

Institut für Photogrammetrie und GeoInformation
Leibniz Universität Hannover

Masterarbeit

**Untersuchungen zur automatischen
Detektion von Flüssen mittels
markierter Punktprozesse**

Christian Kruse

Betreuer

Prof. Dr.-Ing. Christian Heipke
M. Sc. Alena Schmidt

Hannover, 19. November 2015

MASTERARBEIT

- Kandidat** B.Sc. Christian Kruse
- Aufgabe** Untersuchungen zur automatischen Detektion von Flüssen mittels markierter Punktprozesse
(Investigations on the automatic detection of rivers using marked point processes)
- Motivation** Eine vielversprechende Methode für die Beschreibung zufälliger Vorgänge liefern stochastische Prozesse. Hierzu gehören markierte Punktprozesse, welche die Zielsetzung haben, aus einer großen Menge von Daten diejenige Untermenge herauszufinden, die ein beobachtetes Phänomen am besten beschreibt. In der Fernerkundung finden markierte Punktprozesse bei der Detektion von unregelmäßig verteilten Objekten in satelliten- oder flugzeuggestützten Daten Verwendung. Hierfür ist die Modellierung der Objekte erforderlich, welche zusätzlich mit Merkmalen versehen werden können. Mit Hilfe eines Markov-Chain-Monte-Carlo-Verfahrens werden dann zufällig Objektkonfigurationen generiert und die Lösung iterativ optimiert.
- Ansatz** Ziel der Masterarbeit ist die Anwendung von markierten Punktprozessen für die Detektion von Flussnetzwerken in luftgestützten Laserscannerdaten. Hierfür soll am Institut entwickelte Software genutzt und erweitert werden. Neben der Analyse der Einflüsse verschiedener Parametereinstellungen auf das Ergebnis für unterschiedliche Datensätze, sollen Methoden entwickelt werden, mit welchen sich die Detektion von Flussnetzwerken optimieren lässt.
- Skills** Für die Arbeit ist ein vertieftes Wissen in Statistik und der Auswertung von Laserdaten erforderlich. Weiterhin ist die Kenntnis einer Programmier- oder Skriptsprache (C++, Matlab) notwendig.

Ansprechpartner



Institut für Photogrammetrie und Geoinformation

M.Sc. Alena Schmidt

alena.schmidt@ipi.uni-hannover.de
0511 - 762 2566

www.ipi.uni-hannover.de

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Hannover, 19. November 2015

Christian Kruse

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Zielsetzung und Aufbau der Arbeit	3
2	Grundlagen	5
2.1	Airborne Laserscanning.....	5
2.2	Ableitung von digitalen Geländemodellen.....	6
2.3	Stochastische Prozesse	8
2.3.1	Markierte Punktprozesse	9
2.3.1.1	Poisson-Punktprozesse	11
2.3.1.2	Wahrscheinlichkeitsdichte und Gibbs-Energie	12
2.3.2	Monte-Carlo-Verfahren	14
2.3.2.1	Markov-Chain-Monte-Carlo-Verfahren	17
3	Methodik	27
3.1	Methode zur Flussdetektion	27
3.2	Untersuchungen hinsichtlich Zufallszahlengeneratoren.....	30
3.3	Erweiterung der Energiefunktion	32
4	Analyse der Ergebnisse	39
4.1	Testdaten	39
4.2	Initialisierung des Generators sowie Laufzeitvergleich	41
4.3	Untersuchungen zur Verteilung der Parameter.....	44
4.4	Untersuchungen zur modellierten 3D-Komponente	53
4.4.1	Synthetische Daten.....	54
4.4.2	Daten des deutschen Wattenmeeres.....	67
4.4.3	Daten aus Vorarlberg in Österreich	73
5	Zusammenfassung und Ausblick	77
	Literaturverzeichnis	81

1 Einleitung

1.1 Motivation

Seit mehreren Jahrzehnten ist die automatische Extraktion von Linien-Netzwerken aus Bildern in verschiedensten Disziplinen ein zentrales Thema. So weisen beispielsweise in der Medizin Adern und Venen netzwerkartige Strukturen auf (Sun et al., 2007; Lesage et al., 2009; Marin et al., 2011). Ein klassisches Beispiel aus der Fernerkundung ist die Straßenextraktion. Vorreiter für nachfolgende Arbeiten sind hierbei u. a. McKeown und Denlinger (1988) sowie Mayer et al. (1998) gewesen. Auch in anderen Bereichen, wie der Geographie und Ökologie, sind solche netzwerkartigen Gebilde zu finden. In Wattenmeeren werden diese durch Priele repräsentiert, welche dynamischen Veränderungen aufgrund der Gezeiten unterliegen. So sind zum Beispiel Verlagerungen, Verschmälerungen und Formveränderungen der Priele die Folge und es kommt ebenfalls zu einem Einfluss auf die Morphologie der Küstengewässer (Milbradt et al., 2009). Um den außergewöhnlichen Landschaftsraum mit seinen zahlreichen Pflanzen- und Tierarten zu erhalten und andererseits zum Küstenschutz beizutragen, ist die detaillierte Kenntnis der Prielcharakteristiken und dessen Veränderungen über die Zeit notwendig. Sind diese gegeben, können zuverlässige Vorhersagen getroffen werden (Schmidt et al., 2014).

Aufgrund der stetig steigenden Datenmengen, die auf die höheren Auflösungen und Abdeckungen dieser zurückgeführt werden können, sind bei der Prozessierung geeignete Methoden in Betracht zu ziehen. Zur automatischen Detektion der beobachteten Phänomene innerhalb der Bilder oder auch aus von (luftgestützten) Laserscannerdaten abgeleiteten digitalen Geländemodellen sind verschiedene Ansätze in der Literatur zu finden. So werden beispielsweise *perceptual grouping* und Segmentierungstechniken (Poullis und You, 2010), neuronale Netze (Mnih und Hinton, 2010), aktive Konturen (Rochery et al., 2006) oder auch Erfolg versprechende probabilistische Ansätze verwendet. Probabilistische Modelle, eingeführt von Baddeley und Lieshout (1993), nutzen Zufallsvariablen, dessen Realisierungen Konfigurationen geometrischer bzw. parametrischer Objekte sind. Die Aufgabe stochastischer Prozesse ist es, aus oftmals großen Datenmengen die Untermenge zu finden, welche das beobachtete Phänomen am besten beschreibt. Dieses weist häufig den zuvor beschriebenen zufälligen Charakter auf. Zu den zufallsbedingten Prozessen gehören Punktprozesse. Dabei entsprechen die Punkte des zufallsbedingten Prozesses in

der Regel den Positionen von Objekten innerhalb der Daten. Hierdurch ist jedes Objekt einem Punkt innerhalb der Szene zugeteilt. Die Anzahl der Objekte selbst ist ebenso eine Zufallsvariable und muss deshalb nicht von einem Nutzer festgelegt oder geschätzt werden. Erhalten die Objekte noch zusätzlich Merkmale, welche sich von Punkt zu Punkt verschieden ausprägen können, handelt es sich um markierte Punktprozesse. Ein vielfach in der Literatur verwendetes Objekt sind Rechtecke. So werden diese beispielsweise bei Ortner et al. (2007) genutzt, um die Umrisse von Gebäuden zu extrahieren. Tournaire et al. (2007) gebrauchen markierte Punktprozesse, ebenfalls modelliert durch Rechtecke, zur Detektion von Fahrbahnmarkierungen, sodass jeder Punkt des Prozesses für einen Streifen der Markierung steht. Abhängig von der Aufgabenstellung werden neben Rechtecken auch andere geometrische Objekte verwendet. Descamps et al. (2008) nutzen Ellipsen zur automatischen Detektion der Population von Flamingos. Ferner werden Segmente, um Linien-Netzwerke zu extrahieren (Lacoste et al., 2005) oder Zylinder zur Menschendetektion in 3D (Utasi und Benedek, 2011) benutzt. Gleichermaßen sind Kombinationen dieser und weiterer geometrischer Objekte, wie sie Lafarge et al. (2010) nutzen, denkbar.

Ein Punktprozess benötigt die Formulierung einer Wahrscheinlichkeitsdichte, um die Qualität einer Objektkonfiguration zu messen. Dabei ist die Dichte üblicherweise eine Kombination zweier Terme. Ersterer bewertet die Übereinstimmung der Objektkonfigurationen mit den Daten, wobei der andere Term Interaktionen zwischen den Objekten berücksichtigt. Das heißt, es können bestimmte Konfigurationen auf Basis von Vorwissen befürwortet oder bestraft werden. Ferner wird angenommen, dass für jede Objektkonfiguration eine Wahrscheinlichkeit bestimmt werden kann, die gesamte Wahrscheinlichkeitsverteilung für die Objektkonfiguration jedoch unbekannt ist. Ziel ist das Auffinden dieser optimalen Konfiguration, welche die Dichte maximiert. Die Realisierung erfolgt auf Basis eines Monte-Carlo-Samplers. In den meisten Fällen wird ein Markov-Chain-Monte-Carlo-Algorithmus (Metropolis et al., 1953; Hastings, 1970) verwendet, welcher die Optimierung der Zufallsvariablen durch eine Approximation der wahren Verteilung mittels einer ausreichend großen Anzahl an Stichproben ermöglicht. Eine Erweiterung, um mit einer unbekanntem Anzahl an Objekten umgehen zu können, liefert das Reversible-Jump-Markov-Chain-Monte-Carlo-Verfahren (Green, 1995). Indem die Parameter der Objekte (Positionen und Ausprägungen) mittels zuvor definierter Operationen verändert werden, lässt sich die Objektkonfiguration iterativ optimieren. Um dessen Wahrscheinlichkeitsverteilung zu modellieren, kann die Gibbs-Energie, die sich aus den beiden oben genannten Termen zusammensetzt, verwendet werden. Kommt es durch die Modifikationen der Objektparameter zu einem Energiezuwachs, so sind die Änderungen mit einer hohen Wahrscheinlichkeit zu verwerfen, andernfalls anzunehmen. Häufig erfolgt noch eine Kopplung des Samplers mit einem *simulated annealing* (simulierte Abkühlung) (Salamon et

al., 2002). Hierbei handelt es sich um eine Art globales Optimierungswerkzeug, damit das Minimum der Energie gefunden werden kann, ohne dabei in lokalen Minima festhängen zu bleiben.

Die experimentellen Ergebnisse der aufgeführten und auch zahlreichen weiteren, im folgenden nur exemplarisch genannten Paper, zeigen das Potential und die Leistungsfähigkeit stochastischer Modelle basierend auf markierten Punktprozessen in verschiedensten Anwendungsfeldern der Bildanalyse (Perrin et al., 2005; Perrin et al., 2006; Ortner et al., 2008; Mallet et al., 2010). Insbesondere werden zudem die Flexibilität und Optimierbarkeit der stochastischen Prozesse ersichtlich. So verwenden zum Beispiel Tournaire et al. (2010) das Rahmenkonstrukt des markierten Punktprozesses von Ortner et al. (2007) zur Detektion der Grundrisse von Gebäuden. Schmidt et al. (2014) passen die Methode von Tournaire et al. (2010) wiederum an, um Flussnetzwerke zu finden. Neben einer flexiblen Adaption sind die Methoden im Hinblick auf weitere Aspekte, wie beispielsweise die Parameter und die Rechenzeit optimierbar. Lafarge et al. (2010) oder auch Tournaire et al. (2010) berücksichtigen diese Eigenschaften durch einen generalisierten markierten Punktprozess – bei dennoch ähnlicher Genauigkeit im Vergleich zu vorherigen Methoden. Ferner ermöglichen Verfahrensweisen auf Basis von Graphen netzwerkartige Strukturen aus den Daten, wie Straßen, Blutgefäße oder Flüsse, zu extrahieren. Durch die graphenbasierte Repräsentation bleibt die Netzwerkstruktur erhalten, da jeder Knoten entweder zu einem Kreuzungs- oder Endpunkt gehört (Chai et al., 2013).

1.2 Zielsetzung und Aufbau der Arbeit

Innerhalb dieser Arbeit werden markierte Punktprozesse genutzt, um netzwerkartige Strukturen in luftgestützten Laserscannerdaten zu detektieren. Es handelt sich dabei um einen Datensatz des Wattenmeeres der Nordsee, der während Niedrigwasser erhoben und zu einem digitalen Geländemodell weiterverarbeitet worden ist. Darüber hinaus werden Daten des Bundeslandes Vorarlberg in Österreich verwendet. Im Rahmen der Anwendung der Punktprozesse soll eine vom Institut für Photogrammetrie und GeoInformation entwickelte Software genutzt werden. Hierfür sind neben Analysen zur Verteilung der Parameter die Einflüsse unterschiedlicher Einstellungen dieser auf das Ergebnis zu untersuchen. Weiterhin gilt es, Experimente mit Zufallszahlengeneratoren, welche einen zentralen Bestandteil der Software darstellen, durchzuführen. Abschließend sollen Erweiterungen am Programm vorgenommen werden, wodurch die Detektion netzwerkartiger

Strukturen mithilfe von entwickelten Methoden optimiert werden soll. Um diese besser evaluieren zu können, werden zusätzlich synthetische Daten herangezogen.

Die Arbeit gliedert sich hierbei wie folgt.

In Kapitel 2 werden die Grundlagen beschrieben, die für die nachfolgenden Untersuchungen der unterschiedlichen Datensätze und das Verständnis der Arbeit erforderlich sind. Dabei werden die Ausführungen zum flugzeuggestützten Laserscanning und der Ableitung von digitalen Geländemodellen sehr kurz gehalten, während auf die stochastischen Prozesse detailliert eingegangen wird.

Eine Erläuterung von Methoden, welche im Rahmen dieser Arbeit Anwendung finden, erfolgt im dritten Kapitel. Vorangestellt wird eine Beschreibung des von Schmidt et al. (2015) entwickelten Algorithmus, welcher die Grundlage der späteren Untersuchungen bildet. Es folgen Erklärungen der Methodik im Hinblick auf Zufallszahlengeneratoren, der Verteilung von Parametern und der Modellierung einer 3D-Komponente, welche das bestehende Programm erweitert.

In Kapitel 4 erfolgt zunächst eine Beschreibung der verwendeten Datensätze, welche daraufhin auf verschiedene Weise untersucht und deren Ergebnisse dargestellt werden. Die Analysen zur Verteilung der Parameter und der Laufzeitvergleich zwischen zwei Arten von Zufallszahlengeneratoren beruhen fast ausschließlich auf den Daten des deutschen Wattenmeeres. Die Evaluierung des 3D-Modells erfolgt zunächst auf Basis der synthetischen Daten. Anschließend wird das Fließmodell ebenfalls auf die Datensätze aus Deutschland und Österreich angewendet. Die Untersuchungen zur Initialisierung des Generators sind von den Daten losgelöst. Alle Analysen werden zwecks Detektion netzwerkartiger Strukturen aus Laserscannerdaten vorgenommen.

Schließlich erfolgt in Kapitel 5 eine Zusammenfassung der Ergebnisse. Darüber hinaus wird ein kurzer Ausblick hinsichtlich weiterer möglicher Untersuchungen der vorgestellten Methoden – insbesondere im Zusammenhang mit der modellierten 3D-Komponente – gegeben.

2 Grundlagen

2.1 Airborne Laserscanning

Das flugzeuggestützte Laserscanning, auch Airborne Laserscanning (ALS) genannt, liefert eine quantitative, dreidimensionale Repräsentation der Geländeoberfläche, zum Beispiel in Form einer Punktwolke. Dabei erfasst der Laserscanner die geometrischen, gegebenenfalls auch strukturellen Informationen der sichtbaren Objektoberflächen auf der Erde und speichert diese. Durch die berührungslose Aufnahme der Objekte auf der Erdoberfläche sowie der Kombination aus schneller Messung mit hoher Genauigkeit besitzt dieses Verfahren viele Vorteile gegenüber einer terrestrischen Aufnahme. Dies sind Gründe, weshalb das ALS innerhalb der letzten Jahrzehnte einen starken Aufschwung erfahren und sich aus technischer Sicht stetig verbessert hat (Beraldin et al., 2010).

Die Anwendungsbereiche des ALS sind umfangreich. Mithilfe der gewonnenen 3D-Informationen der Erdoberfläche aus den Laserscanneraufnahmen können sowohl digitale Oberflächen- als auch digitale Geländemodelle erstellt werden. Neben der Rekonstruktion von Gebäuden sind auch die Überwachung von Stromleitungen sowie der Küstenschutz zentrale Bereiche, in denen dieses Verfahren Anwendung findet. Beispielsweise ist eine Simulation von Flutereignissen und der dadurch auftretenden Schäden möglich (Sörgel, 2013).

Weitere Informationen zum ALS können bei Bedarf den nachfolgend genannten Quellen entnommen werden. So geben beispielsweise Wagner et al. (2003) einen Überblick über die Eigenschaften des Lasers und das Aufnahmeprinzip des Laserscanners. Neben der Entfernung zum Objekt auf der Erdoberfläche, die es zu bestimmen gilt (beispielsweise in Kraus (2004) formuliert), müssen auch noch die variable Position und Raumrichtung des Flugzeuges bekannt sein. Informationen über die beim ALS notwendigen Komponenten geben zum Beispiel Beraldin et al. (2010). Aufgrund der unterschiedlichen Referenzsysteme der Komponenten ist eine Georeferenzierung notwendig. Näheres hierzu und praktische Erfahrungen können in (Würländer et al., 2005) sowie (Würländer und Wenger-Oehn, 2007) nachgelesen werden.

2.2 Ableitung von digitalen Geländemodellen

Nach der Georeferenzierung können aus der gegebenen Punktwolke, die eine unregelmäßige Struktur hinsichtlich der Punkteverteilung aufweist, digitale (mathematische) Modelle der Erde erstellt werden. Zu diesen gehören das digitale Oberflächenmodell (DOM) sowie das digitale Geländemodell (DGM). Hier liegen die Punkte in einer Datenstruktur vor. Gängige Strukturen sind das regelmäßige Raster oder die Dreiecksvermaschung (Triangulierung).

Das DOM ist eine digitale Repräsentation der Erdoberfläche inklusive der Gebäude und der Vegetation. Von Häusern wird die Oberfläche der Dächer erfasst. Waldgebiete werden durch Flächen, die durch ineinander greifende Baumkronen entstehen, repräsentiert. Für offenes Gelände ist das DOM identisch mit dem DGM.

Das DGM hingegen stellt eine Repräsentation der Erdoberfläche ohne die Gebäude und die Vegetation dar – vielmehr werden die Höhen des Geländes beschrieben. Es kann Geripplinien, Geländekanten und markante Punkte enthalten (Pfeifer, 2003).

In Abbildung 2.1 ist die perspektivische Sicht eines Gebietes, jeweils mit einer Rasterweite von 1 m, dargestellt.

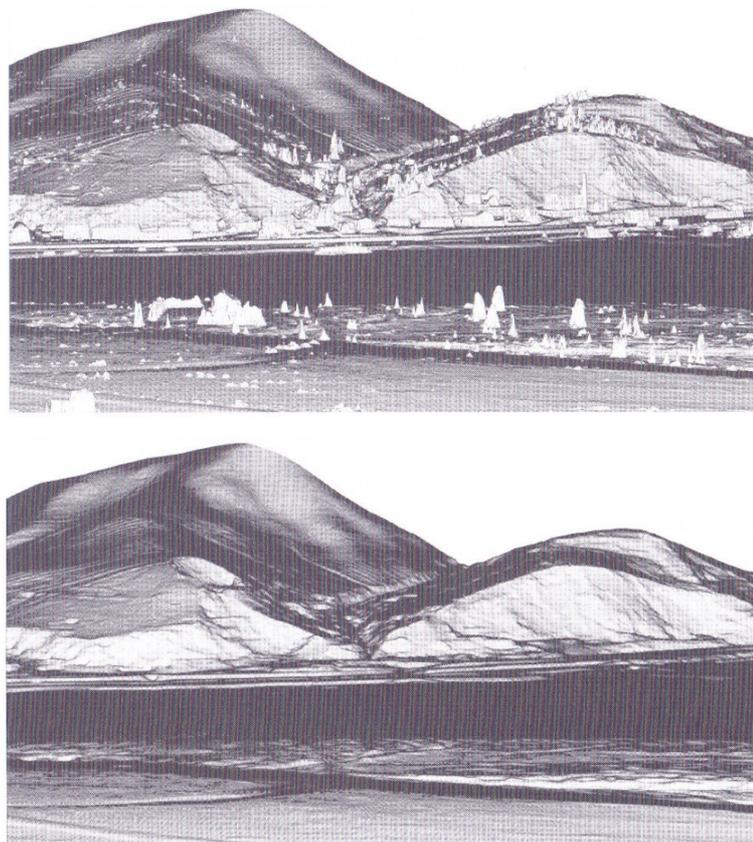


Abbildung 2.1: Darstellung des DOM (oben) und des DGM (unten)
(Briese, 2010)

Für die Erstellung des DOMs sind im Allgemeinen zwei wesentliche Schritte erforderlich: die Filterung (Trennung von Gelände- und Nichtgeländepunkten) sowie die Interpolation der rohen Punktwolke in ein regelmäßiges Raster.

Bei der Erfassung des Geländes fallen hohe Datenmengen an. Aus diesem Grund ist für die Filterung eine Automatisierung nötig, um die Gelände- von den Nichtgeländepunkten zu trennen. Zur Ableitung des DGMs existieren diverse Filterungsansätze. Sithole und Vosselman (2004) sowie Sithole (2005) geben einen Überblick über verschiedene Filteralgorithmen. Einige Filteralgorithmen arbeiten auf Rasterdatenstrukturen, andere direkt mit der originalen Punktwolke oder mit triangulierten Daten. Erstere haben den Vorteil, dass schnelle und einfache Operationen hinsichtlich der Nachbarschaft unter einem gewissen Genauigkeitsverlust möglich sind. Durch die Verarbeitung der originalen Punktwolke können hingegen einige Geländestrukturen, wie z.B. Bruchkanten, besser erhalten bleiben. Manchmal werden die beiden zuerst genannten Algorithmen kombiniert genutzt, um eine bessere Qualität hinsichtlich der Klassifikation in Gelände- und Nichtgeländepunkte zu erzielen (Briese, 2010).

Im nächsten Schritt ist die unregelmäßige Punktwolke in eine Rasterstruktur mithilfe einer Interpolationsmethode zu überführen. Durch die Interpolation werden für jeden Punkt des Rasters Werte aus einer begrenzten Anzahl von Referenzdatenpunkten abgeleitet, um beispielsweise Höhenwerte zu erhalten. Hierbei existieren verschiedene Ansätze.

Lokale Methoden, zu welchen u. a. die Inverse Distanzwichtung oder die Nutzung des gewichteten Mittels zählen, basieren auf der Annahme, dass jeder Punkt die resultierende Oberfläche nur bis zu einer gewissen endlichen Distanz beeinflusst. Neben diesen gibt es noch geostatistische Ansätze, zu denen das Kriging zählt und sogenannte Variationsmethoden. Diesen liegt die Annahme zugrunde, dass die Interpolationsfunktion durch die Datenpunkte oder zumindest in deren Nähe verläuft und gleichzeitig so glatt wie möglich sein soll. Weitere Informationen können Mitas und Mitasova (1999) und den hierin enthaltenen Referenzen entnommen werden. Jedoch macht die Rechenkomplexität dieser Methoden es häufig unmöglich, sie direkt auf größeren Datenmengen zu nutzen. Hier setzen zahlreiche Segmentierungsschemata an, um die Daten in Bereiche aufzuspalten, die dann jeweils eine kleinere Anzahl an Punkten aufweisen. Hierdurch kann die Interpolation für jedes Segment unabhängig durchgeführt werden. Mit dieser Thematik haben sich zum Beispiel Agarwal et al. (2006) befasst, welche einen sogenannten E/A-effizienten Algorithmus auf Basis einer Quadtree-Segmentation vorstellen.

2.3 Stochastische Prozesse

Ein stochastischer Prozess besteht aus Zufallsvariablen $X(t)$, welche von einem deterministischen Parameter t abhängen. Dieser durchläuft einen Parameterraum T , welcher ein Zeitintervall oder aufeinanderfolgende Zeitpunkte darstellt. Oftmals wird der Parameter t als Zeit interpretiert. Der Wert einer Zufallsvariable $X(t)$, welche ein zentrales Objekt aus der Wahrscheinlichkeitstheorie darstellt, hängt vom Zufall ab und folgt somit keiner Verteilungsfunktion. Nach Bewersdorff (2012) handelt es sich bei einer Zufallsvariablen um eine Zuordnungsvorschrift, wodurch jedem Ergebnis eines Zufallsexperimentes eine Größe zugeordnet wird. Häufig steht dabei das Interesse an einer Funktion für das Ergebnis und nicht das Ergebnis selbst im Vordergrund. So können beispielsweise beim Werfen von zwei fairen Würfeln nicht die Augenzahlen der beiden Würfel sondern vielmehr deren gemeinsame Würfelaugensumme von Bedeutung sein. In diesem Fall ist die Zufallsvariable durch die Summe der beiden fairen Würfel definiert (Ross, 1997). Derartige reelle Funktionen führen dazu, dass die Bezeichnung der zufälligen Funktionen ebenfalls zutreffend ist. Jedoch ist in der Literatur meist der Begriff des stochastischen oder auch zufälligen Prozesses zu finden (Beichelt und Montgomery, 2003).

Beichelt und Montgomery (2003) oder auch Ross (1996) definieren stochastische Prozesse wie folgt: Ein stochastischer Prozess mit dem Parameterraum T (auch Indexmenge genannt) und dem Zustandsraum Z ist eine Sammlung von Zufallsvariablen $\{X(t), t \in T\}$, wobei Z die Menge aller Zustände (Werte) bezeichnet, die die $X(t)$ für alle $t \in T$ annehmen können. Werden alle Werte der Zufallsvariablen über den gesamten Zeitraum erfasst, erhält man eine reelle Funktion $x = x(t)$, $t \in T$, welche als Trajektorie oder Realisierung des stochastischen Prozesses bezeichnet wird.

Eine Differenzierung dieser Prozesse in verschiedene Prozessklassen kann auf Basis ihrer stochastischen Eigenschaften erfolgen. Exemplarisch aufgeführt werden können hier die Gauß- (Rasmussen und Williams, 2006) oder auch stochastischen Integrationsprozesse (Kuo, 2006). Die am weitesten verbreitete Klasse ist die der Markov-Prozesse (Kap. 2.3.2.1). Darüber hinaus lassen sich stochastische Prozesse auch in Abhängigkeit vom Parameter- und Zustandsraum in vier unterschiedliche Klassen aufteilen. Bei stochastischen Prozessen mit diskreter Zeit liegt eine abzählbar unendliche oder auch eine kleinere, also endliche Parametermenge, vor. Ist T ein Intervall, handelt es sich um einen stochastischen Prozess mit stetiger Zeit. Weiterhin liegt ein diskreter stochastischer Prozess vor, wenn sein Zustandsraum eine endliche oder abzählbar unendliche Menge ist. In Analogie wird von stetigen stochastischen Prozessen gesprochen, wenn Z ein Intervall ist. Somit ergeben sich für die vier Prozessklassen unterschiedliche Trajektorien. Die Trajektorien eines stetigen stochastischen Prozesses mit stetiger Zeit sind beliebige Funktionen, wohingegen diejenigen

mit diskreter Zeit Folgen reeller Zahlen sind. Für einen diskreten Prozess mit stetiger Zeit sind nur Treppenfunktionen als Trajektorien, welche in Abbildung 2.2 dargestellt sind, denkbar (Beichelt und Montgomery, 2003).

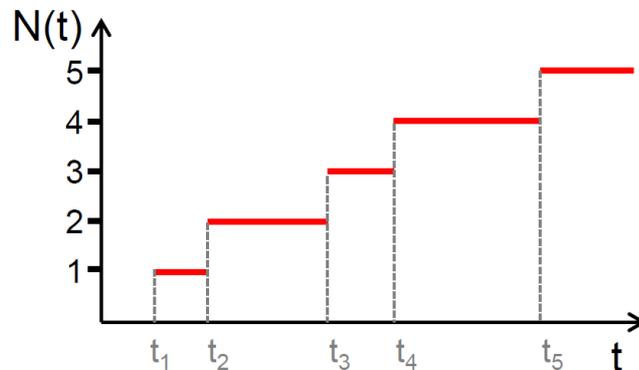


Abbildung 2.2: Treppenfunktion als Trajektorie eines Punktprozesses (Beichelt und Montgomery, 2003)

Für die Untersuchungen innerhalb dieser Arbeit sind diejenigen stochastischen Prozesse von primärer Bedeutung, für welche der Zustandsraum endlich oder abzählbar ist, da diese wertediskreten Prozesse auch Punktprozesse genannt werden. Weitere Details zu stochastischen Prozessen, wie u. a. deren Eigenschaften oder Kenngrößen sowie Beispiele und Aufgaben zu dieser Thematik, können Beichelt und Montgomery (2003) sowie Ross (1997) entnommen werden.

2.3.1 Markierte Punktprozesse

Die nachfolgenden Erläuterungen basieren im Wesentlichen auf (Lacoste et al., 2005) und (Schmidt et al., 2014). Für die klassische Theorie der Punktprozesse und weitere Details sei zum Beispiel auf Cressie (1993), Daley und Vere-Jones (2003), Resnick (2007) sowie van Lieshout (2000) verwiesen.

Punktprozesse gehören zu den stochastischen Prozessen. Diese haben die Aufgabe, aus oftmals großen Datenmengen die Untermenge zu finden, welche ein beobachtetes Phänomen am besten charakterisiert. Sie bieten die Möglichkeit, eine Szene mittels einer ungeordneten Menge an Punkten innerhalb einer begrenzten Region $F \subset R^k$, wobei R^k ein k -dimensionaler Raum ist, zu beschreiben. Für Anwendungen im Bereich der Bildanalyse wird es sich in der Regel um einen 2-dimensionalen Raum handeln, da Operationen auf digitalen Bildern ablaufen. Die Anzahl der Punkte selbst ist ebenfalls eine Zufallsvariable. Für

$n = 1, 2, \dots, N$ ergibt sich die Menge an Konfigurationen $\Omega_n \{x_1, \dots, x_j\}$, welche aus j ungeordneten Punkten $x_i \in F$ besteht. Ein Punktprozess auf F ist dann eine Abbildung X von einem Wahrscheinlichkeitsraum auf die Menge an Konfigurationen $\Omega = \bigcup_{n=1}^{\infty} \Omega_n$, sodass für alle beschränkten *Borel sets* $A \subset F$ (Croft et al., 1991) die Anzahl der in A fallenden Punkte $N_X(A)$ eine endliche Zufallsvariable darstellen (Abb. 2.3a).

Da häufig Punkte allein eine Szene nur schwer beschreiben können, besteht die Möglichkeit, jeden Punkt x_i um zusätzliche Parameter m_i zu erweitern. Auf diese Weise kann der Punkt mit einem geometrischen Objekt $u_i = (x_i, m_i)$ in Verbindung gebracht werden. Folglich wird die Position des Objektes u_i durch den Punkt x_i fixiert (oft der Massenmittelpunkt des Objektes) und das Merkmal m_i enthält weitere Informationen zu dem Objekt, wie beispielsweise die Form oder Größe. Üblicherweise handelt es sich hierbei um mehrere Merkmale m_i , die zu einem Parametervektor zusammengefasst werden. Dieser beschreibt dann ein Objekt eines bestimmten Typs und stellt eine multidimensionale Zufallsvariable dar. Der Prozess des Hinzufügens weiterer Informationen, wie zum Beispiel der Objektparameter zu jedem Punkt, wird als markierter Punktprozess oder auch Objektprozess bezeichnet. Formal ist ein markierter Punktprozess in F , mit Merkmalen m_i in einem Raum M , ein Punktprozess in $F \times M$. Das heißt, dass jeder markierte Punktprozess auch als einfacher Punktprozess aufgefasst werden kann. Anschaulicher ausgedrückt lässt sich ein markierter Punktprozess als ein stochastisches Modell von Konfigurationen einer unbekanntem Anzahl von Objekten des Typs $u_i = (x_i, m_i)$ in F vorstellen. Dabei können sich die Merkmale von Punkt zu Punkt verschieden ausdragen (Abb. 2.3b).

Die Möglichkeiten, die Objekte mithilfe von Merkmalen zu beschreiben, sind zahlreich. Wie bereits in der Motivation aufgeführt, eignen sich zur Repräsentation von Objekten Segmente (Lacoste et al., 2005; Sun et al., 2007), Ellipsen (Descamps et al., 2008; Perrin et al., 2005) oder Rechtecke (Ortner et al., 2007; Schmidt et al., 2014; Tournaire et al., 2007; Tournaire et al., 2010) und finden aus diesem Grund auch immer wieder Anwendung im Zusammenhang mit markierten Punktprozessen. So können beispielsweise Ellipsen über ihren Mittelpunkt, die große und kleine Halbachse sowie ihre Orientierung beschrieben werden. In Abbildung 2.3c ist die Definition eines Rechtecks über dessen Zentrum $c_i = (x_i, y_i)$, seine große Halbachse \vec{v}_i und das Verhältnis der Rechteckseiten $r_i = l_i / L_i$, dargestellt. Derart haben es Tournaire et al. (2010) in ihrem Ansatz realisiert. Somit weicht diese Parametrisierung des Rechtecks von der üblichen Repräsentation, nämlich derjenigen über den Mittelpunkt sowie die Breite, Höhe und Orientierung, ab.

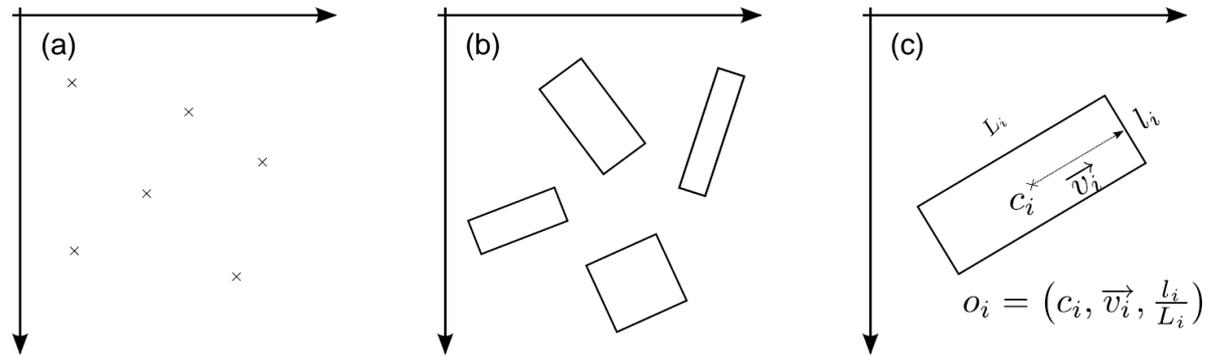


Abbildung 2.3: Realisierungen von Punktprozessen

- (a) Realisierung eines Punktprozesses in F
 (b) Realisierung eines markierten Punktprozesses unter Verwendung von Rechtecken in $F \times M$
 (c) Ein Objekt (Rechteck) des markierten Punktprozesses mit seiner Definition
 (Tournaire et al., 2010)

2.3.1.1 Poisson-Punktprozesse

Der natürlichste Punktprozess ist der homogene Poisson-Punktprozess. Die Anzahl der Punkte des homogenen Prozesses folgt einer diskreten Poisson-Verteilung (Formel 2.1). Diese diskrete Wahrscheinlichkeitsverteilung ist von Bedeutung, damit die Anzahl von Ereignissen modelliert werden kann. Die Wahrscheinlichkeit für die Positionen der Punkte dieses stochastischen Prozesses folgt einer Gleichverteilung. Darüber hinaus sind die Punkte stochastisch unabhängig in F verteilt. Der Parameter λ aus Formel 2.1 beschreibt dabei die erwartete Anzahl an Punkten bzw. Objekten innerhalb von F . (Verdie und Lafarge, 2012).

$$P_\lambda(n) = \frac{\lambda(F)^n}{n!} e^{-\lambda(F)} \quad (2.1)$$

mit

n : Anzahl der Punkte

F : begrenztes Gebiet

λ : Intensität

Mathematisch ausgedrückt ist die Abbildung X genau dann ein homogener Poisson-Prozess in R^k , falls für alle beschränkten Borel sets $A \subset R^k$ die Anzahl der Punkte $N_X(A)$ eine Poisson-Verteilung mit dem Parameter $\lambda > 0$ besitzt. Es werden pro Zeiteinheit exakt λ Sprünge erwartet (Abb. 2.2), weshalb der Erwartungswert der Poisson-Verteilung ebenso $\lambda|A|$ ist. Dabei ist die Sprunghöhe eins und zwischen den Sprüngen sind die Zeiten

exponentialverteilt. Folglich ist der Poisson-Prozess ein diskreter stochastischer Prozess mit stetiger Zeit (vgl. Abb. 2.2). Sind weiterhin A_1, \dots, A_j disjunkte beschränkte Mengen, dann sind $N_X(A_1), \dots, N_X(A_k)$ unabhängig. Aus dieser Definition resultieren die bereits oben erwähnten Eigenschaften, dass n Punkte, die in A fallen, stochastisch unabhängig sind. Zum anderen ist die Wahrscheinlichkeit für die Positionen der Punkte in A gleichverteilt. Demzufolge liegt eine rein zufällige Anordnung der vom Poisson-Prozess erzeugten Punkte, das heißt ohne jegliche Interaktionen untereinander, vor.

Falls weitere Informationen hinsichtlich der Poisson-Prozesse von Interesse sind, wie u. a. deren Eigenschaften, so können diese zum Beispiel in Beichelt und Montgomery (2003) nachgelesen werden. Neben den Poisson- existieren viele weitere Prozesse, auf die im Rahmen dieser Arbeit nicht näher eingegangen wird. Zu nennen ist hier beispielsweise der Neyman-Scott-Prozess, welcher zu den Cluster-Prozessen gehört. Dieser resultiert aus homogenem, unabhängigen Clustering, angewandt auf einen homogenen Poisson-Prozess. Weiterhin gibt es die Hard-Core- sowie Gibbs-Punktprozesse, wobei letzterer aus den Theorien der statistischen Physik hervorgegangen ist. Näheres zu den angesprochenen, anderen sowie Poisson-Punktprozessen ist in Chiu et al. (2013) zu finden.

2.3.1.2 Wahrscheinlichkeitsdichte und Gibbs-Energie

Eine Möglichkeit, um die Qualität einer Punkt- bzw. Objektkonfiguration messen zu können, ist die Formulierung der Wahrscheinlichkeitsdichte $h(\cdot)$ des markierten Punktprozesses. Dies kann mit Bezug auf einen Referenz-Punktprozess erreicht werden, welcher üblicherweise als Poisson-Prozess definiert ist. In der Objekterkennung ist es das Ziel, die wahrscheinlichste Konfiguration von Objekten innerhalb einer Szene bei gegebenen Daten zu finden, welche die Dichte $h(\cdot)$ maximiert. Dabei wird davon ausgegangen, dass für jede Objektkonfiguration eine Wahrscheinlichkeit bestimmt werden kann – die gesamte Wahrscheinlichkeitsverteilung für die Konfiguration jedoch nicht bekannt ist. Indem die Parameter der Objekte $u_i = (x_i, m_i)$, also deren Positionen x_i und Merkmale m_i , mittels zuvor definierter Operationen verändert werden, lässt sich die Objektkonfiguration iterativ optimieren. Die Wahrscheinlichkeitsdichtefunktion $h(\cdot)$ kann mithilfe der Gibbs-Energie $U(\cdot)$ ausgedrückt werden. Die Dichte $h(u)$ für eine Konfiguration u lässt sich unter Nutzung der Gibbs-Gleichung wie folgt formulieren, wobei Z eine Normalisierungskonstante darstellt (Lafarge et al., 2010).

$$h(u) = \frac{\exp(-U(u))}{Z} \quad (2.2)$$

Die Gibbs-Energie kann dabei mittels der Summe zweier Komponenten modelliert werden – die der Datenenergie $U_D(\cdot)$ und zum anderen die der Priorenergie $U_P(\cdot)$. Die erste Energie $U_D(\cdot)$ bewertet die Übereinstimmung der Objektkonfigurationen mit den Daten, wobei die andere $U_P(\cdot)$ Interaktionen zwischen den Objekten berücksichtigt. Hierdurch können bestimmte Konfigurationen auf Basis von Vorwissen befürwortet oder bestraft werden. Kommt es durch die Modifikationen der Objektparameter zu einem Energiezuwachs, so sind die Änderungen mit hoher Wahrscheinlichkeit zu verwerfen, andernfalls anzunehmen. Die Verwerfung erfolgt dabei nicht grundsätzlich, um unter Umständen nicht in einem lokalen Minimum zu verharren. Eine ebenfalls gängige Alternative in der Bildverarbeitung zur Modellierung der markierten Punktprozesse bzw. der Dichte $h(\cdot)$ durch die Gibbs-Energie, stellt die Bayessche Statistik dar. Dieses Modell muss imstande sein, eine Likelihood-Funktion zu besitzen, welche die Verteilung eines Bildes beschreibt. Die Optimierung wird hier durch die a-posteriori-Dichte realisiert, welche die Multiplikation der Likelihood mit einer a-priori-Dichte liefert. Beispiele aus dem Bereich der Bildverarbeitung, die sich mit Punktprozessen im Bayesschen Zusammenhang befassen, sind in u. a. bei Pevatolo und Green (1998) sowie Rue und Hurn (1999) zu finden.

Die optimale Objektkonfiguration $u^* = \{u_1, \dots, u_n\}$ kann bestimmt werden, indem die Wahrscheinlichkeitsdichte $h(\cdot)$ maximiert wird, also $u^* = \arg \max h(\cdot)$. Diese Vorgehensweise entspricht ebenfalls der Konfiguration, welche die Gibbs-Energie $U(\cdot)$ minimiert, das heißt $u^* = \arg \min U(\cdot)$. Diese Technik zum Auffinden des Optimums ist von besonderem Interesse, da hierdurch die Dichte $h(\cdot)$ nicht normalisiert werden muss. Somit ist die komplexe Berechnung der Normalisierungskonstanten Z nicht erforderlich (Lafarge et al., 2010). Hieraus lässt sich für die nicht normalisierte Dichte $h(\cdot)$ Gleichung 2.3 schlussfolgern.

$$h(\cdot) = \exp - U(\cdot) \tag{2.3}$$

Jedoch handelt es sich hierbei keinesfalls um ein gewöhnliches Optimierungsproblem, da die Wahrscheinlichkeitsdichte $h(\cdot)$ in der Regel multi-modal und darüber hinaus im Konfigurationsraum, welcher eine variable Dimension besitzt, definiert ist. Aus diesem Grund wird zum Schätzen des globalen Minimums üblicherweise ein Markov-Chain-Monte-Carlo-Sampler in Kombination mit dem *simulated annealing* verwendet (Kap. 2.3.2.1), um eine Approximation der optimalen Objektkonfiguration $u^* = \{u_1, \dots, u_n\}$ zu finden bzw. die Energie $U(\cdot)$ zu minimieren (Chai et al., 2013). Neben den Monte-Carlo-Verfahren als approximative Verfahren existieren auch exakte Monte-Carlo-Sampling-Methoden (Propp und Wilson, 1996). Diese sind jedoch für die meisten probabilistischen Modelle mit praktischem Interesse schwer zu bewältigen und deshalb nicht tragbar. Aus diesem Grund

muss häufig auf approximative Methoden zur statistischen Inferenz zurückgegriffen werden. Für die Monte-Carlo-Verfahren, welche im nachfolgenden Kapitel näher erläutert werden, erfolgt die Inferenz auf Basis des numerischen Samplings.

2.3.2 Monte-Carlo-Verfahren

Monte-Carlo-Verfahren (MC-Verfahren) sind unterschiedliche stochastische Techniken, welche auf der Verwendung einer großen Anzahl von Zufallszahlen (allg. Zufallsvariablen) und der Wahrscheinlichkeitsrechnung aufbauen. Es handelt sich hierbei in der Regel um rechenintensive Verfahren, die sich auf Probleme in den verschiedensten Bereichen anwenden lassen. Zu diesen zählen beispielsweise die Wirtschafts- und Naturwissenschaften oder auch medizinische Anwendungen. Problemstellungen, die durch deterministische Algorithmen schwer oder überhaupt nicht lösbar sind, werden mithilfe einer MC-Simulation numerisch ermittelt. Im Wesentlichen bildet dabei das Gesetz der großen Zahlen die Basis der MC-Verfahren. Es besagt, dass sich die relative Häufigkeit eines Zufallsergebnisses immer mehr der theoretischen Wahrscheinlichkeit für dieses Ergebnis annähert, je häufiger das entsprechende Zufallsexperiment bei gleichen Voraussetzungen durchgeführt wird (Georgii, 2004).

Die Realisierung dieser Zufallsexperimente erfolgt in der heutigen Zeit mittels Computer, da der Berechnungsaufwand für die große Anzahl der zu generierenden Zufallszahlen hoch ist. Vorreiter des MC-Verfahrens sind die Mathematiker N. Metropolis, J. v. Neumann und S. Ulam in den 1940er Jahren gewesen (Metropolis, 1987). Im September 1949 erschien die erste Veröffentlichung zum MC-Verfahren (Metropolis und Ulam, 1949).

Zufallszahlengeneratoren (Random Number Generators (RNGs)) bilden die Basis der MC-Verfahren. Sie erzeugen eine Sequenz von Zahlen innerhalb eines geschlossenen Intervalls $[\min; \max]$, sodass die Werte dieser Zahlen unvorhersehbar sind. Neben dieser sind noch zwei weiteren Charakteristika eines RNGs von Bedeutung. Ein neuer Wert muss von seinem vorherigen statistisch unabhängig sein. Zum anderen hat die Gesamtverteilung der aus dem Intervall gewählten Zahlen gleichverteilt zu sein (alle Zahlen sind gleichwahrscheinlich und kommen nicht häufiger oder seltener als andere vor). In Abhängigkeit von der Problemstellung kann zwischen nicht-deterministischen und Pseudo-RNGs unterschieden werden. Bei letzteren handelt es sich um deterministische Algorithmen, welche üblicherweise in Software implementiert sind. Diese berechnen eine nur scheinbar zufällige Zahlensequenz und weisen Periodizitäten auf. Das heißt eine Zahlensequenz wird sich in Abhängigkeit von der Länge des Zyklus des Generators früher oder später

wiederholen. Ein solcher Generator benötigt einen Saatpunkt (auch Saatwert bzw. Startwert), um den Zustand des zugrunde liegenden Modells zu initialisieren. Somit weist eine generierte Zahlensequenz zwar Unabhängigkeit und Gleichverteilung als Eigenschaften auf, jedoch ist das gesamte Verhalten des PRNGs vorhersehbar. Diese Generatoren werden häufig in der Numerik oder für Tests verwendet. Eine generelle Einführung zu Zufallszahlen im Hinblick auf die numerische Mathematik kann Press et al. (2007), Kapitel 7, entnommen werden. Nicht-deterministische oder auch wahre RNGs werden beispielsweise für die Schlüsselerzeugung bei Sicherheitsapplikationen genutzt – also dort, wo der deterministische Charakter ein Problem darstellt. Anstelle eines mathematischen Modells zur Erzeugung von Zufallszahlen verwendet dieser Generator eine physikalische Quelle, um Zufälligkeit (Entropie) aus dieser zu gewinnen. Beispiele für solche physikalischen Phänomene sind der Kernzerfall, das Würfeln oder das Werfen einer Münze (Mechalas, 2014).

Die nachfolgenden Ausführungen zu den MC-Methoden beruhen hauptsächlich auf Inhalten aus den Büchern von Bishop (2006) und MacKay (2003) und können bei Interesse dort nachgelesen werden.

Die Ziele der MC-Methoden sind es, eine oder beide der folgenden zwei Aufgaben zu lösen: Das Erzeugen von Samples $\{x^{(r)}\}_{r=1}^R$ aus einer gegebenen Wahrscheinlichkeitsverteilung $P(x)$ und weiterhin das Schätzen der Erwartungswerte von Funktionen $\phi(x)$ gemäß der Verteilung $P(x)$. Die Komponenten von x umfassen diskrete, kontinuierliche oder Kombinationen beider Variablen. Für kontinuierliche Variablen gilt es,

$$E[\phi] = \int P(x) \phi(x) d^N x \quad (2.4)$$

zu berechnen. Für diskrete Variablen wird das Integral durch eine Summe ersetzt. In Abbildung 2.4 ist eine zu Formel 2.4 zugehörige Funktion $\phi(x)$ schematisch dargestellt, deren Erwartungswert bezüglich einer Wahrscheinlichkeitsverteilung $P(x)$ zu schätzen ist. Dabei wird generell angenommen, dass $P(x)$ hinreichend komplex ist, sodass die Erwartungswerte nicht exakt mittels analytischer Techniken geschätzt werden können.

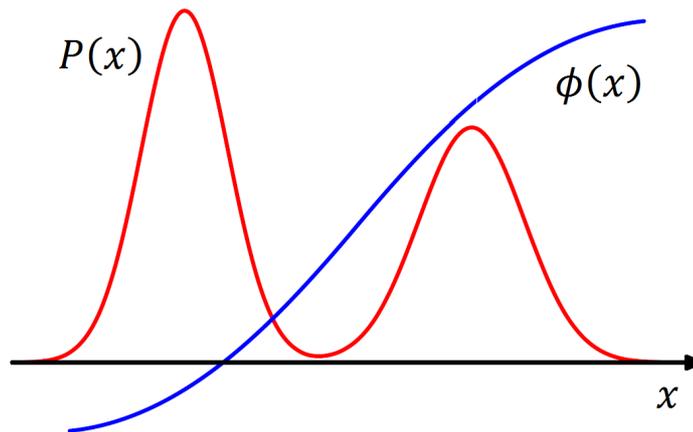


Abbildung 2.4: Darstellung einer Funktion $\phi(x)$ (blau), deren Erwartungswert in Bezug zu einer Verteilung $P(x)$ (rot) zu schätzen ist (Bishop, 2006)

Indem die erste Aufgabe, also das Erzeugen von zufälligen Stichproben $\{x^{(r)}\}_{r=1}^R$ aus $P(x)$, gelöst wird, lässt sich auch das zweite Problem mithilfe dieser lösen. Solange die Samples über $P(x)$ gezogen werden, ist $E[\hat{\phi}] = E[\phi]$. Somit lässt sich Formel 2.4 über eine endliche Summe darstellen (Formel 2.5), die Varianz $var[\hat{\phi}]$ dieses Schätzers ist gegeben durch Formel 2.6.

$$\hat{\phi} = \frac{1}{R} \sum_{r=1}^R \phi(x^{(r)}) \quad (2.5)$$

$$var[\hat{\phi}] = \frac{1}{R} E[(\phi - E[\phi])^2] \quad (2.6)$$

Schlussfolgernd hängt die Genauigkeit der MC-Schätzung (Formel 2.5) nicht von der Dimension von x ab, sondern einzig von der Varianz von ϕ . Dies ist eine wichtige Eigenschaft von MC-Methoden. Somit genügt im Allgemeinen eine kleine Anzahl an unabhängigen Stichproben $\{x^{(r)}\}$, um den Erwartungswert einer Funktion mit ausreichender Genauigkeit zu schätzen. In der Praxis sind dies in etwa zehn bis zwanzig. Jedoch ist es häufig nicht einfach, unabhängige Samples von einer gegebenen Verteilung $P(x)$ zu bekommen, da die Normalisierungskonstante meist unbekannt ist. Zum anderen ist das Ziehen von Stichproben aus $P(x)$ insbesondere in Räumen höherer Dimension anspruchsvoll, da es keinen offensichtlichen Weg gibt, ohne viele oder alle möglichen Zustände durchzählen zu müssen. Korrekte Stichproben werden dazu neigen, von Orten zu kommen, an denen die $P(x)$ groß ist. Hieran knüpft die Frage an, wie diese Orte identifiziert werden können, ohne dass dabei die Wahrscheinlichkeitsverteilung überall auszuwerten ist. Ein anschauliches Beispiel aus MacKay (2003) in Kapitel 29.1 zeigt, dass es unmöglich, jede

Position im Zustandsraum zu besuchen. Der notwendige Berechnungsaufwand hierfür liegt oberhalb einer realistischen Grenze.

Die Überlegung, das zweite Problem zu lösen, führt zum *Uniform Sampling*. Hierbei handelt es sich um eine Methode zum Schätzen des Erwartungswertes einer Funktion. Die Erzeugung von Stichproben aus einer Wahrscheinlichkeitsverteilung ist jedoch nicht möglich. Nur unter der Annahme, dass $\phi(x)$ eine sich nur geringfügig verändernde Funktion ist, die den Charakter von $P(x)$ aufweist, käme diese Methode in Frage. Auch zeigt sich, dass das *Uniform Sampling* für die meisten hochdimensionalen Probleme nicht geeignet ist, es sei denn die Wahrscheinlichkeitsverteilung ist gleichförmig, wie beispielsweise die Gauß-Verteilung (MacKay, 2003). Weitere Informationen zu dieser Thematik und darüber hinaus praktische Erfahrungen zu u. a. Markov-Chain-Monte-Carlo-Verfahren, die in Kapitel 2.3.2.1 vorgestellt werden, können beispielsweise Gilks et al. (1996) sowie Neal (1993) entnommen werden.

Es existieren noch zwei weitere geläufige Sampling-Techniken, das *Importance Sampling* und *Rejection Sampling*, welche ebenfalls gewisse Einschränkungen aufweisen. Von zentraler Bedeutung ist hier die Dimension des Zustandsraumes. Bei beiden Verfahren wird eine einfachere Dichte ergänzt, die üblicherweise als Vorschlagsdichte $Q(x)$ (*proposal density*) bezeichnet wird. Hierdurch können aus dieser Stichproben generiert und $Q(x)$ innerhalb einer Multiplikationskonstanten ausgewertet werden. Gute Resultate lassen sich nur dann erzielen, wenn die Dichte $Q(x)$ (nahezu) gleich mit $P(x)$ ist. Jedoch ist es für komplexe Aufgaben schwierig, ein $Q(x)$ mit dieser Eigenschaft zu erzeugen. Hier knüpfen die sogenannten Markov-Chain-Monte-Carlo-Methoden als ein mächtiges Instrument an. Sie erlauben es, aus einer großen Gruppe an Verteilungen Stichproben zu entnehmen und können weiterhin gut mit hohen Dimensionen umgehen. Das nachfolgende Kapitel befasst sich mit den Markov-Chain-Monte-Carlo-Verfahren. Für Details zum *Importance Sampling* und *Rejection Sampling* sei neben MacKay (2003) auf Andrieu et al. (2003) und Bishop (2006) verwiesen.

2.3.2.1 Markov-Chain-Monte-Carlo-Verfahren

Mithilfe der Markov-Chain-Monte-Carlo- (MCMC-) Verfahren lassen sich spezielle Wahrscheinlichkeitsverteilungen, deren Simulation für gewöhnliche MC-Methoden zu komplex wäre, simulieren. Wie bereits oben beschrieben sind dies Verteilungen, deren Dichten komplizierte Normierungsterme enthalten, oder Wahrscheinlichkeitsverteilungen einer mehrdimensionalen Zufallsvariable. Da in beiden Fällen die direkte Simulation aus der marginalen Verteilung bei vielen Problemen nicht möglich ist (Geyer, 1998), wird stattdessen

eine Markov-Kette generiert. Die stationäre Verteilung dieser Kette stellt die unbekannte Verteilung dar. Anders als zuvor dient die Realisation der Markov-Kette als Simulationsstichprobe. Aus diesem Grund hat Geyer (1991) die Bezeichnung MCMC gewählt. Für die Eignung von MCMC-Verfahren innerhalb von Anwendungen ist es von Bedeutung, wie schnell die simulierte Verteilung gegen die gesuchte Verteilung konvergiert. Hierdurch werden Anhaltspunkte für die Auswahl eines Verfahrens gewonnen und es kann die Länge des Simulationslaufes festgelegt werden. Vorreiter der MCMC-Verfahren waren Metropolis et al. (1953). Eine Verallgemeinerung des 1953 veröffentlichten Papers lieferte Hastings (1970) mit dem Metropolis-Hastings-Algorithmus. Eine Erweiterung zur Konstruktion eines flexiblen, umkehrbaren Markov-Ketten-Samplers, welcher zwischen Parameterunterräumen verschiedener Dimension hin und her springen kann, gibt Green (1995) mit dem Reversible-Jump-Markov-Chain-Monte-Carlo-Verfahren (Dreesman, 1998).

Markov-Ketten

Markov-Ketten, gelegentlich auch als zeitdiskreter Fall der Markov-Prozesse bezeichnet, eignen sich gut zur Modellierung von zufälligen Zustandsänderungen eines Systems. Wie bei den stochastischen Prozessen (Kap. 2.3), kann zwischen Markov-Ketten in diskreter und stetiger Zeit bzw. Zustandsraum unterschieden werden. Der Poisson-Prozess (Kap. 2.3.1.1) stellt ein Beispiel einer Markov-Kette in stetiger Zeit für abzählbare Zustandsräume dar. Innerhalb des Poisson-Prozesses folgt die Anzahl der Punkte einer Exponentialverteilung, der Poisson-Verteilung (Formel 2.1), welche wegen ihrer Gedächtnislosigkeit eine wichtige Rolle spielt. Diese wird auch als Markov-Eigenschaft bezeichnet und besagt, dass die Zukunft eines stochastischen Prozesses unabhängig von der Vergangenheit bei gegebenem gegenwärtigen Zustand ist. Somit können unter der alleinigen Kenntnis des aktuellen Zustands genauso gut Vorhersagen über die Zukunft gemacht werden, als wäre die komplette vorherige Entwicklung des Prozesses schon bekannt. Hierdurch wird sich die Kette häufiger in den wichtigsten Gebieten aufhalten. Anders ausgedrückt sorgt die Markov-Kette dafür, dass die Samples $x^{(t)}$ Stichproben, die aus der Zielverteilung $P(x)$ entnommen wurden, simulieren. An dieser Stelle wird von der vorherigen Notation $x^{(r)}$ abgewichen und stattdessen der Exponent $t = 1, \dots, T$ verwendet, um die Folge der Zustände in einer Markov-Kette zu benennen. Diese Darstellung wird genutzt, da nun keine unabhängigen, sondern vielmehr voneinander abhängige Stichproben einer Verteilung vorliegen. So werden bei einer Metropolis-Hastings-Simulation, die im nächsten Abschnitt behandelt wird, von T Iterationen nicht T unabhängige Samples aus der Zielverteilung $P(x)$ erzeugt (Andrieu et al., 2003).

Werden Markov-Ketten im endlichen Zustandsraum Z , in dem die $x^{(t)}$ nur eine gewisse Anzahl n an diskreten Werten $x^{(t)} \in Z = \{x_1, x_2, \dots, x_n\}$ annehmen können, eingeführt, so handelt es sich um eine Markov-Kette, falls

$$P(x^{(t+1)} | x^{(t)}, x^{(t-1)}, \dots, x^{(1)}) = T(x^{(t+1)} | x^{(t)}) \quad (2.7)$$

gilt. Folglich hängen die Übergangswahrscheinlichkeiten innerhalb der Übergangsmatrix T nur vom aktuellen Zustand $x^{(t)}$ und nicht von der Historie der Kette $\{x^{(1)}, x^{(2)}, \dots, x^{(t-1)}\}$ ab. Die Markov-Kette heißt homogen, falls T für alle Zeitpunkte t unverändert bleibt. In diesem Zusammenhang wird auch von einer Kette mit stationären Übergangswahrscheinlichkeiten gesprochen. Dieses Attribut der Markov-Ketten, dass die bedingte Verteilung von $x^{(t+1)}$, bei gegebenem $x^{(t)}$, für alle t identisch ist, ist auch in zahlreicher Literatur zu den MCMC-Verfahren zu finden (Andrieu et al., 2003). Zu nennen sind hier beispielsweise Meyn und Tweedie (1993).

In Abbildung 2.5 ist der Übergangsgraph bzw. das Prozessdiagramm für ein Beispiel einer Markov-Kette mit drei Zuständen ($n = 3$) dargestellt. Die Übergangsmatrix T und der Wahrscheinlichkeitsvektor für den Anfangszustand $\mu(x^{(1)})$ seien für dieses Beispiel wie folgt gegeben. Die Spaltensummen ergeben dabei jeweils Eins.

$$T = \begin{bmatrix} 0 & 0 & 0.6 \\ 1 & 0.1 & 0.4 \\ 0 & 0.9 & 0 \end{bmatrix} \quad ; \quad \mu(x^{(1)}) = \begin{pmatrix} 0.5 \\ 0.2 \\ 0.3 \end{pmatrix}$$

Wird das Produkt $T \cdot \mu(x^{(1)})$ gebildet, ergibt sich für den nachfolgenden Zustand $\mu(x^{(2)}) = (0.18 \ 0.64 \ 0.18)^T$. Wird das Prinzip der Multiplikation des jeweils neuen Zustandes mit der Transitionsmatrix T wiederholt, so konvergiert das Produkt unabhängig von der gewählten Anfangsverteilung $\mu(x^{(1)})$ gegen die Zielverteilung $P(x) = (0.2 \ 0.4 \ 0.4)^T$. Die Voraussetzung für die Konvergenz der Verteilung $P(x)$ ist, dass die Übergangsmatrix T gewisse Eigenschaften befolgt (Andrieu et al., 2003).

Zu diesen Eigenschaften, an denen sich auch das Verhalten der Markov-Ketten analysieren lässt, zählen vor allem die Irreduzibilität, Rekurrenz, Aperiodizität und Reversibilität. Die Attribute werden im Folgenden nur knapp erläutert. Weitere Informationen sind zum Beispiel im dritten und vierten Kapitel von Gilks et al. (1996) zu finden. Die Irreduzibilität ist mit der Aperiodizität eine der wichtigsten Eigenschaften für die Konvergenz gegen eine stationäre Verteilung. Erstere besagt, dass jede Teilmenge des Zustandsraumes von jedem Anfangszustand aus erreichbar ist. Somit ist die Wahrscheinlichkeit, in einer endlichen Zeit vom einem zum anderen Zustand zu kommen, echt positiv. Aperiodizität beinhaltet, dass die

Markov-Kette kein zyklisches Verhalten aufweist. Hierdurch ist es nicht möglich, exakte Vorhersagen über den Aufenthaltsort einer Teilmenge zu einem bestimmten Zeitpunkt vorzunehmen. Aufgrund dieser Eigenschaft wird unterbunden, dass verschiedene Zustände im Raum in einem regelmäßigen periodischen Abstand erreicht werden. Eine Markov-Kette ist rekurrent, wenn die entsprechenden Teilmengen ausgehend von allen Startwerten fast sicher unendlich oft erreicht werden können. Folglich charakterisiert die Rekurrenz das Langzeitverhalten einer Markov-Kette. Ist die gemeinsame Verteilung mehrerer Zufallsvariablen bei Zeitumkehr invariant, so wird von Reversibilität gesprochen (Dreesman, 1998).

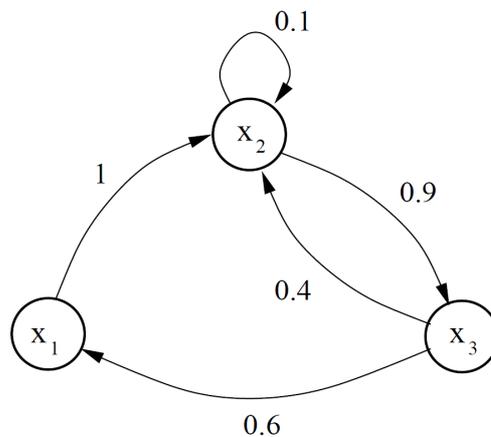


Abbildung 2.5: Prozessdiagramm für die beispielhafte Markov-Kette mit $Z = \{x_1, x_2, x_3\}$ (Andrieu et al., 2003)

Es existieren verschiedene Algorithmen für MCMC-Sampler, deren Basis die eben erläuterte Erzeugung einer Markov-Kette ist. Der nachfolgende Abschnitt wird sich mit dem Metropolis-Hastings-Algorithmus (Metropolis et al., 1953; Hastings, 1970) befassen. Auf andere gängige MCMC-Methoden, wie das Gibbs Sampling (Geman und Geman, 1984) sowie das Slice Sampling (Neal, 2003), wird im Rahmen dieser Arbeit nicht weiter eingegangen.

Metropolis-Hastings-Verfahren

Das Metropolis-Hastings-Verfahren (MH-Verfahren) wird nachfolgend auf Grundlage von MacKay (2003) erläutert. Hierbei hängt die Dichte, welche im Folgenden als Vorschlagsdichte $Q(x)$ bezeichnet wird, vom aktuellen Zustand $x^{(t)}$ und nicht von den vorherigen Zuständen ab (Markov-Eigenschaft). Neben $x^{(t)}$ beinhaltet die Vorschlagsdichte noch einen möglichen neuen Zustand x' , sodass für die Dichte $Q(x'|x^{(t)})$ folgt. Dabei kann die Vorschlagsdichte jede beliebige stationäre Dichte sein (zum Beispiel eine Gauß-

Verteilung), von welcher Samples entnommen werden können. Anders als beim *Importance Sampling* sowie *Rejection Sampling* ist es nicht notwendig, dass eine Ähnlichkeit zur Wahrscheinlichkeitsverteilung $P(x)$ besteht, damit der Algorithmus praktikabel ist. Abbildung 2.6 zeigt ein Beispiel für zwei verschiedene Zustände $x^{(1)}$ und $x^{(2)}$ einer Vorschlagsdichte im eindimensionalen Fall. Innerhalb dieser Abbildung ändert sich die Form von $Q(x'|x^{(t)})$ bei Modifikation des Zustandes, was im praktischen Fall für die Vorschlagsdichte nicht typisch ist. Unter der Annahme, dass $P^*(x)$ für jedes x berechnet werden kann, wird ein vorläufiger neuer Zustand x' aus $Q(x)$ erzeugt. Darüber hinaus wird die Akzeptanzwahrscheinlichkeit a (Gleichung 2.8) bestimmt, um zu entscheiden, ob x' angenommen wird

$$a = \frac{P^*(x') \cdot Q(x^{(t)}|x')}{P^*(x^{(t)}) \cdot Q(x'|x^{(t)})} \quad (2.8)$$

Falls $a \geq 1$ wird der neue Zustand in jedem Fall akzeptiert, andernfalls mit der Wahrscheinlichkeit a . Weiterhin führt eine Akzeptanz zu $x^{(t+1)} = x'$, eine Ablehnung zu $x^{(t+1)} = x^{(t)}$. Somit wird $x^{(t)}$, sollte der neue Zustand abgelehnt werden, erneut auf die Liste der Stichproben geschrieben. Für den Fall, dass es sich bei der Vorschlagsdichte beispielsweise um eine über dem aktuellen Zustand $x^{(t)}$ zentrierte Gauß- oder eine andere einfache symmetrische Verteilung handelt, so ist der hintere Faktor aus Gleichung 2.8 Eins. Eben dies ist die von Metropolis et al. (1953) entwickelte Metropolis-Methode, bei welcher lediglich der Wert der Zielverteilung $P(x)$ an den beiden Punkten verglichen werden muss.

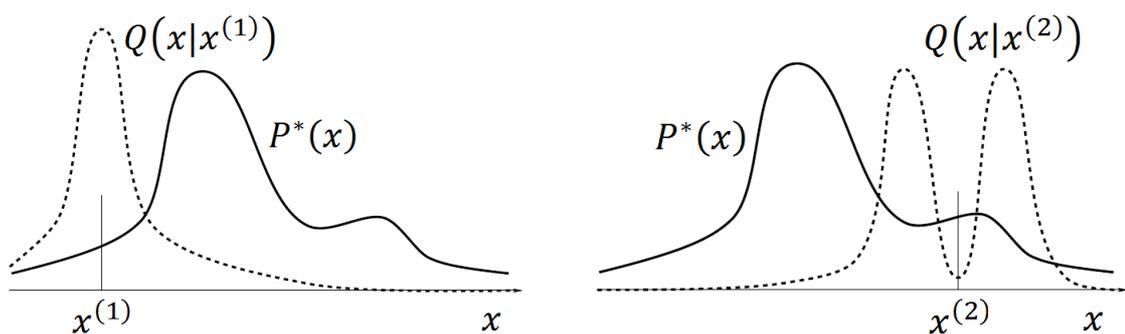


Abbildung 2.6: MH-Algorithmus mit Vorschlagsdichte $Q(x'|x^{(t)})$ für zwei unterschiedliche Zustände $x^{(1)}$ und $x^{(2)}$ (MacKay, 2003)

Wie im vorherigen Abschnitt zu den Markov-Ketten bereits angemerkt, wird bei den MCMC-Methoden eine Sequenz von Zuständen $\{x^{(t)}\}$ erzeugt, wobei jede Stichprobe $x^{(t)}$ eine Wahrscheinlichkeitsverteilung besitzt, die vom vorherigen Wert $x^{(t-1)}$ abhängt. Durch diese

Abhängigkeit der aufeinanderfolgenden Samples muss der Algorithmus unter Umständen erheblich lange laufen, um unabhängige Stichproben von der Zielverteilung zu erzeugen. Dies führt einerseits zur der Problematik bei der Beurteilung, wie lange für die Erzeugung der unabhängigen Samples gewartet werden muss. Ferner ist auch die Frage, wann die MCMC-Methode konvergiert ist, also wann die Wahrscheinlichkeitsverteilung von $x^{(t)}$ gegen $P(x) = P^*(x)/Z$ für jede $Q(x'|x) > 0 \forall x, x'$ mit $t \rightarrow \infty$ strebt, als kritisch zu bewerten.

Beim MH-Verfahren besteht im Gegensatz zum *Importance Sampling* und *Rejection Sampling* keine derart starke Abhängigkeit von der Dimension des Zustandsraumes. Resultate werden somit in einer kürzeren Zeit geliefert. Jedoch kann es aufgrund von Zufallsbewegungen (*random walk*) zu langen Simulationen kommen. Näheres zu dieser Thematik und ein anschauliches Beispiel, wie langsam eine Zufallsbewegung den Zustandsraum erkunden kann, sind bei MacKay (2003) in Kapitel 29.4 zu finden. Im übernächsten Abschnitt werden Möglichkeiten genannt, mit denen die Zufallsbewegungen unterdrückt werden können. Eine von diesen ist das *simulated annealing*. Hierdurch kann bei MCMC-Verfahren die Zeit bis zur Konvergenz und die benötigte Zeit, um unabhängige Samples zu erzeugen, verkürzt werden. Zuvor wird im nachfolgenden Abschnitt auf eine Erweiterung der MCMC-Methoden eingegangen, das sogenannte Reversible-Jump-Markov-Chain-Monte-Carlo-Verfahren (Green, 1995). Bei dieser Methode muss die Dimension von Objekten nicht notwendigerweise fest sein.

Reversible-Jump-Markov-Chain-Monte-Carlo-Verfahren

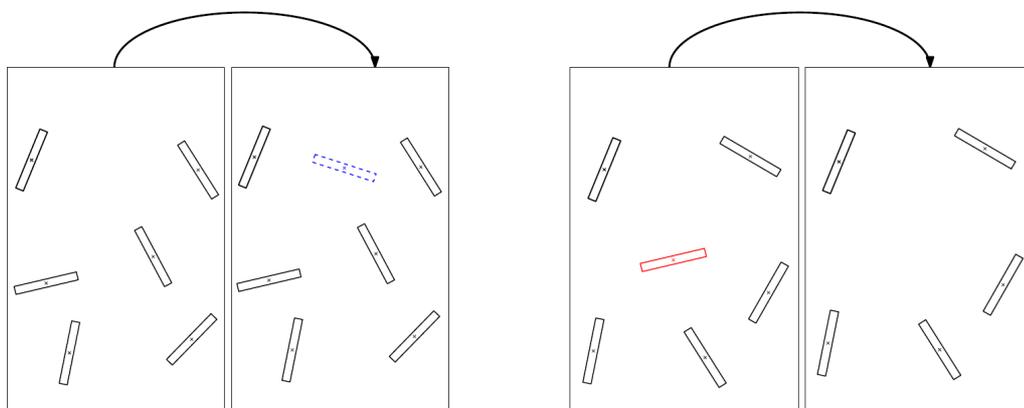
Das in Abbildung 2.6 gezeigte Beispiel beschränkt sich auf den eindimensionalen Fall. Der Übergang auf eine konstante höhere Dimension der Objektkonfiguration ist für den MH-Algorithmus ebenfalls handhabbar, jedoch ist dabei die Anzahl an Objekten innerhalb der Szene immer fest. Hier setzen die Reversible-Jump-Markov-Chain-Monte-Carlo-Verfahren (RJMCMC-Verfahren) an, welche die Modellierung von Szenen mit variierender Anzahl an Objekten erlauben. Diese Erweiterung des MH-Algorithmus, um mit Zustandsräumen verschiedener Dimensionen umgehen zu können, wurde erstmals von Green (1995) vorgeschlagen. Änderungen der Dimension zwischen Stichproben in einer Markov-Kette werden als Dimensionssprünge bezeichnet (Smith, 2011).

Der RJMCMC-Sampler erlaubt die Simulation einer Markov-Kette $X^{(t)}$ im Zustandsraum, welche zwischen verschiedenen Dimensionen hin und her springt. Dies wird durch die Definition einer Menge an Bewegungen/Änderungen, den sogenannten reversiblen Sprüngen (*reversible jumps*), erreicht. Theoretisch können beliebige Arten an Änderungen, solange sie umkehrbar sind, definiert werden. Diese Flexibilität macht die RJMCMC-

Methoden zu einem mächtigen Werkzeug. Reversibilität bedeutet beispielsweise, dass es in einer späteren Änderung möglich sein muss, zum vorherigen Zustand zurückkehren zu können. In jeder Iteration t schlägt der Sampler eine Änderung der aktuellen Objektkonfiguration aus jener vordefinierten Menge an Bewegungen (*jumps*) vor. Dabei ist jede Art der Änderung einer Dichtefunktion Q_m zugehörig, welche auch als Kernel (Kern) bezeichnet wird. Dieser Prozess führt von einer Objektkonfiguration $X^{(t)}$ zu einer neuen Konfiguration $X^{(t+1)}$, gemäß einer Wahrscheinlichkeit $Q_m(X^{(t)} \rightarrow X^{(t+1)})$. Es handelt sich hierbei um kleine Sprünge zwischen Räumen variabler Dimension, wobei lediglich ein einzelnes Objekt der globalen Konfiguration von einem neuen Vorschlag betroffen sein wird. Die Änderung bzw. neue Konfiguration wird mit einer gewissen Akzeptanzwahrscheinlichkeit α (Formel 2.9), die von der Energievariation zwischen $X^{(t)}$ und $X^{(t+1)}$ abhängt, angenommen. Die Auswahl eines Kernels Q_m kann dabei zufällig oder entsprechend einer vorgegebenen Wahrscheinlichkeit erfolgen. U ist die Gibbs-Energie aus Kapitel 2.3.1.2.

$$\alpha = \min \left(1, \frac{Q_m(X^{(t+1)} \rightarrow X^{(t)})}{Q_m(X^{(t)} \rightarrow X^{(t+1)})} \cdot \exp - \left(U(X^{(t+1)}) - U(X^{(t)}) \right) \right) \quad (2.9)$$

Jedem der Kernel Q_m sind gewöhnlich spezifische Modifikationen zugehörig. Auf der einen Seite existieren die Geburts- und Todeskernel (*birth- and death-kernels*), welche für das Hinzufügen bzw. Entfernen von Objekten im Raum möglicher Konfigurationen zuständig sind. Diese folgen einer Poisson-Verteilung (Kap. 2.3.1.1). Geburten (Abb. 2.7, links) entsprechen Sprüngen in Räume höherer Dimension, Tode (Abb. 2.7, rechts) in Räume niedrigerer Dimension.



**Abbildung 2.7: Geburtskern (links) und Todeskernel (rechts)
(Tourenaire et al., 2007)**

Zum anderen gibt es Kernel, die für die Veränderung der Parameter eines Objektes zuständig sind – häufig als Änderungskernel (*perturbation kernels*) bezeichnet. Diese sind wichtig, um die Positionierung der Objekte anzupassen. Exemplarisch genannt werden können Bewegungen wie die Translation, Rotation oder Dilatation (Abb. 2.8). An dieser Stelle sei auf die Eigenschaften der Markov-Ketten hingewiesen. So ist es u. a. von Bedeutung, dass jede Konfiguration im Zustandsraum innerhalb einer endlichen Anzahl an Modifikationen wieder erreicht werden kann (Irreduzibilität).

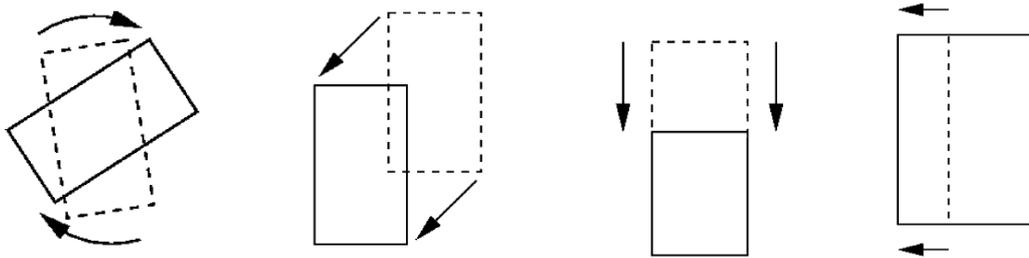


Abbildung 2.8: Änderungskernel
v.l.n.r.: Rotation, Translation, Dilatation (Länge), Dilatation (Breite)
(Ortner et al., 2007)

Der Algorithmus des RJMCMC-Samplers wird im Folgenden kurz zusammengefasst:

Ausgehend von einer Startkonfiguration $X^{(0)}$, die aufgrund der Eigenschaften der Markov-Ketten zufällig gewählt werden kann, werden die nachfolgenden Schritte solange wiederholt, bis ein Konvergenzkriterium erfüllt ist. Dies sind (1) zufällig einen Kernel Q_m entsprechend einer bestimmten Wahrscheinlichkeit auswählen, (2) eine neue Objektkonfiguration $X^{(t+1)}$, ausgehend von $X^{(t)}$ erzeugen, (3) die Akzeptanzwahrscheinlichkeit bzw. -rate α gemäß Formel 2.9 bestimmen und (4) die neue Konfiguration $X^{(t+1)}$ entweder mit der Wahrscheinlichkeit α akzeptieren oder mit $1 - \alpha$ zurückweisen (Mallet et al., 2010; Smith, 2011; Verdie und Lafarge, 2012).

Simulated Annealing

Es existieren verschiedene Möglichkeiten, das Konvergenzverhalten bei MCMC- bzw. RJMCMC-Methoden zu beschleunigen. Eine ist die Hybrid-Monte-Carlo-Methode (Duane et al., 1987), auch als Hamiltonian-Monte-Carlo-Verfahren bezeichnet, welches in kontinuierlichen Zustandsräumen im Zusammenhang mit der MH-Methode anwendbar ist. Zur Reduktion der Zufallsbewegungen werden Informationen von Gradienten genutzt, die auf die Richtung, in welcher sich Zustände mit größerer Wahrscheinlichkeit befinden, hinweisen.

Für das Gibbs-Sampling gibt es die Methode der *overrelaxation* (Adler, 1981), um die benötigte Zeit zur Erzeugung von unabhängige Samples zu reduzieren.

Der Algorithmus des *simulated annealing* geht auf die Beobachtung zurück, dass das Lösen von kombinatorischen Optimierungsproblemen analog zum Abkühlen von Stoffen in der Festkörperphysik ist (Metropolis et al., 1953; Kirkpatrick et al., 1983). Ziel der physikalischen Abkühlung ist es, einen minimalen Energiezustand der Feststoffe durch deren Schmelzen in einem Wärmebad zu erreichen. Durch die langsame Verringerung der Temperatur können sich die Teilchen des Stoffes in einer kristallinen Gitterstruktur neu anordnen, welche einem minimalen Energiezustand für den Feststoff entspricht. Jenes Prinzip machten sich Cerny (1985) und Kirkpatrick et al. (1983) zunutze und schlugen die Kombination eines MCMC-Algorithmus mit einem Temperaturparameter vor, um das globale Minimum einer Energiefunktion zu finden. Die Temperatur gleicht einer Wahrscheinlichkeit, mit der sich ein Zwischenergebnis der Optimierung auch verschlechtern darf. Wesentlicher Vorteil hiervon ist, dass ein lokales Optimum wieder verlassen werden kann um ein besseres zu finden. Dies ist bei einem lokalen arbeitenden Algorithmus nicht möglich. Seitdem die Analogie entdeckt wurde, findet sie Verwendung in zahlreichen Optimierungsproblemen, die auch Anwendungen in der Bildanalyse beinhalten. Diese sind beispielsweise bei Winkler (2003) zusammengestellt (Perrin et al., 2005; Varanelli, 1996).

Indem Formel 2.9 um den Temperaturparameter $T^{(t)}$ erweitert wird, lässt sich das Optimum der Energie finden

$$\alpha = \min \left(1, \frac{Q_m(X^{(t+1)} \rightarrow X^{(t)})}{Q_m(X^{(t)} \rightarrow X^{(t+1)})} \cdot \exp - \left(\frac{U(X^{(t+1)}) - U(X^{(t)})}{T^{(t)}} \right) \right). \quad (2.10)$$

$T^{(t)}$, als eine Sequenz von Temperaturen, strebt gegen Null, während t gegen unendlich strebt. Zu Beginn des Algorithmus, wenn die Temperatur hoch ist, ist das Verfahren nicht selektiv. Bei abnehmender Temperatur wird der Prozess selektiver, das heißt Konfigurationen mit einer hohen Dichte werden bevorzugt. Der Abkühlungsprozess kann über eine logarithmische Abnahme realisiert werden, welche theoretisch Konvergenz zum globalen Optimum für jede beliebige Anfangskonfiguration $X^{(0)}$ gewährleistet. Aufgrund der hohen Berechnungszeiten wird in der Praxis üblicherweise ein schnelleres geometrisches Schema zur Abkühlung verwendet. Dieses liefert ein approximatives Resultat, welches nahe an der optimalen Lösung liegt. Die zwei wesentlichen Parameter innerhalb des *simulated annealing* sind die Anfangs- $T^{(0)}$ und Endtemperatur $T^{(end)}$. Weitere Informationen können zum Beispiel Salamon et al. (2002) sowie Van Laarhoven und Aarts (1987) entnommen werden (Lafarge et al., 2006).

Das nachfolgende Beispiel (Abb. 2.9) aus Lafarge et al. (2008) zeigt die Arbeitsweise des *simulated annealing* anhand zweier Simulationsbeispiele. Zu Beginn, wenn die Temperatur hoch ist, werden die Maxima der Dichtefunktion erforscht und Konfigurationen mit einer hohen Dichte favorisiert. In diesem Beispiel bestehen die Gebäude aus allen möglichen 3D-Modellen. Mit zunehmender Zeit (abnehmender Temperatur) wird der Prozess selektiver, sodass die Gebäude nun strukturiert werden können. Wenn die Konfiguration nahe dem Optimum ist, was für niedrige Temperaturen gilt, wird es kaum mehr zu Veränderungen kommen. Für diesen Fall heißt dies eine detaillierte Anpassung der 3D-Block-Parameter, welche das Gebäude modellieren.

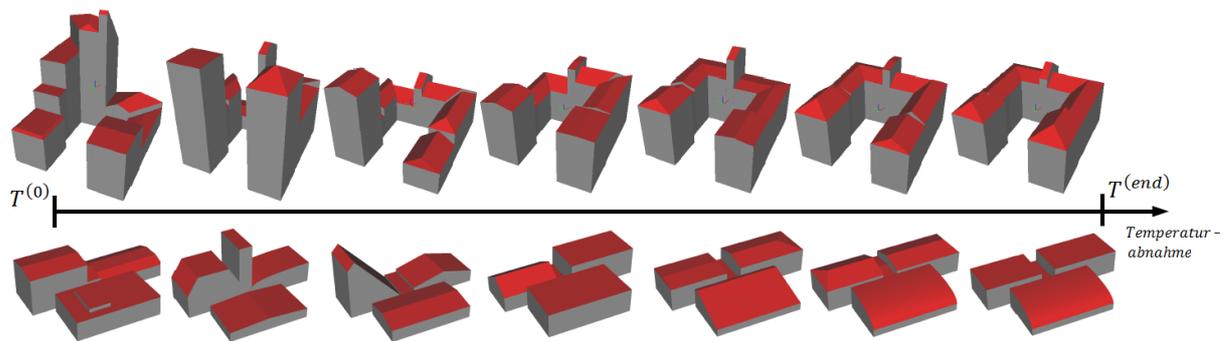


Abbildung 2.9: Entwicklung der Konfiguration bei abnehmender Temperatur (Lafarge et al., 2008)

3 Methodik

Im Folgenden sollen Untersuchungen zur automatischen Detektion von Flüssen durchgeführt werden. Die Beschreibung der Daten, auf denen die Analysen erfolgen, ist in Kapitel 4.1 zu finden. Grundlage für die Untersuchungen bildet der von Schmidt et al. (2015) entwickelte Algorithmus, welcher sich gut auf die eigene Problemstellung anwenden lässt. Dessen Methodik und vorkommende Parameter werden in Kapitel 3.1 vorgestellt. Das nachfolgende Kapitel 3.2 befasst sich mit Untersuchungen zu Zufallszahlengeneratoren. Betrachtete Aspekte sind hier die Initialisierung des Generators sowie das Laufzeitverhalten eines Pseudo- gegenüber einem nicht-deterministischen Zufallszahlengenerators. Insbesondere wird in diesem Kapitel auch der Frage nachgegangen, ob Parameter entsprechend einer vorgegebenen Verteilungsfunktion gewählt werden. In Kapitel 3.3 wird die Energiefunktion um eine 3D-Komponente erweitert, wodurch in das bereits vorhandene Modell für die Priorenergie weiteres Vorwissen einfließt. Hierbei wird die Funktionsweise des Fließmodells anhand von Beispielen verdeutlicht.

3.1 Methode zur Flussdetektion

Die innerhalb dieser Arbeit verwendete Methode zur Flussdetektion basiert auf dem von Schmidt et al. (2015) entwickelten Modell, welches einen graphenbasierten Ansatz zur automatischen Detektion von Prielen mittels markierter Punktprozesse (Kap. 2.3.1) nutzt. Dieser Algorithmus wird im Folgenden kurz beschrieben. Dabei soll auf einige Aspekte, die Bestandteil der Untersuchungen in den nachfolgenden Kapiteln sind, hingewiesen werden. Zu diesen zählen beispielsweise Parameter des Modells sowie Kernel, die Änderungen an der Objektkonfiguration ermöglichen. Ferner wird die Energiefunktion aufgezeigt, welche in Kapitel 3.3 um einen weiteren Term für die Priorenergie ergänzt wird. Sind neben den hier aufgeführten Gesichtspunkten noch weitere Details notwendig, können diese dem Paper entnommen werden.

Objektmodell

Zu detektierende Prielsysteme aus DGMs werden bei Schmidt et al. (2015) durch einen ungerichteten, azyklischen Graphen repräsentiert. Die Verwendung eines Graphen ermöglicht den Erhalt einer Netzwerkstruktur, da jeder Knoten des Graphen entweder zu einem Kreuzungs- oder Endpunkt gehört. Die Knoten bzw. Objekte, die über ihre Bildkoordinaten beschrieben werden, stellen Kreuzungspunkte (Chai et al., 2013) des netzwerkartigen Systems dar. Somit werden unter diesen sowohl Kreuzungs- als auch Endpunkte verstanden, von denen eine variierende Anzahl n von Segmenten s_j ausgeht (Abb. 3.1). Idealerweise sollten die Segmente die Priele im DGM repräsentieren. Hierbei kann ein Schwellwert für die maximale Anzahl an abgehenden Richtungen pro Kreuzungspunkt eines neuen oder bereits vorhandenen Knotens bei einem Geburtsereignis gesetzt werden. Im Graphen sind Knoten über Kanten verbunden. Eine Kante entspricht dabei einem Segment s_j , welches über seine Breite b_j und seine Richtung β_j beschrieben ist. Die Richtung der Mittelachse eines Segments ist über den entgegen des Uhrzeigersinns gezählten Winkel relativ zur positiven y-Achse definiert.

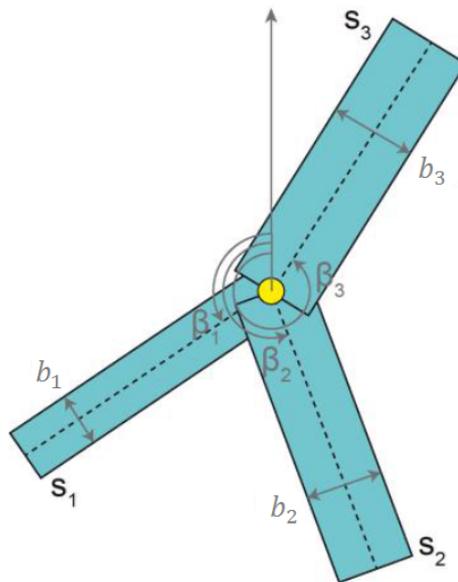


Abbildung 3.1: Das dem Algorithmus zugrunde liegende Objektmodell des Kreuzungspunktes (mit $n=3$) (Schmidt et al., 2015)

Änderungen der Objektkonfiguration

Um eine optimale Objektkonfiguration zu finden, wird der Graph iterativ generiert und den Daten durch ein RJMCMC-Sampling in Kombination mit dem *simulated annealing* angepasst (Kap. 2.3.2.1). Die Methode stellt vier Änderungsarten der Konfiguration bereit, wobei die Modifikationen mithilfe zugehöriger Kernel ermöglicht werden.

Zum einen erlauben **Geburts-** und **Todes-Kernel** das Hinzufügen bzw. Entfernen eines Objektes zur bzw. von der aktuellen Objektkonfiguration (Geburt-Todes-Kernel). Für den Fall einer Geburt werden die Position und die Anzahl der Segmente eines neuen Objektes generiert. Hierbei können Parameter für die minimale und maximale Breite der Segmente s_j gesetzt werden, die im Folgenden mit b_{min} bzw. b_{max} bezeichnet werden. Ferner ist es möglich, einen Schwellwert z für den maximalen Höhen- bzw. Grauwert, bis zu welchem potentielle Orte für ein Geburtseignis gewählt werden, festzulegen. Auf diese Weise kann das Verfahren an die jeweiligen Daten angepasst werden. Der Kreuzungspunkt wird dem Graphen als ein Knoten hinzugefügt. Anschließend wird für Kreuzungspunkte innerhalb einer lokalen Nachbarschaft mit dem vorgegebenen Radius r geprüft, ob das Hinzufügen einer Kante zwischen diesen und dem neuen Knoten möglich ist. Grundlage hierfür bilden drei Kriterien, auf welche an dieser Stelle nicht weiter eingegangen wird. Sollte der benachbarte Knoten den drei Bedingungen genügen, werden die beiden Knoten über eine Kante miteinander verbunden. Beim Todesevent wird ein Knoten zuzüglich seiner Kanten aus dem Graphen entfernt.

Darüber hinaus lassen sich die Parameter eines Objektes der derzeitigen Konfiguration mithilfe der **Translations-** und **Modifikations-Kernel** verändern. Bei ersterem wird für einen zufällig gewählten Knoten im Graphen ein Verschiebungsvektor erzeugt. Sollte dieser dieselben Kriterien wie beim Geburtsevent nicht verletzen, so wird der Kreuzungspunkt verschoben und die Position und Richtungen der Segmente angepasst. Gleichmaßen werden auch beim Modifikations-Kernel ein zufälliger Kreuzungspunkt und eines seiner Segmente bestimmt, dessen Breite dann verändert wird.

Energiefunktion

In jedem Iterationsschritt wird die aktuelle Objektkonfiguration X_t an Knoten und Kanten auf Basis einer globalen Energiefunktion $U(X_t)$ verändert. Diese bewertet die Übereinstimmung des derzeitigen Zustandes mit einem vordefinierten Modell für die Priele und wird während des Samplings minimiert. Die Gibbs-Energie setzt sich aus zwei Teilen, der Daten- und der Priorenergie, zusammen. Der relative Einfluss kann durch einen Parameter β modelliert werden, um eine den Daten angepasste Gewichtung vornehmen zu können

$$U(X_t) = \beta \cdot U_D(X_t) + (1 - \beta) \cdot U_P(X_t). \quad (3.1)$$

Hierbei entspricht $U_D(X_t)$ der Datenenergie, welche die Kongruenz der Objektkonfiguration mit den Eingangsdaten vergleicht. Die Energie wird berechnet, indem Gradienten in den Daten berücksichtigt werden. Das von Schmidt et al. (2015) implementierte Modell favorisiert

hohe Magnituden der Gradienten am Rand der Priele bzw. Segmente, da Priele durch lokal geringere Höhen im Vergleich zur Umgebung gekennzeichnet sind.

Die Priorenergie $U_P(X_t)$ des Modells, mithilfe derer bestimmte Konfigurationen auf Basis von Vorwissen befürwortet oder bestraft werden können, setzt sich aus drei Termen zusammen

$$U_P(X_t) = k \cdot U_K(X_t) + s \cdot U_S(X_t) + w \cdot U_W(X_t). \quad (3.2)$$

Der erste Term der Priorenergie bewertet die Konnektivität des Graphen. Dabei werden Segmente bestraft, die mit dem Prielnetzwerk nur auf einer und nicht auf beiden Seiten mit einem Knoten verbunden sind. Hierdurch wird das Ziel verfolgt, dass die Objektkonfiguration aus einem einzigen Graphen besteht. Um die Ansammlung von vielen Objekten in Regionen mit hoher Datenenergie zu vermeiden, werden sich schneidende Segmente im zweiten Term bestraft. Hierbei werden Flächen von Segmenten, die zum selben Knoten gehören, nicht berechnet, da Segmente sich zwangsläufig in der Nähe eines Kreuzungspunktes überlappen werden. Schließlich erfolgt im letzten Term eine Bestrafung der Kanten, die einen für das Prielsystem unüblichen Schnittwinkel aufweisen. So werden beispielsweise für einen Knoten, der über zwei ausgehende Kanten verfügt, Zwischenwinkel von 180° oder 135° bevorzugt und davon abweichende Konfigurationen bestraft. Mithilfe der Parameter k , s und w (Konnektivität, sich schneidende/überlappende Segmente, Winkelabweichungen) können die Terme unterschiedlich gewichtet werden.

Im Rahmen dieser Arbeit soll die Priorenergie um einen weiteren Term erweitert und die Auswirkung auf die Ergebnisse untersucht werden. Die entsprechende Methodik ist in Kapitel 3.3 dargestellt.

3.2 Untersuchungen hinsichtlich Zufallszahlengeneratoren

Wie bei den MC-Verfahren beschrieben, erfolgt die Realisierung von Zufallsexperimenten aufgrund der großen Anzahl der zu generierenden Zufallszahlen in der heutigen Zeit mithilfe des Computers. Computersysteme weisen von Natur aus ein deterministisches Verhalten auf. Aus diesem Grund ist es deutlich schwieriger, Zufallszahlen mit den drei Charakteristika (statistische Unabhängigkeit, Gleichverteilung und Unvorhersehbarkeit) zu erzeugen, als es zu sein scheint.

Innerhalb dieser Arbeit werden Zufallszahlengeneratoren (Random Number Generators (RNGs)) verwendet, um Zufallszahlen für das MCMC-Verfahren (Kap. 2.3.2.1) zu erzeugen.

Weil es sich hierbei um eine Simulation handelt, unterliegt das Verfahren der Numerik, für welche der Einsatz eines Pseudo-RNGs (PRNGs) in der Regel ausreichend ist. Im Speziellen wird der Generator *Mersenne Twister mt19937* benutzt. Dieser Generator ist ein weit verbreiteter und gleichzeitig einer der besten PRNGs der heutigen Zeit. Im Vergleich zu anderen ist er schnell, bei gleichzeitig guter Qualität. Durch seine riesige Periodizität von $2^{19937} - 1$ kann die eigentliche Schwäche der sich wiederholenden Zahlensequenz praktischerweise vernachlässigt werden (Matsumoto und Nishimura, 1998). Der PRNG *mt19937* wird von der *Boost Random Number Generator Library* bereitgestellt. Diese Bibliothek bietet zusätzlich zum *mt19937* viele weitere PRNGs an. Eine komplette Liste dieser, welche u. a. Laufzeitvergleiche beinhaltet sowie weitere Informationen über deren Implementierung, können Maurer (2015a) entnommen werden.

Diese Sorte von Generatoren muss mit einem Saatpunkt initialisiert werden, was auf verschiedene Arten erfolgen kann. So kann der Saatpunkt beispielsweise mit konstanten Werten, über die Systemzeit oder über einen kryptographischen (nicht-deterministischen) RNG gesetzt werden. Ferner ist es möglich, den Saatwert vor dem Sampling zu initialisieren oder ihn während des Samplings in jeder Iteration neu zu setzen. Untersuchungen hierzu sind sinnvoll, da die unterschiedlichen Möglichkeiten dazu führen können, dass die vom Computer erzeugten Zufallszahlen unter Umständen keine zufälligen Eigenschaften aufweisen. Dies wäre für das Verfahren des Samplings fatal.

Neben dem Generator *mt19937* als PRNG wird zu Testzwecken ein nicht-deterministischer RNG verwendet, um eine Sequenz von Zufallszahlen zu erzeugen. Bei diesen Recherchen liegt das Augenmerk auf deren Laufzeiten, aus welchen sich die Wahl des Generators folgern lässt. Längere Laufzeiten sind für Simulationsansätze von Nachteil, kürzere daher zu bevorzugen.

Neben der Wahl eines solchen RNGs muss eine Verteilung bestimmt werden, die eine weitere Stellschraube bereitstellt. Auch hier bietet die Bibliothek unterschiedliche Verteilungsfunktionen an, welche eine Verteilung (häufig eine von einem Generator gelieferte Gleichverteilung) auf eine andere abbildet. Eine Auflistung möglicher Verteilungsfunktionen, wie der Gleich- oder Normalverteilung, ist in Maurer (2015a) aufgeführt. Indem für Parameter unterschiedliche Verteilungen verwendet werden, kann Vorwissen über bestimmte Charakteristika des untersuchten Gebietes in die Analysen mit einfließen. Dies ist sinnvoll, denn nicht für alle Parameter stellt dieselbe Verteilungsfunktion die optimale Wahl dar. Wird angenommen, dass ein überwiegend geradliniger Flussverlauf vorhanden ist, kann beispielsweise die Normalverteilung geeignet sein, um den Winkel $\beta_{i+1} - \beta_i$ (Abb. 3.1) zwischen zwei Segmenten zu beschreiben. Der Mittelwert des Zwischenwinkels beträgt dann 180° . Die Wahl einer geeigneten Verteilung hat den Vorteil, dass der Algorithmus gegebenenfalls schneller gegen das Energieminimum konvergiert.

Wesentliche Erkenntnisse im Hinblick auf mögliche Probleme, die bei einer ungünstigen Initialisierung des Generators auftreten können und wie es sich mit der Laufzeit eines PRNGs gegenüber einem nicht-deterministischen RNG verhält, sind in Kapitel 4.2 dargestellt. Kapitel 4.3 befasst sich mit der Fragestellung, ob Parameter des in Kapitel 3.1 beschriebenen Modells entsprechend der vorgegebenen Wahrscheinlichkeiten gewählt werden.

3.3 Erweiterung der Energiefunktion

Durch die Modellierung einer 3D-Komponente soll das bereits vorhandene Modell der Priorenergie (Formel 3.2) mithilfe von weiterem Vorwissen durch einen vierten Term $U_F(X_t)$ gestützt werden

$$U_{P'}(X_t) = k \cdot U_K(X_t) + s \cdot U_S(X_t) + w \cdot U_W(X_t) + f \cdot U_F(X_t). \quad (3.3)$$

Dieser bewertet die Übereinstimmung der Objektkonfiguration mit einem Fließmodell. Sollten die im Fließmodell formulierten Bedingungen nicht erfüllt sein, wird ein Strafterm gesetzt. Wie stark die Objektkonfiguration bestraft werden soll, hängt von der Wahl des Faktors f ab. Der Grundgedanke hinter dem Modell liegt darin, dass Wasser nicht bergauf fließen kann. Dies drückt sich darin aus, dass in einen Knoten beliebig viele Flüsse hinein-, jedoch nur einer herauslaufen darf. Somit ist es nicht möglich, dass gar keiner oder mehr als einer der Nachbarn vom aktuell betrachteten Knoten i mit der Höhe h_i eine geringere Höhe aufweist. Ist der Knoten i mit den Nachbarknoten j , k und l verbunden, so muss $h_i > h_j$ und ferner $h_i \leq h_k$ sowie $h_i \leq h_l$ für die Objektkonfiguration gelten. Diese Bedingung lässt sich aus den berechneten Höhendifferenzen schlussfolgern und gilt es in jeder Iteration zu überprüfen. Ist das Kriterium nicht erfüllt, so wird diese vom Modell abweichende Objektkonfiguration bestraft. Hierbei ist anzumerken, dass in der Konfiguration nur Teilgraphen berücksichtigt werden, die aus mehr als zwei Knoten bestehen.

Abbildung 3.2 zeigt exemplarisch verschiedene Objektkonfigurationen in synthetischen Daten, wobei Knoten in grün und Kanten in rot dargestellt sind. Die Fließrichtung des Wassers ist durch rote Pfeile gekennzeichnet. Hellblaue Pfeile weisen bei Bedarf zusätzlich auf den interessanten Bereich hin. Auf die Darstellung der zu den jeweiligen Kanten zugehörigen Segmente wird verzichtet, da diese für das Prinzip nicht von Bedeutung sind.

Zunächst wird auf die hellblau umkreisten Bereiche eingegangen. Bestrafungen sind möglich, wenn die aktuelle Objektkonfiguration durch verschiedene Kernel (Kap. 3.1) modifiziert wird. In Frage kommen der Geburt-Todes-Kernel und der Translations-Kernel. Das Modifikationsevent wird mit einer möglichen Veränderung der Segmentbreite nicht zu einer Abweichung von der aktuellen Konfiguration des Graphen führen. Somit bleibt die Höhe der Knoten unverändert. Die Abbildung 3.2b zeigt die Geburt eines neuen Knotens, welcher dem bestehenden Teilgraphen aus Abbildung 3.2a hinzugefügt wird. Durch das Hinzufügen des Knotens 3 gilt $h_1 < h_2$ und $h_1 < h_3$. Somit ist die Bedingung des Modells in der neuen Konfiguration nicht mehr erfüllt. Weiterhin kann ein Todesevent dazu führen, dass die bestehende Objektkonfiguration (Abb. 3.2c) durch das Entfernen eines Knotens mit seinen benachbarten Kanten nicht mehr bestraft wird (Abb. 3.2d). Durch die Elimination von Knoten 3 besteht der Teilgraph nunmehr aus zwei Knoten, sodass diese Konfiguration im Modell nicht weiter berücksichtigt wird. Abschließend kann die Verschiebung eines Knotens beim Translations-Kernel als Konsequenz eine Bestrafung der neuen (Abb. 3.2f), im Gegensatz zur vorherigen Objektkonfiguration (Abb. 3.2e), mit sich bringen. Während in Abbildung 3.2e die Bedingung mit $h_1 < h_2 < h_3$ noch erfüllt ist, ist diese aufgrund der Verschiebung des Knotens 1' auf den Boden des Wattenmeeres nicht mehr gegeben. Die Höhe $h_{1'}$ ist nun ebenfalls größer als h_2 , sodass das Wasser jetzt von beiden Seiten in Richtung des Knotens 2 fließt.

Nun werden die orange umkreisten Bereiche betrachtet. Die für die vier Abbildungen (Abb. 3.2a-d) identische Konfiguration wird nicht bestraft, da der Teilgraph aus weniger als drei Knoten besteht. Ebenso ist die Bedingung in der Abbildungen 3.2e erfüllt, denn für die drei Knoten gilt $h_1 < h_2 < h_3$. Dennoch entsprechen die Objektkonfigurationen nicht den gewünschten. Wie zu erkennen, können Kanten des Graphen von einem Ausläufer des Priels über den Wattboden zu einem anderen Ausläufer verlaufen.

Aus diesem Grund wird das bestehende Modell, zukünftig als Fließmodell 1 bezeichnet, um ein weiteres ergänzt bzw. optimiert. Jetzt werden nicht nur die Höhe des Knotens und die seiner jeweiligen Nachbarknoten betrachtet, sondern auch alle Höhenwerte dazwischen, die entlang der Mittelachse des Segments verlaufen. Anders als bei der vorherigen Variante betrachtet das Fließmodell 2 bereits Teilgraphen, die lediglich aus zwei Knoten bestehen.

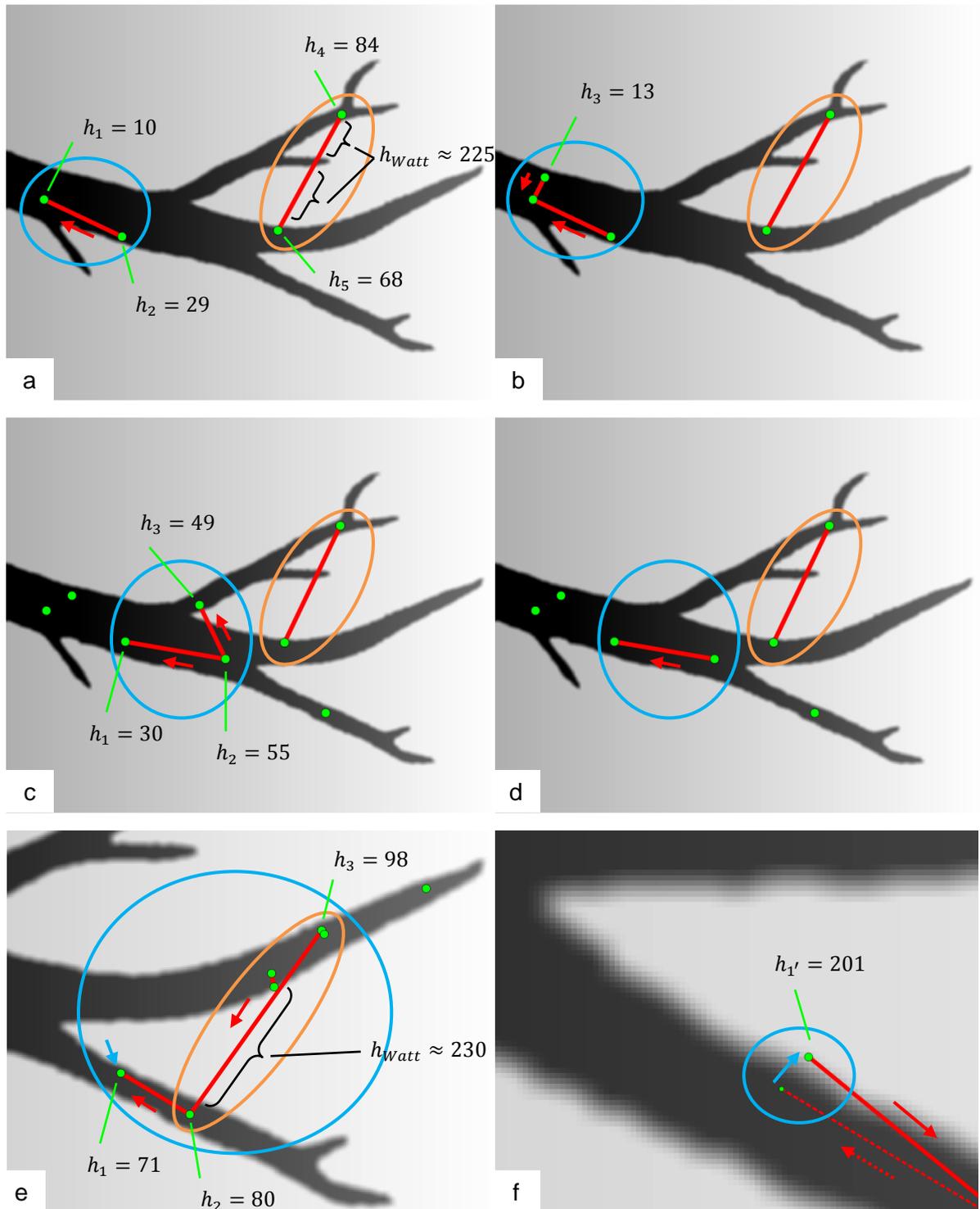


Abbildung 3.2: Hellblaue umkreiste Bereiche verdeutlichen die Funktionsweise des ersten Fließmodells

(a) und (b): Die Geburt des Knotens 3 führt zur Bestrafung der Objektkonfiguration, da Knoten 1 nun eine geringere Höhe als die beiden Nachbarn 2 und 3 aufweist.

(c) und (d): Vor dem Todesevent wird die alte Konfiguration bestraft, weil die beiden Nachbarn des Knotens 2 tiefer als dieser liegen.

(e) und (f): Die Translation des Knotens 1 aus der alten Objektkonfiguration (gestrichelte Linie) auf den Wattboden führt dazu, dass ein Strafterm gesetzt wird. Dies ist der Fall, da nun die beiden benachbarten Höhen $h_{1'}$ und h_3 oberhalb der betrachteten Höhe h_2 liegen.

Im Hinblick auf das zweite Fließmodell muss jede Pixelhöhe h_p zwischen der Höhe des Anfangsknotens h_A und der des Endknotens h_E liegen (Abb. 3.3). Somit muss $h_A \leq h_p \leq h_E$ für $h_A < h_E$ bzw. $h_A \geq h_p \geq h_E$ für $h_A > h_E$ gelten. Abbildung 3.3 stellt den ersteren Fall dar. Zusätzlich muss ein Gefälle bzw. Anstieg oder zumindest eine gleichbleibende Höhe über den gesamten Weg entlang der Kante vorhanden sein. Dabei vergleicht jeder Pixel h_{p_i} seine Höhe mit der des vorherigen $h_{p_{i-1}}$, bis der Endknoten h_E erreicht ist. Für den ersten Vergleich der Pixelhöhen gilt $h_{p_{i-1}} = h_A$. Demnach muss für $h_A < h_E$ die Bedingung $h_{p_{i-1}} \leq h_{p_i}$ und analog für $h_A > h_E$ die Bedingung $h_{p_{i-1}} \geq h_{p_i}$ gelten. Die Pixel werden dabei entlang der Kante in einer 8er-Nachbarschaft abgelaufen und sind entsprechend des Pfades hellgrau eingefärbt. Für Abbildung 3.3a ergibt sich die Grauwertfolge 10, 13, 15, 17, 22, 29, 31, 36, 38 und 40. Somit ist einerseits die Bedingung $h_A \leq h_p \leq h_E$ erfüllt und es liegt weiterhin ein kontinuierlicher Anstieg entlang der Kante vor. Deshalb erfolgt keine Bestrafung der Objektkonfiguration. In Abbildung 3.3b hingegen werden beide Kriterien verletzt. So liegt der Grauwert 42 nicht innerhalb des Intervalls $[h_A; h_E]$ und $h_{p_{i-1}} \leq h_{p_i}$ gilt nicht entlang des gesamten Pfades.

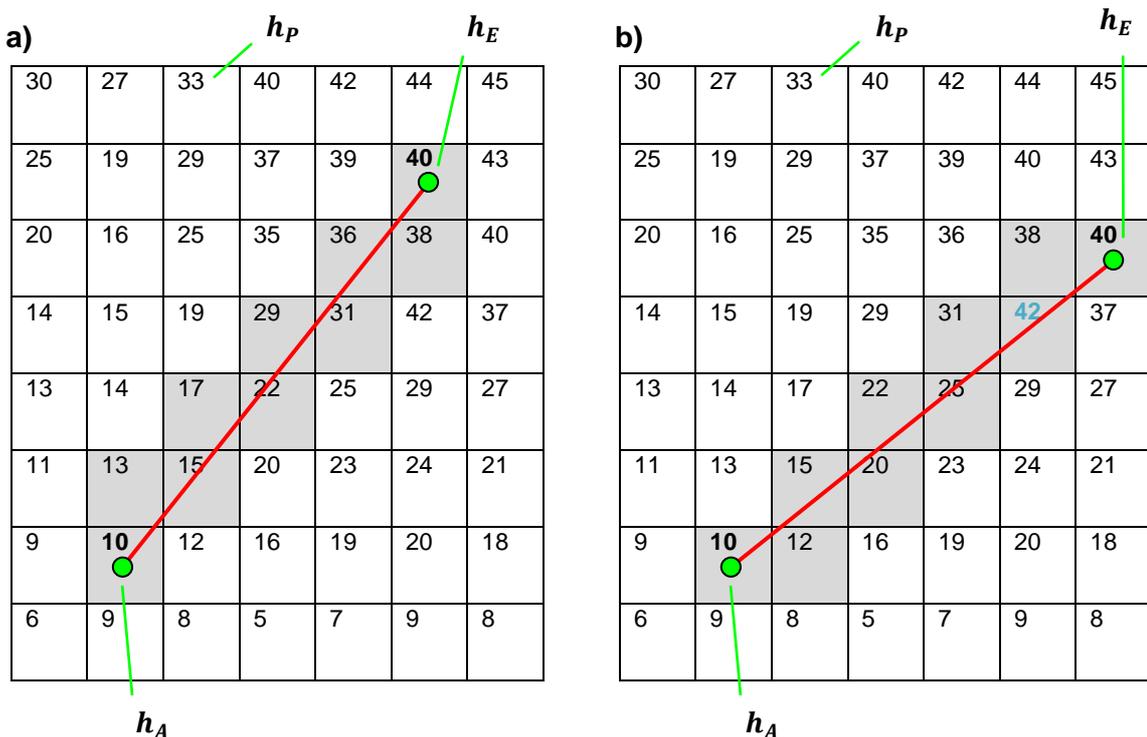


Abbildung 3.3: Schematische Darstellung, welche die Funktionsweise des zweiten Fließmodells verdeutlicht
(a): Die Bedingungen $h_A \leq h_p \leq h_E$ und $h_{p_{i-1}} \leq h_{p_i}$ sind erfüllt
(b): Die Bedingungen $h_A \leq h_p \leq h_E$ und $h_{p_{i-1}} \leq h_{p_i}$ sind nicht erfüllt

Aufgrund des Fließmodells 2 werden nun auch die in der Abbildung 3.2 orange umkreisten Objektkonfigurationen bestraft. Einerseits liegen die Höhen der Pixel nicht zwischen der des Anfangsknotens 5 und des Endknotens 4, da die Kante über den Boden des Wattenmeeres ($h_{Watt} \approx 225$) verläuft (Abb. 3.2a). Somit ist die Bedingung $h_A \geq h_P \geq h_E$ nicht eingehalten. Ferner liegt kein kontinuierliches Gefälle vor. Ähnlich verhält sich dies mit der Kante in Abbildung 3.2e, sodass ebenso beide Bedingungen nicht erfüllt sind. Abschließend ist anzumerken, dass auch hier – ebenso wie beim ersten Fließmodell – Bestrafungen der Objektkonfiguration durch ein Geburt- oder Todesevent oder eine Veränderung der Position des Knotens erfolgen können.

Das erste und zweite Fließmodell werden kombiniert genutzt, um das bereits vorhandene Modell der Priorenergie zu erweitern. Das (kombinierte) Fließmodell $U_F(X_t)$ setzt sich dabei aus den Bedingungen, die im Fließmodell 1 bzw. 2 formuliert worden sind, zusammen.

$$U_F(X_t) = \text{Fließmodell 1} + \text{Fließmodell 2} \quad (3.4)$$

Nach Formel 3.4 kommt es zu einer Aufsummierung von Bestrafungen, falls sowohl das erste als auch das zweite Fließmodell nicht erfüllt sind. Jede Bestrafung wird dabei auf den Wert Eins gesetzt. So wird unter Berücksichtigung aller Teilgraphen die Objektkonfiguration in Abbildung 3.2b insgesamt zwei Mal bestraft ($U_F(X_t) = 2$). Der orange umkreiste Bereich verletzt beide Bedingungen des zweiten Fließmodells. Unabhängig davon, ob die Objektkonfiguration beiden oder nur einer der Bedingungen widerspricht, wird die Konfiguration einmal bestraft. Die andere Bestrafung im hellblau umkreisten Gebiet resultiert aus der nicht erfüllten Bedingung des Fließmodells 1.

Dadurch, dass die formulierten Bedingungen für alle von Änderungen betroffenen Knoten gelten müssen, kann die Bestrafung im Fließmodell auch deutlich höher ausfallen. Hierdurch sollte diese Konfiguration im Endeffekt schneller verworfen werden. Eine größere Wahl des Faktors f wird dies noch zusätzlich beschleunigen. Beispielsweise wird die Objektkonfiguration aus Abbildung 3.4 fünf Mal bestraft. Bereiche, in denen Bestrafungen erfolgen, die auf das erste Fließmodell zurückzuführen sind, sind hellblau umkreist (Abb. 3.4a). Entsprechend zeigen orange umkreiste Gebiete Objektkonfigurationen auf, welche mindestens einer Bedingung des zweiten Fließmodells nicht genügen (Abb. 3.4b). Dabei entspricht jeder Kreis genau einer Bestrafung, die die jeweilige Konfiguration hervorruft. Insgesamt werden drei Mal die Bedingung des Fließmodells 1 und zwei Mal beide Bedingungen des Fließmodells 2 verletzt.

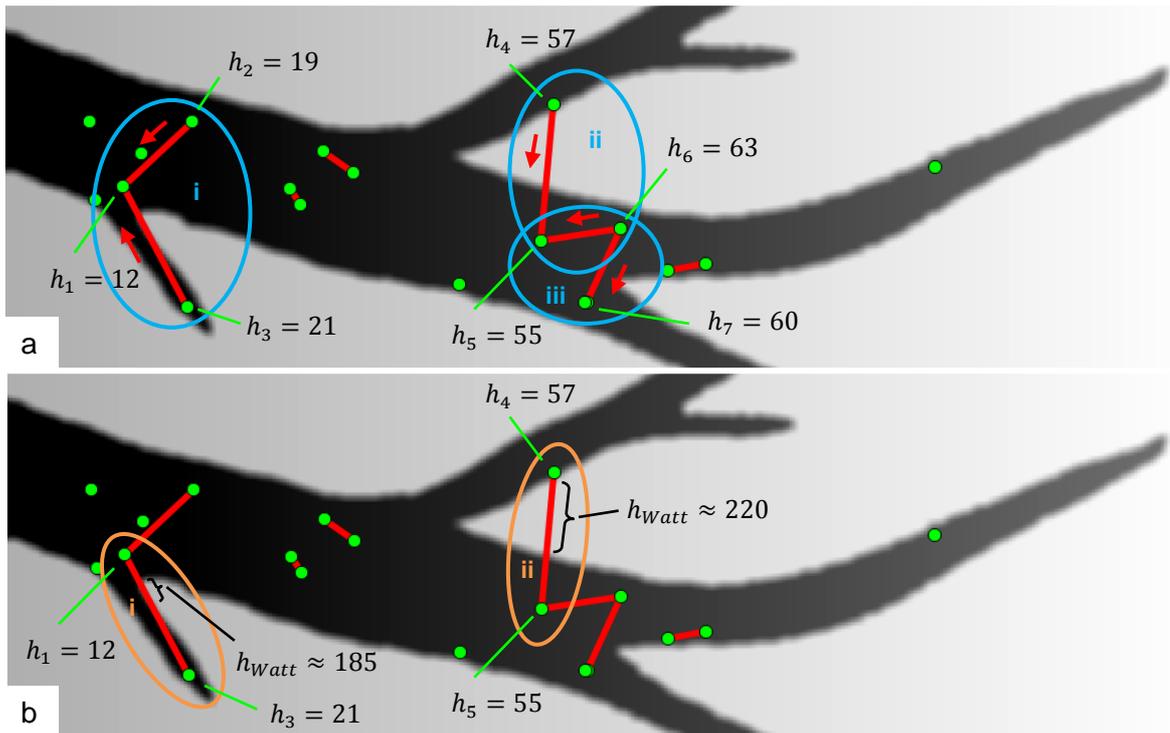


Abbildung 3.4: Ausgewähltes Beispiel, welches die Funktionsweise des Fließmodells im Priorterm verdeutlicht (Kombination des Fließmodells 1 und 2)

- (a): Bestrafungen, die durch das Fließmodell 1 hervorgerufen werden
- (i) Die Nachbarn 2 und 3 von Knoten 1 sind höher
 - (ii) Knoten 5 bildet eine Senke, sodass die Nachbarn 4 und 6 höher liegen
 - (iii) Die Höhe des Knotens h_6 liegt oberhalb derer von h_5 und h_7
- (b): Bestrafungen, die durch das Fließmodell 2 hervorgerufen werden
- (i + ii) Die Höhe des Wattenmeeres h_{Watt} führt dazu, dass beide Bedingungen verletzt werden

Unter Umständen kann es sinnvoll sein, für die Untersuchungen nicht die an der Position des Knotens aus den Eingangsdaten abgegriffene Höhe zu verwenden. Zu nennen sind hier beispielsweise Interpolationsfehler, die bei der Erstellung des DGMs aufgetreten sein können. Darüber hinaus kann das DGM eine inhomogene Oberfläche aufweisen, sodass sich die Grauwerte benachbarter Pixel unter Umständen stark unterscheiden. Dies könnte dazu führen, dass die aktuelle Konfiguration auf Grundlage des aufgestellten Fließmodells (fast) immer bestraft werden würde. Hierbei ist die Chance sehr hoch, dass die Höhe eines Pixels nicht entlang des gesamten Weges in das Höhenintervall zwischen Anfangs- und Endpunkt fällt und überdies ein ständiges/r Gefälle bzw. Anstieg vorhanden ist (Fließmodell 2). Daher kann es sinnvoll sein, das Modell abzuschwächen. Hierzu werden die beiden Bedingungen des Fließmodells 2 um einen konstanten Wert a erweitert. Abhängig davon, ob die Höhe des Anfangsknotens h_A ober- oder unterhalb derer des Endknotens h_E liegt, unterscheiden sich die Vorzeichen. Für $h_A < h_E$ muss dann $h_A - a \leq h_p \leq h_E + a$ und ferner $h_{p_{i-1}} \leq h_{p_i} + a$ gelten. Im Fall von $h_A > h_E$ ändern sich die Vorzeichen zu $h_A + a \leq h_p \leq h_E - a$ sowie $h_{p_{i-1}} \leq h_{p_i} - a$.

Auf der anderen Seite ist es möglicherweise von Vorteil, die Umgebung des betrachteten Pixels zu berücksichtigen und eine gemittelte Höhe zu bestimmen, um dem Einfluss durch ein inhomogenes Gelände entgegenzuwirken. Zu diesem Zweck kann eine Medianfilterung als Vorverarbeitungsschritt durchgeführt werden. Abschließend ist noch anzumerken, dass durch die im Modell geforderten Annahmen das Vorkommen von Inseln (oder auch Sandbänken im Fall der Priele) vernachlässigt wird.

4 Analyse der Ergebnisse

Die in Kapitel 3.3 dargestellte Methode zur Modellierung einer 3D-Komponente im Priorterm sowie die Untersuchungen im Hinblick auf Zufallszahlengeneratoren (Kap. 3.2) werden auf die im nachfolgenden Kapitel 4.1 aufgeführten Daten angewendet. Dabei erfolgen zunächst Analysen bezüglich der Initialisierung des Generators und des Laufzeitvergleichs zwischen einem nicht-deterministischen und einem Pseudo-Zufallszahlengenerator (Kap. 4.2). Daran schließen Untersuchungen an, ob Parameter entsprechend der vorgegebenen Verteilung gewählt werden (Kap. 4.3). Zu diesen exemplarisch näher untersuchten Parametern zählen die für die Winkel gewürfelten Richtungen der vom Kreuzungspunkt ausgehenden Segmente und die Geburtsorte für einen neuen Knoten bei einem Geburtsereignis. Ferner wird überprüft, ob sich der Algorithmus gemäß der gewählten Verteilungsfunktion für den jeweiligen Kernel (Geburt-Todes-, Translations- und Modifikations-Kernel) entscheidet. Kapitel 4.4 befasst sich mit den Analysen der im Priorterm modellierten 3D-Komponente. In diesem Zusammenhang wird geklärt, inwiefern die Erweiterung des bestehenden Modells der Priorenergie um das Fließmodell einen Einfluss auf die Resultate hat. Weiterhin wird der Frage nachgegangen, ob Komponenten der Priorenergie weggelassen werden können, um die Anzahl der Parameter des Algorithmus zu reduzieren. Hierbei wird ebenfalls untersucht, inwiefern sich die Terme auf die Laufzeit des Programms auswirken.

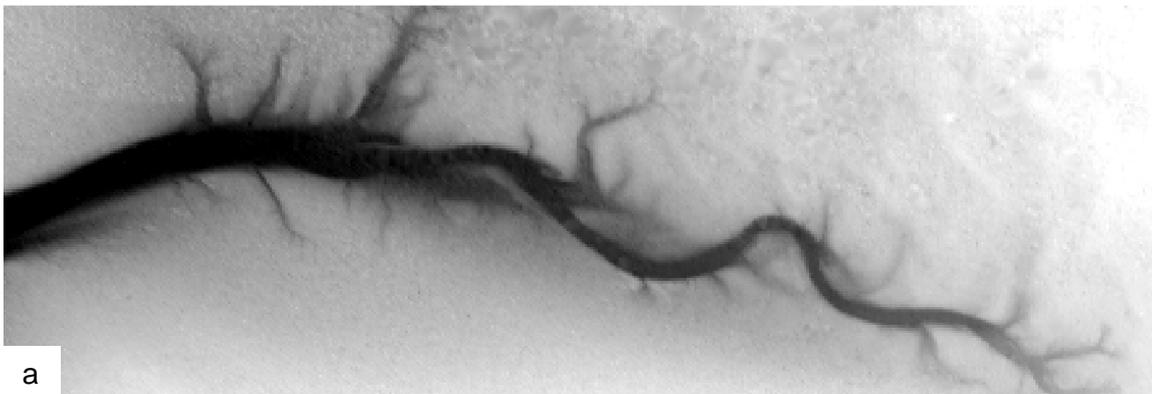
4.1 Testdaten

Innerhalb dieser Arbeit werden die Analysen der Ergebnisse an unterschiedlichen Daten durchgeführt, da sich hierdurch an bestimmten Stellen die Auswirkungen der verschiedenen Untersuchungen besser erkennen und verdeutlichen lassen. Alle nachfolgend genannten Daten bzw. Bilder sind mit acht Bit quantisiert worden, sodass eine Differenzierung zwischen 256 Grauwerten möglich ist.

Ein Datensatz, welcher auch von Schmidt et al. (2015) zur Evaluierung ihrer Methode verwendet wird, wurde im deutschen Wattenmeer auf Basis von LIDAR-Daten (Light Detection an Ranging) erhoben. Das Testgebiet erstreckt sich im Süden der Ostfriesischen

Insel Norderney über eine Distanz von $440 \times 150 \text{ m}^2$ (Abb. 4.1a). Die Aufnahmen des Wattenmeeres erfolgten mithilfe eines LMS-Q560 Sensors der Firma RiegI während einer Flugkampagne im Frühjahr 2012. Die Höhen der rohen Punktwolke, welche eine durchschnittliche Punktdichte von 5,9 Punkten je Quadratmeter besitzt, wurden interpoliert, um ein DGM mit einer Rasterweite von 1 m zu erzeugen. Die Höhen nehmen dabei Werte zwischen $-0,02 \text{ m}$ und $1,72 \text{ m}$ ein, sodass ein Grauwert in etwa $0,007 \text{ m}$ entspricht.

Neben diesen Daten werden auch synthetische Daten für verschiedenste Experimente verwendet. Zum einen wird ein stark vereinfachtes Modell eines Priels erzeugt, bei dem die Grauwerte lediglich in Nord-Süd-Richtung variieren, in Ost-West-Richtung jedoch konstant sind (Abb. 4.1b). Zum anderen wird ein komplexeres Prielmodell verwendet. Das Bild mit einer Ausdehnung von $700 \times 550 \text{ Pixeln}$ ist derart erzeugt worden, dass sich der für das Wattenmeer charakteristische Verlauf von Prielen wiederfinden lässt (Abb. 4.1c). Ebenfalls in Anlehnung an die in der Flugkampagne erhobenen Daten ist ein kontinuierliches Gefälle in Richtung Westen vorhanden. Dabei besitzen die Grauwerte innerhalb des Hauptarms des Priels im äußersten Westen einen Wert von Null, welcher im Osten bei 120 liegt. Die Höhendifferenz zwischen benachbarten Pixeln beträgt maximal Eins. Die Grauwerte außerhalb des Priels bewegen sich im Intervall $[179; 255]$. An den Prielkanten liegt ein linearer Höhenanstieg entsprechend des Wertebereichs zwischen dem Grauwert in bzw. außerhalb des Priels vor.



a



b



c

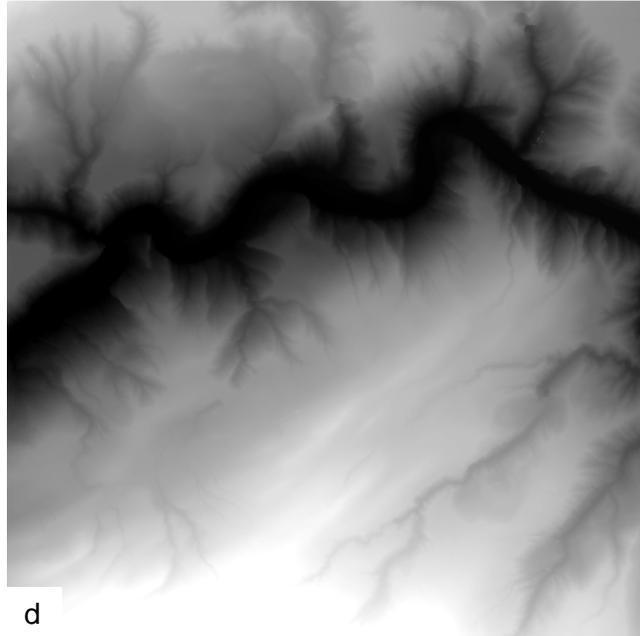


Abbildung 4.1: Testdaten
(a): Daten des deutschen Wattenmeeres
(b): stark vereinfachtes Prielmodell (synthetische Daten)
(c): komplexeres Prielmodell (synthetische Daten)
(d): Daten des österreichischen Bundeslandes Vorarlberg

Schließlich werden Daten aus Österreich herangezogen, welche ebenfalls auf Grundlage von LIDAR-Daten erfasst worden sind. Die Aufnahmen des insgesamt 545 km^2 umfassenden Gebietes des österreichischen Bundeslandes Vorarlberg erfolgten in einem dreijährigen Projekt von 2003 bis 2005 und wurden von der Firma Topscan GmbH durchgeführt. Das Testgebiet befindet sich im Norden von Vorarlberg nördlich des Ortes Heimen und erstreckt sich über eine Distanz von $2500 \times 2500 \text{ m}^2$ (Abb. 4.1d). Die Höhen der rohen Punktwolke wurden interpoliert, um ein DGM mit einer Rasterweite von $0,5 \text{ m}$ zu erzeugen. Für die Höhen ergeben sich Werte zwischen $431,9 \text{ m}$ und $829,9 \text{ m}$, sodass ein Grauwert in etwa $1,55 \text{ m}$ entspricht.

4.2 Initialisierung des Generators sowie Laufzeitvergleich

Im Folgenden wird die Generierung der Zufallszahlen untersucht. Dabei wird zunächst der Fall betrachtet, dass der Saatpunkt innerhalb der Schleife über die Anzahl der Iterationen des Samplings gesetzt wird. Die Initialisierung des Pseudo Random Number Generators (PRNGs) mit denselben Werten innerhalb der Schleife führt zu identischen zurückgegebenen Zufallszahlen. Somit muss der Saatpunkt für den Random Number Generator (RNG) so

gesetzt werden, dass er nicht jedes Mal vom selben Ort startet. Dies kann beispielsweise mithilfe der aktuellen (System-) Zeit realisiert werden. Durch diese Initialisierung wird der Saatpunkt so verändert, dass die Sequenz von Zufallszahlen sich bei jedem Durchlauf des Programms ändert. Jedoch ist hierbei zu beachten, dass die interne Zeit des Computers gegebenenfalls eine geringe Auflösung besitzen kann (Sekundenintervalle). Dies würde dazu führen, dass das Programm bei seiner Ausführung unter Umständen mehrfach in Zeitabständen deutlich unterhalb einer Sekunde durchlaufen wird, was aufgrund der heutigen Geschwindigkeit der Computer durchaus realistisch ist. Folglich wird die Systemzeit dieselben Werte zurückgeben, da der PRNG mit demselben Saatpunkt initialisiert worden ist. Die Konsequenz hiervon wäre, dass man innerhalb der einen Sekunde identische Parameter herausbekommen würde. Eine Möglichkeit hiermit umzugehen, stellt das erneute Setzen des Saatpunktes in jedem Schleifendurchlauf dar, jedoch benötigt dies u. a. mehr Rechenzeit.

Aus den oben genannten Gründen ist davon abzuraten, den Saatpunkt innerhalb der Schleife zu setzen. Stattdessen sollte ein einziges Mal eine Initialisierung des Generators außerhalb der Schleife des Samplings vorgenommen werden. Auch bei dieser Herangehensweise kann es passieren, dass das Setzen des Saatwertes über die Zeit zu identischen Zufallszahlen führt – wenn auch aus einem anderen Grund. Sollte das Programm zum Beispiel zu Testzwecken von Parametern mehrfach zur selben Zeit (innerhalb einer Sekunde) ausgeführt werden, so wird die erzeugte Zahlensequenz dieselbe für alle in dieser Sekunde ausgeführten Programme sein. Um dieser Einschränkung entgegenzuwirken, empfiehlt sich die Initialisierung von *mt19937* oder einem anderen PRNG mithilfe eines nicht-deterministischen bzw. wahren RNGs (True RNG (TRNG)). Die Klasse *random_device* (Maurer, 2015b), ebenfalls von der *boost*-Bibliothek bereitgestellt, ist ein Modell für einen solchen Generator. Dieser nutzt stochastische Prozesse, um eine Sequenz von gleichverteilten, nicht-deterministischen Zufallszahlen zu erzeugen. Das Setzen eines Saatpunktes ist nicht notwendig, da zufällige Daten vom eigenen Betriebssystem abgefragt werden. Weitere Informationen bzw. Empfehlungen zur Verwendung von PRNG-Techniken sind in Eastlake et al. (2005) dargestellt.

Für die nachfolgenden Untersuchungen wird die Funktion im Programm, welche in Abhängigkeit von der gewürfelten Zufallszahl einen der drei Kernel (Geburt-Todes-, Translations-, Modifikations-Kernel) auswählt (Kap. 3.1), verwendet. Zu Testzwecken, wie es sich mit den Laufzeiten eines TRNGs und eines PRNGs verhält, wird die Funktion mehrfach ausgeführt und der Laufzeitunterschied zwischen den beiden Arten von Generatoren ermittelt. Motivation dieser Untersuchung ist es, zu schauen, ob die Verwendung eines TRNGs zu verlängerten Laufzeiten führt. Hierdurch wäre der TRNG für den Einsatz der innerhalb dieser Arbeit verfolgten Aufgabe der Simulation von Nachteil. Obwohl

beispielsweise in der Einführung von Maurer (2015a) die Rede davon ist, TRNGs für Sicherheitsapplikationen und nicht für Simulationen zu verwenden, wurden keine unmittelbaren Informationen über den Laufzeitvergleich aufgeführt. Lediglich auf einigen Internetforen und -seiten, welche sich mit dieser Thematik befasst haben, ist die Rede von verlängerten Laufzeiten für TRNGs, jedoch nicht in quantitativer Hinsicht.

Auf Basis der auch von Schmidt et al. (2015) verwendeten Daten wurden insgesamt eine Millionen Iterationen der Funktion pro Durchlauf bei insgesamt 100 Durchläufen ausgeführt. Der Vergleich der Laufzeiten für den TRNG (*random_device*, blau) und den PRNG (*mt19937*, rot) ist in Abbildung 4.2 dargestellt. Auf der Rechtsachse ist die Nummer der Durchläufe und auf der Hochachse die Laufzeit in Sekunden aufgetragen.

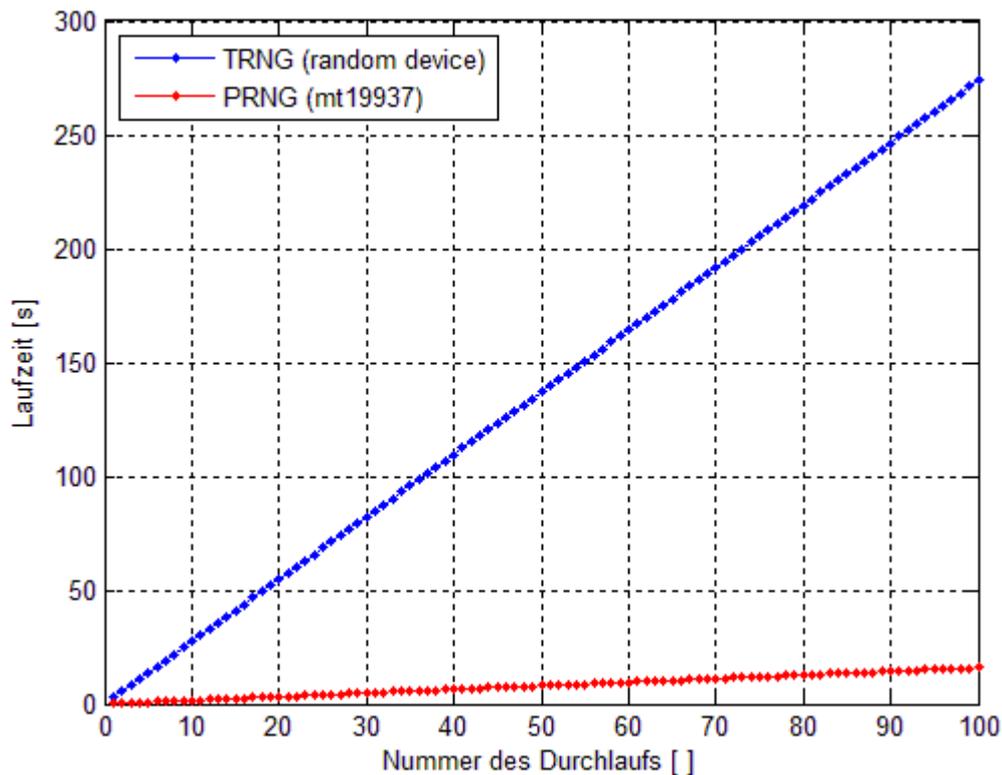


Abbildung 4.2: Vergleich der Laufzeiten eines TRNGs und PRNGs

Wie Abbildung 4.2 zu entnehmen ist, ist sowohl für den TRNG als auch den PRNG ein linearer Laufzeitanstieg festzustellen. In diesem speziellen Beispiel benötigt letzterer Generator im Durchschnitt (Mittelwert) für einen Durchlauf 0,16 Sekunden, während der TRNG mit 2,74 Sekunden deutlich länger braucht. Dies entspricht mehr als der 17-fachen Laufzeit des PRNGs, weshalb ein TRNG aufgrund seiner stark erhöhten Laufzeit für weitere Analysen ungeeignet ist.

4.3 Untersuchungen zur Verteilung der Parameter

Für alle weiteren Untersuchungen innerhalb dieses Kapitels werden die Beobachtungen aus Kapitel 4.2 hinsichtlich der Initialisierung eines RNGs und den benötigten Laufzeiten eines PRNGs im Vergleich zum TRNG berücksichtigt. Das heißt, es wird ein PRNG (*mt19937*) genutzt, welcher außerhalb der Schleife des Samplings oder der gerade untersuchten Funktion mittels eines TRNGs (*random_device*) einmalig initialisiert wird. Diese Initialisierung über *random_device* ist für nachfolgende Untersuchungen sinnvoll, damit das Programm zum selben Zeitpunkt mehrfach ausgeführt werden kann. Auf diese Weise wird obendrein dem Problem entgegengewirkt, dass eine exakte Kopie der Zufallszahlensequenz erzeugt wird und die Ergebnisse für alle Experimente identisch sind. Grundlage für die Experimente stellt der ebenfalls von Schmidt et al. (2015) verwendete Datensatz dar (Abb. 4.1a).

Die folgenden Analysen beruhen zunächst ebenfalls auf der in Kapitel 4.2 untersuchten Funktion, welche in Abhängigkeit von der gewürfelten Zufallszahl einen der drei Kernel (0: Geburt-Todes-Kernel, 1: Translations-Kernel, 2: Modifikations-Kernel) auswählt. Das Verfahren ist derzeit so aufgestellt, dass das Vorkommen aller Kernel unterschiedlich gewichtet werden kann. Wird das Gewicht identisch gewählt, liegt eine Gleichverteilung (GV) vor. Somit sollten im Idealfall, abhängig von den gewählten Wahrscheinlichkeiten, bestimmte Kernel beispielsweise doppelt so häufig wie andere, aller gleichermaßen, usw. gewürfelt werden.

Für die erste Untersuchung sind die Gewichte für das Auftreten eines Kernels im Verhältnis 1:2:2 gesetzt. Das heißt, das Geburt-Todes-Ereignis sollte im Vergleich zu den anderen beiden Events nur halb so oft vorkommen, während Translations- und Modifikations-Kernel gleichermaßen gewählt werden müssten (Abb. 4.3a). Gleiches wird für identische Wahrscheinlichkeiten der drei Kernel wiederholt und ist in Abbildung 4.3b dargestellt. Die Anzahl der Funktionsdurchläufe liegt für beide Untersuchungen bei 100. Auf der Hochachse ist die relative Häufigkeit des Auftretens eines bestimmten Kernels aufgetragen, während die Rechtsachse den gewählten Kernel zeigt. Die Zuordnung der Kernel lässt sich der Legende entnehmen.

In Abbildung 4.3a ist zu erkennen, dass die Auswahl der Kernel den ihnen vorgegebenen Wahrscheinlichkeiten folgt. Der Geburt-Todes-Kernel wurde mit einer relativen Häufigkeit von 0,19 nur rund halb so oft gewählt, wie die beiden übrigen Kernel mit relativen Häufigkeiten von 0,39 und 0,41. Somit folgen für diesen Fall die Ergebnisse den gewählten Wahrscheinlichkeiten.

Dem anderen Fall (Abb. 4.3b) kann diese Eigenschaft nicht zugesprochen werden. Normalerweise sollten alle Kernel gleich oft gewürfelt werden. Infolge der GV, welche dem

Algorithmus als Verteilungsfunktion zugrunde liegt, sollte jeder Kernel eine Häufigkeit von rund 33 % aufweisen. Die relativen Häufigkeiten (in der Reihenfolge der Nummerierung) liegen bei 0,42; 0,31 und 0,27. Somit folgt die Kernauswahl nicht den vordefinierten identischen Wahrscheinlichkeiten für jeden Kernel.

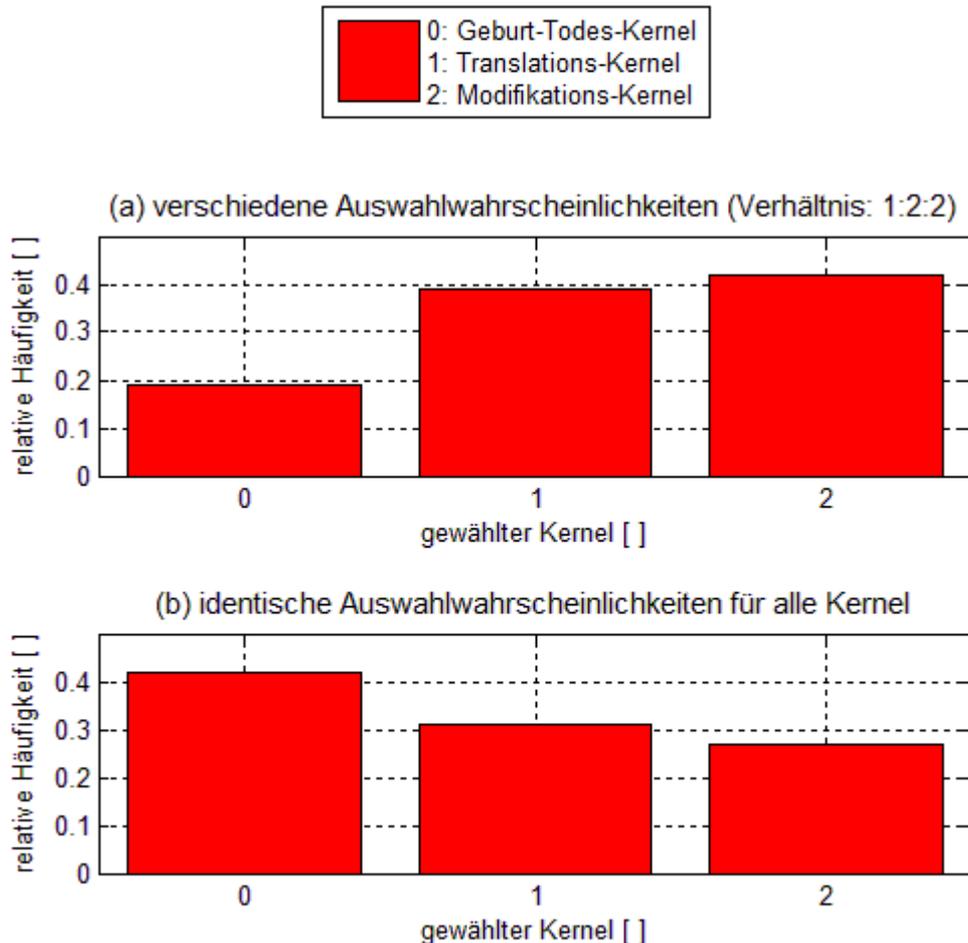


Abbildung 4.3: Vergleich der Auswahl eines Kernels für 100 Iterationen
(a) unterschiedliche Gewichtungen der Kernel (Verhältnis 1:2:2)
(b) identische Gewichtungen der Kernel (GV)

Es ist zu vermuten, dass sich diese Abweichung im Wesentlichen auf zwei Aspekte zurückführen lässt. Einerseits handelt es sich bei den durchgeführten Untersuchungen um Zufallsexperimente, sodass sich die Ergebnisse von Versuch zu Versuch unterscheiden werden und eine Reproduzierbarkeit nicht möglich ist. Folglich könnte bereits beim nächsten Programmdurchlauf die beobachtete Übereinstimmung der Ergebnisse (Abb. 4.3a) mit den Erwartungen nicht mehr kongruieren. Diese Vermutung konnte durch weitere Experimente bestätigt werden, die an dieser Stelle nicht gezeigt werden. Die zweite Annahme ist, dass die Kernel nicht den Auswahlwahrscheinlichkeiten folgen, da die Anzahl der Iterationen mit 100 deutlich zu gering ist. Erst ab einer größeren Anzahl an Funktionsdurchläufen werden alle gleichermaßen ausgewählt. Dieser Vermutung wird im Folgenden nachgegangen.

Abbildung 4.4 zeigt die gewählten Kernel bei identischer Auswahlwahrscheinlichkeit für eine unterschiedliche Anzahl an Iterationen. Auch hier bildet die Grundlage des Prozesses die GV. Für Abbildung 4.4a wurden 10.000 Funktionsdurchläufe realisiert. Die Anzahl der Durchläufe wird bis zu 10.000.000 um je einen Faktor 10 (Abb. 4.4b-d) erhöht. Da die relative Häufigkeit für jeden der drei Kernel bei $1/3$ (blaue Linien) liegen sollte, wird aus Gründen der Anschaulichkeit der Bereich der Hochachse auf das Intervall $[0,32; 0,34]$ begrenzt. Die Rechtsachse stellt den gewählten Kernel (s. Legende) dar.

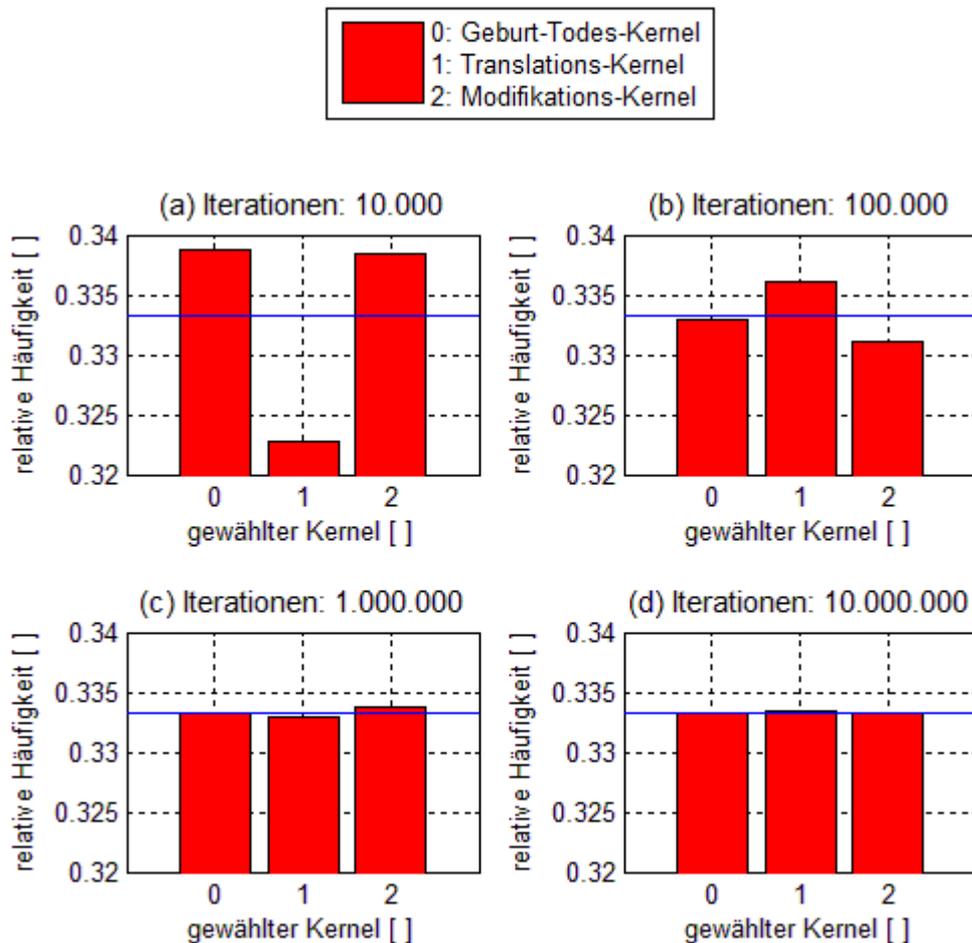


Abbildung 4.4: Gewählte Kernel für eine unterschiedliche Anzahl an Iterationen unter der Annahme einer Gleichverteilung

Es zeigt sich, dass mit steigender Anzahl an Iterationen die Auswahl der Kernel immer deutlicher einer GV folgt. Wird die Funktion 10.000 Mal durchlaufen, so liegt die relative Häufigkeit jedes Kernels im gewählten Darstellungsintervall. Da das Zufallsexperiment bei gleichen Voraussetzungen wiederholt wird, ist mit steigender Anzahl an Iterationen zu erkennen, dass sich die relative Häufigkeit um die theoretische Wahrscheinlichkeit des Zufallsergebnisses stabilisiert (Annäherung an die blaue Linie). Genau das besagt auch das Gesetz der großen Zahlen, welches im Wesentlichen die Basis der MC-Verfahren bildet

(Kap. 2.3.2). Tabelle 4.1 zeigt die zugehörigen Zahlenwerte der absoluten Differenzen zu der erwarteten Wahrscheinlichkeit für das Auftreten eines Kernels von je 1/3. Dabei werden die Differenzen mit steigender Anzahl an Iterationen kleiner. Aufgeführt sind neben den absoluten Differenzen auch die Mittelwerte dieser. Die Nummerierung der Kernel ist analog zu Abbildung 4.4.

Kernel	Iterationen	Absolute Differenz [10 ⁻³]	Mittelwert der abs. Differenz [10 ⁻³]
0	10.000	5,47	7,09
1		10,63	
2		5,17	
0	100.000	0,43	1,82
1		2,74	
2		2,30	
0	1.000.000	0,10	0,31
1		0,37	
2		0,47	
0	10.000.000	0,01	0,01
1		0,02	
2		0,01	

Tabelle 4.1: Quantitative Darstellung der Differenzen zur theoretischen Wahrscheinlichkeit aus Abbildung 4.4

Wird in dem von Schmidt et al. (2015) entwickelten Modell ein neuer Knoten der Objektkonfiguration hinzugefügt, so werden u. a. die Richtungen der Winkel für die ausgehenden Segmente von dem Kreuzungspunkt (Abb. 3.1) gewürfelt. Im Folgenden wird angenommen, dass die Zwischenwinkel zwischen zwei Segmenten an einen Knoten im Mittel 180° betragen. Ferner folgen sie einer Normalverteilung (NV) mit einem Mittelwert von 180 und einer Standardabweichung von 1. Die Winkel werden aus anschaulichen Gründen jeweils in Intervalle eingeteilt.

Im Falle der Abbildung 4.5 wird die Anzahl der Intervalle auf 40 gesetzt. Hierdurch wird der Bereich zwischen dem minimal und maximal vorkommenden Winkel in Intervalle identischer Breite eingeteilt. Auf der Rechtsachse ist der jeweils vom Algorithmus gewürfelte Winkel in Grad, auf der Hochachse die zugehörige relative Häufigkeit des entsprechenden Intervalls aufgetragen. Die Anzahl der Iterationen wird von 100 (Abb. 4.5a) jeweils um den Faktor 10 (Abb. 4.5b-c) erhöht, bis in Abbildung 4.5d 100.000 Zufallszahlen generiert worden sind.

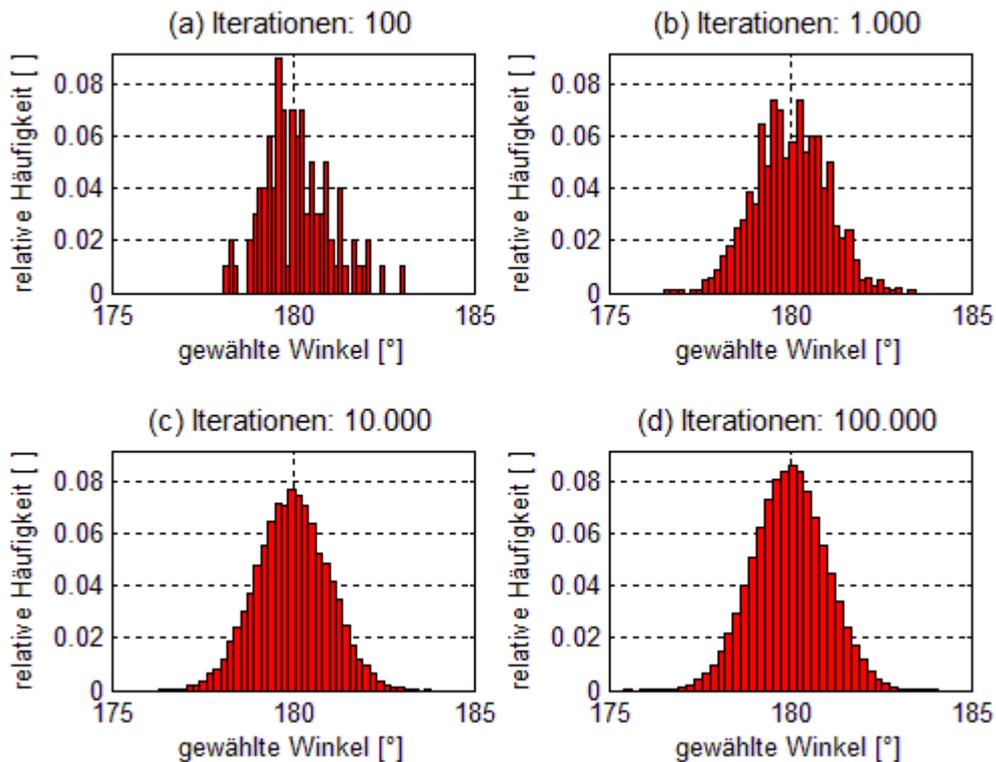


Abbildung 4.5: Gewählte Winkel für eine unterschiedliche Anzahl an Iterationen unter der Annahme einer Normalverteilung $NV(180,1)$

Wie der Abbildung 4.5 zu entnehmen ist, folgt auch hier mit zunehmender Anzahl an Iterationen der Prozess immer deutlicher einer NV. Bei 100 Iterationen (Abb. 4.5a) ist für die relative Häufigkeit des Auftretens eines Winkelintervalls der Charakter einer NV noch nicht gut zu erkennen. Einzig die Tatsache, dass keine Winkel im minimalen bzw. maximalen Winkelbereich gewürfelt worden sind, ist für die NV typisch. Ab 10.000 Iterationen (Abb. 4.5c) lässt sich im Wesentlichen die Glockenkurve erkennen, welche mit steigender Anzahl an Iterationen (Abb. 4.5d) immer mehr Form annimmt. Somit greift auch hier das Gesetz der großen Zahlen.

Abbildung 4.6 zeigt die Verteilung für die gewählten Winkel nach 10 Millionen Iterationen. Die Anzahl der gewählten Intervalle liegt diesmal bei 120. Die Achseneinteilung wird identisch zu der aus Abbildung 4.5 gewählt. Das dargestellte Intervall für den gewürfelten Winkel aus Abbildung 4.6b wird auf den Bereich $[178,5; 181,5]$ und zusätzlich die Hochachse auf das Intervall $[0,015; 0,036]$ beschränkt. Indem dieser Ausschnitt mit einer $NV(180,1)$ überlagert wird (blaue Kurve), zeigen sich die geringen Abweichungen zwischen den beiden Plots. Somit werden auch für den Fall der NV die Winkel entsprechend der gewählten Verteilungsfunktion ausgesucht, sodass die Erzeugung der Sequenz von Zufallszahlen mittels PRNG zufällig erfolgt.

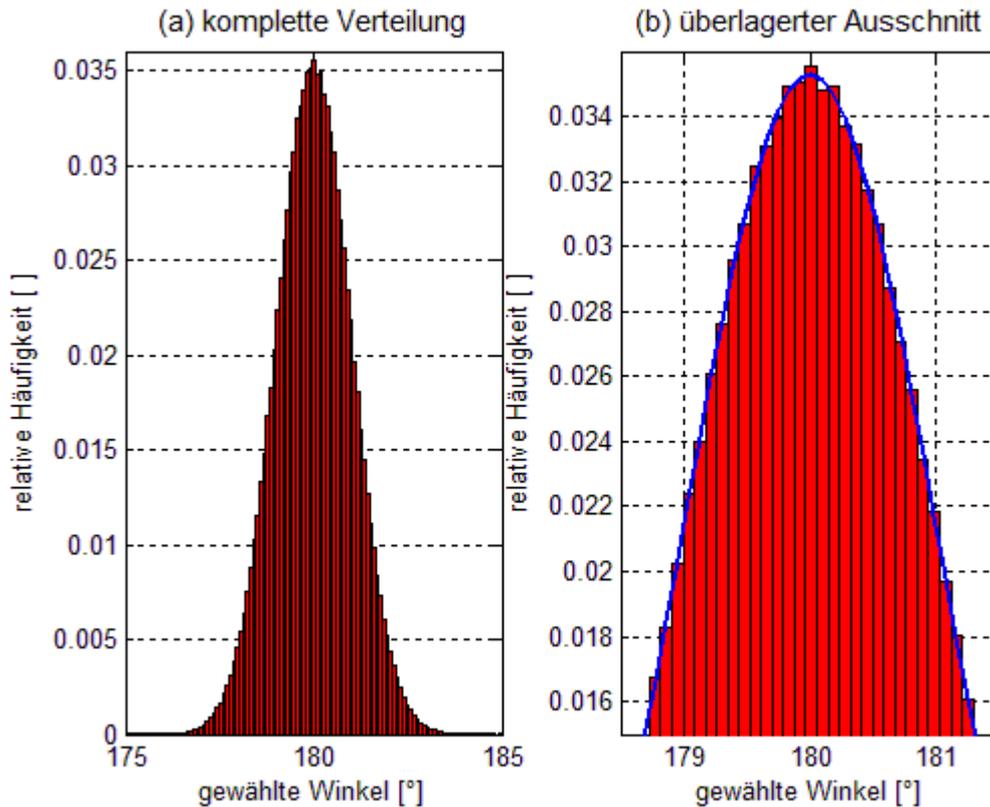


Abbildung 4.6: Gewählte Winkel bei 10.000.000 Iterationen unter der Annahme einer Normalverteilung

Als letztes werden die Geburtsorte untersucht. Ähnlich wie bei den Winkeln wird der Ort (Bildkoordinaten des Kreuzungspunktes) zufällig bestimmt, an welchem der neue Knoten erzeugt wird. Auch hier gilt es zu überprüfen, ob die Geburtsorte der gewählten Verteilungsfunktion folgen. Anders als bei den bisherigen Analysen zu den gewählten Kernen und Winkeln, erfolgen hier die Untersuchungen weniger auf quantitativer sondern auf visueller Ebene. Insbesondere soll gezeigt werden, wie unterschiedliche Verteilungsfunktion (hier GV und NV) genutzt werden können, damit sich der Algorithmus auf bestimmte, interessante Bereiche des Bildes (in diesem Fall die Priele) konzentriert.

Zunächst werden die Bildkoordinaten der Geburtsorte bzw. -positionen auf Basis einer GV gewürfelt. In Anlehnung an Abbildung 2.3a sollten die Positionen dieser Knoten des Punktprozesses gleichmäßig und unabhängig im betrachteten Gebiet verteilt sein. In diesem Fall ist der Bereich das aus den Laserscannerdaten erzeugte DGM des deutschen Wattenmeeres. Die Anzahl der Geburtseignisse (rote Punkte) ist für Abbildung 4.7a auf 500, für Abbildung 4.7b auf 30 festgelegt. Die Anzahl an Iterationen wurde für dieses Beispiel bewusst klein gehalten, damit der zufällige Charakter des Prozesses ersichtlich wird.

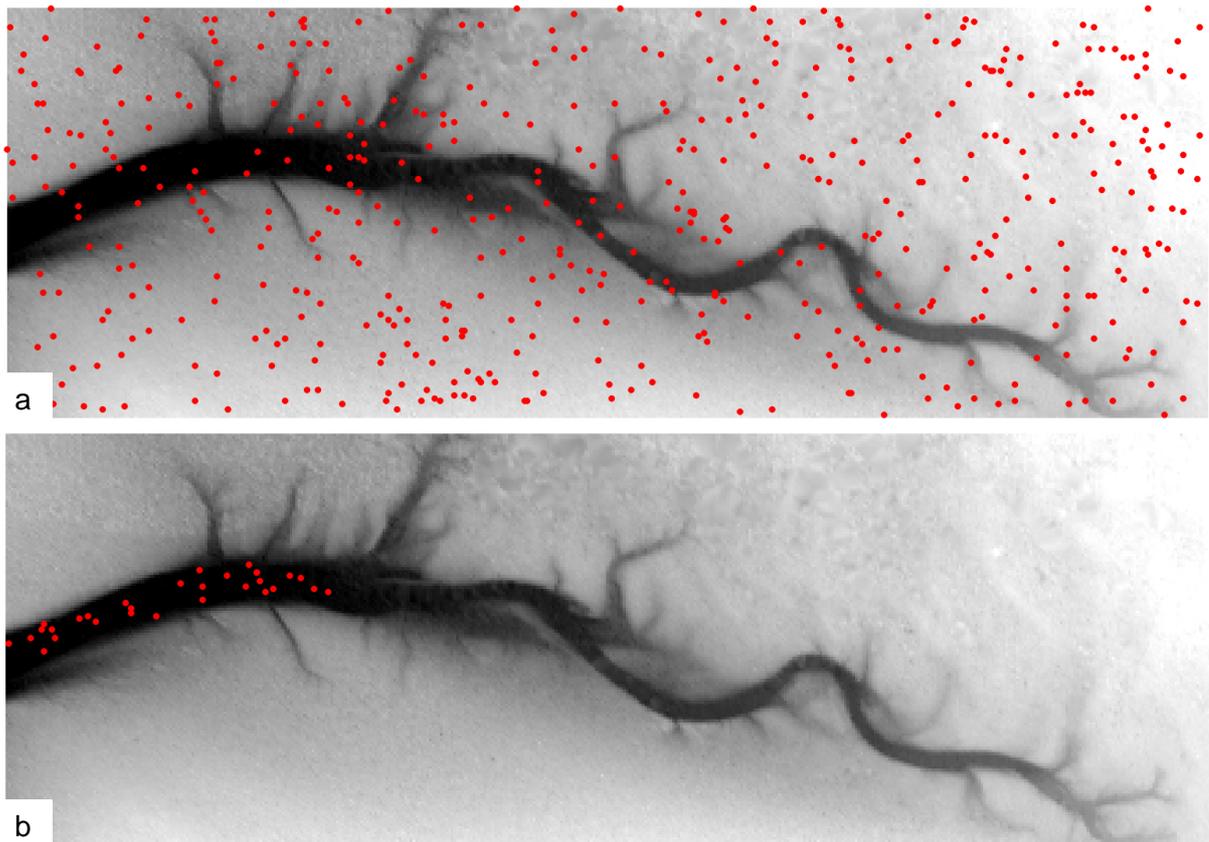


Abbildung 4.7: Geburtseignisse mit 500 (a) bzw. 30 (b) Geburten (rot) für einen Poisson-Punktprozess auf Basis einer GV

Abbildung 4.7a zeigt, dass kein Ort bevorzugt aufgesucht wird. Somit handelt es sich um einen homogenen Poisson-Punktprozess (Kap. 2.3.1.1). Da die Geburtsorte nicht vom DGM selbst abhängen, hätte zur Darstellung auch ein weißer Hintergrund mit den Dimensionen des Bildes verwendet werden können. Anders ist es für Abbildung 4.7b, bei welcher der Schwellwert z für den maximalen Höhenwert festgelegt wird, bis zu welchem potentielle Orte für ein Geburtseignis gewählt werden (Kap. 3.1). Ein Grauwert innerhalb des DGMs entspricht einem bestimmten Höhenwert. Hierdurch werden nur in bestimmten Bereichen des Priel – in Abhängigkeit des Schwellwertes – neue Knoten erzeugt. Auch hier lässt sich kein Trend beobachten.

Im Folgenden wird die NV an einem weiteren Beispiel untersucht. Grundlage bilden auch hier die Orte für die Geburtseignisse, deren Bildkoordinaten nun auf Basis der NV gewürfelt werden. Die Idee dahinter ist es, dass in der Mitte des Priel öfters ein Knoten erzeugt werden soll, als zum Rand hin, wo der Priel flacher wird. Wird für den Mittelwert der NV der tiefste Wert des Testgebietes (Prielmitte) verwendet, sollten die Geburtseignisse hier häufiger erfolgen. Dies wäre von Vorteil, da hierdurch die von den Knoten ausgehenden Segmente den Prielverlauf schneller korrekt wiedergeben würden, weil der Versatz zwischen

dem Ist- und dem Soll-Zustand bereits geringer wäre. Je nach gewählter Standardabweichung werden weitere Orte zugelassen.

Um diese Überlegung zu verifizieren, wird ein stark vereinfachtes Prielmodell erzeugt. Der Mittelwert der NV wird auf den tiefsten Wert (schwarz) gesetzt und die Standardabweichung derart gewählt, dass auch in den höheren Bereichen des Priels (hellgrau) noch Knoten geboren werden. Insgesamt treten 250 Geburtereignisse auf. Diese sind in Abbildung 4.8 entsprechend der fünf verschiedenen Höhen, die in diesem Beispiel vorkommen, eingefärbt. Die absolute Häufigkeit der jeweiligen Höhe ist ebenfalls aufgeführt.

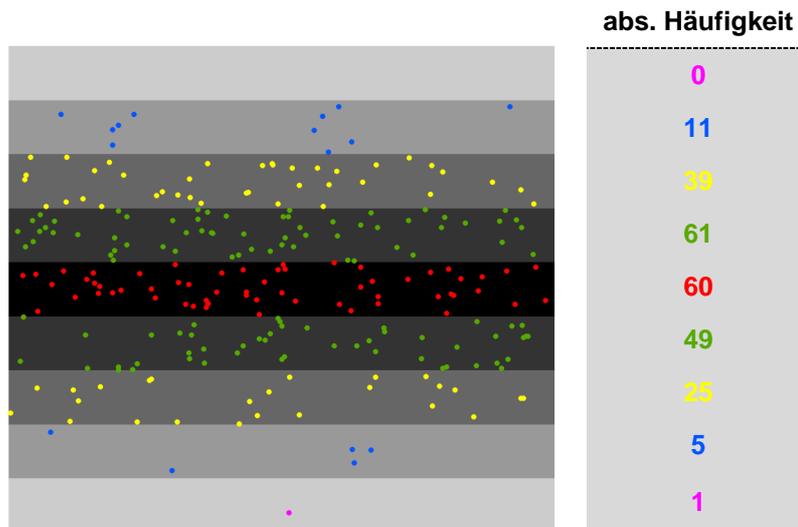


Abbildung 4.8: 250 Geburtereignisse auf Basis einer Normalverteilung für ein stark vereinfachtes Prielmodell

Wie zu erwarten, nehmen die Geburtsorte zum Rand des Priels hin ab. Somit spiegelt sich die modellierte Verteilung in den synthetischen Daten wider, wenn auch in diesem Beispiel der nördliche Bereich des Priels mehr Knoten als der südliche enthält. Diese Abweichung von der NV kann auf die geringe Strichprobenanzahl zurückgeführt werden.

Nachfolgend wird das Verfahren auf die Daten des Wattenmeeres angewendet (Abb. 4.9). Der Mittelwert liegt für dieses Beispiel bei dem minimal vorkommenden Grauwert innerhalb des DGMs, die Standardabweichung wird auf 25 gesetzt. Die Anzahl der Iterationen liegt bei 1000.

In Abbildung 4.9 lässt sich die Charakteristik, dass viele Knoten an der tiefsten Stelle entstehen und diese zum Rand hin abnehmen, nicht beobachten. Dies kann durch die Eigenschaften der echten Daten begründet werden. Das DGM verfügt über eine raue Oberfläche, sodass sich auch der Rand der Priele nicht wie bei den synthetischen Daten ausprägen wird. Somit unterscheiden sich die Höhenwerte von benachbarten Pixeln teils

stark. Ferner zeigt sich in der Ergebnissen, dass im westlichen Bereich des Priels deutlich häufiger ein neuer Knoten entsteht, als es im östlicheren Bereich der Fall ist. Dies ist auf die zunehmende Höhe der Priele von Westen nach Osten zurückzuführen. Durch eine abweichende Wahl der Parameter der NV können in anderen Bereichen mehr Knoten erzeugt werden.

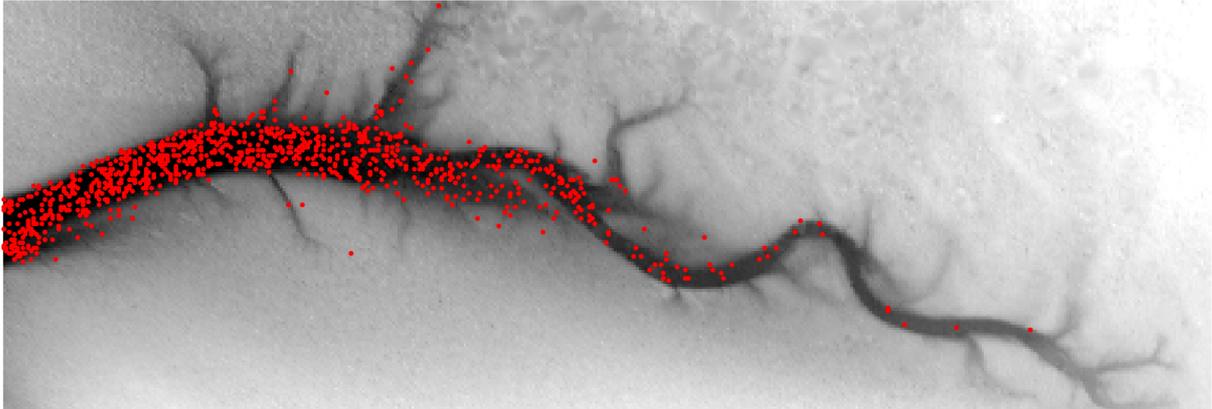


Abbildung 4.9: 1000 Geburtseignisse (rot) auf Grundlage einer Normalverteilung $NV(49, 25)$

Aufgrund der unregelmäßigen Höhen der Priele in echten Daten und der üblicherweise vorhandenen Steigung im Gelände ist es fraglich, ob die NV geeignet für die Wahl dieses Parameters ist. Zumal sollte in der Regel das Ziel sein, das gesamte Liniennetzwerk (aus Priele) und nicht nur einen Ausschnitt zu detektieren. Wird in den tieferen Bereichen der Priele sehr häufig und in den flacheren nur selten ein neuer Knoten geboren, so kommt es bei den tieferen Priele zu einer Häufung von Knoten. Demnach ist bei der Modellierung der Geburtsorte von der NV abzuraten, zumal auch für die Wahl des Mittelwertes und der Standardabweichung viel Vorwissen über die Daten vorhanden sein muss.

Zusammenfassung der Analysen zur Verteilung der Parameter

Indem unterschiedliche Verteilungsfunktionen verwendet werden, kann Vorwissen über bestimmte Merkmale des untersuchten Gebietes in die Analysen mit einfließen. Durch die individuelle Anpassung an die Problemstellung ist es möglich, dass der Algorithmus gegebenenfalls schneller gegen das Energieminimum konvergiert. Nicht für alle Parameter stellt dieselbe Verteilungsfunktion die optimale Wahl dar. So bietet sich in dem von Schmidt et al. (2015) entwickelten Modell zum Beispiel die GV bei der Auswahl der Kernel an. Zur Modellierung der Zwischenwinkel zwischen benachbarten Segmenten kann dort die NV verwendet werden, wo die Winkel zwischen angrenzenden Segmenten bzw.

Flussabschnitten im Testgebiet in der Regel gleichgroß (zum Beispiel 180°) sind. Weiterhin könnte die NV für einen Flussverlauf mit wenigen Abzweigungen bei der Anzahl der abgehenden Segmente von einem bereits vorhandenen Knoten (Abb. 3.1) genutzt werden (Mittelwert = 2). Darüber hinaus ist es denkbar, dass sich der Algorithmus mithilfe der NV auf bestimmte Bereiche des Bildes (z. B. tiefere im Vergleich zu höheren Priel) mehr konzentriert. Deshalb ist es wichtig, dass im Allgemeinen die Parameter entsprechend der gewünschten Verteilung gewählt werden, was in den obigen Untersuchungen gezeigt werden konnte. Diese Schlussfolgerung lässt sich neben den exemplarisch untersuchten Parametern auch auf die anderen Parameter bzw. die richtige Arbeitsweise des verwendeten PRNGs *mt19937* übertragen.

Zwei wesentliche Gesichtspunkte sind hierbei zu berücksichtigen. Einerseits handelt es sich um Zufallsexperimente, sodass sich die erwartete Verteilung in den Ergebnissen mal besser und mal schlechter – insbesondere für einen geringen Stichprobenumfang – widerspiegelt. Allerdings wird sich nach dem Gesetz der großen Zahlen die relative Häufigkeit um die theoretische Wahrscheinlichkeit des Zufallsergebnisses bei zunehmender Anzahl an Iterationen stabilisieren (Abb. 4.4d, 4.6b, Tab. 1). Das Sampling beruht auf exakt dieser Eigenschaft, da erst durch eine sehr große Anzahl an Wiederholungen die Energie gegen ein Minimum konvergiert (zweiter Gesichtspunkt). Bei Schmidt et al. (2015) liegt diese Anzahl der Iterationen beispielsweise im Bereich von 30 Millionen.

4.4 Untersuchungen zur modellierten 3D-Komponente

Wie in Kapitel 4.2 erwähnt, kann die Initialisierung eines Pseudo-Zufallszahlengenerators derart erfolgen, dass die erzeugte Sequenz von Zufallszahlen für jedes Sampling identisch ist. Indem der Generator mit demselben Wert initialisiert wird, kann eine Reproduzierbarkeit von Ergebnissen erreicht werden. Diese ist von Vorteil, um die Resultate besser untereinander vergleichen und einfacher Unterschiede bei der Wahl verschiedener Parameter erkennen zu können. Ferner hilft dies auch beim Auffinden von Fehlern. Diese Eigenschaften wurden bereits in Kapitel 3.3 zur Erzeugung der dortigen Beispiele sowie zur Implementierung des Fließmodells genutzt. Darüber hinaus ist die Reproduzierbarkeit insbesondere für die nachfolgenden Analysen von Vorteil, da auf diese Weise zum Beispiel gezeigt werden kann, inwiefern sich die zusätzlich modellierte 3D-Komponente unter gleichen Bedingungen auf das Ergebnis auswirkt.

In den folgenden Kapiteln werden verschiedenste Untersuchungen im Hinblick auf das in Kapitel 3.3 erläuterte Fließmodell durchgeführt. Hierfür werden zunächst die synthetischen Daten mit dem für das Wattenmeer charakteristischen Verlauf verwendet (Kap. 4.4.1). Im ersten Abschnitt werden Resultate der ausschließlichen Verwendung des Fließmodells gezeigt. Hierbei soll im Speziellen geklärt werden, ob das Fließmodell ohne die anderen drei Komponenten der Priorenergie und den Datenterm aus Schmidt et al. (2015) zur Detektion des Prielverlaufes geeignet ist. Daran schließen Untersuchungen an, inwiefern die Erweiterung in Kombination mit dem bestehenden Modell genutzt werden und hierdurch bereits ohne Datenterm ein Bezug zu den Prielen geschaffen werden kann. Daraufhin folgen Analysen von Ergebnissen, bei denen der Datenterm berücksichtigt worden ist. In diesem Zusammenhang wird untersucht, ob Terme der Priorenergie weggelassen werden können, um die Anzahl der Parameter des Algorithmus zu reduzieren. Ebenfalls wird der Frage nachgegangen, inwiefern sich die Terme auf die Laufzeit des Programms auswirken. Im letzten Abschnitt werden die Untersuchungen zusammengefasst.

Grundlage für die darauf folgenden Analysen in Kapitel 4.4.2 bilden die im deutschen Wattenmeer erhobenen LIDAR-Daten. Die Parametereinstellungen werden dabei wie bei Schmidt et al. (2015) gesetzt. Die Untersuchungen sind ähnlich zu denen der synthetischen Daten, wobei die Darstellung der Ergebnisse kürzer gefasst wird. Zunächst wird analysiert, inwiefern das Fließmodell einen Einfluss auf die Resultate hat. Der darauffolgende Abschnitt zeigt Auswirkungen auf die Ergebnisse, falls Komponenten der Priorenergie weggelassen werden. Eine Zusammenfassung der Resultate wird am Ende des Kapitels gegeben.

Abschließend wird die 3D-Komponente auf den vorarlbergischen Datensatz angewendet (Kap. 4.4.3). Im Gegensatz zu den beiden vorherigen Kapiteln wird hierbei lediglich getestet, inwiefern sich das Fließmodell auf die Ergebnisse auswirkt.

4.4.1 Synthetische Daten

Alle nachfolgenden Untersuchungen innerhalb dieses Kapitels basieren auf den synthetischen Daten des komplexeren Prielmodells (Abb. 4.1c). Hierbei sind die Parameter zunächst wie bei Schmidt et al. (2015) gesetzt. So wird u. a. ähnlich Abbildung 4.7b der Parameter z für den maximalen Höhen- bzw. Grauwert, bis zu welchem potentielle Orte für ein Geburtsereignis gewählt werden, anders festgelegt (Kap. 3.1). Dieser wird auf 120 eingestellt, was dem maximalen Grauwert innerhalb des Priels entspricht. Weiterhin werden der Schwellwert für die minimale und maximale Breite eines Segmentes unter Verwendung von Vorwissen über die Daten auf $b_{min} = 10 \text{ m}$ bzw. $b_{max} = 90 \text{ m}$ gesetzt (Abb. 3.1).

Ausschließliche Verwendung des Fließmodells

Im Folgenden wird untersucht, inwiefern das (kombinierte) Fließmodell ohne die anderen drei Komponenten der Priorenergie und den Datenterm aus Schmidt et al. (2015) zur Detektion des Prielverlaufs geeignet ist. Der Intensitätsparameter λ des Poisson-Prozesses (Kap. 2.3.1.1) wird auf 30 und der Radius r (Kap. 3.1) auf 100 m gesetzt. Der Ausschnitt ist aus Gründen der Anschaulichkeit auf den für die nachfolgenden Analysen wesentlichen Bereich des Priels beschränkt. Gleichermaßen wird auf die Darstellung der Segmente verzichtet, zumal diese aufgrund des fehlenden Datenterms keinen Bezug zum Priel besitzen.

Abbildung 4.10a zeigt ein Ergebnis, nachdem der Algorithmus konvergiert ist. Die dargestellte Objektkonfiguration wird nicht bestraft, da alle im Fließmodell geforderten Bedingungen (Kap. 3.3) erfüllt sind. Die Auf- und Abbewegungen der Kanten innerhalb des Priels resultieren daher, dass die Grauwerte in nordsüdlicher Richtung bis zum Beginn des Prielrandes überall denselben Wert aufweisen. Hierdurch wird der Algorithmus sich nicht bevorzugt auf einen Bereich weiter südlich oder nördlich konzentrieren. Hauptsache ist, dass die einzelnen Bedingungen des Fließmodells in Summe erfüllt sind. Jedoch ist das Resultat insofern unbrauchbar, da das Prielnetzwerk nicht durch die modellierte 3D-Komponente detektiert wird. Zwar führt das Fließmodell dazu, dass unerwünschte Objektkonfigurationen vermieden werden, allerdings kommt es durch dessen ausschließliche Verwendung nicht zu zweckmäßigen Resultaten.

Dies gilt auch für weitere Untersuchungen unter gleichen Parametereinstellungen (andere Zufallszahlensequenz). Exemplarisch ist in Abbildung 4.10b ein weiteres Ergebnis dargestellt. Auch hier erfüllen die Teilgraphen zwar die geforderten Bedingungen. Es entsteht jedoch keinesfalls ein gewünschter zusammenhängender Graph, welcher mehr oder weniger die gesamte Prielstruktur erfasst. Dieses Verhalten des Algorithmus lässt sich dadurch begründen, dass die Kanten des Graphen durch den fehlenden Datenterm nicht in die Mitte des Priels gezogen werden. Hierdurch würden die Rechtecke im optimalen Fall mit dem Prielrand abschließen. Zum anderen erlauben die fehlenden Terme der Priorenergie aus Schmidt et al. (2015) eine beliebige Anordnung der Kanten bzw. Segmente. Erstere dürfen daher willkürliche Winkel zueinander aufweisen und letztere sich nach Belieben überlappen. Ferner müssen die Segmente nicht verbunden sein (Formel 3.2). Hierdurch ist es unmöglich, dass der Algorithmus gegen eine brauchbare Lösung konvergiert.

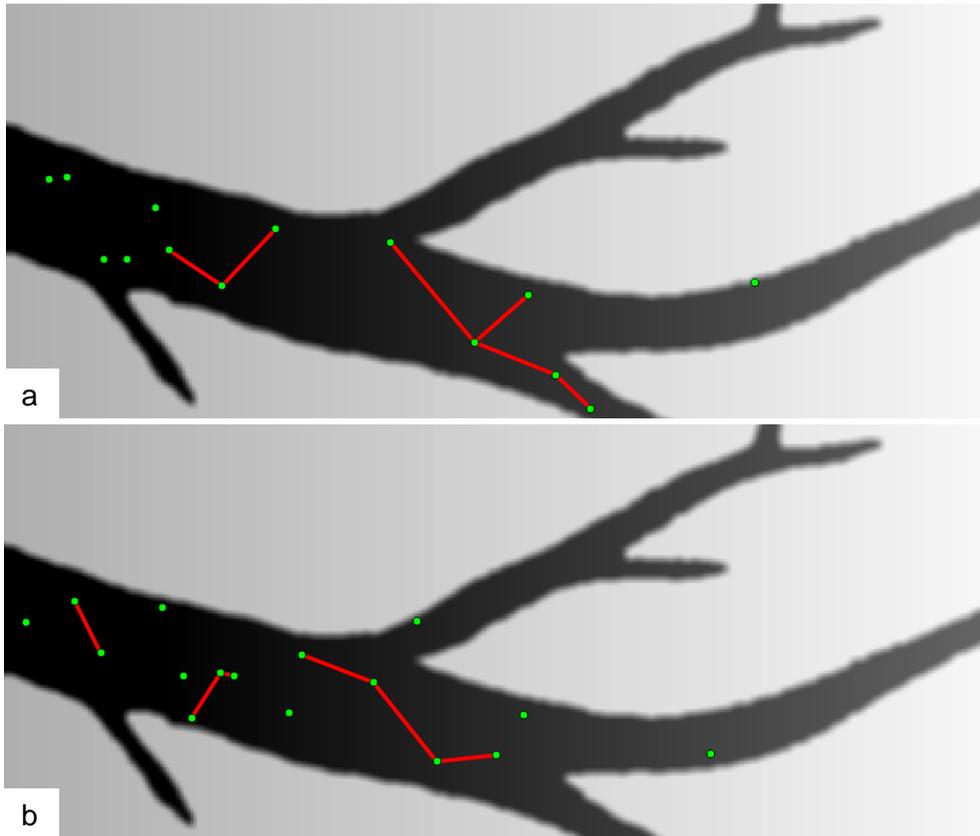


Abbildung 4.10: Ergebnisse unter ausschließlicher Verwendung des Fließmodells

Unterm Strich kann festgehalten werden, dass die 3D-Komponente ohne weitere Modelle nicht genutzt werden kann, um akzeptable Ergebnisse zu erzeugen. Aus diesem Grund werden im folgenden Unterkapitel neben dem Fließmodell die von Schmidt et al. (2015) genutzten Priorterme verwendet und geschaut, inwiefern hierdurch eine Optimierung erzielt werden kann.

Kombination aus Priortermen und dem Fließmodell

Nachfolgend wird analysiert, inwiefern sich das Modell der drei Terme der Priorenergie aus Schmidt et al. (2015) in Kombination mit der 3D-Komponente auf den beim Sampling erzeugten Graphen auswirkt. In Abbildung 4.11 werden zugehörige Ergebnisse aufgezeigt. Die Wahl der Parameter p und r ist identisch zum vorherigen Kapitel und auch der Datenterm wird nicht verwendet. Der Parameter des Fließmodells f (Formel 3.3) wird ausgehend von $f = 1$ für Abbildung 4.11b jeweils um den Faktor 10 erhöht (Abb. 4.11c-d). Für Abbildung 4.11a wird das Fließmodell nicht berücksichtigt ($f = 0$).

In Abbildung 4.11a ist das Ergebnis ein zusammenhängender Graph, der keine sich überlappenden Segmente (aus Gründen der Übersichtlichkeit nicht gezeigt) und keine von der favorisierten Winkelkonstellation abweichende Verhältnisse besitzt. Dies zeigt die korrekte Funktionsweise dieser Modelle. Aufgrund des fehlenden Datenterms ist kein Bezug zu den Daten (Hintergrund) vorhanden. Bei einer (zunehmenden) Bestrafung der vom Fließmodell abweichenden Objektkonfiguration passt sich der Graph dem Priel insofern an, dass die Konfiguration nach deren Konvergenz geringer (Abb. 4.11b-c) oder gar nicht mehr (Abb. 4.11d) bestraft wird. Hierdurch kann ein Bezug zu den Daten hergestellt werden, da die Bedingungen des Modells verhindern, dass das Prielnetzwerk beispielsweise von den Kanten des Graphen gekreuzt wird. Anzumerken ist, dass im Allgemeinen der Algorithmus nicht gegen eine Lösung konvergieren muss, bei welcher Terme der Priorenergie Null werden. In Abhängigkeit von der Wahl der Faktoren werden die Bestrafungen – wie für den Term $U_F(X_t)$ in Abbildung 4.11 gezeigt – unterschiedlich ausfallen.

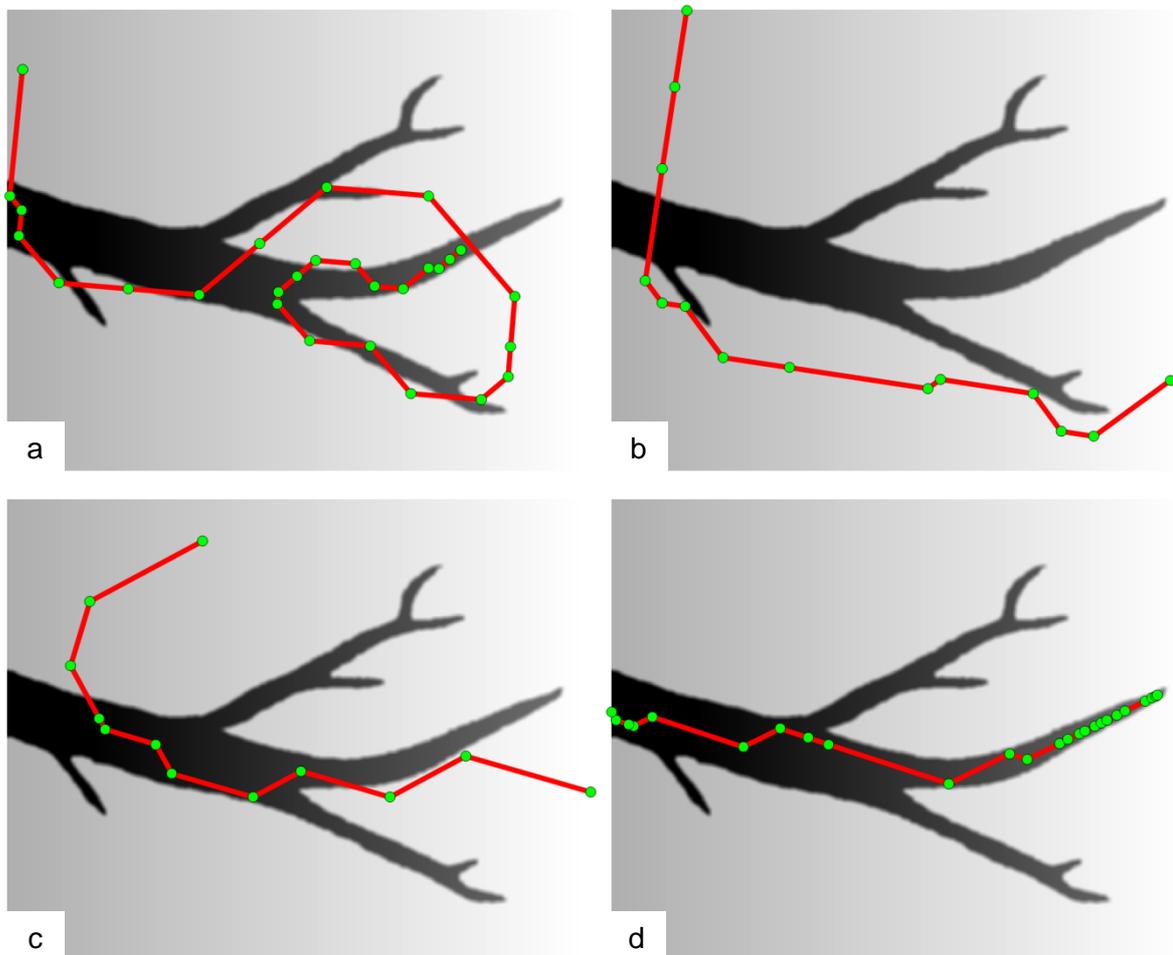


Abbildung 4.11: Priortermine aus Schmidt et al. (2015) mit dem Fließmodell kombiniert
 (a): ausschließliche Verwendung der Priortermine aus Schmidt et al. (2015)
 (b) bis (d): Priortermine aus Schmidt et al. (2015) um das 3D-Modell erweitert, wobei der Faktor der Bestrafung für die vom Modell abweichenden Objektkonfigurationen von b nach d zunimmt

Bei den Untersuchungen ist aufgefallen, dass beim Herauslaufen des Graphen aus dem Priel (Abb. 4.12a, hellblauer Kreis) die Objektkonfiguration in bestimmten Fällen bestraft wird, in anderen jedoch nicht. Ursächlich hierfür ist das Prinzip des Fließmodells 2 (Abb. 3.3). Da u. a. für jeden Pixel entlang der Kante die aktuelle Höhe mit der vorherigen verglichen wird, wirkt sich der Verlauf der Kante auf das Ergebnis aus. So wird die Objektkonfiguration nicht bestraft werden, falls beispielsweise der diagonale Weg über die Pixelecken genommen wird (Abb. 4.12b). Für dieses Beispiel (links unten beginnend) ergibt sich die Grauwertfolge 112, 113, 113, 124, 137 und 151, die alle Bedingungen erfüllt. Falls der Verlauf des Graphen nur leicht anders ist, kommt es in diesem Beispiel bereits zu einer Bestrafung (Abb. 4.12c). Auch hier werden alle von der Kante geschnittenen Pixel bei der Betrachtung der Grauwerte herangezogen, sodass sich als Teilfolge 112, 112, 113, 114 und 113 ergibt. Da hierdurch eine Bedingung des Fließmodells 2 verletzt wird, wird die Konfiguration bestraft. Die Abbildungen 4.12b und 4.12c beziehen sich dabei auf den in Abbildung 4.12a durch einen hellblauen Pfeil gekennzeichneten Bereich des Prielrandes.

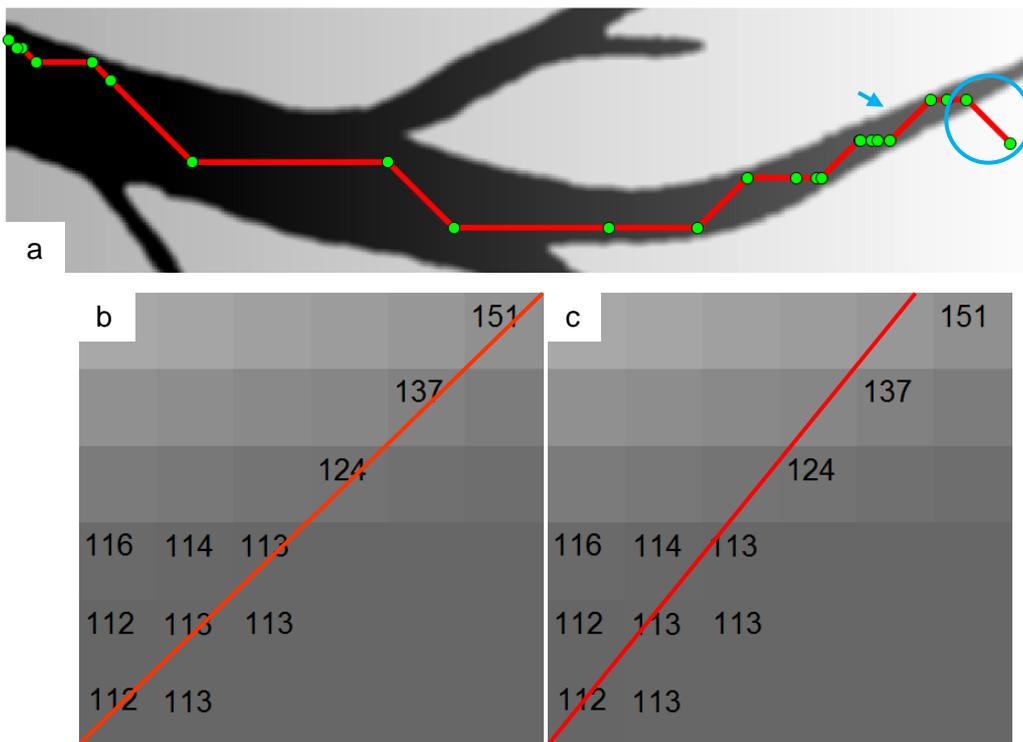


Abbildung 4.12: Problem des Überlaufs einer Kante des Graphen über den Prielrand

**(a): Beispielhaftes Ergebnis
(b) und (c): Veranschaulichung der Problematik**

Damit das Überlaufen der Kante über den Rand des Priels einheitlich behandelt werden kann, wird das zweite Fließmodell um eine zusätzliche Bedingung erweitert. Hierzu wird obendrein gefordert, dass sich der Höhenunterschied zwischen dem Grauwert des aktuell

betrachteten und des vorherigen Pixels entlang der Kante im Betrag nicht um mehr als einen Grauwert unterscheiden darf. Demnach muss für $h_A < h_E$ die Bedingung $(h_{P_i} - h_{P_{i-1}}) \leq 1$ und analog für $h_A > h_E$ die Bedingung $(h_{P_{i-1}} - h_{P_i}) \leq 1$ gelten. Die Annahme kann für die synthetischen Daten getroffen werden, da innerhalb des Priel diese Forderung immer erfüllt ist. Sobald die Kante auf den Prielrand trifft, wird die Bedingung verletzt und die aktuelle Objektkonfiguration bestraft. Es hat sich gezeigt, dass durch diese Erweiterung dem unvorhersehbaren Verhalten an Prielrändern entgegengewirkt werden kann. Aus diesem Grund wird die zusätzliche Bedingung für die hier untersuchten synthetischen Daten beibehalten. Die Resultate aus Abbildung 4.11 wurden bereits unter deren Verwendung erzeugt.

Im Folgenden soll das Ziel verfolgt werden, möglichst das gesamte Prielnetzwerk mithilfe des beim Sampling erzeugten Graphen zu detektieren. Dabei sollte der Graph im optimalen Fall aus einer vorzugsweise geringen Anzahl von Teilgraphen bestehen und die Objektkonfiguration das gesamte Prielnetzwerk in zweckmäßiger Weise erfassen. Das heißt, es sollten möglichst keine sich überlappenden oder freien Segmente oder andere dem Modell widersprechende Konfigurationen vorhanden sein. Wie auch zuvor wird der Datenterm aus Schmidt et al. (2015) nicht verwendet. Zur Detektion wird eine Optimierung der Parameter vorgenommen, da zurzeit lediglich der Hauptarm des Priel gefunden wird (Abb. 4.11d). Im Wesentlichen sind dies der Radius r , der Intensitätsparameter λ , sowie die vier Parameter s, w, k, f (Formel 3.3) der Priorenergie.

Während der Experimente hat sich gezeigt, dass die Ergebnisse stark von der Wahl der Parameter abhängen. Ein geeignetes Resultat ergibt sich für $\lambda = 120$ und $r = 25$, während die einzelnen Komponenten der Priorenergie auf $s = 20, w = 10, k = 50$ und $f = 10$ gesetzt werden. In Abbildung 4.13a ist das Ergebnis für diese Parameterwahl dargestellt. Es ist zu beobachten, dass der Graph aus drei Teilen besteht, die das Prielnetzwerk weitestgehend abdecken. Die kleineren Ausläufer werden dabei bis auf eine Ausnahme nicht gefunden. Der Algorithmus konvergiert gegen eine Lösung, bei welcher die Bestrafung für das Fließmodell nicht Null ist. Dies begründet die Tatsache, dass der eine Teilgraph über den Prielrand hinaus laufen kann (hellblauer Pfeil). Auffällig sind die nicht über eine Kante miteinander verbundenen Teilgraphen im Bereich des hellblauen Kreises. Dieses Problem, dass ein Knoten nicht über drei ausgehende Kanten verfügt (Dreierkreuzung), konnte bereits bei den Ergebnissen von Schmidt et al. (2015) festgestellt werden. Bei Untersuchungen im nächsten Abschnitt sind hierfür mögliche Gründe aufgeführt.

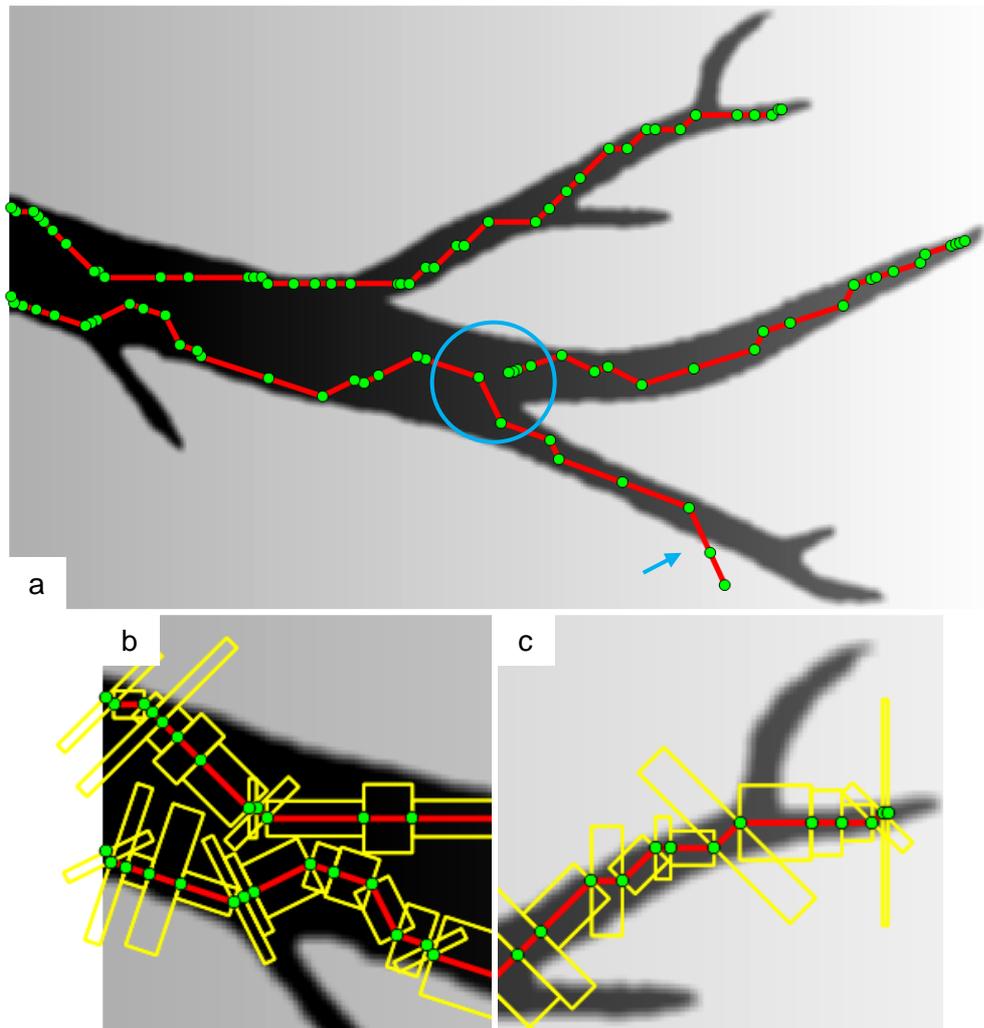


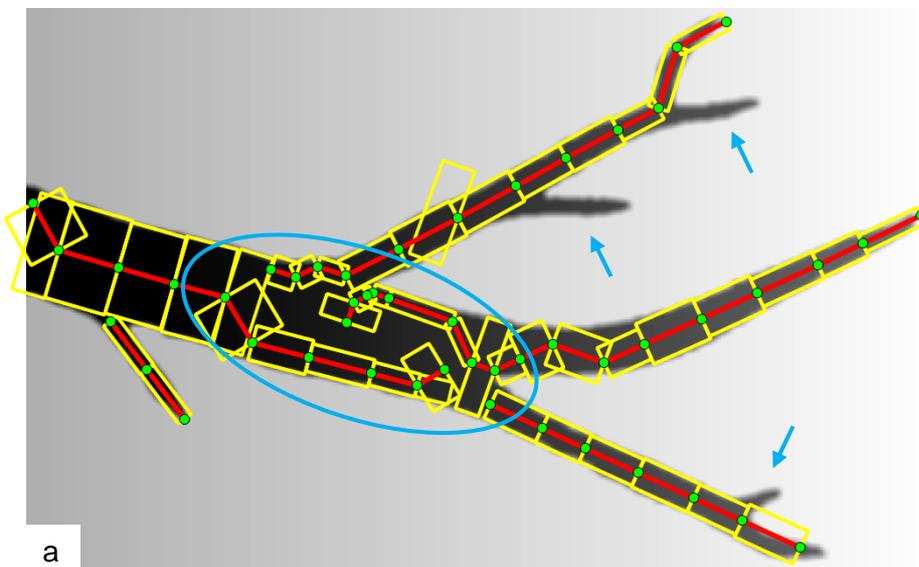
Abbildung 4.13: Ergebnis unter ausschließlicher Verwendung der vier Komponenten der Priorenergie ($s = 20$, $w = 10$, $k = 50$, $f = 10$)

Ferner ist in Abbildung 4.13a zu erkennen, dass im Westen des Priels zwei Teilgraphen übereinander verlaufen. Dies ist zwar keine gewollte Objektkonfiguration, jedoch ist diese nicht vermeidbar. Wie Abbildung 4.13b zeigt, sind die Segmente (gelb) derart angeordnet, dass sie sich nicht überlappen. Somit spricht nichts dafür, dass der Algorithmus diese Konfiguration unterbindet, zumal in diesem Bereich auch die beiden anderen Modelle erfüllt sind. Weiterhin fällt der fehlende Bezug der Segmente zum Priel bzw. dessen Rand auf (Abb. 4.13b, c), welcher die Motivation für die nachfolgenden Untersuchungen darstellt. Aus diesem Grund werden die im Datenterm berechneten Gradienten genutzt, um die bestehende Objektkonfiguration – insbesondere den Bezug zum Prielrand – zu optimieren. Hierdurch müsste zum Beispiel eine Konfiguration wie in Abbildung 4.13b verhindert und der Kantenverlauf in die Mitte des Prielnetzwerks verschoben werden.

Kombination aus Daten- und erweiterter Priorenergie

Nachfolgend wird überprüft, ob mithilfe des Datenterms der Bezug des Graphen zum Prielnetzwerk hergestellt werden kann. Dafür werden in diesem Abschnitt die Parametereinstellungen aus dem vorherigen Kapitel verwendet und zusätzlich die Daten- zur Gesamtenergie hinzugezogen. Hierbei kann der Datenterm über den Parameter β (Formel 3.1) unterschiedlich stark gewichtet werden. In Abbildung 4.14 sind Ergebnisse für verschiedene Gewichtungen des Parameters aufgeführt, wobei diese von Abbildung 4.14a zu Abbildung 4.14c zunimmt.

In Abbildung 4.14 ist zu erkennen, dass durch die Verwendung des Datenterms der Bezug der Segmente zum Prielnetzwerk hergestellt werden kann. Die Kanten des Graphen verlaufen nun überwiegend geradlinig durch die Prielmitte, da auf diese Weise die Gradientenbeträge auf beiden Seiten der Segmente maximal werden. Darüber hinaus lässt sich innerhalb des gesamten Graphen kein Knoten finden, welcher über mehr als zwei ausgehende Kanten verfügt. Aus diesem Grund entstehen an allen potentiellen Dreier- und zusätzlich auch einigen Zweierkreuzungen neue Teilgraphen. Dabei ist die Anzahl der Teilgraphen bereits im Vergleich zu Abbildung 4.13a erhöht und nimmt bei stärkerer Gewichtung des Datenterms weiter zu. Folglich muss sich der Datenterm im negativen Sinne auf die Anzahl der Teilgraphen auswirken. Daneben hat der größere Einfluss des Datenterms zusätzliche freie Segmente zur Folge und wirkt sich auch anderweitig auf die Objektkonfiguration aus. So verläuft der Graph nun vereinzelt über den Prielrand, während freie Segmente in diesen Bereichen mit den Prielrändern abschließen (orange Kreise). Dies zeigt den nun vergleichsweise stärkeren Einfluss der Daten- gegenüber der Priorenergie. Positiv zu bewerten sind die zusätzlich gefundenen Ausläufer der Nebenarme (hellblaue Pfeile). Auch wird der vorher nicht von Segmenten erfasste Bereich geschlossen (hellblauer Kreis).



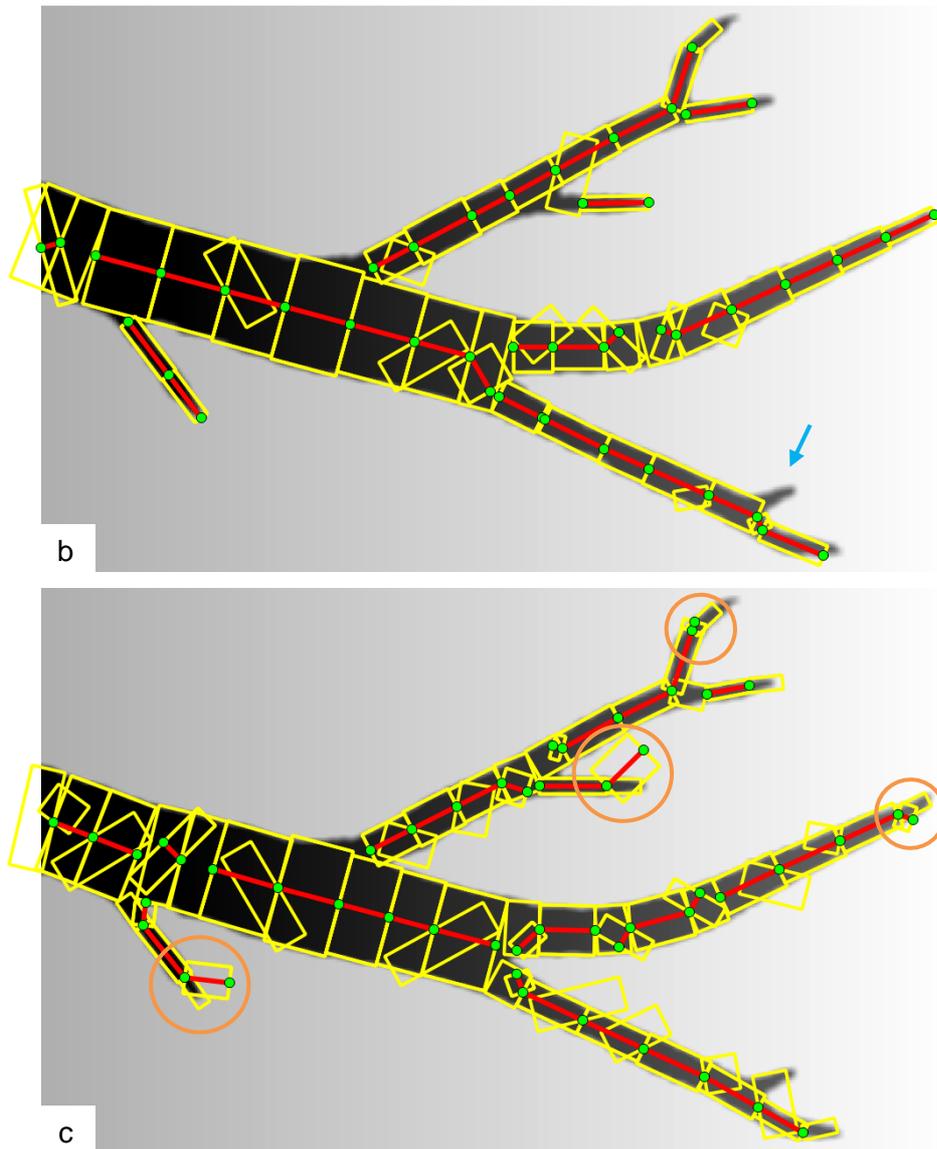


Abbildung 4.14: Ergebnis unter Verwendung der Daten- sowie der erweiterten Priorenergie ($s = 20$, $w = 10$, $k = 50$, $f = 10$)
(a): Einfluss des Datenterms: $\beta = 0.05$
(b): Einfluss des Datenterms: $\beta = 0.10$
(c): Einfluss des Datenterms: $\beta = 0.20$

Insgesamt zeigt sich, dass die Wahl der Parameter beim Verfahren der markierten Punktprozesse einen erheblichen Einfluss auf die Ergebnisse hat. Es ist mitunter mit viel Aufwand verbunden, eine für die Szene geeignete Parameterwahl zu finden. Dabei unterscheiden sich bereits bei geringer Variation der Gewichtung des Datenterms die Ergebnisse teils stark (Abb. 4.14). Da neben diesem noch zahlreiche andere Parameter vorhanden sind, kann eine falsche Wahl schnell zu unbrauchbaren Ergebnissen führen. Aus diesem Grund wird das Weglassen von Termen getestet. Die Überlegung dahinter ist es, einem potentiellen Nutzer ein Programm bereitzustellen, welches über möglichst wenige Stellschrauben verfügt. Hierzu gehört beispielsweise die bestmögliche Reduktion der zu

wählenden Parameter des Algorithmus, wobei die Qualität der Ergebnisse nicht zu sehr beeinflusst sein darf.

Im Folgenden wird untersucht, ob einzelne Terme der Priorenergie weggelassen werden können. Die Basis für die weiteren Analysen stellt die Parameterwahl aus Abbildung 4.14b dar, da diese den besten Kompromiss zwischen gefundenen Objekten (Ausläufer u. Ä.) und geeignetem Kantenverlauf darstellt.

Abbildung 4.15 zeigt einen Ausschnitt des nördlichen Nebenarms, wobei für dieses Ergebnis der Term $U_S(X_t)$, nicht berücksichtigt worden ist. Durch die fehlende Bestrafung von sich überlappenden Segmenten können Geburten neuer Knoten in unmittelbarer Nähe zueinander stattfinden. Aufgrund der stetig zunehmenden Anzahl an Segmenten, wird sich auch die Energie des Datenterms von Iteration zu Iteration minimieren. Folglich werden durch den dominierenden Datenterm die Bestrafungen der Priortermine kontinuierlich zunehmen und das gesamte zugrunde liegende Modell versagt. Dies gilt konsequenterweise auch dann, wenn zusätzlich noch andere Terme der Priorenergie nicht berücksichtigt werden.

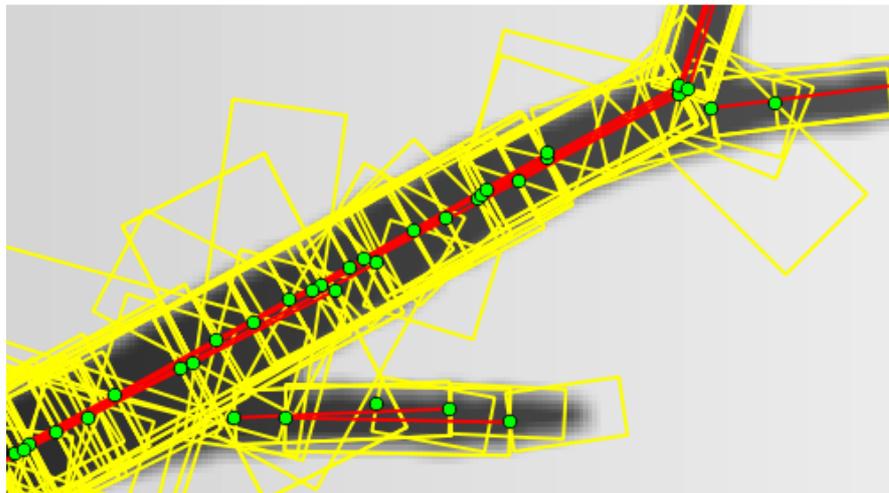


Abbildung 4.15: Ergebnis, falls sich überlappende Segmente als eine Komponente der Priorenergie nicht berücksichtigt werden

Bei weiteren Analysen hat sich ergeben, dass ohne die Bestrafung der Winkelabweichungen sowie freien Segmente brauchbare Ergebnisse erzielt werden können. Abbildung 4.16 zeigt diese, falls die beiden Terme $U_W(X_t)$ und $U_K(X_t)$ weggelassen werden. Es ist zu erkennen, dass so gut wie an jedem Knoten ein freies Segment vorhanden ist. Auf der anderen Seite sind viele Aspekte vergleichsweise als positiv zu bewerten. So werden alle Ausläufer der Nebenarme gefunden. An den Enden des nördlichen und südlichen Nebenarms sind Knoten mit jeweils drei ausgehenden Kanten vorhanden (hellblaue Kreise), wodurch die Anzahl der

Teilgraphen im Vergleich zu Abbildung 4.14b reduziert ist. Dies lässt vermuten, dass das Winkelkriterium unter Umständen einen negativen Einfluss auf die Erzeugung von Dreierkreuzungen mit sich bringt. Falls die Szene bevorzugte Zwischenwinkel (Kap. 3.1) nicht ermöglicht, muss immer derart von der priorisierten Konfiguration abgewichen werden, dass es aufgrund der Bestrafung zur Verwerfung kommt. Jedoch kann dies keinesfalls der einzige Grund sein, da ansonsten ein vollständig zusammenhängender Graph entstehen müsste. Es liegt nahe, dass zusätzlich der Datenterm einen großen Einfluss auf die Geschlossenheit des Graphen, wie in den Abbildungen 4.14a-c gezeigt werden konnte, hat. So verhindert beispielsweise ein freies Segment die Erzeugung einer Dreierkreuzung, da es ansonsten zu einer Überlappung kommen würde (hellblauer Pfeil). Dies zeigt erneut die starke Abhängigkeit der Ergebnisse von den Parametern und auch den gegenseitigen Einfluss untereinander, sodass in diesem Zusammenhang noch weitere Untersuchungen notwendig wären.

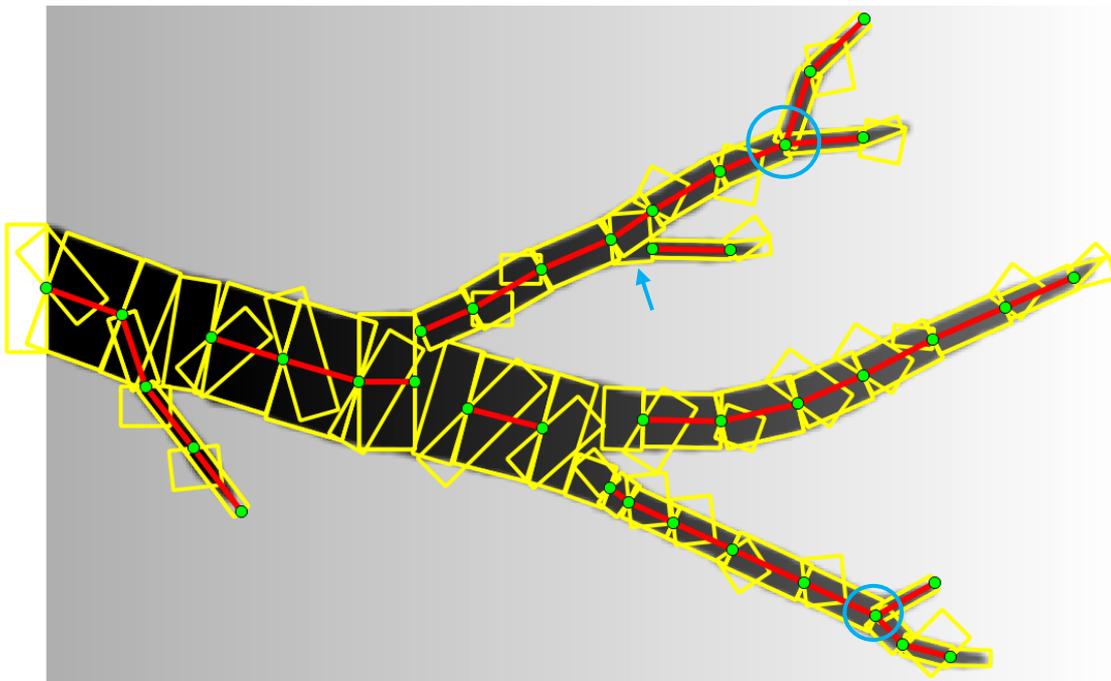


Abbildung 4.16: Ergebnis, falls die Kombination aus Winkelabweichungen und freien Segmenten als Terme der Priorenergie nicht einbezogen wird

Wird das Fließmodell bei gleichen Parametereinstellungen wie in den Abbildungen 4.14b und 4.16 nicht berücksichtigt, lassen sich keine merklichen Unterschiede feststellen. Auf die Darstellung der Resultate wird an dieser Stelle verzichtet. In Kapitel 4.4.2 werden Ergebnisse solcher Untersuchungen aufgezeigt.

Es ist zu vermuten, dass dies u. a. mit den zweckmäßigen Prielrändern (hohe Gradientenbeträge) der synthetischen Daten zusammenhängt. Auf diese Weise kann die

Prielstruktur bereits durch den Datenterm gut erfasst werden. Ein weiterer wesentlicher Aspekt ist, dass bereits geeignete Parameter für die gegebenen Daten gefunden worden sind. Insbesondere dann, wenn die Wahl der Parameter noch nicht in dieser Form erfolgt ist, unterstützt das Fließmodell dabei, ungewollte Objektkonfigurationen zu vermeiden. Abbildung 4.17 zeigt ein Beispiel, bei dem die Parameterwahl zu den vorherigen Ergebnissen vergleichsweise ungeeignet ist. So ist beispielsweise der Datenterm verhältnismäßig gering gewichtet worden. Die Abbildung zeigt lediglich den interessanten Ausschnitt des gesamten Bildes.

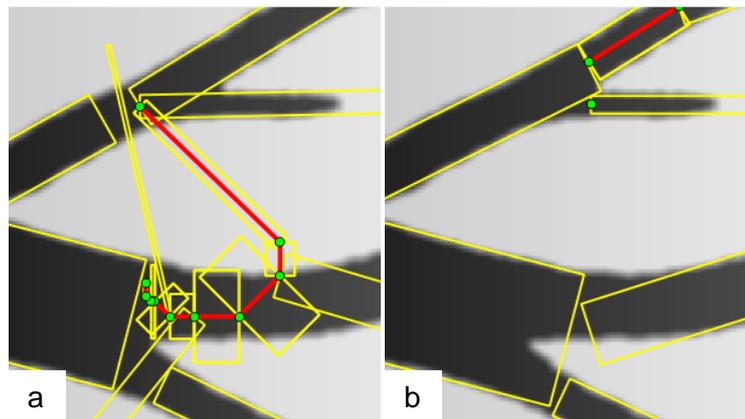


Abbildung 4.17: Beispiel unter Verwendung der Daten- sowie der erweiterten Priorenergie, welches einen möglichen Nutzen des Fließmodells aufzeigt
(a): keine Berücksichtigung des Fließmodells
(b): Berücksichtigung des Fließmodells

Laufzeitanalysen legen dar, dass zusätzliche Komponenten der Priorenergie die Rechenzeit des Algorithmus beschleunigen. Werden die Winkelabweichungen und die freien Segmente bei den Berechnungen nicht mit herangezogen, ist die Laufzeit um rund zehn Prozent erhöht. Wird darüber hinaus noch der Parameter f für das Fließmodell zu Null gesetzt, erhöht sich diese abermals um wenige Prozentpunkte. Dies klingt wenig, jedoch sind in Abhängigkeit der Parameterwahl und der gegebenen Daten Berechnungen über mehrere Stunden keine Seltenheit.

Zusammenfassung der Analysen auf Basis der synthetischen Daten

Im ersten Abschnitt des Kapitels 4.4.1 konnte gezeigt werden, dass die ausschließliche Verwendung der modellierten 3D-Komponente keine brauchbaren Ergebnisse liefert. Zwar führt die Kombination des ersten und zweiten Fließmodells dazu, dass unerwünschte Objektkonfigurationen vermieden werden können, jedoch wird das Prielnetzwerk als solches unzureichend detektiert (Abb. 4.10).

Im nachfolgenden Abschnitt wurden neben dem Modell zur Berücksichtigung der Fließrichtung die von Schmidt et al. (2015) genutzten Priorterme verwendet. Hierdurch ließ sich ein Bezug zu den Daten herstellen, da die Bedingungen des 3D-Modells verhindern, dass es beispielsweise zu einem Überlauf von Kanten über die Ränder des Prielnetzwerks kommt (Abb. 4.11). In diesem Zusammenhang wurde das vorhandene Fließmodell um eine weitere Bedingung ergänzt. Auf diese Weise kann dem unvorhersehbaren Verhalten an Prielrändern im Hinblick auf die Bestrafung entgegenwirkt und diese vereinheitlicht werden (Abb. 4.12). Abschließend konnte festgestellt werden, dass es durch die um die vierte Komponente erweiterte Priorenergie (ohne Datenterm) möglich ist, den Prielverlauf mehr oder weniger wiederzugeben. Jedoch stehen die Segmente dabei in keinem Zusammenhang mit den Prielen bzw. dessen Rand (Abb. 4.13).

Aus diesem Grund wird der Datenterm aus Schmidt et al. (2015) zu den Berechnungen hinzugezogen. Hierdurch konnten die Kanten des Graphen in die Mitte des Priels gezogen werden, sodass sich die Segmente in Form und Lage in das Prielnetzwerk einfügen. Voraussetzung hierfür ist eine für die Szene geeignete Parameterwahl, wobei das Auffinden dieser mitunter mit erheblichem Aufwand verbunden ist. Zusätzlich hat die Wahl einen starken Einfluss auf die Ergebnisse, was Analysen für die Variation bereits eines einzelnen Parameters (Gewichtung des Datenterms mithilfe des Parameters β) gezeigt haben (Abb. 4.14). Es folgten Untersuchungen, ob einzelne Komponenten der Priorenergie weggelassen werden können, um die Anzahl der Parameter zu reduzieren. Hierdurch soll einem potentiellen Nutzer die Bedienbarkeit des Programms erleichtert werden. Es konnte gezeigt werden, dass der Term $U_S(X_t)$ zur Bestrafung von sich überlappenden Segmenten von großer Bedeutung ist. Ansonsten wird das Prielnetzwerk im Laufe des Programmdurchlaufs mehrfach an denselben oder ähnlichen Stellen detektiert (Abb. 4.15). Weitere Analysen legten dar, dass die Komponenten $U_W(X_t)$ und $U_K(X_t)$ der Priorenergie zur Erzeugung von geeigneten Ergebnissen nicht notwendigerweise benötigt werden. Ferner konnte gezeigt werden, dass das Winkelkriterium negative Auswirkungen auf Dreierkreuzungen mit sich bringt (Abb. 4.16). In diesem Zusammenhang wurde erneut auf die starke Abhängigkeit der Ergebnisse von der Wahl der Parameter und auch auf den gegenseitigen Einfluss untereinander hingewiesen. Neben diesen Untersuchungen wurde geschaut, ob das 3D-Modell die Ergebnisse wesentlich beeinflusst. Im Großen und Ganzen konnte kein bedeutsamer Einfluss bei der zusätzlichen Verwendung des Fließmodells für die synthetischen Daten festgestellt werden (weiteres in Kap. 4.4.2 und 4.4.3). Jedoch kann beispielsweise die Berücksichtigung des Fließmodells von Vorteil sein, falls noch keine geeigneten Parameter gefunden worden sind, um ungewollte Konfigurationen zu umgehen (Abb. 4.17). Abschließend wurden Laufzeitanalysen durchgeführt. Diese haben ergeben, dass Komponenten der Priorenergie die benötigte Zeit bis zur Konvergenz des Algorithmus

reduzieren. Aus diesem Grund gilt es abzuwägen, ob Terme bei den Berechnungen mit einbezogen werden sollen, jedoch hierdurch die Anzahl der zu wählenden Parameter erhöht ist.

4.4.2 Daten des deutschen Wattenmeeres

Da die verwendeten synthetischen Daten aus Kapitel 4.4.1 keine raue Oberfläche aufweisen, ist zu vermuten, dass die Bedingungen des Fließmodells für echte Daten zu hart formuliert sind. Zu hohe Grauwertdifferenzen benachbarter Pixel innerhalb der Daten des deutschen Wattenmeeres werden zu einer Bestrafung der Objektkonfiguration führen. Hierdurch können sich die Segmente dem Prielnetzwerk nur bedingt anpassen. Würde das Gewicht der Bestrafung zu hoch gewählt werden, wäre es unmöglich, einen zusammenhängenden Graphen zu detektieren. Aus diesem Grund werden die Eingangsdaten im Vorfeld mit einem Medianfilter geglättet, wobei der Median aus den 25 benachbarten Pixeln (Pixel im Zentrum mit eingeschlossen) bestimmt wird (Kap. 3.3). Auch die zusätzliche Bedingung, die bei den Analysen der synthetischen Daten eingeführt worden ist, wird vernachlässigt. Jedoch ist trotz der Einschränkungen zu mutmaßen, dass das Fließmodell in dieser Form nach wie vor zu hart formuliert ist. Dieser Vermutung und inwiefern das Fließmodell die Objektkonfiguration beeinflussen kann, wird zu Beginn des Kapitels nachgegangen. Der folgende Abschnitt setzt sich mit der Frage auseinander, wie sich Komponenten der Priorenergie auf Ergebnisse auswirken. Abschließend werden die Resultate zusammengefasst. Basis der Untersuchungen bilden die bei Schmidt et al. (2015) gewählten Parametereinstellungen. Ansonsten wird in den nachfolgenden Abschnitten gesondert darauf hingewiesen.

Untersuchungen des 3D-Modells

Abbildung 4.18a zeigt das Resultat für eine identische Parameterwahl, wie sie bei Schmidt et al. (2015) verwendet wurde. Da die Initialisierung des Zufallszahlengenerators nicht mit demselben Wert erfolgt ist, unterscheidet sich das Ergebnis von dem des Papers leicht. Ausgehend von diesem Resultat wird im Folgenden untersucht, inwiefern sich die Erweiterung um die 3D-Komponente auf die Ergebnisse auswirkt. Auch hier wird der Pseudo-Zufallszahlengenerator für alle nachfolgenden Analysen mit demselben Wert, wie er für Abbildung 4.18a gewählt wurde, initialisiert. In Abbildung 4.18b werden die vom

Fließmodell abweichenden Objektkonfigurationen mit dem Faktor $f = 10$ – in Anlehnung an die Resultate bei den synthetischen Daten – bestraft.

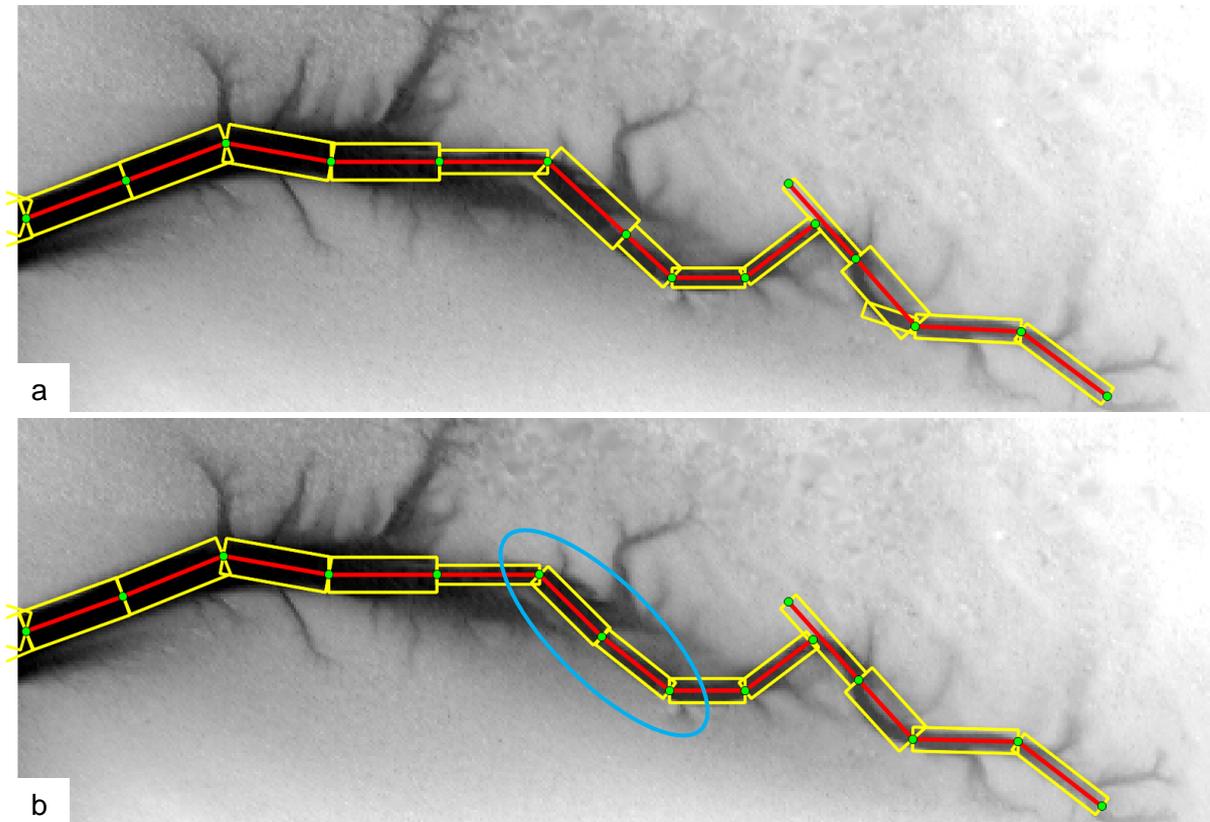


Abbildung 4.18: Ergebnis für eine identische Parameterwahl wie bei Schmidt et al. (2015)
(a): keine Berücksichtigung des Fließmodells
(b): Berücksichtigung des Fließmodells mit $f = 10$

Wie zu entnehmen, ähneln sich die Ergebnisse der Abbildungen 4.18a und 4.18b stark, was bereits in Untersuchungen der synthetischen Daten festgestellt werden konnte. Interessant ist der Verlauf des Graphen für das durch einen hellblauen Kreis hervorgehobene Gebiet. Aufgrund der Bedingungen des Fließmodells hat sich der mittlere Knoten vom Wattboden etwas nach Südwesten in den Priel verschoben. Zusätzlich passen sich die Segmente dem Prielrand durch eine geringere Breite besser an.

Jedoch ist es von großer Bedeutung, dass die Gewichtung der 3D-Komponente nicht zu groß ausfällt. Abbildung 4.19 zeigt das Ergebnis, falls der Term $U_F(X_t)$ der Priorenergie mit $f = 100$ gewichtet wird. Es ist zu beobachten, dass der aus vorher zwei Teilgraphen bestehende Graph jetzt aus insgesamt fünf Abschnitten besteht und Lücken aufweist. Weiterhin wird der Hauptarm des Priel insgesamt deutlich schlechter detektiert.

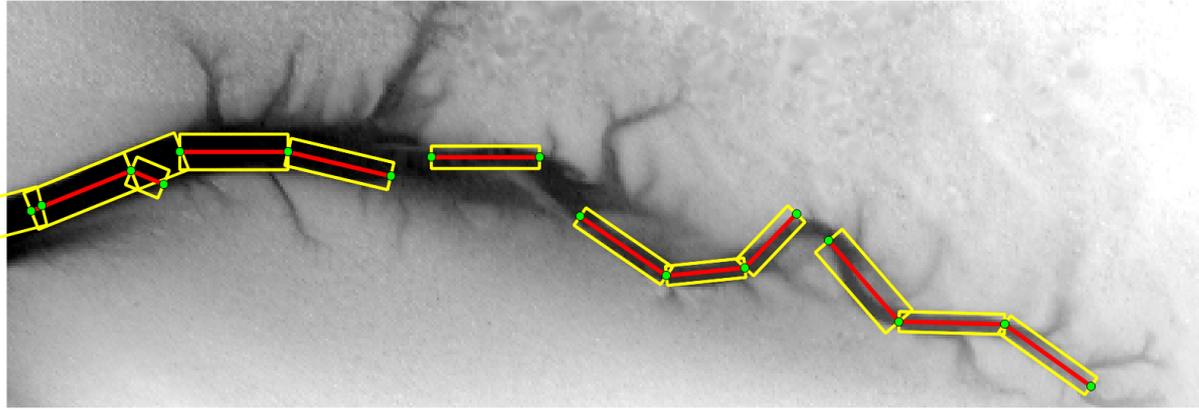


Abbildung 4.19: Ergebnis für eine identische Parameterwahl wie bei Schmidt et al. (2015) und zusätzlichem Fließmodell ($f = 100$)

Wie obige Ergebnisse zeigen, sind diese Bedingungen für die gegebenen Daten – trotz der Verwendung des Medianbildes – zu hart formuliert. Insbesondere führen die Forderungen des Fließmodells 2 aufgrund der unregelmäßigen Grauwerte benachbarter Pixel zu zahlreichen Bestrafungen. Aus diesem Grund wird das bestehende Modell abgeschwächt und der Parameter a auf $a = 10$ gesetzt (Kap. 3.3). Hierdurch kann der Aufspaltung des Graphen in Teilgraphen entgegengewirkt werden (Abb. 4.20). Die abgeschwächten Bedingungen bleiben für nachfolgende Analysen erhalten. Wie bereits bei den Untersuchungen der synthetischen Daten festgestellt werden konnte, besteht auch hier die Objektkonfiguration im Bereich des hellblauen Kreises nicht aus einer Dreierkreuzung. Während in allen vorherigen Abbildungen dieses Kapitels das Problem des Verlaufs einer Kante über den Wattboden vorhanden ist, ist dies nun nicht mehr der Fall (hellblauer Pfeil).

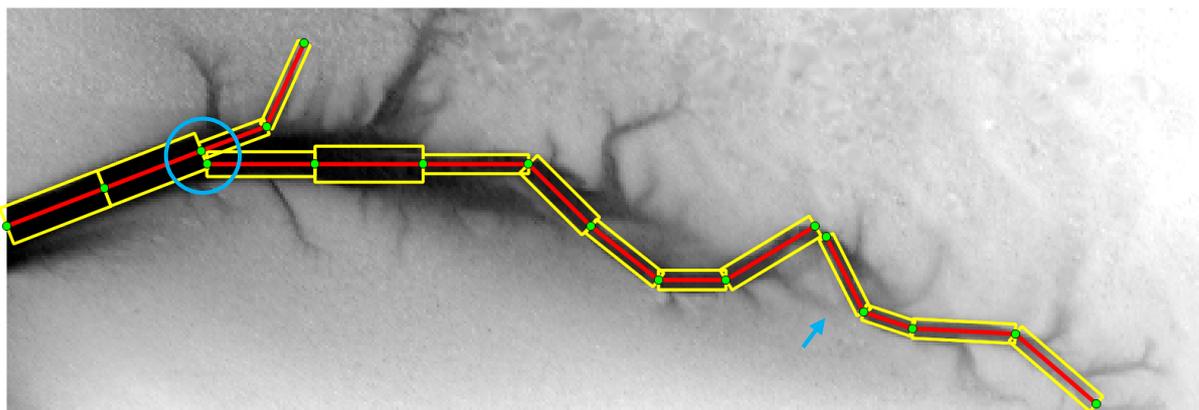


Abbildung 4.20: Ergebnis für eine identische Parameterwahl wie bei Schmidt et al. (2015) und zusätzlichem Fließmodell mit abgeschwächten Bedingungen ($f = 100$)

Mit Bezug zu vorherigen Untersuchungen, bei denen der positive Einfluss des Fließmodells bei geringer Gewichtung des Datenterms getestet worden ist (Abb. 4.17), sollen diese Analysen ebenfalls für die Wattdaten realisiert werden. Hierzu wird das Gewicht für den

Datenterm von $\beta = 0.3$ auf $\beta = 0.13$ reduziert. In Abbildung 4.21a ist das Fließmodell nicht und in Abbildung 4.21b mit $f = 100$ berücksichtigt.

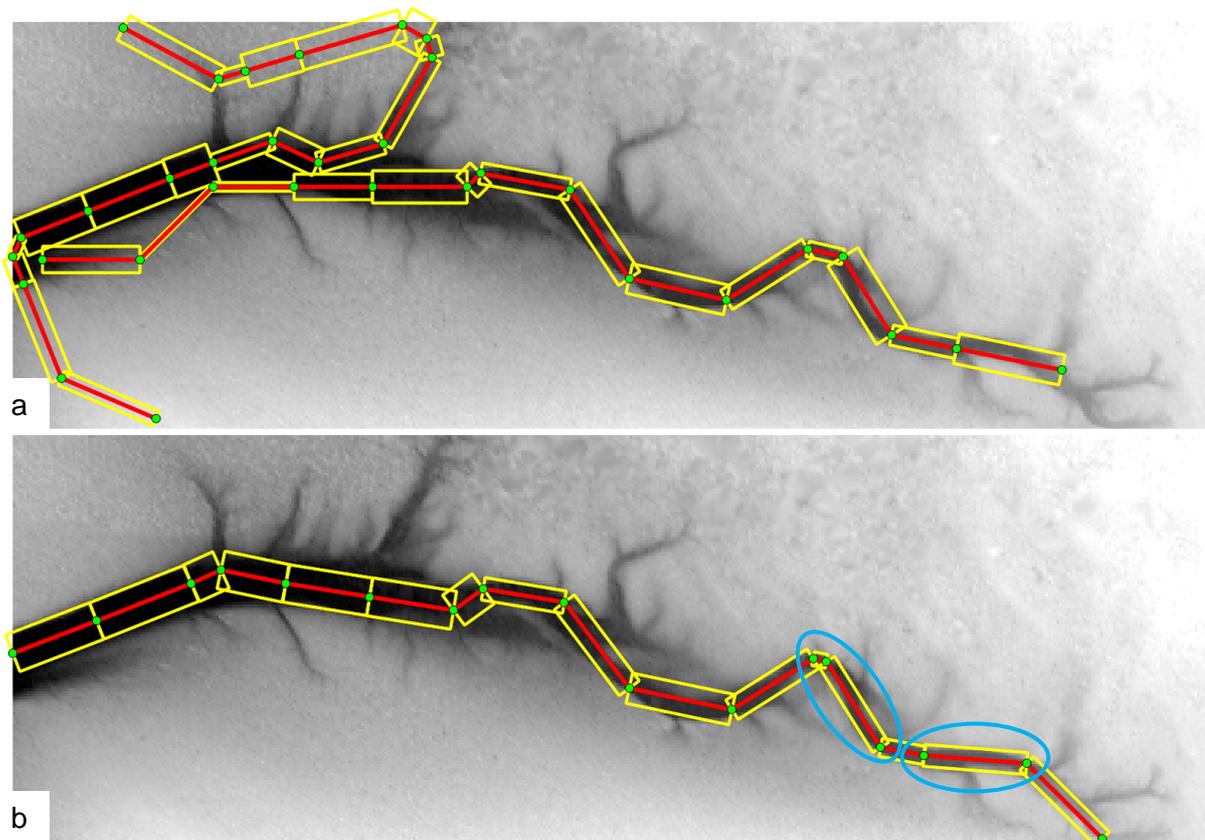


Abbildung 4.21: Ergebnis für eine identische Parameterwahl wie bei Schmidt et al. (2015), wobei der Datenterm mit $\beta = 0.13$ gewichtet ist
(a): keine Berücksichtigung des Fließmodells
(b): Berücksichtigung des abgeschwächten Fließmodells mit $f = 100$

Wie Abbildung 4.21a zu entnehmen ist, kommt es im westlichen Bereich des Priels zu zwei übereinander verlaufenden Teilgraphen, die an einigen Stellen aus dem Priel herauslaufen. Die Verwendung des Fließmodells führt zur Behebung dieser ungewollten Objektkonfiguration (Abb. 4.21b). Darüber hinaus lassen sich der Konfiguration noch weitere positive Eigenschaften zusprechen. So verfügt diese über einen einzigen Teilgraphen, der den Verlauf des Priels geeigneter wiedergibt. Beispielsweise passen sich im Bereich der hellblauen Kreise die Segmente besser dem wahren Prielverlauf an, sodass die Kanten nicht mehr über den Wattboden verlaufen. Somit zeigt sich auch für die echten Daten, dass das Fließmodell zu einer Optimierung der Ergebnisse führen kann. Insbesondere stützt es den Datenterm, wenn dieser verhältnismäßig gering gewichtet wird.

Untersuchungen der Notwendigkeit von Prioritermen

Im Folgenden werden in Anlehnung an Untersuchungen bei den synthetischen Daten Terme der Priorenergie weggelassen. Es soll geschaut werden, wie sich dies auf die Ergebnisse auswirkt und ob hierdurch gegebenenfalls die Anzahl der Parameter reduziert werden kann. In Abbildung 4.22 sind die Terme $U_W(X_t)$, $U_K(X_t)$ und $U_F(X_t)$ nicht berücksichtigt worden. Dies führt zu einer deutlich ungeordneteren Objektkonfiguration als zum Beispiel in Abbildung 4.20. Es sind zahlreiche Teilgraphen vorhanden, deren Segmente sich zusätzlich noch an einigen Stellen überlappen. Auch lassen sich aufgrund der fehlenden Komponenten der Priorenergie zwei Dreierkreuzungen (hellblaue Kreise), diverse freie Segmente und detektierte Ausläufer beobachten. Einige der Ausläufer sind exemplarisch durch hellblaue Pfeile gekennzeichnet. Die Detektion dieser wird u. a. dadurch begünstigt, dass freie Segmente nicht bestraft werden. Ferner spielt der nun verhältnismäßig stärkere Einfluss der Daten- gegenüber der Priorenergie eine Rolle. Hierdurch wird es erst möglich, Priele zu detektieren, die über nicht so stark ausgeprägte Ränder bzw. Gradienten verfügen. Dies trifft im Allgemeinen bei Ausläufern in echten Daten zu.

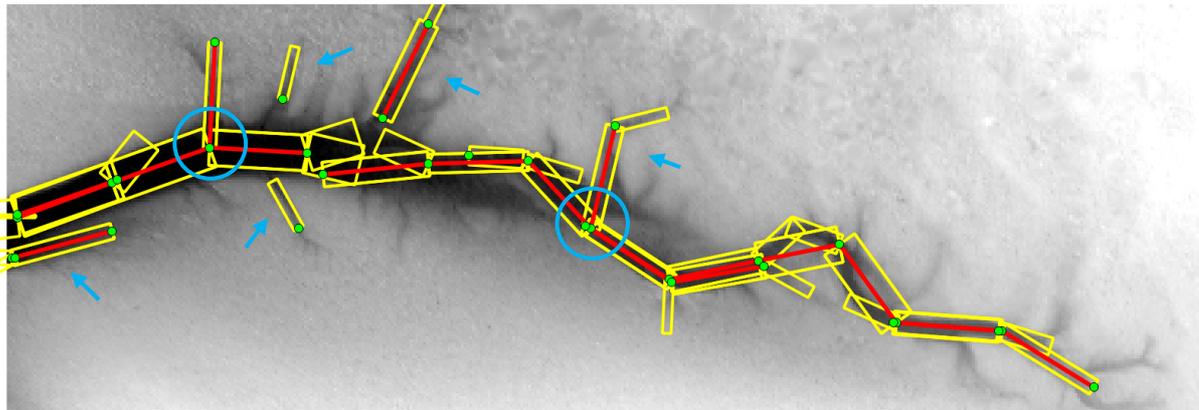


Abbildung 4.22: Ergebnis für eine identische Parameterwahl wie bei Schmidt et al. (2015), wobei die Parameter w , k und f jeweils zu Null gesetzt werden

Die Objektkonfiguration aus Abbildung 4.22 lässt sich durch eine andere Wahl der Parameter sicherlich noch optimieren, was im Folgenden jedoch nicht weiter verfolgt wird. Vielmehr wird an Untersuchungen aus Kapitel 4.4.1 angeknüpft, inwiefern die zusätzliche Verwendung der 3D-Komponente die Daten- und Priorenergie stützt.

Abbildung 4.23 zeigt das Ergebnis, falls die vorherigen Parametereinstellungen um das Fließmodell erweitert werden. Der zugehörige Parameter wird auf $f = 100$ gesetzt. Insgesamt ähneln sich die beiden Objektkonfiguration, allerdings lassen sich auch Unterschiede erkennen. So werden mehrere Dreierkreuzungen gefunden (hellblaue Kreise) und einige der vorher freien Segmente gehören jetzt zum entsprechenden Teilgraphen dazu.

Daneben kann der Verlauf der Kante über den Wattboden abgefangen werden (hellblauer Pfeil), wenn auch hierdurch die in Abbildung 4.22 noch vorhandene Konnektivität der Kanten in diesem Bereich zerstört wird.

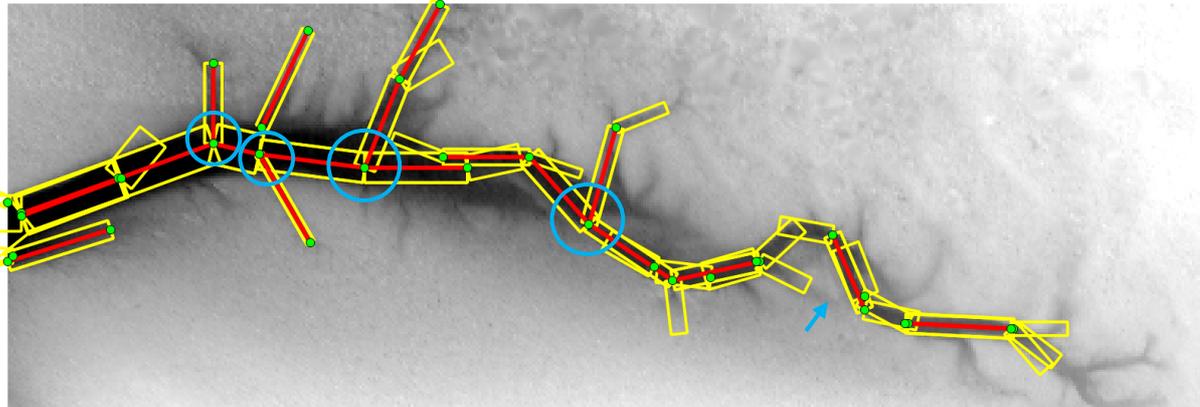


Abbildung 4.23: Ergebnis für eine identische Parameterwahl wie bei Schmidt et al. (2015), wobei die Parameter w und k jeweils zu Null gesetzt werden

Zusammenfassung der Analysen auf Basis der Wattdaten

Es konnte gezeigt werden, dass die Bedingungen des Fließmodells für echte Daten zu hart formuliert sind. Aus diesem Grund wurde eine Anpassung an die Daten vorgenommen. Hierdurch ließ sich einer Aufspaltung des Graphen (Abb. 4.19) entgegenwirken und es konnten ähnliche Ergebnisse wie bei Schmidt et al. (2015) erzeugt werden (Abb. 4.20). An einigen Stellen, wie zum Beispiel im Bereich der Kreise in Abbildung 4.21b, ist eine Optimierung der Objektkonfiguration festzustellen. Ebenfalls haben Untersuchungen bestätigt, dass die 3D-Komponente helfen kann, falls beispielsweise der Datenterm gering gewichtet worden ist (Abb. 4.21).

Ferner ließ sich feststellen, dass das Stützen des Datenterms durch Vorwissen von Bedeutung ist. Insbesondere prägen sich bei echten Daten die Gradienten nicht derart aus, wie es in den synthetischen Daten der Fall ist. Durch die Terme der Priorernergie ist es möglich, einen weitestgehend zusammenhängenden Graphen zu erzeugen, welcher die wesentlichen Bereiche des Priels erfasst (Abb. 4.20). Wird neben dem Datenterm nur das Modell der sich überlappenden Flächen verwendet, erscheint das Ergebnis visuell schlechter zu sein, da es zu vielen Teilgraphen und freien Segmenten kommt (Abb. 4.22). Von Vorteil sind beispielsweise gefundene Dreierkreuzungen und auch Ausläufer des Hauptarms. Die zusätzliche Berücksichtigung des abgeschwächten Fließmodells erlaubt es, dass ungewollte Konfigurationen gegebenenfalls vermieden werden (hellblauer Pfeil). Jedoch kann dies auch

zu zusätzlichen Teilgraphen führen, was der Vergleich der Abbildungen 4.22 und 4.23 im selbigen Bereich zeigt.

Laufzeitanalysen bestätigen die Feststellungen bei den Untersuchungen der synthetischen Daten aus Kapitel 4.4.1. Werden Terme der Priorenergie weggelassen, erhöht sich die Laufzeit. So benötigt der Algorithmus mit den zu Abbildung 4.23 zugehörigen Parametereinstellungen rund ein Fünftel mehr Rechenzeit, als bei der zusätzlichen Verwendung der Terme $U_W(X_t)$ und $U_K(X_t)$ (Abb. 4.20). Wird darüber hinaus noch das Fließmodell $U_F(X_t)$ nicht berücksichtigt (Abb. 4.22), erhöht sich die Laufzeit abermals um fünf Prozentpunkte. Für die synthetischen Daten hat der Laufzeitunterschied lediglich bei ungefähr der Hälfte gelegen. Dies spricht daher umso mehr für eine Berücksichtigung aller Komponenten der Priorenergie. Von Nachteil sind die zusätzlich zu definierenden Parameter, wobei für die Implementierung der 3D-Komponente überdies noch weiteres Vorwissen bezüglich der Stärke der Abschwächung benötigt wird. In Anlehnung an die zusammenfassenden Ausführungen zu den synthetischen Daten gilt es – auch im Hinblick auf die gewünschte Objektkonfiguration – abzuwägen, ob Terme der Priorenergie bei den Berechnungen mit einbezogen werden sollen.

4.4.3 Daten aus Vorarlberg in Österreich

Die nachfolgenden Untersuchungen basieren auf den Daten aus dem österreichischen Bundesland Vorarlberg (Abb. 4.1d). Grundlage der Untersuchungen bilden die bei den Analysen der Wattdaten gewählten Parametereinstellungen. Unter Verwendung von Vorwissen über die Daten werden einige Parameter variiert, um sie dem vorarlbergischen Datensatz anzupassen. Zu diesen zählen der Schwellwert für die minimale und maximale Breite eines Segmentes, welche auf $b_{min} = 10\text{ m}$ bzw. $b_{max} = 200\text{ m}$ gesetzt werden (Abb. 3.1) Ebenfalls wird der Radius r (Kap. 3.1) aufgrund der vergleichsweise deutlich höheren Abmessungen des Bildes von $2500 \times 2500\text{ m}^2$ auf $r = 125\text{ m}^2$ gesetzt. Ferner werden von der favorisierten Winkelkonstellation abweichende Verhältnisse mit $w = 1$ bestraft. Auf die Wahl des Schwellwertes z (Kap. 3.1) für den maximalen Höhen- bzw. Grauwert, bis zu welchem potentielle Orte für ein Geburtereignis gewählt werden, wird an der jeweilige Stelle gesondert hingewiesen. Schließlich wird in Anlehnung an die Untersuchungen der Daten des Wattenmeeres das Fließmodell abgeschwächt und der Parameter a auf $a = 3$ gesetzt (Kap. 3.3).

Im Folgenden soll der Einfluss des Fließmodells auf die Ergebnisse untersucht werden. Damit die Auswirkungen der 3D-Komponente besser untereinander verglichen werden

können, wird auch hier der Pseudo-Zufallszahlengenerator für alle nachfolgenden Analysen mit demselben Wert initialisiert. Zusätzlich werden die Ergebnisse mit Referenzdaten unterlegt, die den wahren Verlauf der Flüsse (blau) repräsentieren.

In Abbildung 4.24a wird der Parameter z zunächst auf $z = 25$ festgelegt, um sich im Wesentlichen auf den Fluss und weniger auf dessen Abzweigungen zu konzentrieren. Für Abbildung 4.24b wird zusätzlich die 3D-Komponente berücksichtigt. Vom Fließmodell abweichende Objektkonfigurationen werden mit dem Faktor $f = 10$ bestraft. Der Ausschnitt ist aus anschaulichen Gründen auf das relevante Gebiet des Flusses beschränkt.

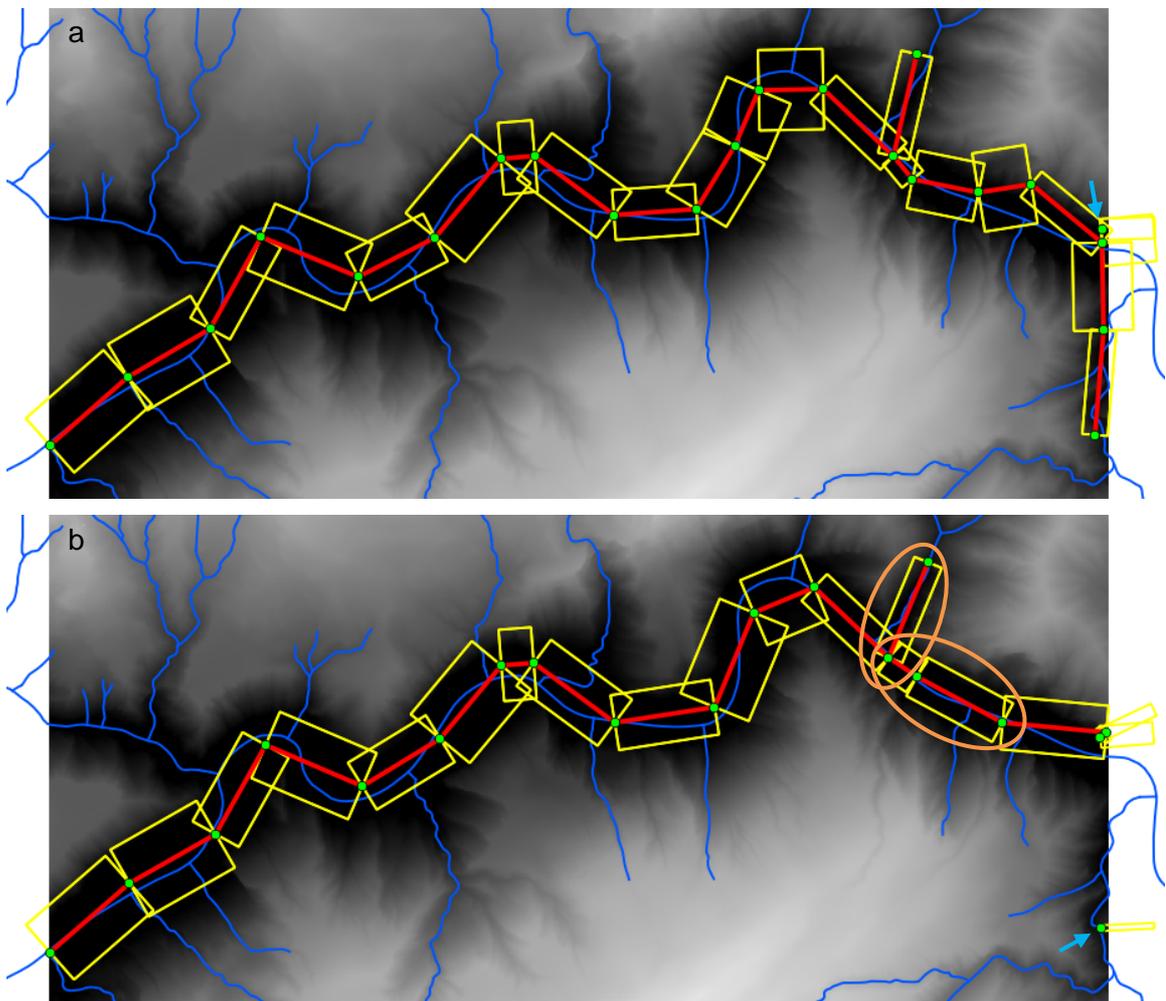


Abbildung 4.24: Ergebnis für $z = 25$
(a): keine Berücksichtigung des Fließmodells
(b): Berücksichtigung des abgeschwächten Fließmodells mit $f = 10$

Es ist zu beobachten, dass sich die Ergebnisse in Abbildung 4.24 ähneln. Insgesamt wird der Hauptarm des Flusses durch einen einzigen zusammenhängenden Graphen repräsentiert. Lediglich ein freies Segment lässt sich jeweils erkennen (hellblauer Pfeil). Im Gegensatz zu

den Untersuchungen der synthetischen Daten und Wattdaten werden Dreierkreuzungen gefunden, obwohl der Term $U_W(X_t)$ berücksichtigt wird. Ein Grund hierfür wird die nicht vorhandene Gewichtung sein ($w = 1$). Im westlichen Bereich ist der Verlauf des Graphen für beide Abbildungen nahezu identisch. Jedoch lassen sich insbesondere südöstlich der Dreierkreuzung Unterschiede feststellen. Unter Berücksichtigung des Fließmodells (Abb. 4.24b) geben die Kanten bzw. Segmente des Graphen an dieser Stelle den wahren Flussverlauf deutlich besser wieder. Dies gilt auch für die an der Dreierkreuzung nordöstlich verlaufende Kante (orangefarbene Kreise). Diese optimierenden Eigenschaften konnten bereits bei den Untersuchungen der Daten des Wattenmeeres und den synthetischen Daten festgestellt werden.

Im Folgenden wird der Parameter z auf 50 erhöht, sodass auch einige weitere Ausläufer des Flusses detektiert werden sollten. Abbildung 4.25a zeigt das Ergebnis ohne Berücksichtigung des Fließmodells $U_F(X_t)$, während es in Abbildung 4.25b zusätzlich mit dem Faktor $f = 10$ multipliziert wird.

Wie angenommen, werden in Abbildung 4.25a im Wesentlichen drei weitere Ausläufer gefunden (hellblaue Pfeile). Ebenso wird das detektierte Flussnetzwerk durch einen einzigen zusammenhängenden Graphen repräsentiert, welcher ebenfalls über Dreierkreuzungen verfügt. Wird das Fließmodell berücksichtigt (Abb. 4.25b), so werden deutlich weniger Bereiche des Flusses detektiert. Dieser Nachteil kann derart begründet werden, dass insbesondere in den (kleineren) Ausläufern des Flusses die Bedingungen des Fließmodells öfters verletzt werden. Hierdurch werden diese Konfigurationen unter Umständen verworfen. Auf der anderen Seite lässt sich wie auch bei den vorherigen Untersuchungen dieser und der anderen Daten der positive Effekt der Fließmodells beobachten. Die Kanten des Graphen geben nun den wahren Verlauf des Flusses im Bereich des orangenen Kreises nahezu perfekt wieder. Ferner zeigt der Vergleich der beiden Ergebnisse einen weiteren Vorteil (weißer Kreis in Abb. 4.25a). Anhand der Referenzdaten lässt sich erkennen, dass die Objektkonfiguration den wahren Flussverlauf nicht korrekt repräsentiert. Die Dreierkreuzung (weißer Pfeil) müsste weiter westlich liegen. Zwar wird an dieser Stelle der Fluss durch Hinzunahme des Fließmodells nicht weiter detektiert, jedoch ließ sich hierdurch eine ungewollte Objektkonfiguration vermeiden. Abschließend ist anzumerken, dass durch eine Erhöhung des Parameters z versucht worden ist, weitere Ausläufer des Flussnetzwerks zu detektieren. Jedoch konnten unter diesen Parametereinstellungen keine brauchbaren Ergebnisse erzeugt werden.

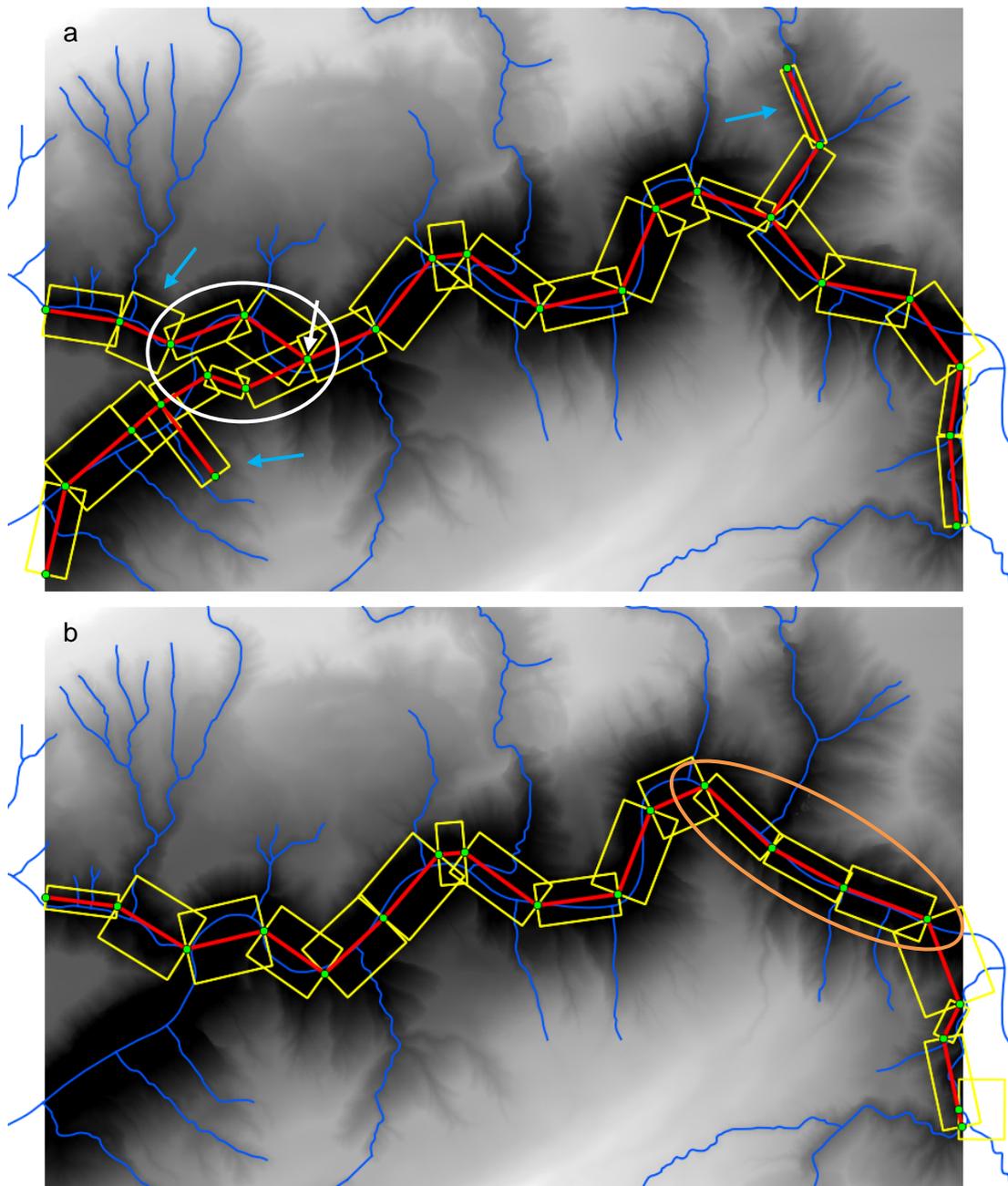


Abbildung 4.25: Ergebnis für $z = 50$
(a): keine Berücksichtigung des Fließmodells
(b): Berücksichtigung des abgeschwächten Fließmodells mit $f = 10$

Zusammenfassend ist festzuhalten, dass das Fließmodell zu einer Optimierung der Objektkonfiguration beitragen kann. Unter Berücksichtigung der 3D-Komponente geben die Kanten bzw. Segmente des Graphen den wahren Flussverlauf in der Regel besser wieder, wobei eine höhere Gewichtung zu weniger detektierten Abschnitten des Netzwerks führen kann. Somit konnte der mögliche Nutzen des Fließmodells an einem weiteren Datensatz gezeigt werden.

5 Zusammenfassung und Ausblick

Innerhalb dieser Arbeit werden verschiedene Datensätze genutzt, um mithilfe von markierten Punktprozessen netzwerkartige Strukturen in den Daten zu detektieren. Für die Analysen der Daten wird ein digitales Geländemodell des deutschen Wattenmeeres und des österreichischen Bundeslandes Vorarlberg, welches aus flugzeuggestützten Laserscannerdaten abgeleitet worden ist, verwendet. Ferner werden synthetische Daten hinzugezogen, um selbst implementierte Methoden auf geeignete Weise evaluieren zu können. Grundlage aller Untersuchungen bildet eine vom Institut für Photogrammetrie und GeoInformation entwickelte und genutzte Software. Im Zusammenhang mit den eigenen Erweiterungen am Programm wird untersucht, inwiefern die Detektion der netzwerkartigen Strukturen optimiert werden kann. Diesen Analysen werden Untersuchungen zu den Einflüssen unterschiedlicher Parametereinstellungen auf das Ergebnis sowie der Verteilung der Parameter vorangestellt. Weiterhin werden Experimente zu Zufallszahlengeneratoren und deren Laufzeitverhalten durchgeführt, da diese die Basis der Software bilden.

Die Untersuchungen zu den Zufallszahlengeneratoren zeigen die starke Abhängigkeit der Ergebnisse von der Art der Initialisierung. Innerhalb dieser Arbeit werden Pseudo-Zufallszahlengeneratoren verwendet, da nicht-deterministische Zufallszahlengeneratoren vergleichsweise hohe Laufzeiten besitzen. Erstere benötigen zur Initialisierung einen Saatpunkt. Erfolgt das Setzen des Saatpunktes mit demselben Wert, führt dies zu identischen Zufallszahlen und somit identischen Parametern für alle erzeugten Ergebnisse. Auf diese Weise lassen sich Resultate besser untereinander vergleichen und Unterschiede bei der Wahl verschiedener Parameter können einfacher erkannt werden. Um eine unterschiedliche Sequenz von Zufallszahlen zu erzeugen, ist die Initialisierung des Pseudo-Zufallszahlengenerators mithilfe eines wahren Zufallszahlengenerators sinnvoll. Das Setzen eines Saatpunktes ist hier nicht notwendig, da zufällige Daten vom eigenen Betriebssystem abgefragt werden.

Am Beispiel der Gleich- und Normalverteilung ist untersucht worden, ob Zufallszahlen entsprechend der vorgegebenen Verteilungsfunktion erzeugt werden. Die Generierung der Zufallszahlen ist dabei mithilfe der *boost*-Bibliothek erfolgt. Es hat sich gezeigt, dass bei

hinreichender Anzahl an Stichproben die Verteilung gut wiedergegeben wird. Zudem erlaubt die Verwendung unterschiedlicher Verteilungsfunktionen für Parameter eine individuelle Anpassung an die Problemstellung. Hierdurch kann Vorwissen über bestimmte Charakteristika des untersuchten Gebietes in die Analysen mit einfließen.

Die innerhalb dieser Arbeit verwendete Software nutzt einen graphenbasierten Ansatz zur automatischen Detektion von Flussnetzwerken aus digitalen Geländemodellen mittels markierter Punktprozesse. Das Netzwerk wird dabei durch einen ungerichteten, azyklischen Graphen repräsentiert. Im Graphen sind Knoten bzw. Objekte über Kanten verbunden, wobei eine Kante einem Segment mit einer bestimmten Breite und Ausrichtung entspricht.

Um eine optimale Objektkonfiguration zu finden, wird der Graph iterativ generiert. In jedem Iterationsschritt wird die aktuelle Objektkonfiguration auf Basis einer globalen Energiefunktion verändert, welche die Übereinstimmung des derzeitigen Zustandes mit einem vordefinierten Modell für das Flussnetzwerk bewertet. Die Energiefunktion wird während des Samplings minimiert und setzt sich im dargestellten Verfahren aus zwei Teilen, der Daten- und der Priorenergie, zusammen. Die Datenenergie vergleicht die Übereinstimmung der Objektkonfiguration mit den Eingangsdaten, indem Beträge von Gradienten berücksichtigt werden. Die Priorenergie des Modells, mithilfe derer bestimmte Konfigurationen auf Basis von Vorwissen befürwortet oder bestraft werden können, setzt sich aus drei Termen zusammen. Dabei werden Segmente bestraft, die mit dem Flussnetzwerk nur auf einer und nicht auf beiden Seiten mit einem Knoten verbunden sind. Ebenfalls erfolgt eine Bestrafung von sich überlappenden Segmenten sowie von Kanten, die einen für das Flusssystem unüblichen Schnittwinkel aufweisen.

Im Rahmen dieser Arbeit ist die Priorenergie um einen zusätzlichen Term erweitert worden. Dieser bewertet die Übereinstimmung der Objektkonfiguration mit einem Fließmodell und besagt, dass in einen Knoten beliebig viele Flüsse hinein-, jedoch nur einer herauslaufen darf. Ferner wird gefordert, dass jede Pixelhöhe entlang der Kante zwischen der Höhe des Anfangsknotens und der des Endknotens liegt. Zusätzlich muss ein Gefälle bzw. Anstieg oder zumindest eine gleichbleibende Höhe über den gesamten Weg vorhanden sein. Sollten die im Fließmodell formulierten Bedingungen nicht erfüllt sein, wird ein Strafterm gesetzt. Mithilfe von Faktoren können die einzelnen Terme der Priorenergie unterschiedlich gewichtet werden.

In den Analysen ist der Einfluss des Fließmodells auf die Ergebnisse untersucht worden. Es hat sich gezeigt, dass die ausschließliche Verwendung des Fließmodells keine brauchbaren Ergebnisse liefert. Dies gilt auch bei Hinzunahme der drei Komponenten der Priorenergie, da der Bezug der Segmente zu den Rändern des Flussnetzwerks noch nicht gegeben ist. Wird darüber hinaus der Datenterm verwendet, können sich die Segmente in Form und Lage dem

Netzwerk einfügen. Werden die bestehenden Modelle um die 3D-Komponente erweitert, kann das Fließmodell in bestimmten Situationen zu einer Optimierung der Objektkonfiguration beitragen. Beispielsweise können ungewollte Kantenverläufe des Graphen über den (Watt-) Boden vermieden werden. Ferner kann die 3D-Komponente ermöglichen, dass die Kanten bzw. Segmente des Graphen den wahren Flussverlauf besser wiedergeben. Insbesondere hilft das Fließmodell auch dann, wenn die Wahl der Parameter für die gegebene Szene noch nicht optimal erfolgt ist. So kann bei geringer Gewichtung des Datenterms eine ungeeignete Konfiguration durch die zusätzliche Verwendung der 3D-Komponente behoben werden.

Weitere Untersuchungen legen dar, dass bestimmte Komponenten der Priorenergie für die Erzeugung von brauchbaren Ergebnissen nicht notwendigerweise benötigt werden. Das Weglassen von Prioritermen führt zu einer reduzierten Anzahl von Parametern, was einem potentiellen Nutzer die Benutzung des Programms erleichtern wird. Auf der anderen Seite zeigen Laufzeitanalysen, dass der Algorithmus durch zusätzliche Komponenten der Priorenergie schneller konvergiert, was für eine Berücksichtigung aller Terme spricht.

Während der Analysen haben sich die starke Abhängigkeit der Ergebnisse von der Wahl der Parameter und auch der gegenseitige Einfluss untereinander gezeigt. Das Auffinden einer geeigneten Parameterwahl für die Szene ist mitunter mit erheblichem Aufwand verbunden.

Die Untersuchungen haben gezeigt, dass sich viele Faktoren auf die erzeugte Objektkonfiguration auswirken. Im Hinblick auf zukünftige Untersuchungen ist die Verwendung weiterer Datensätze für umfassendere Tests sinnvoll, da hierdurch das Zusammenspiel von Parametern besser verstanden werden kann. Denkbar ist auch eine Beurteilung der erzeugten Konfiguration mithilfe von Zahlenwerten. Als Qualitätsmaße können die Größen Vollständigkeit und Korrektheit herangezogen werden (Heipke et al., 1997). Dabei kann die Bewertung auf Pixel- oder Objektebene in Bezug zu Referenzdaten erfolgen. Erstere Variante berücksichtigt alle Pixel innerhalb der Segmente des Graphen, während bei der objektbasierten Evaluierung die Anzahl der detektierten Objekte (z. B. die Ausläufer des Flusses) ausgewertet wird.

Um die Detektion der netzwerkartigen Strukturen verbessern zu können, sind beispielsweise Optimierungen hinsichtlich der Konnektivität des Graphen sinnvoll. In diesem Zusammenhang wäre ein Kriterium von Vorteil, welches freie Enden der Teilgraphen bei Bedarf auf geeignete Weise miteinander verbindet. Realisiert werden könnte dies über einen weiteren Änderungskernel. Hierbei würde zunächst einen zufälliger Knoten ausgewählt werden, welcher nur auf einer Seite über eine ausgehende Kante verfügt. Anschließend könnte innerhalb einer lokalen Nachbarschaft geschaut werden, ob dieser Knoten

gegebenenfalls mit einem anderen verschmolzen werden kann. Eine Alternative wäre es, in der Nähe des ausgewählten Knotens einen neuen Knoten vorzuschlagen. In Abhängigkeit der Energievariation wird der neue Zustand entweder angenommen oder verworfen werden. Ferner könnten Dreierkreuzungen eher entstehen, falls typische Konstellationen der Zwischenwinkel aus Trainingsdaten gelernt werden. Diese sowie auch die Parameter von Kreuzungspunkten werden zurzeit auf Basis von Vorwissen gesetzt. Darüber hinaus könnte das Fließmodell insofern verändert werden, dass Kanten entsprechend ihrer Anzahl an Bestrafungen gewichtet werden. Kanten könnten unter Umständen öfters bestraft werden, wenn jedes Mal der gesamte Weg entlang der Kante abgelaufen wird, auch wenn bereits eine Bedingung verletzt worden ist. Wie sich dies auf das Auffinden einer optimalen Objektkonfiguration auswirkt, müsste noch untersucht werden. In jedem Fall wäre die Laufzeit des Programms erhöht. Abschließend könnten zur Modellierung der Objekte Trapeze gegebenenfalls geeigneter sein, da insbesondere Prielnetzwerke zum Ende hin schmaler werden.

Um die Detektion von Ausläufern des Flussnetzwerks zu verbessern, könnte anstelle des Schwellwertes für den maximalen Höhen- bzw. Grauwert, bis zu welchem potentielle Orte für ein Geburtsereignis gewählt werden, ein anderer Vorverarbeitungsschritt genutzt werden. Denkbar wäre die Verwendung von Gradientenbeträgen, da die Höhe des Betrages einen Hinweis auf das zu detektierende Flussnetzwerk gibt. Ist der Betrag hoch, so wird sich an dieser Stelle mit großer Wahrscheinlichkeit eine Gewässerkante befinden. Somit ist es sinnvoll, zwischen zwei hohen, parallel verlaufenden Gradientenbeträgen neue Knoten zu erzeugen. Um die Detektion von Ausläufern weiterhin zu fördern, würde ein zusätzliches Kriterium in der Form helfen, dass der Algorithmus mit größerer Wahrscheinlichkeit einen neuen Knoten in der lokalen Nachbarschaft eines bereits erzeugten Knotens schätzt.

Insgesamt hat sich gezeigt, dass die markierten Punktprozesse ein geeignetes Verfahren zur Detektion von Flussnetzwerken in luftgestützten Laserscannerdaten darstellen. Die Methode ist sehr flexibel und lässt sich in vielerlei Hinsicht erweitern. So kann Vorwissen zum Aufstellen von zusätzlichen Modellen genutzt werden, welche dabei unterstützen, die relevanten Gebiete der Szene zu extrahieren.

Literaturverzeichnis

- Adler, S. L. (1981). Over-relaxation method for the Monte Carlo evaluation of the partition function for multiquadratic actions. *Physical Review D*, 23(12), 2901.
- Agarwal, P., Arge, L., & Danner, A. (2006). From Point Cloud to Grid DEM: A Scalable Approach. In A. Riedl, W. Kainz, & G. Elmes (Eds.), *Progress in Spatial Data Handling*, 771–788. Springer Berlin Heidelberg.
- Andrieu, C., Freitas, N. de, Doucet, A., & Jordan, M. I. (2003). An Introduction to MCMC for Machine Learning. *Machine learning*, 50(1-2), 5–43.
- Baddeley, A. J., & van Lieshout, M. N. (1993). Stochastic geometry models in high-level vision. *Journal of Applied Statistics*, 20(5-6), 231–256.
- Beichelt, F. E., & Montgomery, D. C. (Eds.) (2003). *Lehrbuch Mathematik. Teubner-Taschenbuch der Stochastik: Wahrscheinlichkeitstheorie, stochastische Prozesse, mathematische Statistik* (1. Aufl.). Stuttgart: Teubner.
- Beraldin, J.-A., Blais, F., & Lohr, U. (2010). Laser Scanning Technology. In G. Vosselman & H. G. Maas (Eds.), *Airborne and Terrestrial Laser Scanning*. Dunbeath: Whittles Publishing.
- Bewersdorff, J. (2012). *Glück, Logik und Bluff: Mathematik im Spiel - Methoden, Ergebnisse und Grenzen*. Wiesbaden: Vieweg+Teubner Verlag.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning* (1. Aufl.). *Information Science and Statistics*. New York, USA: Springer.
- Briese, C. (2010). Extraction of Digital Terrain Models. In G. Vosselman & H. G. Maas (Eds.), *Airborne and Terrestrial Laser Scanning*. Dunbeath: Whittles Publishing.
- Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1), 41–51.
- Chai, D., Forstner, W., & Lafarge, F. (2013). Recovering Line-Networks in Images by Junction-Point Processes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1894–1901.
- Chiu, S. N., Stoyan, D., Kendall, W. S., & Mecke, J. (2013). *Stochastic Geometry and Its Applications* (3. Aufl.). *Wiley Series in Probability and Statistics*. New York: Wiley.
- Cressie, N. A. C. (1993). *Statistics for Spatial Data* (2. Aufl.). *Wiley Series in Probability and Statistics*. New York: Wiley.
- Croft, H. T., Falconer, K. J., & Guy, R. K. (1991). *Unsolved Problems in Geometry* (Vol. 2). New York, NY: Springer.
- Daley, D. J., & Vere-Jones, D. (2003). *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods* (2. Aufl.). *Probability and its Applications*. New York, NY: Springer.
- Descamps, S., Descombes, X., Bechet, A., & Zerubia, J. (2008). Automatic Flamingo detection using a multiple birth and death process. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1113–1116.

- Dreesman, J. (1998). *Zur statistischen Inferenz in Markov-Feldern: Markov-Chain-Monte-Carlo-Verfahren und Modelle mit räumlich variierenden Koeffizienten*. München: Herbert Utz Verlag.
- Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*, 195(2), 216–222.
- Eastlake, D., Schiller, J., & Crocker, S. (2005). *Randomness Requirements for Security: RFC-editor 4086*.
- Geman, S., & Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6), 721–741.
- Georgii, H. O. (2004). *Stochastik: Einführung in die Wahrscheinlichkeitstheorie und Statistik* (2. Aufl.). *De Gruyter Lehrbuch*. Berlin: De Gruyter.
- Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood. In E. M. Keramidas (Ed.), *Computing Science and Statistics: The 23rd symposium on the interface*, 156–163. Fairfax: Interface Foundation of North America.
- Geyer, C. J. (1998). Markov chain Monte Carlo lecture notes. Unveröffentlicht.
- Gilks, W. R., Richardson, S., & Spiegelhalter, D. (1996). *Markov Chain Monte Carlo in Practice* (Neuaufgabe). *Chapman & Hall/CRC Interdisciplinary Statistics*: Taylor & Francis.
- Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4), 711–732.
- Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1), 97–109.
- Heipke, C., Mayer, H., Wiedemann, C., & Jamet, O. (1997). Evaluation of automatic road extraction. *International Archives of Photogrammetry and Remote Sensing*, 32(3-4W2), 151–160.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671–680.
- Kraus, K. (2004). *Photogrammetrie, Band 1: Geometrische Informationen aus Photographien und Laserscanneraufnahmen* (7. vollst. bearb. und erw. Aufl.). *De Gruyter Lehrbuch*. Berlin: De Gruyter.
- Kuo, H. H. (2006). *Introduction to Stochastic Integration*. New York: Springer.
- Lacoste, C., Descombes, X., & Zerubia, J. (2005). Point processes for unsupervised line network extraction in remote sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), 1568–1579.
- Lafarge, F., Descombes, X., Zerubia, J., & Pierrot-Deseilligny, M. (2006). A structural approach for 3D building reconstruction. *Research Report*.
- Lafarge, F., Descombes, X., Zerubia, J., & Pierrot-Deseilligny, M. (2008). Building reconstruction from a single DEM. *Computer Vision and Pattern Recognition (CVPR)*, 1-8. Anchorage, United States.
- Lafarge, F., Gimelfarb, G., & Descombes, X. (2010). Geometric feature extraction by a multimarked point process. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 1597–1609.
- Lesage, D., Angelini, E. D., Bloch, I., & Funka-Lea, G. (2009). A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. *Medical Image Analysis*, 13(6), 819–845.
- MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*. United Kingdom: Cambridge University Press.

- Mallet, C., Lafarge, F., Roux, M., Soergel, U., Bretar, F., & Heipke, C. (2010). A marked point process for modeling lidar waveforms. *IEEE Transactions on Image Processing*, 19(12), 3204–3221.
- Marin, D., Aquino, A., Gegundez-Arias, M. E., & Bravo, J. M. (2011). A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features. *IEEE Transactions on Medical Imaging*, 30(1), 146–158.
- Matsumoto, M., & Nishimura, T. (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1), 3–30.
- Maurer, J. (2015a). *Chapter 24. Boost.Random*.
http://www.boost.org/doc/libs/1_58_0/doc/html/boost_random.html#boost_random.introduction.
 Besucht: Juli 2015
- Maurer, J. (2015b). *Chapter 24. Boost.Random: Class random_device*.
http://www.boost.org/doc/libs/1_58_0/doc/html/boost/random/random_device.html.
 Besucht: Juli 2015
- Mayer, H., Laptev, I., & Baumgartner, A. (1998). Multi-scale and snakes for automatic road extraction. In G. Goos, J. Hartmanis, J. van Leeuwen, H. Burkhardt, & B. Neumann (Eds.), *Lecture notes in computer science. Computer Vision — ECCV'98*, 720–733. Berlin, Heidelberg: Springer Berlin Heidelberg.
- McKeown, D. M., & Denlinger, J. L. (1988). Cooperative methods for road tracking in aerial imagery. In *Proceedings CVPR '88: The Computer Society Conference on Computer Vision and Pattern Recognition*, 662–672.
- Mechalas, J. (2014). *Intel Digital Random Number Generator (DRNG) Software Implementation Guide*.
<https://software.intel.com/en-us/articles/intel-digital-random-number-generator-drng-software-implementation-guide/>. Besucht: Juli 2015
- Metropolis, N. (1987). The beginning of the Monte Carlo method. *Los Alamos Science*, 15(584), 125–130.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6), 1087–1092.
- Metropolis, N., & Ulam, S. (1949). The monte carlo method. *Journal of the American Statistical Association*, 44(247), 335–341.
- Meyn, S. P., & Tweedie, R. L. (1993). *Markov Chains and Stochastic Stability*. London: Springer.
- Milbradt, P., Dorow, C., & Asche, C. (2009): *Identifikation morphologischer Tendenzen und Geschwindigkeiten im Küstenbereich (ImTG)*. Abschlussbericht, Institut für Bauinformatik, Leibniz Universität Hannover.
- Mitas, L., & Mitasova, H. (1999). Spatial interpolation. *Geographical information systems: principles, techniques, management and applications*, 1, 481–492.
- Mnih, V., & Hinton, G. (2010). Learning to Detect Roads in High-Resolution Aerial Images. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Lecture notes in computer science. Computer Vision – ECCV 2010*, 210–223. Springer Berlin Heidelberg.
- Neal, R. M. (1993). *Probabilistic Inference using Markov Chain Monte Carlo Methods*. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, Canada.
- Neal, R. M. (2003). Slice Sampling. *The Annals of Statistics*, 31(3), 705–741.

- Ortner, M., Descombes, X., & Zerubia, J. (2007). Building Outline Extraction from Digital Elevation Models Using Marked Point Processes. *International Journal of Computer Vision*, 72(2), 107–132.
- Ortner, M., Descombes, X., & Zerubia, J. (2008). A marked point process of rectangles and segments for automatic analysis of digital elevation models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1), 105–119.
- Perrin, G., Descombes, X., & Zerubia, J. (2005). Adaptive Simulated Annealing for Energy Minimization Problem in a Marked Point Process Application. In A. Rangarajan, B. Vemuri, & A. Yuille (Eds.), *Lecture notes in computer science. Energy Minimization Methods in Computer Vision and Pattern Recognition*, 3–17. Springer Berlin Heidelberg.
- Perrin, G., Descombes, X., & Zerubia, J. (2006). 2D and 3D Vegetation Resource Parameters Assessment using Marked Point Processes. In *18th International Conference on Pattern Recognition (ICPR'06)*, 1–4.
- Pfeifer, N. (2003). Oberflächenmodelle aus Laserdaten. *Österreichische Zeitschrift für Vermessung und Geoinformation (VGI)*, 91. Jahrgang(4), 243–252.
- Pievatolo, A., & Green, P. J. (1998). Boundary detection through dynamic polygons. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(3), 609–626.
- Poullis, C., & You, S. (2010). Delineation and geometric modeling of road networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(2), 165–181.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes: The art of scientific computing* (3. Aufl.). Cambridge: Cambridge Univ. Press.
- Propp, J. G., & Wilson, D. B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures & Algorithms*, 9(1-2), 223–252.
- Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning* (3. Aufl.). *Adaptive Computation and Machine Learning*. Cambridge, Massachusetts: MIT Press.
- Resnick, S. I. (2007). *Heavy-Tail Phenomena: Probabilistic and Statistical Modeling*. *Springer Series in Operations Research and Financial Engineering*. New York: Springer.
- Rochery, M., Jermyn, I., & Zerubia, J. (2006). Higher Order Active Contours. *International Journal of Computer Vision*, 69(1), 27–42.
- Ross, S. M. (1996). *Stochastic Processes* (2. Aufl.). *Wiley Series in Probability and Statistics*. New York: Wiley.
- Ross, S. M. (1997). *Introduction to Probability Models* (6. Aufl.). San Diego: Academic Press.
- Rue, H., & Hurn, M. A. (1999). Bayesian object identification. *Biometrika*, 86(3), 649–660.
- Salamon, P., Sibani, P., & Frost, R. (2002). *Facts, Conjectures, and Improvements for Simulated Annealing*. *Society for Industrial and Applied Mathematics*. Philadelphia, USA.
- Schmidt, A., Rottensteiner, F., Soergel, U., & Heipke, C. (2014). Extraction of fluvial networks in lidar data using marked point processes. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-3, 297–304.
- Schmidt, A., Rottensteiner, F., Soergel, U., & Heipke, C. (2015). A graph based model for the detection of tidal channels using marked point processes. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-3/W3, 115–121.
- Sithole, G. (2005). *Segmentation and classification of Airborne Laser Scanner Data*. Dissertation, TU Delft, Publications on Geodesy of the Netherlands Commission of Geodesy: Vol. 59.

- Sithole, G., & Vosselman, G. (2004). Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. *Advanced Techniques for Analysis of Geospatial Data*, 59(1–2), 85–101.
- Smith, K. (2011). *Reversible-jump markov chain monte carlo multi-object tracking tutorial*. Technical Report IDIAP-COM-06-07, IDIAP Research Institute. <http://www.kev-smith.com/tutorial/rjmcmmc.php>. Besucht: Juni 2015.
- Sörgel, U. (2013). *Flugzeuglaserscanning*. Vorlesungsskript zur Veranstaltung „Fernerkundung“ des 6. Bachelorsemesters Geodäsie und Geoinformatik an der Leibniz Universität Hannover.
- Sun, K., Sang, N., & Zhang, T. (2007). Marked Point Process for Vascular Tree Extraction on Angiogram. In A. Yuille, S.-C. Zhu, D. Cremers, & Y. Wang (Eds.), *Lecture notes in computer science. Energy Minimization Methods in Computer Vision and Pattern Recognition*, 467–478. Springer Berlin Heidelberg.
- Tournaire, O., Brédif, M., Boldo, D., & Durupt, M. (2010). An efficient stochastic approach for building footprint extraction from digital elevation models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(4), 317–327.
- Tournaire, O., Paparoditis, N., & Lafarge, F. (2007). Rectangular road marking detection with marked point processes. In *Proceedings of the Photogrammetric Image Analysis*, 149–154.
- Utasi, A., & Benedek, C. (2011). A 3-D marked point process model for multi-view people detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3385–3392.
- van Laarhoven, P. J., & Aarts, E. H. (1987). *Simulated Annealing: Theory and Applications* (Neuaufgabe). *Ellis Horwood Series in Mathematics and Its Applications*. Niederlande: Springer.
- van Lieshout, M. N. (2000). Markov point processes and their applications in high-level imaging. In *Imperial College Press*. London.
- Varanelli, J. M. (1996). *On the Acceleration of Simulated Annealing*. Dissertation, University of Virginia, Charlottesville, USA.
- Verdié, Y., & Lafarge, F. (2012). Efficient Monte Carlo Sampler for Detecting Parametric Objects in Large Scenes. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, & C. Schmid (Eds.), *Lecture notes in computer science. Computer Vision – ECCV 2012*, 539–552. Springer Berlin Heidelberg.
- Wagner, W., Ullrich, A., & Briese, C. (2003). Der Laserstrahl und seine Interaktion mit der Erdoberfläche. *Österreichische Zeitschrift für Vermessung und Geoinformation (VGI)*, 91. Jahrgang(4), 223–235.
- Winkler, G. (2003). *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction* (Neuaufgabe). *Stochastic Modelling and Applied Probability*: Springer Berlin Heidelberg.
- Würländer, R., Rieger, W., Drexel, P., & Briese, C. (2005). Landesweite Datenerhebung mit ALS: Technologische Herausforderungen und vielseitige GIS-Anwendungen. In J. Strobl, T. Blaschke, & G. Griesebner (Eds.), *Angewandte Geoinformatik 2005 - Beiträge zum 17. AGIT-Symposium Salzburg*, 800–805. Wichmann Verlag.
- Würländer, R., & Wenger-Oehn, K. (2007). Die verfeinerte Georeferenzierung von ALS-Daten – Methodik und praktische Erfahrung. In J. Strobl, T. Blaschke, & G. Griesebner (Eds.), *Angewandte Geoinformatik 2007 - Beiträge zum 19. AGIT-Symposium Salzburg*, 885–890. Wichmann Verlag.