

Aktualisierung von Geodaten in einer Multiple Representation Database

JAN-HENRIK HAUNERT¹

Zusammenfassung: In zunehmendem Maße werden topographische Datensätze unterschiedlicher Maßstäbe in einer gemeinsamen Datenbank gehalten. Die Fortführung der Daten innerhalb einer solchen Datenbank ist ein aktuelles Problem der Geoinformatik. Im Allgemeinen wird, ausgehend von einem Quelldatensatz, eine Propagierung von Information in einen Zieldatensatz geringerer Auflösung mittels Generalisierung angestrebt.

In der Kartographie wird die Landschaft üblicherweise in Flächen unterschiedlicher Landnutzungs- und Vegetationsklassen gegliedert. Die Aggregation dieser Flächen ist eine wichtige Aufgabe der Modellgeneralisierung. In diesem Beitrag wird ein iteratives Aggregationsverfahren, welches lokale Beziehungen zwischen Objekten berücksichtigt, einem globalen Optimierungsverfahren gegenübergestellt. Die Möglichkeiten zum Einsatz der Verfahren für die Fortführung werden diskutiert.

1 Multiple Representation Database (MRDB)

Datenbanken, welche es ermöglichen, verschiedene Repräsentationen der Realität gemeinsam zu führen, werden im Allgemeinen mit dem Begriff „Multiple Representation Database“ (MRDB) bezeichnet. In dem besonders häufigen Fall der Verwaltung von Geodaten unterschiedlicher Maßstäbe wird auch oft der Begriff „Multiple Resolution Database“ verwendet. Die Datenbestände werden in einer MRDB durch Links zwischen korrespondierenden Objekten der unterschiedlichen Maßstabsebenen verknüpft.

Eine MRDB lässt sich auf zwei verschiedenen Wegen aufbauen. Zum einen ist es möglich, mehrere bestehende Datensätze miteinander zu verlinken. Hierbei liegt die wesentliche Schwierigkeit in der Entwicklung geeigneter Zuordnungsverfahren. Zum anderen kann eine MRDB durch Ableitung der Folgemaßstäbe aus einem Ausgangsdatensatz mittels Generalisierung erzeugt werden. Die Herausforderung liegt dabei in der Entwicklung geeigneter Generalisierungsverfahren, wohingegen sich die Zuordnungen direkt aus der Generalisierung ergeben.

Im ersten Fall existieren zwischen den verschiedenen Datensätzen oft Differenzen, die keiner bestimmaren Regel folgen. Diese können beispielsweise durch eine unterschiedliche Interpretation der Realität bei der Erfassung der verschiedenen Datensätze entstanden sein oder sind von der Natur einer rein geometrischen Ungenauigkeit. Bei der Fortführung ergibt sich hierbei das Problem der geometrischen Datenhomogenisierung. Neue Information, die aus einem größeren Maßstab übernommen werden soll, muss zunächst in den existierenden Datenbestand topologisch korrekt eingepasst werden. Auf dieses Problem soll an dieser Stelle allerdings nicht vertieft eingegangen werden. Als Kernproblem der Fortführungsaufgabe wird stattdessen die automatische Generalisierung behandelt.

2 Generalisierung

Generalisierungsverfahren werden im MRDB-Ansatz für zwei verschiedene Aufgaben eingesetzt. Zum einen ist es möglich eine MRDB durch Erzeugung verschiedener Generalisierungsstufen aus einem detaillierten Datensatz aufzubauen. Zum anderen können

¹ Dipl.-Ing. Jan-Henrik Haurert, Institut für Kartographie und Geoinformatik, Universität Hannover, Appelstraße 9a, 30167 Hannover, e-mail: jan.haurert@ikg.uni-hannover.de

Generalisierungsverfahren eingesetzt werden, um Änderungen in kleinere Maßstäbe zu übertragen und die MRDB somit aktuell zu halten. Im zweiten Fall wird gewöhnlich von „inkrementeller Generalisierung“ gesprochen. Da hierbei von einem existierendem Datenbestand im Zielmaßstab ausgegangen werden kann, sind nur solche Objekte erneut zu generalisieren, die auch von der Aktualisierung betroffen sind.

In diesem Abschnitt werden zunächst die entwickelten Generalisierungsverfahren in ihrer grundlegenden Form dargestellt, bevor in Abschnitt 3 auf Besonderheiten für die Anwendung auf den Aktualisierungsfall eingegangen wird.

2.1 Aggregation

Topographische Datenbanken beinhalten in der Regel Information über die Landbedeckung durch verschiedene Nutzungsarten oder Vegetationsformen. Üblicherweise wird hierfür eine Tessellation der Ebene bzw. ein Polygonmosaik ohne Überlappungen und Lücken definiert, wobei jedes Polygon einer Landbedeckungsklasse zugewiesen ist.

Die Generalisierung einer derartigen Landbedeckungskarte umfasst in der Regel die Generalisierung von Klassen sowie die Zusammenfassung von Objekten. Ein Beispiel für die Generalisierung von Klassen ist, dass Laubwald und Nadelwald im kleineren Maßstab durch die Klasse Wald ersetzt werden. Die Zusammenfassung von Objekten bzw. die Aggregation ist erforderlich, um die geforderten Mindestgrößen des Zielmaßstabs zu erfüllen ohne Lücken entstehen zu lassen.

Im Folgenden wird davon ausgegangen, dass die Zuordnungen zwischen den Klassen des Ausgangs- und Zielmaßstabes bekannt und eindeutig sind. Diese Voraussetzung ist bei topographischen Datensätzen in der Regel gegeben, da die Datenmodelle für verschiedene Maßstäbe aufeinander aufbauen. In diesem Fall kann das Aggregationsproblem isoliert betrachtet werden. Eine ausführliche Diskussion des Aggregationsproblems wird von VAN SMAALEN (2003) geführt.

Bei der Aggregation sind alle Objekte zu topologisch zusammenhängenden Regionen zusammenzufassen, deren Flächeninhalte mindestens den Betrag des Schwellwertes für den Zielmaßstab erreichen müssen. Um diese Bedingung zu erfüllen, ist es erforderlich die Klassenzugehörigkeit einzelner Objekte zu ändern. Generell wird angestrebt möglichst wenige dieser Klassenänderungen durchzuführen, bzw. Änderungen zu Klassen mit möglichst ähnlicher Semantik vorzunehmen (PODRENEK, 2002). Beispielsweise wird man vorziehen, eine Fläche der Klasse „Forst“ in „Wald“ umzuwidmen, anstatt sie der Klasse „See“ zuzuordnen. Ein weiteres Ziel ist es, möglichst kompakte Flächen zu erschaffen, bzw. komplizierte Geometrien zu vermeiden. Die Kompaktheit kann beispielsweise durch die Länge der Flächenbegrenzung ausgedrückt werden, wobei eine kurze Begrenzung für eine kompakte Geometrie spricht. Prinzipiell muss zwischen der semantischen Ähnlichkeit der Klassen und der geometrischen Kompaktheit ein Kompromiss eingegangen werden. Die Leistungsfähigkeit eines lokalen Algorithmus und eines globalen Optimierungsalgorithmus wurde zunächst allerdings für das Ziel ähnlicher Klassen isoliert betrachtet.

2.1.1 Iterativer Algorithmus (Region Growing)

Häufig werden für die Aggregation iterative Algorithmen eingesetzt, die Abwandlungen des bekannten Region Growing Verfahrens sind. Üblicherweise wird in jeder Iteration die kleinste Fläche gewählt und mit einem Nachbarn vereinigt, bis alle Flächen die geforderte Mindestgröße erreichen (VAN OOSTEROM, 1995).

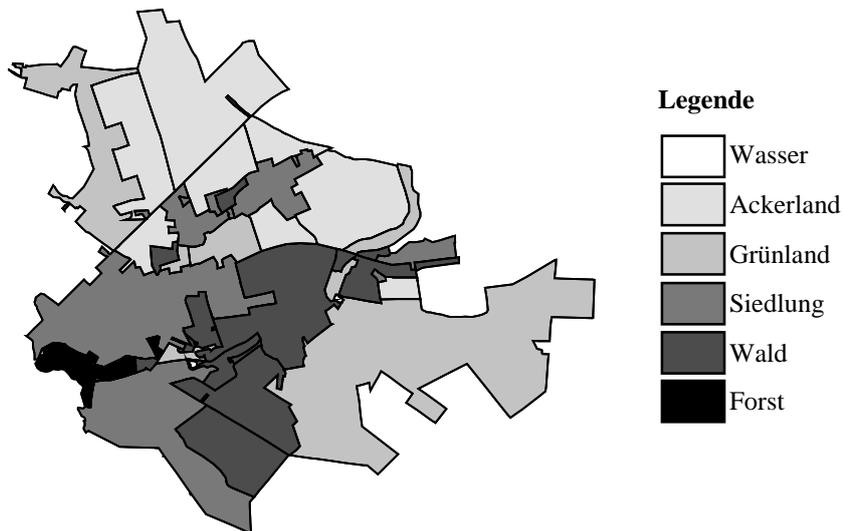


Abb. 1: Ausgangssituation im DLM 50

In unserem Fall wird der Nachbar mit ähnlichster Klasse gewählt. Haben mehrere Nachbarn dieselbe Klasse, so wird davon die größte Fläche gewählt. Auf diese Weise lässt sich eine Lösung in linearer Zeit bestimmen, die allerdings in der Regel nicht die optimale Lösung darstellt. Global betrachtet gibt es unter Umständen Lösungen, die vorzuziehen sind.

Abb. 1 zeigt ein Beispiel, welches aus dem digitalen Landschaftsmodell DLM 50 des Amtlichen Topographisch-Kartographischen Informationssystem (ATKIS) entnommen ist. Der Inhalt des DLM 50 orientiert sich an der topographischen Karte im Maßstab 1:50.000.

Unter der Berücksichtigung semantischer Ähnlichkeiten zwischen den Klassen wurden die Flächen mit dem iterativen Algorithmus aggregiert. Hierbei wurden die Schwellwerte des DLM 250 eingesetzt, welches der topographischen Karte im Maßstab 1:250.000 entspricht.

Die resultierenden Klassen sind in Abb. 2 dargestellt. Flächen, welche die Mindestgröße des Zielmaßstabes nicht erfüllen und keinen direkten Nachbarn derselben Klasse haben, ändern mit großer Wahrscheinlichkeit ihre Klasse. Ein Beispiel hierfür ist die Siedlungsfläche unten links in Abb. 1, die in Wald umgewidmet wird. Diese relativ drastische Änderung ist allerdings nicht erforderlich, da es möglich wäre, durch die Änderung einer vergleichsweise kleinen Waldfläche eine Verbindung zu einer zweiten Siedlungsfläche herzustellen. Dass eine solche Lösung durch einen iterativen Algorithmus gefunden wird, der nur lokal nach der besten Lösung sucht, ist nicht wahrscheinlich. Aus diesem Grund wird im nächsten Abschnitt ein globales Optimierungsverfahren vorgestellt.

2.1.2 Globale Optimierung

In einer zurückliegenden Arbeit (HAUNERT & WOLFF, 2006) wurde ein globales Optimierungsverfahren für die Aggregation entwickelt und getestet. Die Aggregation wird dabei als Färbungsproblem des Nachbarschaftsgraphen formuliert. Hierin wird jede Fläche durch einen Knoten dargestellt. Jede Kante drückt aus, dass die inzidenten Knoten eine gemeinsame Grenze haben. Der Graph ist also planar. Jedem Knoten v ist ein Gewicht $w(v)$ zugeordnet, welches dem Flächeninhalt entspricht. Außerdem hat jeder Knoten v eine Farbe $\gamma(v)$, die der ursprünglichen Klasse entspricht. Gesucht sind nun die neuen Farben $\gamma'(v)$, so dass das gesamte Gewicht einer zusammenhängenden Komponente gleicher Farbe jeweils den Schwellwert des Zielmaßstabes θ erfüllt und

$$\sum_{v \in V} w(v) \cdot d(\gamma(v), \gamma'(v)) \quad \text{minimiert wird.}$$

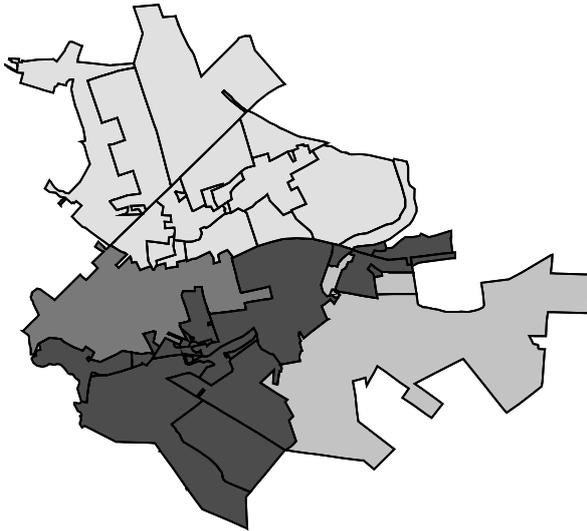


Abb. 2: Ergebnis des iterativen Algorithmus

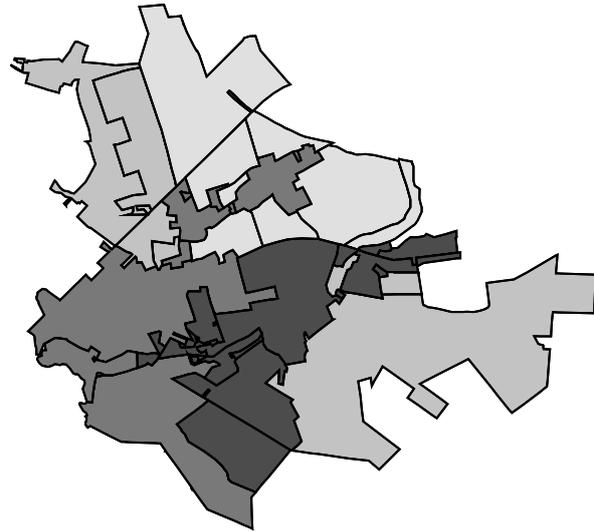


Abb. 3: Ergebnis der globalen Optimierung

Die Distanzfunktion d definiert dabei Kosten, die pro Flächeneinheit für den Wechsel zwischen zwei Farben anfallen. Zusätzlich wird gefordert, dass in jeder Komponente mindestens eine Fläche ihre Farbe beibehält. Durch diese Forderung wird vermieden, dass Farben im Zielmaßstab auftauchen, die nicht im Ausgangsmaßstab vorhanden sind.

Dieses Problem kann als *Mixed Integer Program* (MIP) formuliert werden, welches neben der linearen Zielfunktion Bedingungen in Form linearer Ungleichungen umfasst. Die Variablen hierin dürfen zum großen Teil nur ganzzahlige Werte annehmen.

Ein MIP kann von Standard-Software gelöst werden. Dabei kommen *Branch and bound* Verfahren zum Einsatz, bei denen ausgenutzt wird, dass viele Kombinationen von Werten für Variablen frühzeitig ausgeschlossen werden können. Dieses geschieht vor allem dann, wenn die absehbaren Gesamtkosten unter Annahme dieser Werte in jedem Fall höher sind als die Kosten der optimalen Lösung. Für diese Abschätzung werden untere und obere Schranken für die Zielfunktion benötigt. Untere Schranken können hierbei gewonnen werden, indem Ganzzahligkeitsbedingungen relaxiert werden, da die Kosten nach dieser Veränderung des Problems allenfalls geringer sein können. Hierbei ergeben sich gewöhnliche lineare Programme, die beispielsweise über den Simplex Algorithmus gelöst werden können. Die Grundlagen von MIPs bzw. der gemischt-ganzzahligen Optimierung und Anwendungsmöglichkeiten werden von KALLRATH (2002) dargelegt.

Das MIP für das Aggregationsproblem beruht im Wesentlichen auf der Definition eines Flusses auf den Kanten des Graphen. Ein ähnliches Modell für ein anderes Problem wurde von SHIRABE (2006) vorgestellt. Das MIP liefert die Lösung, die in Abb. 3 dargestellt ist. Nur wenige Flächen mussten verändert werden, um die Voraussetzungen des Zielmaßstabes zu erfüllen. Weitergehende Forschung ist allerdings notwendig, um kompaktere Regionen zu erzielen.

Der Nachweis des Optimums für die Lösung der Instanz mit 50 Knoten benötigt bereits über 20 Stunden. Eine Möglichkeit die Rechenzeit zu verringern besteht darin, die Berechnung bei ausreichend guter Approximation der optimalen Lösung abubrechen. Eine Lösung, die garantiert, dass sie höchstens 10% schlechter ist als das Optimum, ist beispielsweise sicherlich zufriedenstellend. Die Tests zeigten allerdings, dass diese Einschränkung nicht ausreicht um übliche Datensätze in angemessener Zeit zu prozessieren.

Bei vielen Problemen lassen sich zwar effiziente Algorithmen finden, die eine ähnliche Performance-Garantie versprechen. Ein solcher Approximationsalgorithmus ist allerdings für das Aggregationsproblem nicht bekannt. Aus diesem Grund wurden Heuristiken eingeführt,

bei denen unwahrscheinliche Lösungen ausgeschlossen werden. Die Garantie, die optimale Lösung zu finden, geht somit zwar verloren, aber es werden zufriedenstellende Lösungen in angemessener Zeit erzielt. Die optimalen Lösungen, die für kleinere Beispiele ohne den Einsatz von Heuristiken erzielt wurden, dienen hierbei als Referenz, anhand der die getroffenen Annahmen verifizieren werden können. Die Tests ergaben, dass auch Datensätze von 400 Flächen noch in angemessener Zeit nahezu optimal gelöst werden können. Umfangreiche Tests und Laufzeitmessungen wurden durchgeführt (HAUNERT & WOLFF, 2006).

2.2 Kollaps

In einigen Fällen reicht es nicht aus, die Generalisierung einer Landbedeckungskarte als reines Aggregationsproblem anzusehen. Beispielsweise werden Flüsse, die in großen Maßstäben flächenhaft dargestellt sind, in kleineren Maßstäben durch eine linienförmige Geometrie repräsentiert. In diesem Fall ist also ein Geometriotypwechsel zu vollziehen. Gleichzeitig stellt sich allerdings auch hier die Frage, wie die frei gewordenen Flächen auszufüllen ist. Eine Zuweisung zu einem Nachbarn ist hierbei nicht sinnvoll. Stattdessen ist die ursprüngliche Fläche so auf die Nachbarn zu verteilen, dass das erzeugte Linienobjekt die neue Grenze zwischen den Nachbarn bildet.

Generell besteht die Vorgehensweise darin, diese Kollaps-Operationen zuerst durchzuführen und anschließend die Flächen zu aggregieren. Bei der Aggregation ist es möglich Bedingungen einzufügen, die sicherstellen, dass Grenzen, welche durch abgeleitete Linienobjekte gebildet werden, erhalten bleiben. Alternativ können zusätzliche Kosten definiert werden, die anfallen, wenn über diese Grenzen hinaus zusammengefasst wird.

Für den Kollaps wurde ein Operator entwickelt, der auf dem Straight Skeleton beruht (HAUNERT, 2005). Der Operator ist beispielhaft für eine Fläche in Abb. 4 dargestellt. Das Straight Skeleton ist vorteilhaft, da es neben der linienförmigen Repräsentation des Objektes eine Unterteilung der ursprünglichen Fläche in Teilflächen liefert. Hierbei kann jede Teilfläche genau einer Kante des Polygons und somit genau einem Nachbarn zugeordnet werden (Abb. 4b). Das Resultat dieses Kollaps ist in Abb. 4d dargestellt. Außerdem ist es möglich, unterschiedliche Gewichte für Kanten zu definieren und das Skelett somit in eine bevorzugte Richtung zu verlagern (Abb. 4c). Auf diese Weise können auch Nachbarschaftsbeziehungen zu anderen Objekten bewahrt werden.

Das Straight Skeleton ermöglicht einen partiellen Kollaps einer Fläche. Hierbei wird nur für solche Teile der Fläche ein Skelett berechnet, in denen eine Mindestbreite unterschritten wird. Diese Variante ist zum Beispiel für den Geometriotypwechsel von Flüssen relevant, die ab einer bestimmten Breite flächenförmig darzustellen sind.

3 Aktualisierung

Mit den beschriebenen Generalisierungsalgorithmen ist es prinzipiell jederzeit möglich, Flächendatensätze kleinerer Maßstäbe aus einem detaillierten Basisdatensatz abzuleiten. Wird der Basisdatensatz laufend aktuell gehalten, so können also auch aktuelle Daten im kleineren Maßstab erzeugt werden. Die automatische Generalisierung größerer Datenbestände ist allerdings eine zeitintensive Prozedur, die nicht nach jeder kleinen Veränderung des Basisdatensatzes durchgeführt werden sollte. In der Regel ist dieses auch nicht erforderlich, da eine lokale Änderung im Basisdatensatz nur begrenzte Auswirkungen auf den Zieldatensatz hat. Stattdessen reicht es aus, die Teile des Zieldatensatzes neu zu generieren, die von den Änderungen beeinflusst sind. Dieses ist die Zielsetzung der inkrementellen Generalisierung (KILPELÄINEN, T. & SARJAKOSKI, 1995).

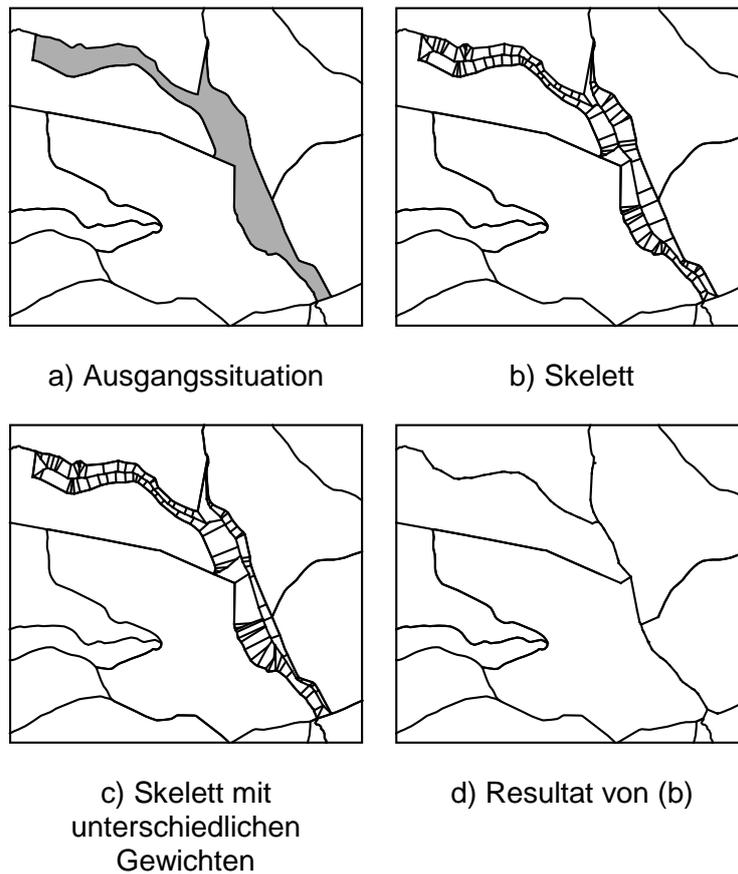


Abb. 4: Kollaps-Operator basierend auf dem Straight Skeleton.

3.1 Aktualisierung durch Auswertung von Links oder Produktionen

Die Propagierung von Änderungen kann durch Links zwischen korrespondierenden Objekten erleichtert werden. Wird ein Objekt im Basisdatensatz geändert, so ist im einfachsten Fall nur das korrespondierende Objekt im Zieldatensatz zu aktualisieren, auf welches effizient über die Links zugegriffen werden kann.

In den seltensten Fällen werden einzelne Objekte bei der Generalisierung allerdings isoliert behandelt. In der Regel fließen Relationen zu anderen Objekten wie Nachbarschaften mit in die Generalisierung ein, so dass die Abhängigkeiten zwischen Objekten unterschiedlicher Datensätze in der Regel sehr komplex sind.

In vielen Fällen reicht es dennoch aus, eine kleine Teilmenge des Datenbestandes zu berücksichtigen, um ein Ergebnis zu generieren, das der Generalisierung des gesamten Datensatzes exakt gleich ist. Eine sehr ausgefeilte Methode für die inkrementelle regelbasierte Generalisierung wurde von SKOGAN & SKAGESTEIN (2005) vorgestellt. Die Herangehensweise hierbei ist, die gesamte Generalisierung in Form von Produktionen bei der globalen Prozessierung aufzuzeichnen. Die Auswertung dieser Produktionen gibt im Fall einer Aktualisierung Auskunft über die bestehenden Abhängigkeiten zwischen Ziel- und Basisdatensatz. Somit kann der Einfluss einer Änderung ermittelt werden. Das Verfahren ist wirksam, wenn der Generalisierungsprozess in Module gegliedert ist, die aus der Auswertung von Regeln und dem darauf folgenden Ausführen überschaubarer Aktionen basiert. Die in Abschnitt 2 vorgestellten Aggregationsverfahren weisen allerdings die Charakteristik auf, den gesamten Datensatz innerhalb relativ abgeschlossener Algorithmen oder global zu

prozessieren. Ähnliche Herangehensweisen werden häufig auch für andere komplexe Generalisierungsprobleme gewählt.

3.2 Inkrementeller Algorithmus für das Region Growing

Der iterative Algorithmus in Abschnitt 2.1.1 weist noch am ehesten eine Ähnlichkeit zu dem regelbasierten System von SKOGAN & SKAGESTEIN (2005) auf, denn auch hier erfolgt die Generalisierung durch schrittweise Analyse und Ausführung von Aktionen. Die Reihenfolge der Prozessierungen ist hierbei allerdings durch die Größe der Flächen fest vorgegeben. Ferner ist bekannt, welche Objekte in einem Iterationsschritt ausgewertet werden, da stets direkte Nachbarn des selektierten Objektes betrachtet werden. Da auch die Kriterien bekannt sind, nach denen der Nachbar für die Zuweisung ausgewählt wird, können die Abhängigkeiten zwischen den Datensätzen rekonstruiert werden, ohne dass eine Aufzeichnung der einzelnen Schritte durchgeführt wurde. Der folgende inkrementelle Algorithmus nutzt diese Tatsache aus. Zunächst sind folgende Definitionen erforderlich:

- Die Region aus Objekten im Zieldatensatz, die alle aktualisierten Objekte im Ausgangsdatsatz überdeckt, heißt „innere Region“.
- Die Region, die alle Nachbarn der inneren Region im Zieldatensatz enthält, heißt „Grenzregion“.
- Alle anderen Objekte bilden den „Außenbereich“.

Wird der iterative Algorithmus auf den aktualisierten Datensatz angewendet, so kann sich nur dann eine Änderung im Außenbereich einstellen, wenn zuvor die Grenzregion von den Änderungen der inneren Region beeinflusst wurde. Dieses kann nur geschehen, wenn ein Objekt der inneren Region einem Objekt der Grenzregion zugewiesen wird oder umgekehrt. In beiden Fällen wird die Trennung von Grenzregion und innerer Region verletzt.

Demnach braucht der iterative Algorithmus zunächst nur auf die Objekte in der inneren Region und der Grenzregion angewendet werden. Tritt eine Vereinigung ein, welche die Trennung beider Regionen verletzt, so können die Regionen nachträglich ausgedehnt werden.

Inkrementeller Algorithmus:

Wende den iterativen Algorithmus auf die Objekte im Ausgangsdatsatz an, die sich innerhalb der inneren Region und der Grenzregion befinden. Wenn die Trennung von innerer Region und Grenzregion durch eine Vereinigung verletzt wird,

- *halte den Algorithmus an,*
- *dehne die innere Region auf die Objekte im Zieldatensatz aus, die an dieser Vereinigung beteiligt waren,*
- *dehne die Grenzregion auf die Nachbarn im Zieldatensatz aus,*
- *hole die Vereinigungen innerhalb der neu hinzugekommenen Flächen nach, welche gemäß der definierten Prozessierungsreihenfolge vor der oben genannten Vereinigung auszuführen sind und*
- *setze die Aggregation der Objekte in innerer Region und Grenzregion fort.*

Das Ergebnis entspricht exakt dem Ergebnis des iterativen Algorithmus bei Anwendung auf den gesamten Datensatz, doch die Anzahl der ausgewerteten Objekte ist sehr begrenzt.

Tests zeigten, dass die Regionen in der Regel nicht weiter als bis zu den Nachbarn zweiter Ordnung der direkt betroffenen Objekte expandiert werden müssen.

4 Ausblick

Die in Abschnitt 3 erläuterten Verfahren der inkrementellen Generalisierung gehen von einer modularisierten oder iterativen Vorgehensweise bei der Generalisierung aus. Nach der Aktualisierung des Basisdatensatzes sind nur wenige Prozessierungsschritte auszuführen, um den Zieldatensatz aufzudatieren, da ein großer Teil der ursprünglichen Generalisierung seine Gültigkeit behält. Die erzeugten Ergebnisse gleichen dabei exakt dem Resultat einer erneuten Prozessierung des gesamten Datensatzes. In vielen Fällen kann allerdings nicht von den getroffenen Voraussetzungen ausgegangen werden. Das globale Optimierungsverfahren aus Abschnitt 2.1.2 ist ein Beispiel, bei dem es schwieriger ist, die Information über eine ursprünglich gefundene Lösung hilfreich für die Prozessierung des aktualisierten Datensatzes einzusetzen. Da *Branch and bound* Verfahren allerdings auf einer Zerlegung des Gesamtproblems in Teilprobleme beruhen und eine optimale Lösung für einen großen Teil des Datensatzes bekannt ist, ist auch hier anzunehmen, dass der Rechenaufwand im Aktualisierungsfall reduziert werden kann.

Ein prinzipielles Problem liegt dagegen vor, wenn der Zieldatensatz ursprünglich nicht durch ein automatisches Generalisierungsverfahren entstanden ist. Wurde er stattdessen unabhängig erfasst, so macht es keinen Sinn zu fordern, dass nach der Übertragung einer Änderung in den kleineren Maßstab dasselbe Ergebnis wie nach einer globalen Generalisierung vorliegen soll. Die unabhängig erfassten Daten weisen schließlich ohnehin Differenzen zum Ergebnis eines automatischen Verfahrens auf. Hier wird man ebenfalls die aktualisierten Objekte und ihren Kontext analysieren und nach einer Generalisierung in den Zielmaßstab einfügen, wobei allerdings nicht klar definiert ist, welchen Ausmaß dieser Kontext hat. Sicher macht es auch hier Sinn, die Menge der betrachteten Objekte räumlich einzugrenzen. Hierfür könnte ähnlich wie beim inkrementellen Algorithmus aus Abschnitt 3.2 eine schrittweise Ausdehnung dieses Einflussgebietes geschehen. Zu diesen offenen Fragen ist weitere Forschung erforderlich.

5 Literaturverzeichnis

- HAUNERT, J.-H., 2005: Geometrietywechsel in einer Multi-Resolution-Datenbank. Mitteilungen des Bundesamtes für Kartographie und Geodäsie, Aga-Tagung 2004.
- HAUNERT, J.-H. & WOLFF, A., 2006: Generalization of land cover maps by mixed integer programming. Eingereicht für: Proc. of 14th International Symposium on Advances in Geographic Information Systems, November 10–11, 2006, Arlington, Virginia, USA.
- KALLRATH, J., 2002: Gemischt-ganzzahlige Optimierung: Modellierung in der Praxis. 1. Auflage, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig.
- KILPELÄINEN, T. & SARJAKOSKI, T., 1995: Incremental generalization for multiple representations of geographic objects. In: Muller, J. C., Lagrange, J. P. & Weibel, R. (Edt.) GIS and Generalization: Methodology and Practise. Taylor & Francis, London, S. 209-218.
- VAN OOSTEROM, P., 1995: The GAP-tree, an approach to ‘on the fly’ map generalization of an area partitioning. In: Muller, J. C., Lagrange, J. P. & Weibel, R. (Editoren) GIS and Generalization: Methodology and Practise. Taylor & Francis, London, S. 120-132.
- PODRENEK, M., 2002: Aufbau des DLM50 aus dem Basis-DLM und Ableitung der DTK50 – Lösungsansatz in Niedersachsen. In: KS, Band 6, Kartographie als Baustein moderner Kommunikation, S.126-130, Bonn.
- SKOGAN, D. & SKAGESTEIN, G., 2005: An Automated Incremental Generalization Method. In: Skogan, D. (Autor) Multiple Resolution and Consistency, PhD Thesis, Department of Informatics, University of Oslo, Norway.

VAN SMAALEN, J.W.N., 2003: Automated Aggregation of Geographic Objects, Dissertation, Wageningen University, Niederlande.