a. 2004] OCHIENG, W. Y. ; QUDDUS, M. A. ; NOLAND, R. B.: Map-Matching in complex urban road networks. In: *Brazilian Journal of Cartography* 55 (2) (2004), S. 1–18
- [Ofstad u. a. 2008] OFSTAD, A. ; NICHOLAS, E. ; SZCODRONSKI, R. ; CHOUDHURY, R. R.: AAMPL: Accelerometer Augmented Mobile Phone Localization. In: *Proceedings of the ACM MELT Workshop (zusammen mit Mobicom)*, 2008

- [OGC 1999] OGC: *OpenGIS Consortium Simple Features Specification for SQL, Revision 1.1*. 1999
- [OpenCloud API 2010] OPENCLOUD API: *Webseite der OpenCloud API*. 2010. – URL <http://opencloud.mcavallo.org/>. – Stand: 27.10.2010
- [pgAdmin 2010] PGADMIN: *Webseite der Software pgAdmin*. 2010. – URL <http://www.pgadmin.org>. – Stand: 27.10.2010
- [Pilgrim 2004] PILGRIM, M.: *Dive Into Python*. Apress, 2004
- [PostGIS 2010] POSTGIS: *PostGIS 1.5.1 Manual (Download)*. 2010. – URL <http://postgis.refractor.net/download/postgis-1.5.1.pdf>. – Stand: 27.10.2010
- [PostgreSQL 2009] POSTGRESQL: *PostgreSQL 8.4.4 Documentation (Download)*. 2009. – URL <http://www.postgresql.org/files/documentation/pdf/8.4/postgresql-8.4.4-A4.pdf>. – Stand: 27.10.2010
- [Quddus u. a. 2006] QUDDUS, M. A. ; NOLAND, R. B. ; OCHIENG, W. Y.: A High Accuracy Fuzzy Logic Based Map Matching Algorithm for Road Transport. In: *Journal of Intelligent Transport Systems* 10 (3) (2006), S. 103–115
- [Quddus u. a. 2007] QUDDUS, M. A. ; OCHIENG, W. Y. ; NOLAND, R. B.: Current map-matching algorithms for transport applications: State-of-the art and future research directions. In: *Transportation Research Part C* 15 (5) (2007), S. 312–328
- [Raper 2008] RAPER, J. F.: *Tracking users of Camineo mobile guides in the Swiss National Park*. Cartographic Conference GeoCart, Auckland. 2008. – Presentation
- [Raper u. a. 2007] RAPER, J. F. ; GARTNER, G. ; KARIMI, H. ; RIZOS, C.: A critical evaluation of location based services and their potential. In: *Journal of Location-Based Services* 1 (1) (2007), S. 5–45
- [Reichenbacher 2004] REICHENBACHER, T.: *Mobile Cartography. Adaptive Visualisation of Geographic Information on Mobile Devices.*, Technische Universität München, Dissertation, 2004
- [Rinzivillo u. a. 2008a] RINZIVILLO, S. ; PEDRESCHI, D. ; NANNI, M. ; GIANNOTTI, F. ; ANDRIENKO, N. ; ANDRIENKO, G.: Visually driven analysis of movement data by progressive clustering. In: *Information Visualization* 7 (3) (2008), S. 225–239
- [Rinzivillo u. a. 2008b] RINZIVILLO, S. ; TURINI, F. ; BOGORNY, V. ; KÖRNER, C. ; KUIJPERS, B. ; MAY, M.: Knowledge Discovery from Geographical Data. In: *Mobility, Data Mining and Privacy. Geographic Knowledge Discovery*. Springer, Berlin, 2008, Kap. 9, S. 243–265
- [RRZN 2004] RRZN: *SQL. Grundlagen und Datenbankdesign*. 2. RRZN-Handbuch, Universität Hannover, 2004

- [Slingsby u. a. 2008] SLINGSBY, A. ; DYKES, J. ; WOOD, J.: Using treemaps for variable selection in spatio-temporal visualisation. In: *Information Visualization* 7 (3) (2008), S. 210–224
- [Slingsby u. a. 2010] SLINGSBY, A. ; WOOD, J. ; DYKES, J.: Treemap Cartography for showing Spatial and Temporal Traffic Patterns. In: *Journal of Maps* v2010 (2010), S. 135–146
- [Snozzi 2009] SNOZZI, J.: *Quantitative Assessment of Context for Mobile Services. A Case Study on the WebPark Service in the Swiss National Park*, Department of Geography, University of Zurich, Diplomarbeit, 2009
- [SNP 1983] SNP: *Verordnung über den Schutz des Schweizerischen Nationalparks (Nationalparkordnung)*. 1983
- [SNP 2010] SNP: *Webseite des Schweizerischen Nationalparks*. 2010. – URL <http://www.nationalpark.ch>. – Stand: 27.10.2010
- [Socialight 2010] SOCIALIGHT: *Webseite von Socialight*. 2010. – URL <http://socialight.com/>. – Stand: 27.10.2010
- [Spasov u. a. 2006] SPASSOV, I. ; BIERLAIRE, M. ; MERMINOD, B.: Map-matching for Pedestrians via Bayesian Inference. In: *European Navigation Conference. Global Navigation Satellite Systems*, 2006
- [Steiniger u. a. 2006] STEINIGER, S. ; NEUN, M. ; EDWARDES, A.: *Foundations of Location Based Services. Lesson 1*. CartouCHE. Lecture Notes on LBS. Institut für Geographie, Universität Zürich. 2006
- [Stolze 2003] STOLZE, K.: SQL/MM Spatial: The Standard to Manage Spatial Data in Relational Database Systems. In: WEIKUM, Gerhard (Hrsg.) ; SCHÖNING, Harald (Hrsg.) ; RAHM, Erhard (Hrsg.): *Datenbanksysteme für Business, Technologie und Web (BTW) : 10. GI-Fachtagung*, Bonner Köllen, 2003, S. 247–264
- [SwissTopo 2005] SWISSTOPO: *Näherungslösungen für die direkte Transformation CH1903 WGS84*. Download. 2005. – URL <http://www.swisstopo.admin.ch/internet/swisstopo/de/home/products/software/products/skripts.html>. – Stand: 27.10.2010
- [SwissTopo 2010] SWISSTOPO: *Schweizerische Bezugssysteme. Webseite*. 2010. – URL <http://www.swisstopo.admin.ch/internet/swisstopo/de/home/topics/survey/sys/refsys/switzerland.html>. – Stand: 27.10.2010
- [Tobler 1970] TOBLER, W. R.: A computer movie simulating urban growth in the Detroit region. In: *Journal of Economic Geography* 46 (2) (1970), S. 234–240
- [Tobler 1987] TOBLER, W. R.: Experiments In Migration Mapping By Computer. In: *Cartography and Geographic Information Science* 14 (2) (1987), S. 155–163

- [Tsai u. a. 2010] TSAI, J. Y. ; KELLEY, P. G. ; CRANOR, L. F. ; SADEH, N.: *Location-Sharing Technologies: Privacy Risks and Controls*. 2010. – im Druck
- [Tucker 2004] TUCKER, C.: *Writing Geoprocessing Scripts With ArcGIS*. ESRI. Redlands, California, 2004
- [Ullenboom 2009] ULLENBOOM, C.: *Java ist auch eine Insel. Programmieren mit der Java Standard Edition Version 6*. 8., aktualisierte Auflage. Galileo Computing, 2009
- [Vasiliev 1997] VASILIEV, I. R.: Mapping Time. In: *Cartographica: The International Journal for Geographic Information and Geovisualization* 34 (2) (1997), S. 1–51
- [Verhein 2009] VERHEIN, F.: Mining Complex Spatio-Temporal Sequence Patterns. In: *Proceedings of the 9th SIAM International Conference on Data Mining (SDM 09)*. Sparks, Nevada, 2009, S. 605–616
- [Verhein und Chawla 2006] VERHEIN, F. ; CHAWLA, S.: Mining Spatio-Temporal Association Rules, Sources, Sinks, Stationary Regions and Thorougfares in Object Mobility Databases. In: *Proceedings of the 11th International Conference on Database Systems for Advanced Applications (DASFAA)* Bd. 3882, Springer, 2006, S. 187–201
- [Virrantaus u. a. 2001] VIRRANTAU, K. ; VEIJALAINEN, J. ; MARKKULA, J. ; KATANOSOV, A. ; GARMASH, A. ; TIRRI, H. ; TERZIYAN, V.: Developing GIS-Supported Location-Based Services. In: *Proceedings of WGIS 2001. First International Workshop on Web Geographical Information Systems*. Kyoto, Japan, 2001, S. 423–432
- [Wall 2006] WALL, T.: *Tree (and tree table) with checkboxes*. 2006. – URL <http://rabbit-hole.blogspot.com/2006/11/tree-and-tree-table-with-checkboxes.html>. – Stand: 27.10.2010
- [Wasinger u. a. 2003] WASINGER, R. ; STAHL, C. ; KRÜGER, A.: M3I in a Pedestrian Navigation and Exploration System. In: *Proceedings of the Fifth International Symposium on Human Computer Interaction with Mobile Devices*, 2003, S. 481–485
- [WebPark 2010] WEBPARK: *Webseite des WebPark-Projekts*. 2010. – URL <http://www.webparkservices.info>. – Stand: 27.10.2010
- [Wikitude 2010] WIKITUDE: *Webseite des Wikitude World Browsers*. 2010. – URL <http://www.wikitude.org>. – Stand: 27.10.2010
- [Wood 2008] WOOD, J.: *Mouse Sensitive Shapes*. Webseite. 2008. – URL <http://www.soi.city.ac.uk/~jwo/processing/shapes/>. – Stand: 27.10.2010
- [Wood u. a. 2009] WOOD, J. ; DYKES, J. ; SLINGSBY, A. ; RADBURN, R.: Flow Trees for Exploring Spatial Trajectories. In: *Proceedings of the GIS Research UK 17th Annual Conference GISRUUK 2009*. University of Durham, Durham, 2009, S. 229–234
- [Yu 2006] YU, H.: Spatio-temporal GIS Design for Exploring Interactions of Human Activities. In: *Cartography and Geographic Information Science* 33 (1) (2006), S. 3–19

[Zhao 1997] ZHAO, Y.: *Vehicle Location and Navigation Systems*. Artech House, 1997

A Anhang

A.1 Action Codes 2007 und 2008

Gruppe	Code	Beschreibung
General	101	MainPage
	102	PrivacyPolicy
	105	MoreMainPage
	602	News
	701	Get Top 20
Info Around	401	AroundMe
	504	Get FOI Categories
	505	Get Update Categories
	506	Get map IFOI list
Info on Route	301	Route information
	302	Route vertical profile
	303	Route map
	304	Route search
	601	VirtualTrails
Orientation	201	Show map (from Home page)
	203	WhereAmI
	605	Thematic maps
	606	Map layer
Content	801	Content: Grasshopper
	802	Content: Wildfire
	803	Content: Vegetation
	804	Content: Baer
	805	Content: Butterfly
	806	Content: Bird
	807	Content: Plants
	808	Content: POI
	809	Content: IFOI
Special Function	501	AdvancedSearch
	502	SearchNaturalLanguage
	503	SearchPOI
	901	Bookmarks
	902	TakePicture
Info on Device	103	BatteryLow
	104	TutorHelp
	603	KeyApps

Action Codes im Jahr 2007.

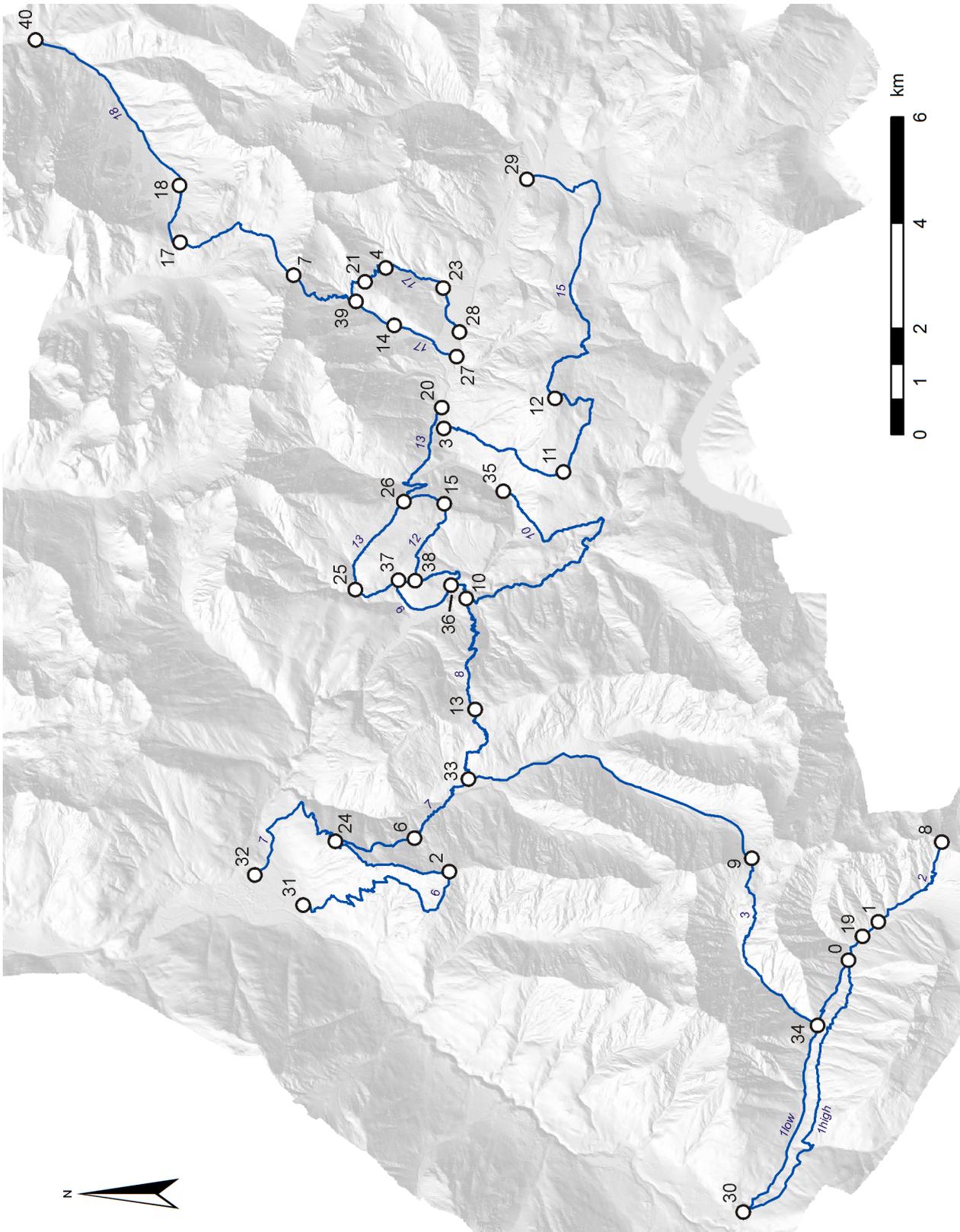
Gruppe	Code	Beschreibung
General	101	HomePage
	102	PrivacyPolicy
	107	Button page
	108	User preferences
	602	News
	701	Get Top 6
Info Around	202	Get map IFOI list
	401	AroundMe
	504	Get Info FOI
	505	Get IFOI
Info on Route	302	Route vertical profile
Content	801	Content: Grasshopper
	803	Content: Vegetation
	805	Content: Butterfly
	806	Content: Bird
Special Function	501	SearchNaturalLanguage
	901	Bookmarks
	902	TakePicture
Info on Device	105	TutorHelp
	106	Logout

Action Codes im Jahr 2008.

A.2 Übersicht aller Regions of Interest

ID	Beschreibung	Route	$m_{Start}[m]$	$m_{Ende}[m]$
0	picnic 0 / intersection lhigh-llow	0lhigh	6092,43	6192,43
		2	0,00	174,68
		0llow	5564,66	5664,66
1	picnic 1	2	836,38	1036,38
2	picnic 2	6	5152,94	5352,94
3	picnic 3 / start 15	15	0,00	129,42
4	picnic 4	17	3879,01	4079,01
6	picnic 6	7	5118,00	5318,00
7	picnic 7	18	8407,20	8607,20
8	picnic 8 / end 2	2	3308,45	3408,45
9	picnic 9	3	6263,75	6463,75
10	picnic 10 / intersection 8-10	10	0,00	100,00
		8	5920,24	6120,24
11	picnic 11	15	2639,07	2839,07
12	picnic 12	15	5149,49	5349,49
13	picnic 13	8	2385,51	2585,51
14	picnic 14	17	1305,37	1505,37
15	picnic 15	12	3057,15	3257,15
17	picnic 17	18	5414,33	5614,33
18	picnic 18	18	4042,51	4242,51
19	picnic 19	2	455,12	655,12
20	picnic 20 / hotel il fuorn / end 13	13	4657,94	4896,29
21	picnic 21	17	3119,86	3319,86
23	picnic 23	17	5141,49	5341,49
24	intersection 6-7	6	7888,99	7988,99
		7	3416,73	3616,73
25	intersection 12-13 (1)	13	0,00	100,00
		12	0,00	100,00
26	intersection 12-13 (2)	13	1918,23	2118,23
		12	3777,22	3877,22
27	start 17	17	0,00	100,00
28	end 17	17	6145,15	6245,15
29	end 15	15	11844,99	11944,99
30	start 1	0llow	0,00	220,97
		0lhigh	0,00	100,00
31	start 6	6	0,00	100,00
32	Zernez / start 7	7	0,00	100,00
33	chuozza hut / intersection 3-7-8	8	26,04	226,04
34	intersection llow-3	3	10427,83	10527,83
		0llow	4014,14	4214,14
35	end 10	10	6363,88	6463,88
36	intersection 8-9	9	1602,15	1802,15
		8	6490,39	6590,39
37	intersection 9-12 (1)	9	0,00	100,00
		12	931,88	1131,88
38	intersection 9-12 (2)	9	2826,44	2926,44
		12	1222,00	1422,00
39	intersection 17-18	17	2099,97	2299,97
		18	10278,37	10378,37
40	start 18	18	0,00	100,00

Liste aller 38 *Regions of Interest* und ihrer Definition im linearen Bezugssystem der Routen.



Übersichtskarte aller 38 *Regions of Interest* im Nationalpark.