

# **Matching von Strassendaten stark unterschiedlicher Massstäbe und Aufbau einer Multirepräsentationsdatenbank**

**Patrick Lüscher**

Diplomarbeit  
Ausgeführt am Geographischen Institut der Universität Zürich

Leitung und Betreuung:  
Prof. Dr. Robert Weibel  
Dr. Dirk Burghardt

Zürich 2006



# Summary

Multi-representation databases (MRDB) contain several geographic datasets covering the same area. In an MRDB, the objects within the different datasets that model the same real-world phenomenon are linked with each other. MRDBs reduce inconsistencies among datasets, promise cheaper updating, and offer new possibilities for geographic analysis and for the visualisation of spatial datasets. One of the important applications is the generation of intermediate scale levels for smooth zoom operations.

This thesis investigates the integration of road datasets into a multi-representation database. Contrary to existing work, datasets that are of significantly different scales are to be used.

When integrating existing data into an MRDB, the links between corresponding objects have to be created by means of automatic matching techniques. Since existent matching processes have been created for datasets that are of the same or at least similar scales, a goal of this work was the development of such a matching technique for road data of greatly different scales. A general conceptual framework for matching processes has been derived by examination of existing matching approaches. Using that framework, a new matching approach has been designed. The approach uses geometric and semantic information, and two algorithmic components: Generation of closest paths and line tracing. The algorithm works in two stages: First, 1:1 assignments between crossroads are created. Subsequently these assignments are converted into road assignments.

The acquired concepts have been implemented as a prototype in Java using the open source technologies JTS and JUMP. With regard to practical applications of MRDBs we were also interested how the user interface could be designed. An algorithm for the visualisation of links between linear objects has been developed. Various tools enable links to be manipulated.

The matching algorithm has been evaluated with road data at the scales of 1:25'000 and 1:200'000, respectively. The results have been compared to manually matched reference data. 75–90% of the road network could be matched automatically, depending on the test dataset. Because the algorithm allows only 1:1 assignments between crossroads, problems arise mainly when roundabouts or single segments collapse into nodes at 1:200'000.

# Zusammenfassung

Multirepräsentationsdatenbanken (MRDB) enthalten verschiedene geographische Datensätze desselben Gebietes. Objekte in den verschiedenen Datensätzen, die dasselbe Realwelt-Phänomen modellieren, sind in einer MRDB miteinander verknüpft. MRDBs verringern Inkonsistenzen zwischen den Datensätzen, versprechen eine günstigere Nachführung und bieten neue Möglichkeiten zur geographischen Analyse und zur Darstellung von räumlichen Daten. Eine wichtige Anwendung ist die Erzeugung von Zwischenmassstäben für gleichmässige Zoom-Operationen.

Die vorliegende Diplomarbeit untersucht die Integration von Strassendaten in eine Multirepräsentationsdatenbank. Anders als bisher sollen dabei Datensätze verwendet werden, die sich stark in ihrem Massstab unterscheiden.

Bei der Integration bestehender Daten müssen die Verknüpfungen zwischen korrespondierenden Objekten durch automatische Matching-Techniken erzeugt werden. Da die existierenden Matching-Verfahren für Datensätze von gleichen oder zumindest ähnlichen Massstäben geschaffen wurden, war ein Ziel der Arbeit die Entwicklung eines Matching-Algorithmus für Strassendaten stark unterschiedlicher Massstäbe. Durch eine Analyse von bestehenden Matching-Ansätzen konnte ein allgemeiner konzeptioneller Rahmen für Matching-Prozesse hergeleitet werden. Mit den Erkenntnissen daraus wurde ein eigener Matching-Ansatz entworfen. Der Ansatz benutzt geometrische und semantische Informationen und zwei algorithmische Komponenten: Die Bildung von nächsten benachbarten Wegen und die Linienverfolgung. Der Algorithmus arbeitet in zwei Phasen: Zuerst werden 1:1-Zuordnungen zwischen Strassenkreuzungen erzeugt. Diese werden anschliessend in Strassenzuordnungen umgesetzt.

Die erarbeiteten Konzepte wurden als Prototyp in Java implementiert. Dafür wurden die Open Source Technologien JTS und JUMP verwendet. Im Hinblick auf den produktiven Einsatz von MRDBs interessiert, wie die Benutzerschnittstelle gestaltet werden kann. Es wurde ein neuer Algorithmus für die Visualisierung von Verknüpfungen zwischen Linienobjekten entwickelt. Dem Benutzer stehen verschiedene Werkzeuge zur Bearbeitung der Verknüpfungen zur Verfügung.

Der Matching-Algorithmus wurde mit Strassendaten in den Massstäben 1:25'000 respektive 1:200'000 evaluiert und mit manuell verknüpften Referenzdaten verglichen. Abhängig vom Datensatz konnten 75–90% des Strassennetzes automatisch zugeordnet werden. Weil das Verfahren nur 1:1-Beziehungen zwischen Strassenkreuzungen erlaubt, ergeben sich Probleme vorallem bei Strassenkreiseln und einzelnen Segmenten, die in 1:200'000 zu einzelnen Knoten kollabieren.

# Danksagung

Ich möchte mich an dieser Stelle ganz herzlich bei allen bedanken, die mich während meines Studiums begleitet und unterstützt haben. Ein besonderer Dank geht an meine Familie, die mir das Studium ermöglicht hat und in schwierigen Phasen stets ein guter Rückhalt war.

Für die Vergabe des interessanten Themas und die ausgezeichnete Betreuung der Arbeit möchte ich mich bei Prof. Dr. Robert Weibel und Dr. Dirk Burghardt bedanken. Matthias Bobzien und Moritz Neun halfen mir bei vielen kleinen Problemchen und trugen mit ihrer Begeisterung für das Thema und zahlreichen Anregungen zum Entstehen dieser Arbeit bei.

Danken möchte ich auch meinen Studienkollegen Matthias Luder und Oliver Pearce für die zahlreichen Gespräche, die mich immer wieder motiviert und mir mit Kritik und Anregungen weitergeholfen haben.

Buchs, im Januar 2006

Patrick Lüscher



# Inhalt

	Abbildungsverzeichnis .....	vii
	Tabellenverzeichnis .....	xi
<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Hintergrund und Motivation .....	1
1.2	Zielsetzung .....	2
1.3	Gliederung der Arbeit .....	3
<b>2</b>	<b>Multirepräsentationsdatenbanken</b>	<b>5</b>
2.1	Multirepräsentationsdatenbanken (MRDB) .....	5
2.1.1	Multiple Repräsentationen .....	5
2.1.2	Was sind MRDB? .....	6
2.2	Globale Schemas für MRDB .....	7
2.2.1	Verknüpfungskardinalitäten .....	7
2.2.2	Modellierung der Verknüpfungen .....	8
2.2.3	Hierarchische Datenmodelle für Multiresolutionsdatenbanken .....	9
2.2.4	Verknüpfung der Objekt-Instanzen .....	11
2.3	Anwendungen von MRDB .....	11
2.3.1	Web-Kartographie und mobile Kartographie .....	11
2.3.2	Inkrementelle Generalisierung .....	12
2.3.3	Fahrzeugnavigation .....	13
2.3.4	Verbesserung/Überprüfung der Datenqualität .....	14
<b>3</b>	<b>Matching von räumlichen Objekten</b>	<b>15</b>
3.1	Ein methodischer Rahmen für Matching-Prozesse .....	15
3.1.1	Begriffserläuterung .....	15
3.1.2	Übersicht über den Matching-Ablauf .....	15
3.1.3	Matching-Teilprozesse .....	17
3.1.4	Zuordnungsfehler .....	19
3.2	Ähnlichkeitsmasse .....	19
3.2.1	Semantische Ähnlichkeitsmasse .....	19
3.2.2	Geometrische Ähnlichkeitsmasse .....	22
3.2.3	Kontextabhängige Ähnlichkeitsmasse .....	25
3.2.4	Verknüpfung von Einzelmassen zu einem Gesamtmass .....	27
3.3	Besprechung bestehender Ansätze zum Matching von Strassendaten .....	28
<b>4</b>	<b>Ansatz zum Matching von Strassendaten stark unterschiedlicher Massstäbe</b>	<b>33</b>
4.1	Gegenüberstellung der Datensätze VECTOR25 und VECTOR200 .....	33
4.1.1	Der Datensatz VECTOR25 .....	33

4.1.2	Der Datensatz VECTOR200 . . . . .	35
4.1.3	Vergleich der Datenmodelle . . . . .	36
4.1.4	Vergleich der geometrischen Erfassung . . . . .	37
4.2	Matching-Prozess . . . . .	39
4.2.1	Motivation zur Entwicklung des Matching-Ansatzes . . . . .	39
4.2.2	Abgrenzung . . . . .	40
4.2.3	Übersicht über den Ablauf. . . . .	40
4.2.4	Module des Matching-Prozesses . . . . .	41
4.2.5	Bildung von Kandidatenmengen . . . . .	49
4.2.6	Matching der Knoten. . . . .	54
4.2.7	Matching der Strassen . . . . .	55
4.2.8	Nachbearbeitung . . . . .	55
<b>5</b>	<b>Implementation des Prototyps</b>	<b>57</b>
5.1	Anforderungen. . . . .	57
5.2	Software-Komponenten. . . . .	57
5.2.1	Java Topology Suite (JTS) . . . . .	58
5.2.2	JUMP Unified Mapping Platform . . . . .	59
5.2.3	Axpond/DRIVE . . . . .	59
5.2.4	Java Matrix Package (JAMA) . . . . .	59
5.3	Modellierung der MRDB . . . . .	59
5.3.1	Modellierung von Repräsentationen . . . . .	59
5.3.2	Modellierung der Strassen-Objekte. . . . .	61
5.3.3	Modellierung von Verknüpfungen . . . . .	61
5.4	Persistenz. . . . .	63
5.5	Matching-Prototyp. . . . .	64
5.5.1	Benutzeroberfläche . . . . .	64
5.5.2	Erzeugung der MRDB. . . . .	65
5.5.3	Manuelle Erzeugung und Nachbearbeitung von Verknüpfungen . . . . .	65
5.5.4	Kandidatenbildung im automatischen Matching-Prozess . . . . .	66
5.5.5	Benutzerinteraktion im automatischen Matching-Prozess . . . . .	68
5.6	Visualisierung der MRDB. . . . .	69
5.6.1	Visualisierung von 1:N-Verknüpfungen zwischen Strassen . . . . .	69
5.6.2	Visualisierung von Kreiseln und kollabierten Strassensegmenten . . . . .	73
<b>6</b>	<b>Evaluation</b>	<b>75</b>
6.1	Übersicht . . . . .	75
6.2	Evaluation des automatischen Matching-Algorithmus . . . . .	77
6.2.1	Zuordnungsrate . . . . .	77
6.2.2	Fehlerrate . . . . .	80
6.3	Zeitverhalten . . . . .	81
6.3.1	Abhängigkeit der Laufzeit von der Dateigrösse . . . . .	82
6.3.2	Anteil der einzelnen Module . . . . .	83
<b>7</b>	<b>Schlussfolgerungen und Ausblick</b>	<b>85</b>
7.1	Erreichtes . . . . .	85
7.2	Diskussion . . . . .	86
7.3	Ausblick. . . . .	87
7.3.1	Ausbau des Prototyps . . . . .	87
7.3.2	Entwicklung einer generellen Matching-Plattform . . . . .	88
	<b>Literaturverzeichnis</b>	<b>91</b>

# Abbildungsverzeichnis

## Kapitel 2

2.1	Je nach Anwendung und Massstab können von demselben Realwelt-Phänomen verschiedene Repräsentationen abgeleitet werden. Luftphoto und Übersichtsplan: GIS-Browser des Amts für Raumordnung und Vermessung des Kantons Zürich. Landeskarten: Aus VECTOR25 resp. VECTOR200. . . . .	6
2.2	Verknüpfungskardinalitäten von Strassen (Sester et al. 1998:342). . . . .	8
2.3	Möglichkeiten, zwei Klassen aus den Datenbanken $T$ und $C$ in eine MRDB zu integrieren (Balley et al. 2004:338). . . . .	9
2.4	Hierarchie, die durch die zunehmende Generalisierung entsteht, am Beispiel einer Kirche. . . . .	9
2.5	(a) Kartenobjekte im Map Cube Model (Timpf 1998:191). (b) Ein Generalisierungsgraph für Area-Objekte (Timpf 1998:196). . . . .	10
2.6	Beispiel eines Drill-downs. Niedrige Auflösung: Aus der Landeskarte 1:25'000. Hohe Auflösung: Aus dem Übersichtsplan des Kantons Zürich. . . . .	12
2.7	(a) Prinzip von Adaptive Zooming (Cecconi 2003:110). (b) Beispiel für eine Interpolation zwischen zwei in der Datenbank gespeicherten Massstabsebenen $s_{25}$ und $s_{200}$ (Cecconi 2003:113). . . . .	13

## Kapitel 3

3.1	Matching zweier Datensätze. . . . .	16
3.2	<i>Buffer Growing</i> -Algorithmus (Walter 1996:64). . . . .	17
3.3	(a) Mittlere Zwischenfläche als Distanzmass zwischen Linien. (b) Fall, wo die mittlere Zwischenfläche versagt (Devogele 1997:39). . . . .	22
3.4	Ein Puffer der Breite $x$ um die „echte“ Küstenlinie wird mit der Vergleichslinie verschnitten, um den Prozentsatz zu bestimmen, zu welchem die Vergleichslinie im Puffer liegt (Goodchild 1997:301). . . . .	23
3.5	Berechnung der Hausdorff-Distanz (Hangouët 1995:2). . . . .	23
3.6	Hausdorffdistanz bei Linien verschiedener Länge. . . . .	24
3.7	(a) Basislinie $b$ einer Linie $L$ . (b) Zwischenwinkel zwischen Segmenten. . . . .	24

3.8	Der Nachbarschaftsgraph für das National Museum of Natural History (Samal et al. 2004:473).....	26
3.9	(a) Matching zweier Nachbarschaftsgraphen (b) Daraus abgeleitete Verschiebungsvektoren (Samal et al. 2004:474). ....	26
3.10	Beispiel für den Gebrauch des Knotengrades im Strassenmatching. Cyan: Referenzlinie. Rot: Kandidatenstrassen (Zhang et al. 2005:4). ....	27
3.11	Gegenüberstellung ATKIS – GDF (Walter 1996:57).....	29

#### Kapitel 4

4.1	Die beiden Kreisel (grün markiert) in VECTOR25 kollabieren zu einem Knoten in VECTOR200. ....	38
4.2	VECTOR25-Segment, das in VECTOR200 zu einem Knoten kollabiert ist (grüner Pfeil).....	38
4.3	Modellierung der Autobahnen mit getrennten Fahrspuren (VECTOR25) bzw. mit einer Mittellinie (VECTOR200). Durchgezogen: Als Autobahn klassierte Strassen. Gestrichelt: Autobahneinfahrten bzw. -ausfahrten. ....	38
4.4	Vergleich der Längen von Strassenstücken. ....	39
4.5	Ablauf des Matching-Prozesses.....	42
4.6	Vergleich zweier Knotenkandidaten mittels Zwischenwinkelsumme. (a) Situation. Grün: VECTOR25-Knotenkandidaten. (b) Zwischenwinkel für Knotenkandidat 1. (c) Zwischenwinkel für Knotenkandidat 2.....	44
4.7	Filterung nach dem kürzesten und nach dem nächsten Weg (Devogele 1997:138)..	44
4.8	Ablauf des Shortest Path-Algorithmus nach Dijkstra. ....	46
4.9	Filterung durch das Modul <i>Nächste benachbarte Wege</i> . Rot: VECTOR200-Strasse. Grün: Knotenkandidaten. Grau/Schwarz: Alle Kandidatenstrassen. Schwarz: Kandidatenstrassen, welche Teil eines nächsten benachbarten Wegs sind.....	47
4.10	Die Linienverfolgung trifft auf eine Gabelung.....	47
4.11	Situation nach der Auflösung der Gabelung durch die Linienverfolgung. Strassenkandidat a fällt weg, dadurch verbleibt nur noch ein Knotenkandidat, der eindeutig zugeordnet werden kann. ....	48
4.12	Flussdiagramm der Phase <i>Bildung von Kandidatenmengen für Strassen und Knoten</i> .....	49
4.13	Beispiel zur Illustration der Kandidatenbildung. Rot: VECTOR200. Blau: VECTOR25. ....	49
4.14	250 m-Puffer um das VECTOR200-Stassensegment (gelb) und VECTOR25-Segmente, die innerhalb dieses Puffers liegen (schwarz). ....	50
4.15	Kandidatenmenge nach der Anwendung der Beschränkung <i>Strassenklasse</i> .....	51

4.16	Umgebung der beiden Endknoten (rote Rechtecke) der Beispiels-Strasse und deren Knotenkandidaten. (a) linker Endknoten (b) rechter Endknoten. . . . .	51
4.17	Histogramm der mittleren Zwischenwinkel für das von Hand zugeordnete Gebiet Pfäffikon. . . . .	52
4.18	(a) Der linke Endknoten und (b) der rechte Endknoten nach der Anwendung der Beschränkungen <i>Knotengrad</i> und <i>mittlere Zwischenwinkelsumme</i> . . . . .	52
4.19	Kandidatenmenge nach der Bildung des Graphen und Entfernen von Sackgassen und der separaten Teilgraphen. . . . .	53
4.20	Kandidatenmenge nach der Bildung der nächsten benachbarten Wege. Die grünen Kreise bezeichnen die Knoten, für welche nächste benachbarte Wege berechnet wurden. . . . .	53
4.21	Nächste benachbarte Wege für die zu den Endknoten angrenzenden Strassen und die verbleibenden Knotenkandidaten. . . . .	54
4.22	Flussdiagramm der Phase <i>Matching der Knoten</i> . . . . .	54
 <b>Kapitel 5</b>		
5.1	Software-Komponenten des Prototyps. . . . .	58
5.2	Axpand/DRIVE Datenmodell einer Repräsentation. . . . .	60
5.3	UML-Klassendiagramm von <i>GenObjectStreet</i> und <i>GenObjectRoundabout</i> . . . . .	61
5.4	Implizite Modellierung einer 1:N-Verknüpfung. . . . .	62
5.5	Explizite Modellierung einer 1:N-Verknüpfung. . . . .	63
5.6	Oberfläche des Matching-Prototyps. . . . .	64
5.7	Dialogbox zum Import von neuen Daten in den <i>RoadMatcher</i> . . . . .	65
5.8	Werkzeugleiste des Matching-Prototyps. . . . .	66
5.9	UML-Klassendiagramme von <i>NodeCandidateSet</i> und <i>NodeCandidate</i> . Es sind nur die wichtigsten Attribute und Methoden aufgeführt. . . . .	67
5.10	UML-Klassendiagramm von <i>StreetCandidateSet</i> mit den wichtigsten Attributen und Methoden. . . . .	68
5.11	Benutzerinteraktion im automatischen Matching-Prozess. . . . .	69
5.12	Repräsentationen in separaten Kartenfeldern (Anders et al. 2003). . . . .	70
5.13	Visualisierung mit flächenhaften Schraffuren. Rot: VECTOR200. Blau: VECTOR25. . . . .	71
5.14	Suche nach einer Geraden, die senkrecht auf dem VECTOR200-Strassenstück (rot) steht. . . . .	72
5.15	(a) Verbindungslinie 2 ist inkorrekt, da sie ein zugeordnetes VECTOR25-Segment überschneidet. (b) Fall mit 3 gültigen Verbindungslinien. . . . .	72
5.16	Situation, wo keine senkrechte Verbindung möglich ist. . . . .	73

5.17	Visualisierung mit senkrechten Verbindungslinien. Rot: VECTOR200. Blau: VECTOR25. ....	73
5.18	(a) Kollabierter VECTOR25-Kreisel (b) Kollabierte VECTOR25-Strasse. ....	74
<b>Kapitel 6</b>		
6.1	(a) Testgebiet „Pfäffikon“ (b) Testgebiet „Winterthur“. ....	76
6.2	Ausschnitt aus dem Gebiet „Pfäffikon“. Links: VECTOR25 (blau) überlagert mit VECTOR200 (rot). Rechts: Extrahierte VECTOR25-Strassen. ....	76
6.3	Ausschnitt aus dem Gebiet „Winterthur“. Links: VECTOR25 (blau) überlagert mit VECTOR200 (rot). Rechts: Extrahierte VECTOR25-Strassen. ....	77
6.4	(a) Erfolgreiches Knotenmatching. (b) – (d) Situationen, wo der korrespondierende Knoten nicht automatisch gefunden wurde. ....	79
6.5	Situation mit stark unterschiedlichen korrespondierenden Strassen. ....	80
6.6	Falschzuordnung von Strassen im Gebiet Winterthur. ....	81
6.7	Laufzeit der Kandidatenbildung in Abhängigkeit von der Datensatzgrösse. ....	82

# Tabellenverzeichnis

## Kapitel 3

3.1	Ähnlichkeitstabelle für das Attribut Stromquelle (Cobb et al. 1998:28). . . . .	20
3.2	String-Anomalien, die beim Matching berücksichtigt werden müssen (Samal et al. 2004:468). . . . .	21

## Kapitel 4

4.1	Attribute der Ebene Strassennetz aus VECTOR25 (Swisstopo 2004a:8). . . . .	34
4.2	Objektarten der Ebene Strassennetz aus VECTOR25 (Swisstopo 2004a:9). . . . .	34
4.3	Attribute der Ebene Verkehrsnetz aus VECTOR200 (Swisstopo 2004b:9). . . . .	35
4.4	Objektarten der Ebene Verkehrsnetz aus VECTOR200 (Swisstopo 2004b:10). . . . .	36
4.5	Vereinbarkeit der Objektklassen (0 = nicht vereinbar, 1 = vereinbar). Zeilen: VECTOR200-Objektklassen. Spalten: VECTOR25-Objektklassen. . . . .	42

## Kapitel 6

6.1	Zuordnungsraten des automatischen Matching-Prozesses für die Testgebiete Pfäffikon und Winterthur. . . . .	78
6.2	Anteil der verschiedenen Module an der Gesamtlaufzeit in der Phase <i>Kandidatenbildung</i> . . . . .	83



# Kapitel 1

## Einleitung

### 1.1 Hintergrund und Motivation

Räumliche Daten sind heute vielfältig und in grosser Anzahl digital vorhanden. Während man sich früher bei der Analyse eines geographischen Problems auf einen oder wenige Datensätze beschränkte, werden heute üblicherweise eine grosse Anzahl verwendet. Die Datensätze werden meistens unabhängig voneinander gehalten, und die Integration geschieht visuell oder durch eine geometrische Überlagerung (*Overlay*). Diese Art der Datenhaltung ist nicht effizient, da Aktualisierungen an allen Datensätzen separat ausgeführt werden müssen. Zudem kann das Potential gemeinsamer Analysen durch die einfachen Integrationsmethoden nicht ausgeschöpft werden.

Bei Multirepräsentationsdatenbanken (MRDBs) wird eine permanente Integration von verschiedenen Repräsentationen räumlicher Daten angestrebt. Diejenigen Objekte, welche in den verschiedenen Datensätzen dasselbe Realwelt-Phänomen darstellen, sind explizit miteinander verknüpft. Änderungen in einem Datensatz können automatisch in alle anderen Datensätze der MRDB übertragen werden. So verbessert sich nicht nur die Konsistenz, auch die Nachführung wird einfacher. Nachdem die nationalen topographischen Ämter ihre Landeskartenwerke auf die digitale Produktion umgestellt haben, stellt sich auch für sie die Frage, wie ihr Datenbestand rationell gespeichert und effizient nachgeführt werden kann. Einige dieser Ämter, beispielsweise die Swisstopo und das Bundesamt für Kartographie und Geodäsie (BKG) in Deutschland, arbeiten derzeit am Aufbau eigener MRDBs.

Multirepräsentationsdatenbanken werden zunehmend auch in der digitalen Kartographie verwendet: Die Produktion einer klassischen Papierkarte ist aufwändig. Daher muss man sich auf wenige, dafür aber flexibel anwendbare und informationsreiche Ausgaben beschränken. In der digitalen Kartographie kann besser auf Benutzerwünsche eingegangen werden, indem Karten *ad hoc* nach Benutzervorgaben generiert werden. Aus einem grossen Datenbestand sollen relevante Datenobjekte extrahiert, zu einem Gesamtmodell fusioniert und dem Benutzer auf kartographisch ansprechende Weise präsentiert werden. Dieser Vorgang muss automatisch und soll möglichst schnell geschehen. Dies wird durch die Integration der Datensätze in eine MRDB möglich.

Für die Integration bestehender Daten in eine MRDB müssen die Objekt-Instanzen, die in den verschiedenen Repräsentationen dasselbe Realwelt-Phänomen modellieren, erkannt werden. Wegen der Grösse der Datensätze kommen meistens nur automatische Methoden in Frage.

*Matching-Algorithmen*, welche Objekte anhand ihrer geometrischen und semantischen Eigenschaften und ihren Beziehungen zu den Nachbarobjekten miteinander vergleichen und bei ausreichender Übereinstimmung eine Verknüpfung erstellen, haben in den letzten Jahren stark an Bedeutung gewonnen.

Strassendaten gehören zu den wichtigsten topographischen Grundlagendaten mit diversen Anwendungen in der Navigation und Planung. Deshalb haben Matching-Verfahren für Strassendaten eine grosse praktische Bedeutung. Die existierenden Verfahren wurden für Datensätze geschaffen, die in gleichen oder zumindest in ähnlichen Massstäben vorliegen. Für Datensätze von unterschiedlichen Massstäben wird vorgeschlagen, den grösseren (detaillierteren) Massstab vor der Zuordnung zu generalisieren (Sester et al. 1998:354). Folgende Argumente sprechen gegen diesen Ansatz:

- Für stark unterschiedliche Massstäbe existieren zur Zeit noch keine vollständig automatische Generalisierungsmethoden. Manuelle oder halb-interaktive Methoden sind jedoch zu aufwändig und bieten keine Vorteile gegenüber manuellem Matching.
- Wenn die beiden zu verknüpfenden Datensätze unabhängig voneinander erfasst worden sind, führt eine Generalisierung des grossmassstäbigeren Datensatzes nicht notwendigerweise dazu, dass beide Datensätze besser vergleichbar werden. Viele der bestehenden Datensätze wurden manuell generalisiert und dieser Prozess unterliegt auch subjektiven Entscheidungen, die mit automatischen Methoden nicht nachgeahmt werden können.

In der vorliegenden Arbeit werden Strassendaten betrachtet, die sich stark bezüglich ihres Massstabs unterscheiden: Die Massstabsebenen 1:25'000 und 1:200'000 sollen in eine MRDB integriert werden. Im Fall von gleichen oder ähnlichen Massstäben werden für das Matching oft Eigenschaften individueller Strassen oder weniger Strassengruppen miteinander verglichen. Für stark unterschiedliche Massstäbe ist dies wenig Erfolg versprechend: Im grösseren Massstab ist das Strassennetz meist viel dichter, folglich korrespondieren viele kurze Strassenstücke mit einer einzigen Strasse aus dem kleineren Massstab. Um überhaupt vergleichbare Einheiten zu erhalten, müssten die kurzen Strassenstücke aus dem grösseren Massstab zuerst gefiltert und aggregiert werden. Weitere Herausforderungen sind, dass sich die Strassenverläufe beider Massstabsebenen nur bedingt ähnlich sind, da sie im kleineren Massstab i. A. stark geglättet wiedergegeben werden. Lokal begrenzt können wegen der erfolgten Verdrängung von nahe liegenden Kartenobjekten grosse geometrische Unterschiede auftreten, z. B. bei engen Strassenkurven. Folglich stellt sich die Frage, welche Methoden zum Matching von Strassendaten, die sich stark im Massstab unterscheiden, geeignet sind, und wie gut sich solche Datensätze überhaupt noch automatisch zuordnen lassen.

## 1.2 Zielsetzung

In der vorliegenden Arbeit soll die Integration von Strassendaten, die sich stark bezüglich ihres Massstabes unterscheiden, in eine Multirepräsentationsdatenbank untersucht werden. Die Ziele der Arbeit sind die folgenden:

1. *Entwicklung eines Matching-Prozesses für Strassendaten von stark unterschiedlichen Massstäben.* Für Entwurf und Test des Matching-Prozesses stehen Ausschnitte aus den Datensätzen VECTOR25 und VECTOR200 des Bundesamtes für Landestopografie (Swisstopo) zur Verfügung.
2. *Aufbau einer Multirepräsentationsdatenbank (MRDB) aus den verknüpften Strassendaten.* Die im Matching-Prozess gefundenen Verknüpfungen müssen explizit modelliert und persistent gespeichert werden.
3. *Implementation der Konzepte in einem Prototyp.* Die Umsetzbarkeit des Matching-Prozesses soll mit einem Prototyp gezeigt werden. Im Zusammenhang mit dem künftigen produktiven Einsatz von Multirepräsentationsdatenbanken sollen Visualisierung von MRDBs und Benutzerinteraktion zur Manipulation der Verknüpfungen untersucht werden.

## 1.3 Gliederung der Arbeit

**Kapitel 2** – Multirepräsentationsdatenbanken werden definiert und alternative Möglichkeiten zu deren Aufbau diskutiert. Die praktische Anwendung von MRDBs wird anhand einiger Beispiele gezeigt.

**Kapitel 3** – Es wird ein konzeptueller Rahmen für das Matching von Objekten aus verschiedenen Repräsentationen erarbeitet. Bestehende Ansätze für das Matching von Strassendaten werden beschrieben.

**Kapitel 4** – Die beiden Datensätze VECTOR25 und VECTOR200 werden im Hinblick auf ihr Datenmodell und auf Unterschiede in der Erfassung miteinander verglichen. Aufbauend auf die in Kapitel 3 besprochenen Verfahren wird ein eigener Prozess für das Matching von Strassendaten stark unterschiedlicher Massstäbe hergeleitet.

**Kapitel 5** – Geht auf die Implementation des Matching-Prototyps ein. Ein neuer Algorithmus zur Visualisierung von Verknüpfungen zwischen Linienobjekten wird erläutert.

**Kapitel 6** – Zeigt die Ergebnisse, die mit dem Matching-Algorithmus in den Testgebieten erreicht wurden. Anhand von Beispielen werden typische Situationen besprochen, die mit dem Matching-Algorithmus nicht lösbar sind. Die Zeitkomplexität des Algorithmus wird diskutiert.

**Kapitel 7** – Enthält die Schlussfolgerungen und den Ausblick.



## Kapitel 2

# Multirepräsentationsdatenbanken

Multirepräsentationsdatenbanken (MRDB) enthalten verschiedene Repräsentationen desselben Gebietes. Sie versprechen eine günstigere Fortführung, erhöhte Konsistenz und einfachere Benutzung von Geodaten. Eine häufige Anwendung ist die Integration von Datensätzen verschiedenen Massstabes. In diesem Kapitel werden Multirepräsentationsdatenbanken definiert und Alternativen für deren Aufbau diskutiert. Die praktische Anwendung von MRDBs wird an Beispielen gezeigt.

## 2.1 Multirepräsentationsdatenbanken (MRDB)

### 2.1.1 Multiple Repräsentationen

Objekte der Realwelt können niemals vollständig erfasst werden, sondern es werden vielmehr nur bestimmte Aspekte davon beschrieben (Glinz 2001:1–3). Wie die geographische Realität repräsentiert wird, hängt von der beabsichtigten Anwendung ab. Gemäss Hangouët (2004:313) besteht eine Repräsentation aus fünf unabhängigen Bestandteilen:

$$\textit{Representation} = (\textit{Phenomenon}, \textit{Attention}, \textit{Medium}, \textit{Inscription}, \textit{Reception})$$

Das Augenmerk (*Attention*) bezeichnet die Abhängigkeit vom Zweck der Erfassung, dem Vorwissen und der Subjektivität des Erfassers. Die Inschrift (*Inscription*) ist das Bild das entsteht, wenn das Realwelt-Phänomen (*Phenomenon*) durch ein bestimmtes Augenmerk auf ein bestimmtes *Medium* übertragen wird. In einem Computersystem wird die Inschrift beispielsweise durch die Daten selbst gebildet, die in einem bestimmten Vektor- oder Rastermodell (dem *Medium*) gespeichert sind. Die Aufnahme (*Reception*) schliesslich ist die Wahrnehmung der Repräsentation von anderen, d. h. durch den menschlichen Benutzer oder durch eine Software, welche die Inschrift verarbeitet.

Verschiedene Interpretationen desselben erdräumlichen Ausschnittes führen dazu, dass ein und dasselbe Realwelt-Phänomen mehrfach unter verschiedenen Aspekten repräsentiert wird – im Formalismus von Hangouët gibt es zwei Repräsentationen  $(P, A_1, M_1, I_1, R_1)$  und  $(P, A_2, M_2, I_2, R_2)$ , wo mindestens eines der Attribute Medium, Augenmerk, Inschrift, oder Aufnahme verschieden ist. Dann liegen *multiple Repräsentationen* vor. Die Objekte aus verschiedenen Reprä-

sentationen, die dasselbe Realwelt-Phänomen repräsentieren, werden als *homologe Objekte*<sup>1</sup> bezeichnet.

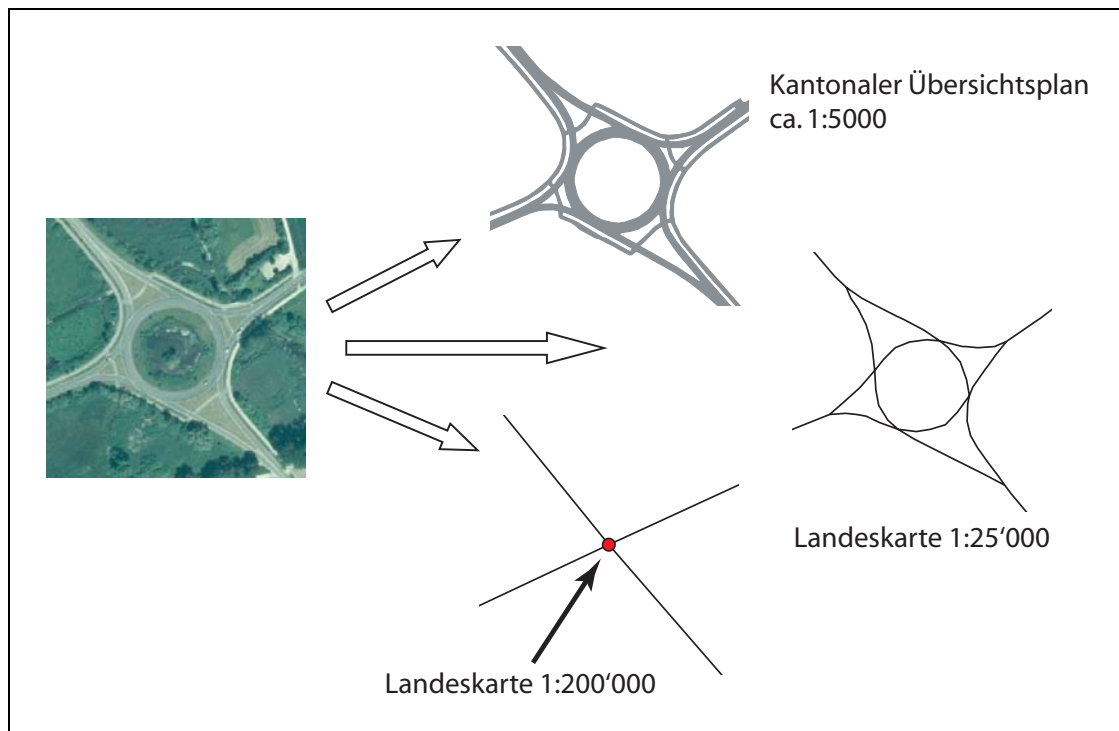


Abbildung 2.1: Je nach Anwendung und Massstab können von demselben Realwelt-Phänomen verschiedene Repräsentationen abgeleitet werden. Luftphoto und Übersichtsplan: GIS-Browser des Amts für Raumordnung und Vermessung des Kantons Zürich<sup>2</sup>. Landeskarten: Aus VECTOR25 resp. VECTOR200.

Gemäss Timpf und Devogele (1997:1382) können sich multiple Repräsentationen unterscheiden bezüglich:

- Massstab<sup>3</sup>: Bezeichnet die Stufe der Abstraktion der Repräsentation.
- Anwendung: Art der Anwendung – diese bestimmt ebenfalls das Raumkonzept.
- Zeit: Zeitpunkt, den die Repräsentation abbildet.
- Räuml. Datenmodell: Modell der Beschreibung von räumlichen Daten.

### 2.1.2 Was sind MRDB?

Für geographische Analysen werden üblicherweise mehrere Repräsentationen verwendet, weshalb multiple Repräsentationen in GIS ein bekanntes Konzept sind. Jedoch werden die Reprä-

1. Aus dem Altgriechischen *homolégeo* = übereinstimmen
2. <http://www.gis.zh.ch/>, Stand 20.12.2005
3. Der Begriff Massstab bezeichnet das Verhältnis zwischen einer Strecke in der Karte und derselben Strecke in Natur. Da ein digitaler Datensatz in jedem beliebigen Massstab dargestellt werden kann, ist die Auflösung (engl. *resolution*), welche das kleinste im Datensatz erfasste Objekt bezeichnet, im digitalen Kontext der bessere Begriff. Da viele Datensätze aus Papierkarten digitalisiert wurden, wird in der vorliegenden Arbeit trotzdem der Begriff Massstab verwendet.

sentationen meist als separate, untereinander nicht verbundene Datensätze gehalten. Dies genügt in vieler Hinsicht nicht; die geographische Analyse, Visualisierung und Nachführung können profitieren, wenn Repräsentationen stärker integriert werden. Es gibt mehrere Stufen der Datenintegration (Devogele 1997:29–31):

- *Integration der Metadaten:* Kataloge beschreiben Datensätze und können dem Benutzer helfen, den passenden Datensatz unter vielen auszuwählen. Metadaten sollen deshalb in einem standardisierten Format erstellt und in einem globalen Katalog verwaltet werden.
- *Integration der Semantik:* Es wird ein integriertes Schema<sup>1</sup> definiert, das die Semantik der ursprünglichen Schemas vereinigt. Zusätzlich müssen Überführungsregeln bestimmt werden, welche die Umwandlung der Daten in das integrierte Schema erlauben.
- *Vollständige Integration:* Bei der semantischen Integration werden zwar die verschiedenen Repräsentationen in einem gemeinsamen Datenmodell vereinigt, aber Objekte, welche dasselbe Realwelt-Phänomen beschreiben, bleiben unverbunden. Bei einer vollständigen Integration werden diese Objekte identifiziert und die Beziehung „entspricht demselben Realwelt-Phänomen“ wird explizit als *Link* in der Datenbank abgebildet.

Die vollständige Integration führt zu einer *Multirepräsentationsdatenbank* (MRDB). Meistens unterscheiden sich die beteiligten Repräsentationen nur im Massstab, so dass eine *Multiresolutionsdatenbank* aus mehreren *Levels of Detail* entsteht.

## 2.2 Globale Schemas für MRDB

### 2.2.1 Verknüpfungskardinalitäten

Ein *Kardinalitätsverhältnis* für einen Gegenstand  $G$ , der an einer Beziehung beteiligt ist, bezeichnet die Anzahl Elemente von  $G$ , welche an dieser Beziehung teilnehmen (Elmasri und Navathe 2004:65). Bei Verknüpfungen zwischen zwei Datensätzen können folgende Kardinalitätsverhältnisse unterschieden werden (Stadler 2004:50–51; Sester et al. 1998:342):

- **1:1**  
Eine 1:1-Beziehung bedeutet, dass eine Instanz aus Datensatz  $A$  einer einzigen Instanz aus Datensatz  $B$  zugeordnet wird. Dies ist der einfachste Fall einer Zuordnung.
- **1:N oder N:1**  
Eine 1:N oder N:1-Beziehung bedeutet, dass je  $n$  Objekte aus einem Datensatz genau einem Objekt aus dem anderen Datensatz zugeordnet werden. Dieser Fall entspricht einer Aggregation. Er tritt häufig auf, wenn die beiden Datensätze unterschiedliche Massstäbe haben.
- **1:0 oder 0:1**  
Bei gleichen Massstäben ist möglicherweise ein Objekt nur in einem Datensatz vorhanden, bei unterschiedlichen Massstäben werden oft unbedeutende Objekte wie Fusswege im kleineren Massstab ganz weggelassen. Dieser Fall kann als 1:0-Beziehung modelliert werden

---

1. In einem Datenbanksystem beschreibt das *Datenschema* das Datenmodell, während der Begriff *Datenbank* den Dateninhalt bezeichnet (Elmasri und Navathe 2004:27)

- **N:M**

Der allgemeinste Fall ist eine N:M-Beziehung, wo eine Menge von  $n$  Objekten aus Datensatz  $A$  einer Menge von  $m$  Objekten aus Datensatz  $B$  zugeordnet wird.

Bei Strassen kann ein Spezialfall davon auftreten, wenn in einem Datensatz eine Strasse durch zwei verschiedene Fahrspuren repräsentiert wird, im anderen Datensatz aber nur durch eine Mittelachse. Dann liegt eine  $N:M_1+M_2$ -Verknüpfung vor.

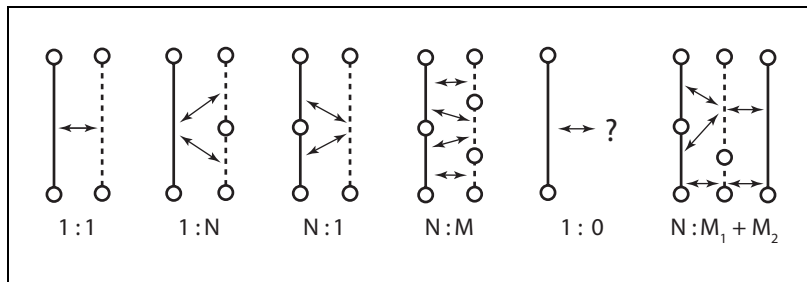


Abbildung 2.2: Verknüpfungskardinalitäten von Strassen (Sester et al. 1998:342).

## 2.2.2 Modellierung der Verknüpfungen

Zur Modellierung der Verknüpfungen gibt es zwei Ansätze: Bei einer *starken Integration* werden die ursprünglichen Schemas in ein globales Schema vereinigt. Korrespondierende Klassen werden zu einer einzigen verschmolzen. Jede Klasse und jedes Attribut des vereinigten Schemas trägt eine Marke, die angibt, für welche Repräsentationen das Element gültig ist. Bei einer *schwachen Integration* bleiben die ursprünglichen Klassen erhalten wie sie sind und werden mit einer Beziehung „korrespondierendes Objekt“ verknüpft. Die Beziehung kann jede der in Kapitel 2.2.1 aufgeführten Kardinalitäten modellieren.

Abbildung 2.3 zeigt die beiden Integrationsarten, wie sie im Projekt MurMur (Multiple Representation-Multiple Resolution) erarbeitet worden sind (Balley et al. 2004). Flussabschnitte sind in der Datenbank  $T$  als eine Klasse „River-Section-T“ modelliert. Sie hat den Typ einer Liniengeometrie (Symbol  $\curvearrowright$ ) mit ihrem zeitlichen Verlauf (Symbol  $f(\odot)$ ). Die Attribute der Klasse sind ein eindeutiger Identifikator und dort, wo ein Wasserbecken vorkommt, eine Punktgeometrie, die den Ort des Wasserbeckens angibt. In Datenbank  $C$  ist ein Flussabschnitt auch als eine Klasse mit Liniengeometrie modelliert und trägt als Attribute nur den eindeutigen Identifikator.

Die linke Ableitung zeigt eine starke Integration der beiden Klassen. Die Marken, die angeben, aus welcher Repräsentation die Attribute stammen, sind mit  $\boxed{T}$  bzw.  $\boxed{C}$  symbolisiert. Das Attribut *Identifizier* gilt für beide Repräsentationen und trägt daher keine Marke. Die rechte Ableitung zeigt eine schwache Integration.

Welche der beiden Möglichkeiten besser geeignet ist, hängt von der Anwendung ab. Eine starke Integration kann bei stark unterschiedlichen Datenmodellen aufwändig oder unmöglich sein und ist weniger flexibel bezüglich späterer Erweiterungen. Ein weiterer Vorteil der schwachen Integration ist, dass die Verknüpfungen mit zusätzlichen Informationen angereichert werden können. Sie können nicht nur anzeigen, mit welcher Sicherheit auf einen Match (siehe Abschnitt 2.2.4) geschlossen werden kann, welche Art von Übergang stattfindet (Vereinfachung, Aggrega-

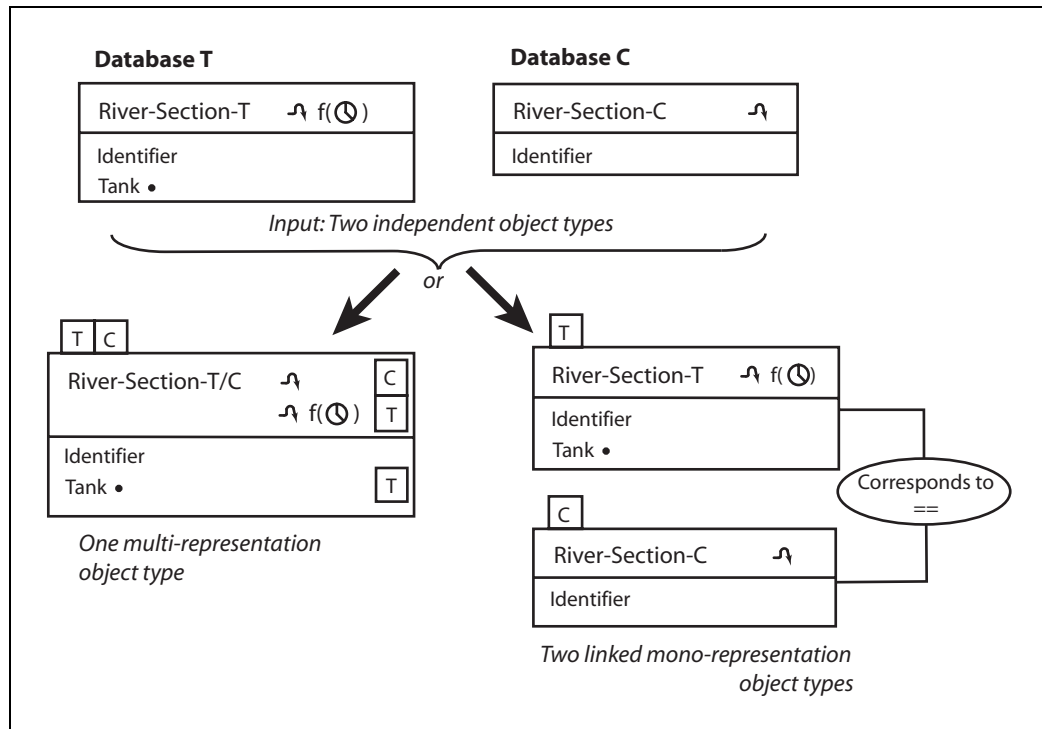


Abbildung 2.3: Möglichkeiten, zwei Klassen aus den Datenbanken *T* und *C* in eine MRDB zu integrieren (Balley et al. 2004:338).

tion eines Kreisels zu einem Kreuzungspunkt, Geometriypwechsel zu einem Symbol etc.), sondern auch zusätzliche kartographische Vergleichsmasse der beiden Repräsentationen festhalten, die Operationen auf die MRDB wie das Interpolieren zwischen Masstabsebenen wesentlich beschleunigen können. Einige kartographische Vergleichsmasse werden in Kapitel 3 besprochen.

### 2.2.3 Hierarchische Datenmodelle für Multiresolutionsdatenbanken

Bei abnehmendem Masstab steht weniger Platz für Objekte zur Verfügung. Sie werden deshalb vereinfacht, mit anderen aggregiert, oder gelöscht. Daher entsteht typischerweise eine hierarchische Beziehung zwischen homologen Objekten in verschiedenen Masstäben.

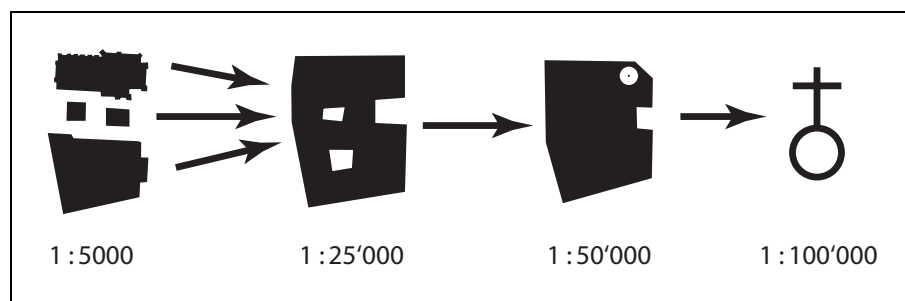


Abbildung 2.4: Hierarchie, die durch die zunehmende Generalisierung entsteht, am Beispiel einer Kirche.

Diese Hierarchie kann in einer Datenbank modelliert werden. Erste Ansätze beschränkten sich darauf, die Geometrie redundanzfrei zu speichern, wie der Multi-Scale Line Tree (Jones und Abraham 1986:387–391). Das Map Cube Model (Timpf 1998) arbeitet mit Hierarchien auf zwei Ebenen:

- **Innerhalb einer Repräsentation.** Die Landschaft wird durch das *Trans-hydro-Netz* (kombiniertes Strassen- und Gewässernetz) in einzelne *Containers* geteilt. *Areas* sind Flächen innerhalb eines Containers, wie Landnutzungsflächen. *Elements* sind wiederum in Areas enthalten. Es sind Kartenelemente wie Häuser, Symbole, Beschriftungen etc.
- **Zwischen Repräsentationen.** Jedes der Kartenelemente bildet über die Massstabebenen hinweg eine Generalisierungshierarchie. Beim Trans-hydro-Netz fallen in kleineren Massstäben unwichtige Strassen und kleinere Bäche weg und damit werden Container miteinander vereinigt. Kartenelemente können vereinfacht werden, sich mit anderen Kartenelementen vereinigen, oder ganz verschwinden. Somit lassen sich für alle Kategorien (Trans-hydro-Netzwerk, Container, Area, und Element) Graphen bilden, welche diese Hierarchie nachbilden.

Somit ist jedes Kartenelement doppelt verknüpft: horizontal mit dem Kartenobjekt, in welchem es enthalten ist (Element in Area, Area in Container) und vertikal mit den homologen Objekten der nächsthöheren und nächsttieferen Massstabebene. Diese doppelte Verknüpfung gibt eine gewisse Kontrolle über die Konsistenz der Topologie. So kann zum Beispiel ein Haus nicht in zwei Massstabebenen auf verschiedenen Strassenseiten liegen, da es dann in zwei verschiedenen, miteinander nicht im Generalisierungsgraphen verbundenen Containern liegen würde.

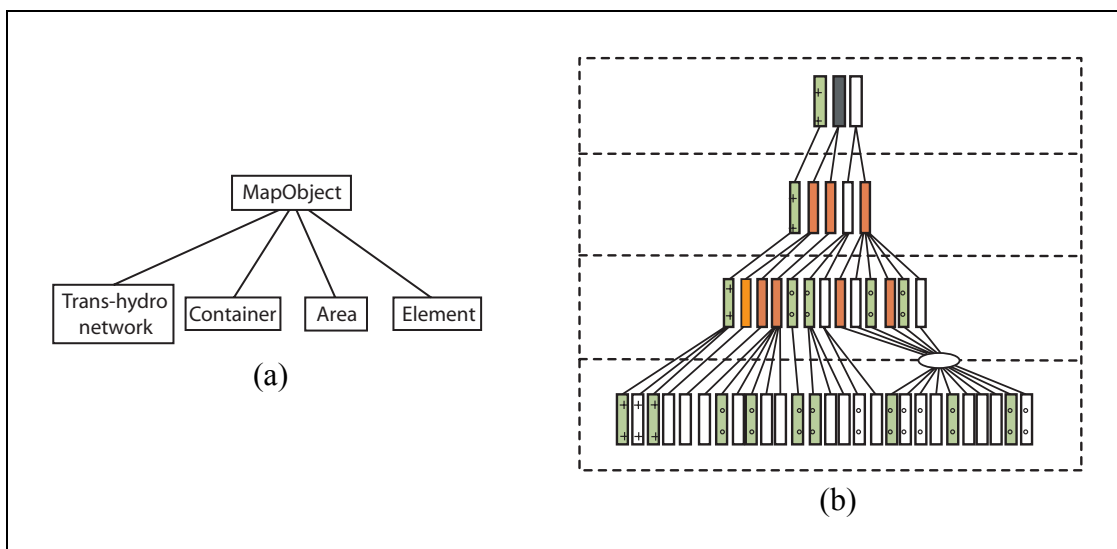


Abbildung 2.5: (a) Kartenobjekte im Map Cube Model (Timpf 1998:191). (b) Ein Generalisierungsgraph für Area-Objekte (Timpf 1998:196).

## 2.2.4 Verknüpfung der Objekt-Instanzen

Nach der Erstellung eines globalen Schemas müssen die eigentlichen Objekt-Instanzen einander zugeordnet werden. Grundsätzlich gibt es zwei Möglichkeiten:

1. Neu-Ableitung von Daten aus Basisdatensätzen
2. Integration von bestehenden Daten

Durch die Neu-Ableitung ergeben sich die Verknüpfungen automatisch – sie müssen nur noch explizit mitgespeichert werden. Trotzdem ist eine Neu-Ableitung meistens nicht machbar:

- Eine Neu-Ableitung ist nicht vollständig automatisch lösbar, da heute die dazu nötigen Prozeduren der automatischen Generalisierung noch unvollständig sind. Sie ist deshalb mit grossen Kosten verbunden.
- Die bestehenden Datensätze wurden mit viel Aufwand und Know-How aufgebaut und fortgeführt. Es ist nicht sinnvoll, dieses wirtschaftliche Potential auszumustern. Stattdessen sollte man auch in einer MRDB weiter davon profitieren.

Es werden deshalb möglichst automatische Methoden gesucht, welche die Objekt-Instanzen aus den verschiedenen Datensätzen, die dasselbe Realwelt-Phänomen beschreiben, einander zuordnen. Diese als *Matching* bekannten Prozesse werden in Kapitel 3 behandelt.

## 2.3 Anwendungen von MRDB

In diesem Abschnitt sollen einige wichtige Anwendungen von MRDBs besprochen werden.

### 2.3.1 Web-Kartographie und mobile Kartographie

Die elektronische Kartographie bietet gegenüber der klassischen analogen Kartographie eine viel grössere Flexibilität: Da der Aufwand für die Erstellung statischer Papierkarten gross ist, gibt es davon wenige Variationen, welche dafür breit einsetzbar sein müssen. Elektronische Karten können dagegen gezielt auf Benutzerbedürfnisse eingehen. Die Ausgabe von Karten geschieht immer öfters online auf dem Web oder auf einem mobilen Gerät. Für diese Nutzungsarten ist eine wichtige Anforderung, dass die Erstellung der Karte in kurzer Zeit (und automatisch) geschehen muss. MRDB bieten eine Lösung für diese Anforderungen, indem Elemente verschiedener Repräsentationen zu einer einzigen Karte fusioniert werden.

Um online zwischen Repräsentationen zu navigieren, wurde der klassische *graphische Zoom um Drill-Operationen* (Bernier et al. 2005) erweitert. Während der graphische Zoom Daten nur aus einer Repräsentation bezieht, erlauben es Drills, Daten aus verschiedenen Repräsentationen zu kombinieren. Drills sind dabei auch nur für einzelne Objektklassen oder gar einzelne Objekte möglich. Ein Drill-down beispielsweise ersetzt ein oder mehrere Objekte mit einer genaueren Repräsentation. So können z. B. auf einer Stadtkarte die öffentlichen Gebäude detailliert dargestellt sein, während die restlichen Gebäude stärker generalisiert sind. In Abbildung 2.6 ist dies in einem Beispiel visualisiert. In der rechten Abbildung stammt ein Häuserblock aus einer höher aufgelösten Massstabsebene, während die Umgebung schlechter aufgelöst dargestellt wird. Diese Darstellung führt zu einer deutlich besseren Übersicht bei der Navigation, wo die unmittelbare Umgebung eines *Point Of Interest* (POI) genauer dargestellt wird als der Rest der Karte.

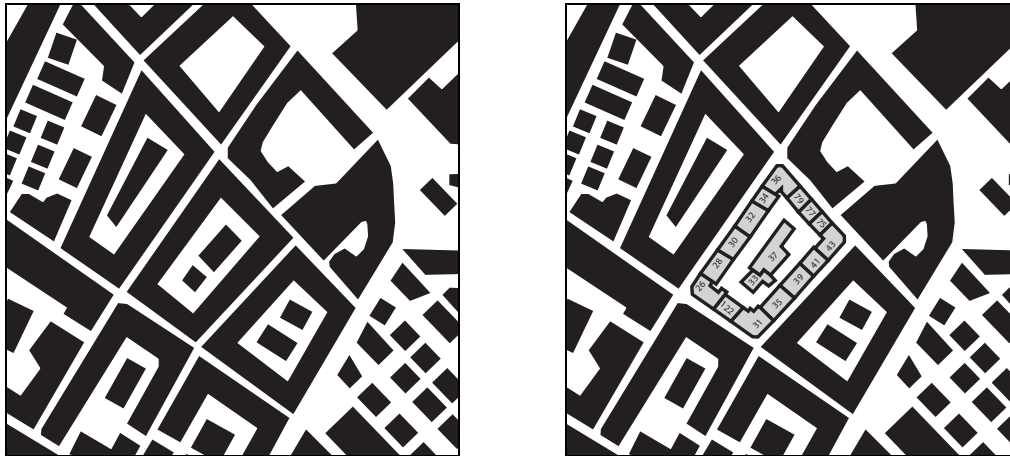


Abbildung 2.6: Beispiel eines Drill-downs. Niedrige Auflösung: Aus der Landeskarte 1:25'000. Hohe Auflösung: Aus dem Übersichtsplan des Kantons Zürich<sup>1</sup>.

Aber auch die Interpolation zwischen Massstabsebenen zur Herleitung einer neuen Geometrie ist möglich: Bei heutigen Internet-Kartendiensten stammt der Inhalt einer Karte jeweils von einer einzigen Massstabsebene pro Zoomstufe. Entsprechend ist das Vergrössern/Verkleinern auf die gespeicherten Massstabsebenen beschränkt, zwischen denen die Darstellung abrupt ändert. Es wurden Methoden entwickelt, um die Kartengeometrie zwischen zwei Massstabsebenen zu interpolieren (Cecconi 2003) und so stufenlose Vergrösserung (*Adaptive Zooming*) innerhalb der Grenzen der gespeicherten Massstabsebenen zu erlauben. Dazu müssen die Massstabsebenen in einer MRDB integriert sein.

Abbildung 2.7a zeigt das Prinzip von Adaptive Zooming für Liniendaten. Vorausgesetzt wird, dass beide Linien dieselbe Anzahl Knoten haben ( $m = n$ ). Über folgende Gleichung können die Knoten der Zwischenlinie berechnet werden:

$$P_j^m = \Delta P, \text{ wobei } \Delta P = |P_j - P'_j| \times s_f \quad (2.1)$$

$s_f$  ist der Skalierungsfaktor, mit  $s_f = 0$  für die Massstabsebene  $s_{25}$  und  $s_f = 1$  für die Massstabsebene  $s_{200}$ .

### 2.3.2 Inkrementelle Generalisierung

Nationale kartographische Ämter sind mit dem Aufbau und der Nachführung der topographischen Landeskarten beauftragt. Das Landeskartenwerk der Schweiz umfasst 6 Massstabsebenen von 1:25'000 bis zu 1:1'000'000. Die Landschaft ist einem stetigen Wandel unterworfen – so rechnet man damit, dass sich 10% der Kartenobjekte pro Jahr ändern (Badard 2000:19).

Die Änderungen müssen zur Zeit für jeden Massstab separat identifiziert und ausgeführt werden – was nicht nur teuer ist, sondern auch die Gefahr der Inkonsistenzen zwischen den Massstabsebenen schafft, denn nicht alle sind immer auf dem gleichen Aktualitätsstand.

1. <http://www.gis.zh.ch/>, Stand 20.12.2005

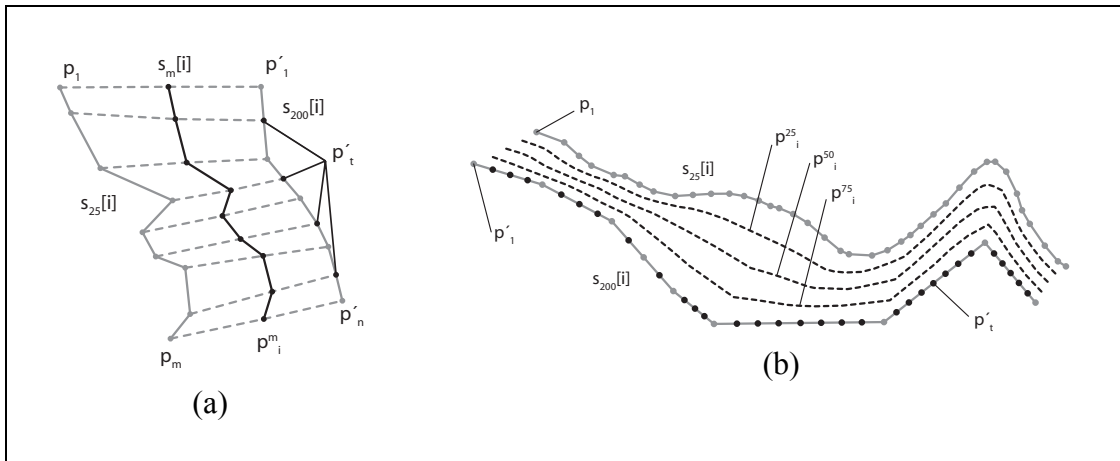


Abbildung 2.7: (a) Prinzip von Adaptive Zooming (Cecconi 2003:110). (b) Beispiel für eine Interpolation zwischen zwei in der Datenbank gespeicherten Massstabsebenen  $s_{25}$  und  $s_{200}$  (Cecconi 2003:113).

Idealerweise sollen deshalb die Nachführungen nur noch im grössten Massstab (dem Basismassstab) vorgenommen werden. Die kleineren Massstäbe sollen automatisch abgeleitet werden. Dazu sind zwei verschiedene Ansätze möglich:

- **Prozessorientiert:** Die kleineren Massstäbe werden jeweils komplett durch automatische Generalisierung vom Basismassstab abgeleitet.
- **Repräsentationsorientiert:** Die bestehenden Massstabsebenen werden in eine MRDB integriert. Änderungen werden nur am Basismassstab vorgenommen und automatisch in die kleineren Massstäbe übertragen. Kilpeläinen hat dafür den Begriff inkrementelle Generalisierung geprägt (Kilpeläinen und Sarjakoski 1995).

Zur Zeit ist eine vollständig automatische Generalisierung mangels geeigneter Technologien noch nicht durchführbar. Zudem sind die vorhandenen topographischen Karten wissensreiche Produkte mit hervorragender Qualität. Es ist deshalb sinnvoll, den repräsentationsorientierten Ansatz zu verfolgen und die bestehenden Datenbestände topographischer Karten in eine MRDB einzubinden (Weibel 1997; Meng und Töllner 2004).

Derzeit bestehen in einigen kartographischen Ämtern Projekte, um die digitalen Geodatenbestände in MRDBs zu überführen (Kreiter 2003). Ein Prototyp für ein solches System wird von Harrie und Hellström (1999) beschrieben und zeigt die Machbarkeit der inkrementellen Generalisierung.

### 2.3.3 Fahrzeugnavigation

Die menschliche Wegfindung ist von Natur aus ein hierarchischer Prozess, bei welchem verschieden detaillierte Ebenen des Strassennetzes involviert sind. Timpf und Kuhn (2003) identifizieren drei verschiedene Ebenen, in denen die Fahrzeugnavigation auf Autobahnen auf stattfindet: Die Ebene der *Planung* beruht nur auf einem grobmaschigen Strassennetz; ein Plan besteht aus einer Liste von Strassen und Orten, die nacheinander befahren werden müssen, um vom Start zum Ziel zu gelangen. Auf der Ebene der *Instruktionen* wird das Strassennetz genauer repräsentiert; die Liste der Instruktionen enthält Informationen, wo Autobahnen befahren, ge-

wechselt oder verlassen werden müssen. Die Ebene des eigentlichen *Fahrvorgangs* ist die detaillierteste; hier werden Entscheidungen zu einzelnen Fahrspurwechseln getroffen. Für jede Ebene wird ein entsprechend detailliertes Strassennetz benötigt. Diese müssen zudem miteinander verknüpft sein, damit Anweisungen zwischen den Ebenen ausgetauscht werden können.

Auch die kartographische Visualisierung eines Fahrzeugnavigationssystems soll flexibel sein: In Ortschaften soll das Strassennetz detailliert dargestellt werden, während auf längeren, geraden Strecken oder auf Autobahnen eine kleinmassstäbigere Darstellung mehr Übersicht gibt.

### **2.3.4 Verbesserung/Überprüfung der Datenqualität**

Es gibt oft räumliche Datensätze mit ähnlichem Inhalt, die aber für unterschiedliche Zwecke oder von unterschiedlichen Organisationen erfasst wurden. Wenn sie in eine MRDB integriert werden, können Inkohärenzen entdeckt und beseitigt werden.

Eine solche Anwendung beschreiben Rosen und Saalfeld (1985) bzw. Saalfeld (1988). Sie kombinierten Datensätze des Bureau of the Census mit den topographischen Daten des United States Geological Survey (USGS). Ziel war es, beide ursprüngliche Karten zu verbessern und Fehler darin aufzudecken. So sollten Attribute wie Strassenarten, Strassennamen und Hausnummern in die USGS-Karten übernommen werden (Walter 1996:60).

Bard (2004) schliesslich beschreibt eine Methode, um die Generalisierungsqualität zu überprüfen. Der ungeneralisierte und der generalisierte Datensatz werden dazu verknüpft und die Objekte werden anhand verschiedener Masszahlen verglichen.

## Kapitel 3

# Matching von räumlichen Objekten

In Kapitel 2 wurde aufgezeigt, wie verschiedene Repräsentationen zu einem Gesamtschema integriert werden können. Für eine Integration müssen jedoch auch die eigentlichen Objekt-Instanzen einander zugeordnet werden. Dies bedeutet, dass homologe Objekte erkannt und miteinander verknüpft werden müssen. Wegen der Grösse der Datensätze kommen für dieses *Matching* nur automatische Methoden in Frage. In diesem Kapitel soll ein methodischer Rahmen für automatisches Matching geschaffen werden. Für Strassendaten lässt sich in der Literatur bereits eine Anzahl von Algorithmen finden. Die wichtigsten davon werden kurz zusammengefasst.

## 3.1 Ein methodischer Rahmen für Matching-Prozesse

### 3.1.1 Begriffserläuterung

Häufig werden Verknüpfungen temporär angelegt, um aus zwei Datensätzen einen dritten Datensatz von besserer Qualität abzuleiten (z. B. indem Attributdaten des einen Datensatzes in den anderen, geometrisch genaueren übernommen werden). Für diese Anwendung ist auch der Begriff *Map Conflation* geläufig (Yuan und Tao 1999:1).

Werden beim Matching nur Eigenschaften der Objekte selbst (Geometrie und Attributdaten) verwendet, spricht man von *feature based matching*. Werden die Beziehungen zwischen der Objekten (Topologie) einbezogen, spricht man von *relational matching*. Meistens werden sowohl Geometrie, Semantik als auch Topologie in einem Matching-Ansatz kombiniert.

### 3.1.2 Übersicht über den Matching-Ablauf

In der Literatur lässt sich eine grosse Anzahl von Ansätzen finden, die jeweils auf ganz bestimmte Datensätze abgestimmt sind. So verschieden diese Ansätze auch sind, so haben sie doch gewisse Gemeinsamkeiten. Konzeptuell kann das Matching in die drei Phasen Vorverarbeitung und geometrische Entzerrung – Matching – Nachbearbeitung gegliedert werden (Abbildung 3.1). Je nach Ansatz und Charakteristik der zu integrierenden Datensätze sind die Phasen verschieden ausgeprägt. Das folgende Schema lehnt sich an Walter und Fritsch (1999) an, ähnliche Schemas sind auch bei Stadler (2004) und Yuan und Tao (1999) zu finden:

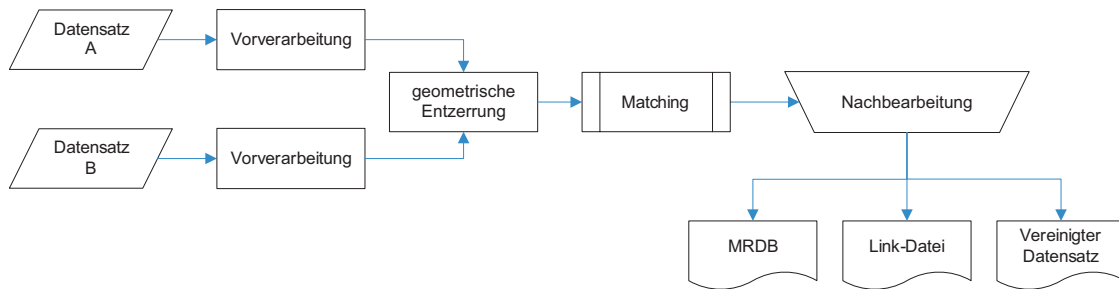


Abbildung 3.1: Matching zweier Datensätze.

### Vorverarbeitung und geometrische Entzerrung

In der Vorverarbeitung sollen die beiden zu integrierenden Datensätze aneinander angepasst werden, damit der eigentliche Matching-Prozess einfacher wird. Dieser Schritt ist somit stark von der Beschaffenheit der beiden Datensätze abhängig oder kann auch ganz wegfallen.

Falls die beiden Datensätze in unterschiedlichem Massstab vorliegen, so wird vorgeschlagen, den grösseren Massstab in der Vorverarbeitung zu generalisieren (Sester et al. 1998:354). Mögliche Schritte einer solchen Vorgeneralisierung wären beispielsweise eine Zusammenfassung von Flächen des grösseren Massstabs, damit sie mit den Flächen des kleineren Massstabs vergleichbar sind (Sester et al. 1998), ein Kollaps von flächenhaft repräsentierter Strassen zu ihren Mittellinien (Petzold et al. 2005) oder von mehrspurig repräsentierten Strassen zu den Mittellinien (Thom 2005).

Mit der *geometrischen Entzerrung* werden globale geometrische Lageunterschiede zwischen den beiden zu verknüpfenden Datensätzen beseitigt. Sie treten durch die Verwendung unterschiedlicher Referenzsysteme oder aufgrund systematischer Fehler bei der Erfassung auf. Die Datensätze lassen sich durch geeignete geometrische Transformationen aneinander angleichen. Die dazu benötigten Kontrollpunkte (engl. *ground control points*) können manuell oder automatisch bestimmt werden (Walter und Fritsch 1999:450).

### Matching

In der eigentlichen Matching-Phase werden die Objekte der zu integrierenden Datensätze aufgrund ihrer geometrischen, semantischen und topologischen Eigenschaften verglichen. Homologe Objekte sollen zuverlässig erkannt und miteinander verknüpft werden. Die Matching-Phase wird in Abschnitt 3.1.3 genauer erläutert.

### Nachbearbeitung

Häufig können mit dem automatischen Matching nicht alle Objekte erfolgreich zugeordnet werden. Möglicherweise sind auch Fehlzugeordnungen gemacht worden (siehe dazu Kapitel 3.1.4). Deshalb muss das Ergebnis des automatischen Teils in der Regel von einem Operator kontrolliert und eventuell ergänzt oder korrigiert werden.

Die Verknüpfungen können auf verschiedene Weise verwertet werden. Möglich ist die Ausgabe eines angereicherten oder fusionierten Datensatzes, welcher Eigenschaften von beiden Quelldatensätzen enthält. Die Verknüpfungen selbst können direkt in einer MRDB gespeichert werden oder zur externen Bearbeitung als sog. Link-File ausgegeben werden.

### 3.1.3 Matching-Teilprozesse

Die eigentliche Matching-Phase kann wiederum in Teilprozesse gegliedert werden:

- A. Suche von potentiellen Matching-Kandidaten
- B. Bilden von Ähnlichkeitmassen
- C. Anwenden von Beschränkungen (Constraints)
- D. Evaluation der Matching-Kandidaten
- E. Benutzerinteraktion

#### A. Suche von potentiellen Matching-Kandidaten

Meistens wird einer der beiden Datensätze als Referenzdatensatz festgelegt. Bei unterschiedlichen Massstäben ist dies derjenige mit dem grösseren Massstab. Zu jedem Objekt aus dem Referenzdatensatz wird eine Menge von Objekten aus dem Vergleichsdatensatz gebildet, die alle potentielle Matching-Kandidaten sind.

Bei Liniendaten wie Strassennetzen kann dies geometrisch geschehen, indem um jedes Strassen-segment im Referenzdatensatz ein Puffer gebildet wird und alle Strassen-segmente des Vergleichsdatensatzes in diesem Puffer als Kandidaten gelten. Mit einfachen Puffer-Operationen können nur 1:N-Beziehungen gebildet werden. Für die Bildung von N:M-Beziehungen werden auch die Objekte im Referenzdatensatz zu einem *virtuellen Referenzobjekt* aggregiert. Für Strassen sind dazu *Buffer Growing-Algorithmen* (Walter 1996:63–64) am weitesten verbreitet.

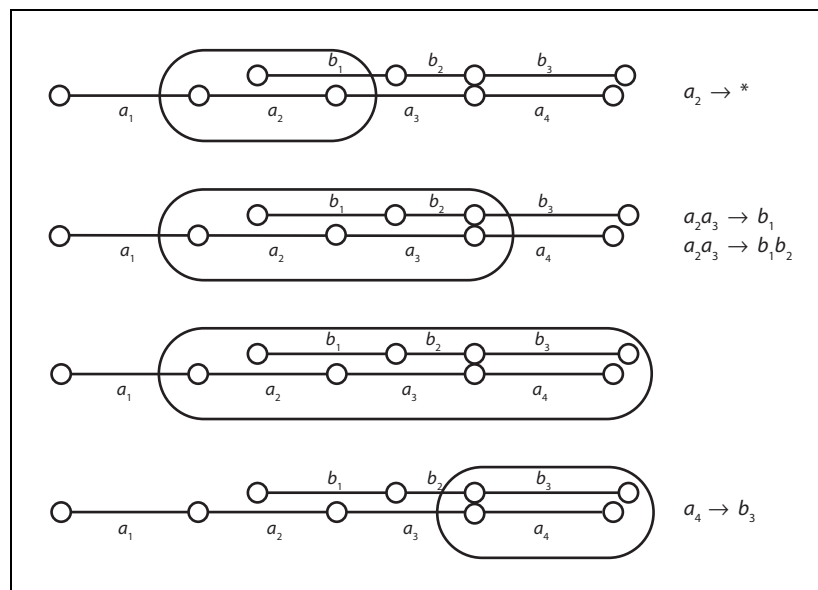


Abbildung 3.2: Buffer Growing-Algorithmus (Walter 1996:64).

In Abbildung 3.2 ist der *Buffer Growing-Algorithmus* illustriert. Linienzug  $A$  ist im Referenzdatensatz, Linienzug  $B$  im Vergleichsdatensatz. Im ersten Schritt liegt kein Segment aus dem Vergleichsdatensatz vollständig im Puffer um die Referenzlinie  $a_2$ . Der Puffer wird deshalb im zweiten Schritt erweitert. Aus den Segmenten  $a_2 a_3$  wird ein neues, virtuelles Referenzobjekt. Nun liegen die Segmente  $b_1 b_2$  vollständig im Puffer um  $a_2 a_3$ . Es kann somit ein N:M-Kandida-

tenpaar gebildet werden. Die Aggregation  $a_2a_3a_4 \rightarrow b_1b_2b_3$  wird indes verworfen, da sie bereits in den zwei kürzeren Zuweisungen  $a_2a_3 \rightarrow b_1b_2$  und  $a_4 \rightarrow b_3$  enthalten ist.

Bei flächenhaften Daten wie Gebäuden oder geologischen Karten wird sehr oft der Prozentsatz der gegenseitigen Überschneidung als Kriterium für die Kandidatenbildung verwendet.

Ein indirektes Kriterium, das bei linienhaften und flächenhaften Daten angewendet wird, ist der Grad der Überschneidung der minimalen umschliessenden Rechtecke (*minimum bounding box*).

## **B. Bilden von Ähnlichkeitsmassen**

Die Kandidatenmenge eines Referenzobjekts kann nach Schritt A inkorrekte Objekte enthalten, welche herausgefiltert werden müssen. Dazu werden für jedes Kandidatenobjekt bestimmte Ähnlichkeitsmasse berechnet. Ähnlichkeitsmasse werden im Abschnitt 3.2 im Detail behandelt.

## **C. Anwenden von Beschränkungen (Constraints)**

Beschränkungen sind Schwellenwerte für Ähnlichkeitsmasse, die nicht überschritten werden dürfen. Beispiele für Beschränkungen sind:

- Der Winkelunterschied zwischen den Basislinien von Referenz- und Kandidatenlinie darf nicht  $> 30^\circ$  sein.
- Die Längendifferenz zwischen Referenz- und Kandidatenlinie darf nicht  $> 30$  m sein.
- Strassen mit dem Attribut „Autobahn“ dürfen nur mit Kandidatenstrassen verknüpft werden, welche ebenfalls das Attribut „Autobahn“ haben.

Die Bestimmung der Schwellenwerte ist problematisch: Sind sie zu gross gewählt, werden zu viele homologe Objekte fälschlicherweise ausgeschieden, sind sie zu klein, sinkt die Rate der automatisch zugeordneten Objekte. Sie können interaktiv durch Kontrolle des Zuordnungsergebnisses bestimmt oder durch statistische Auswertung aus manuell zugeordneten Testgebieten gewonnen werden. Alle Kandidatenobjekte, bei welchen eine oder mehrere Beschränkungen verletzt sind, werden aus den Kandidatenmengen entfernt.

## **D. Evaluation der Matching-Kandidaten**

Selbst nach der Anwendung der Beschränkungen haben Referenzobjekte in der Regel noch mehrere Kandidatenobjekte. In diesem Schritt geht es darum, aus den Kandidatenobjekten dasjenige zu finden, das mit der grössten Wahrscheinlichkeit dasselbe Realwelt-Phänomen darstellt. Dazu werden die Ähnlichkeitsmasse aller Kandidaten eines Referenzobjektes miteinander verglichen. Kann kein eindeutiger Sieger ermittelt werden, so muss die Zuordnung auf später verschoben werden – eventuell hat sich die Situation im nächsten Iterationsschritt vereinfacht, oder der Benutzer muss interaktiv den richtigen Kandidaten auswählen.

## **E. Benutzerinteraktion**

Konnten nicht alle Zuordnungen eindeutig gelöst werden, so wird der Benutzer aufgefordert, eine oder mehrere korrekte Zuordnungen zu machen.

Die Schritte D und E laufen iterativ so lange, bis alle Objekte zugeordnet werden konnten oder keine neuen Zuordnungen mehr erstellt werden können.

### 3.1.4 Zuordnungsfehler

Es können drei Arten von Zuordnungsfehlern auftreten (Saalfeld 1988:223):

- Ein Objekt kann einem korrespondierenden Objekten zugeordnet werden, obwohl es eigentlich keinen Partner hätte (*false positive*)
- Es kann keinem Partner zugeordnet werden, obwohl es eigentlich einen hätte (*false negative*)
- Ein Objekt kann dem falschen Partner zugeordnet werden (*mismatch*).

*False negatives* und *mismatches* sind dabei kritische Fehler, weil sie das korrekte Zuordnen von anderen Objekten verhindern.

## 3.2 Ähnlichkeitsmasse

Zentral beim Matching ist der Begriff der Ähnlichkeit zweier Objekte. Ähnlichkeitsmasse dienen als Beschränkungen und zum Vergleich verschiedener potenzieller Matching-Partner. Auch bei der Bildung von Matching-Kandidatenmengen können einfache Ähnlichkeitsmasse zum Einsatz kommen.

Ähnlichkeit und Distanz stehen in komplementärer Beziehung – oft wird der eine Begriff anhand des anderen definiert (Samal et al. 2004:463):

$$\sigma(A, B) = 1 - \frac{\Delta(A, B)}{U} \quad (3.1)$$

$\sigma$  ist die Ähnlichkeitsfunktion,  $A$  und  $B$  sind Objekte bzw. Eigenschaften davon,  $\Delta$  ist ein Distanzmass und  $U$  ein Normalisierungsfaktor, welcher das Distanzmass auf ein Intervall zwischen 0 und 1 begrenzt.  $U$  wird bestimmt, indem für das Distanzmass eine Beschränkung eingeführt wird (Punkt C in Abschnitt 3.1).  $U$  ist dann gleich dem Schwellwert dieser Beschränkung.

Ähnlichkeitsmasse können in semantische, geometrische und kontextabhängige Masse unterteilt werden.

### 3.2.1 Semantische Ähnlichkeitsmasse

Merkmale von Attributen können von verschiedener Skala sein (Longley et al. 2001:66). Je nachdem sind verschiedene Arten von Ähnlichkeitsmassen sinnvoll.

- *Nominalskala*: Attribute dieses Typs haben nur die Funktion, Entitäten zu identifizieren oder zu unterscheiden. Beispiele sind Strassennamen, Autonummern oder auch Farben. Arithmetische Operationen machen bei nominalskaligen Attributen keinen Sinn.
- *Ordinalskala*: Es existiert eine Rangordnung zwischen den Merkmalsausprägungen, so dass die Relationen  $<$  und  $>$  Sinn machen. Die Abstände zwischen den Rängen sind jedoch nicht festgelegt. Ein Beispiel ist die Einteilung von Böden in die verschiedenen Bodengüteklassen „ärmste Böden“, „geringe Böden“, „mittlere Böden“ und „gute Böden“ und „sehr gute Böden“.
- *Intervallskala*: Die Abstände zwischen den Merkmalsausprägungen lassen sich exakt bestimmen. Die Operationen  $+$  und  $-$  machen deshalb Sinn. Es existiert aber kein natürlicher

Nullpunkt. Beispiel: die Temperatur in Grad Celsius, da der Nullpunkt willkürlich (als Schmelzpunkt von Eis) festgelegt wurde.

- *Verhältnis- oder Ratioskala*: Es existiert ein natürlicher Nullpunkt in der Skala, so dass Verhältnisse gebildet werden können. Beispiele: das Gewicht oder die Länge einer Linie.

### Attribute in Nominal- und Ordinalskala

Hier muss die Behandlung von kategorialen Attributen unterschieden werden von der Behandlung von Zeichenketten, welche Namen bezeichnen.

Cobb et al. (1998:28) beschreiben in ihrem Text eine Methode, um die Ähnlichkeit von kategorialen Attributen zu erfassen. Das Attribut „Stromquelle“ eines Eisenbahn-Datensatzes kann einen der Werte {0 – unbekannt, 1 – elektrifiziert, 3 – elektrifiziert durch eine Oberleitung, 4 – nicht elektrifiziert, 999 – sonstig} annehmen. Semantisch sind sich die beiden Begriffe „elektrifiziert“ und „elektrifiziert durch eine Oberleitung“ ähnlicher als beispielsweise die beiden Begriffe „elektrifiziert“ und „nicht-elektrifiziert“. Man würde deshalb der erstgenannten Kombination einen höheren Ähnlichkeitskoeffizienten zuweisen. Für jede mögliche Kombination der Attribute kann so ein Ähnlichkeitskoeffizient bestimmt und in eine Tabelle eingetragen werden (Tabelle 3.1). Da in diesem Fall die Beziehung symmetrisch ist, muss nur die eine Hälfte der Tabelle ausgefüllt werden. Man beachte, dass die Ähnlichkeit von „0 – unbekannt“ zu „0 – unbekannt“ nicht gleich eins ist, obwohl in diesem Fall beide Attribute denselben Wert haben. „unbekannt“ bedeutet, dass keine Information über das Attribut vorliegt. Wenn bei zwei Objekten aus verschiedenen Datensätzen das Attribut nicht bestimmt ist, bedeutet dies natürlich nicht, dass es sich um homologe Objekte handelt.

Stromquelle	0	1	3	4	999
0	0.2				
1	0.2	1			
3	0.2	0.6	1		
4	0.2	0.1	0.1	1	
999	0.2	0.2	0.2	0.2	0.2

Tabelle 3.1: Ähnlichkeitstabelle für das Attribut Stromquelle (Cobb et al. 1998:28).

Falls es mehrere solche linguistische Attribute gibt, können sie miteinander in Beziehung stehen. Zur Verknüpfung von mehreren solchen Attributen zu einem Gesamtmaß verwenden Cobb et al. (1998) Methoden aus der *Fuzzy Logic*.

Zeichenketten (*Strings*) als Attribute können wie im besprochenen Fall Kategorien bezeichnen, sie können aber auch Namen (Strassennamen, Gebäudennamen etc.) sein. Namen sind oft mehrdeutig oder können Schreibfehler beinhalten (siehe Tabelle 3.2). Solche Anomalien sollten durch das Ähnlichkeitsmaß erfasst werden.

Methoden hierzu sind aus dem Matching von Objekten in nicht-räumlichen Datenbanken bekannt. Die folgende Methode stammt aus (Samal et al. 2004). Zuerst werden die beiden Strings

Fehlertyp	Beispiel 1	Beispiel 2
Weglassen von Wörtern	Abraham Lincoln Memorial	Lincoln Memorial
Ersetzen von Wörtern	Reagan National Airport	Washington National Airport
Andere Wortstellung	National Art Gallery	National Gallery of Art
Abkürzungen	National Archives	Nat'l Archives
Auslassen von Buchstaben	Washingtn Monument	Washington Monument
Ersetzen von Buchstaben	Frear Gallery	Freer Gallery

Tabelle 3.2: String-Anomalien, die beim Matching berücksichtigt werden müssen (Samal et al. 2004:468).

in einzelne Wörter (*Tokens*) zerlegt. Mit einem Lexikon, welches Synonyme und Abkürzungen enthält, können diese Tokens in ein standardisiertes Format gebracht werden. Aus den beiden Strings  $S_1 = \text{„The Portrait Gallery (Smithsonian)“}$  und  $S_2 = \text{„National Gallery of Portraits“}$  können zum Beispiel die Wortmengen  $T_1 = \{\text{„Portrait“}, \text{„Gallery“}, \text{„Smithsonian“}\}$  respektive  $T_2 = \{\text{„National“}, \text{„Gallery“}, \text{„Portrait“}\}$  abgeleitet werden. Die Tokens können nun mit der *Damerau-Levenshtein-Metrik* (auch als *Edit-Distanz* bekannt) verglichen werden. Sie zählt die Anzahl Buchstaben, die gelöscht, hinzugefügt oder ausgetauscht werden müssen, um einen String in einen anderen zu transformieren. Die Damerau-Levenshtein-Metrik kann zwischen allen Tokens nach folgender Formel berechnet und in eine Wort-Wort-Matrix eingetragen werden:

$$\Lambda_{i,j} = \frac{DamLev(i,j)}{Max(Length(i), Length(j))} \quad (3.2)$$

$DamLev(i,j)$  ist die Damerau-Levenshtein-Metrik zwischen Token  $i$  und Token  $j$ ,  $Length(i)$  und  $Length(j)$  bezeichnen die Anzahl Buchstaben des jeweiligen Tokens. Um die Gesamtähnlichkeit zu finden, werden die Distanzen der optimalen Zuweisungen summiert und normiert:

$$\sigma(S_1, S_2) = \frac{\sum_{k=1}^{l_{max}} \Lambda_{optimal}(k)}{\mu} \quad (3.3)$$

$l_{max}$  ist die Kardinalität der grösseren Wortmenge und  $\mu$  das Mittel der Kardinalitäten beider Wortmengen.

### Attribute in Intervall- und Ratioskala

Die Ähnlichkeit von zwei Merkmalen in Intervall- oder Ratioskala kann wie folgt berechnet werden:

$$\sigma(A, B) = 1 - \frac{|a - b|}{|U|} \quad (3.4)$$

$U$  bezeichnet den Wertebereich. Diese Art der Berechnung kommt nicht nur bei Attributen, sondern auch bei geometrischen Ähnlichkeitsmassen häufig vor.

### 3.2.2 Geometrische Ähnlichkeitsmasse

Diese Masse haben ihren Ursprung oft in der Überprüfung der Digitalisiergenauigkeit oder der Effekte der Generalisierung. Dabei soll die Ähnlichkeit eines digitalisierten bzw. generalisierten Datensatzes mit einem als exakt angenommenen Referenzdatensatz verglichen werden. Sie eignen sich aber auch für das Matching. Distanzen können wie besprochen in Ähnlichkeitsmasse umgerechnet werden.

Die geometrische Ausprägung eines Objekts kann punktförmig, linienhaft, oder flächenhaft sein. Im Folgenden sollen Vergleichsmasse, welche zum Matching verwendet werden können, in dieser Klassifikation besprochen werden. Zur Vereinfachung wird angenommen, dass ein planares Koordinatensystem vorliegt und die euklidische Metrik gilt.

#### Ähnlichkeitsmasse für punktförmige Geometrien

Bei punktförmigen Geometrien liegt am wenigsten Information vor. Entsprechend gibt es hier nur ein Mass, nämlich die *Lageähnlichkeit*:

$$\sigma(P_1, P_2) = 1 - \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{U} \quad (3.5)$$

#### Ähnlichkeitsmasse für linienförmige Geometrien

Es gibt mehrere Möglichkeiten, die *Distanz* zwischen zwei Linien zu bestimmen. Ein solches Mass ist die Fläche zwischen den Linien, dividiert durch die Linienlänge.

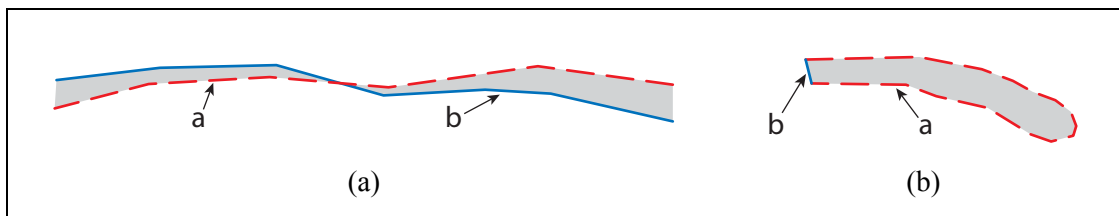


Abbildung 3.3: (a) Mittlere Zwischenfläche als Distanzmass zwischen Linien. (b) Fall, wo die mittlere Zwischenfläche versagt (Devogele 1997:39).

Die so normierte Zwischenfläche darf jedoch nicht alleine verwendet werden, sondern muss immer mit einer Maximaldistanz verknüpft werden, da es Fälle gibt, wo sie einen falschen Eindruck vermittelt. So ist sie in Abbildung 3.3b klein, visuell sind die beiden Linien a und b jedoch sehr verschieden.

Ein weiteres Mass wurde von Goodchild (1997) vorgestellt. Um die Referenzlinie wird ein Puffer der Breite  $x$  gebildet. Es wird gemessen, wieviel Prozent der Vergleichslinie innerhalb dieses Puffers liegt. Das Mass ist nun die Größe  $x$  dieses Puffers, die benötigt wird, damit ein bestimmtes Quantil der Vergleichslinie (z.B. 90% oder 95%) innerhalb des Puffers liegen. Die Puffergröße kann durch eine iterative Prozedur bestimmt werden.

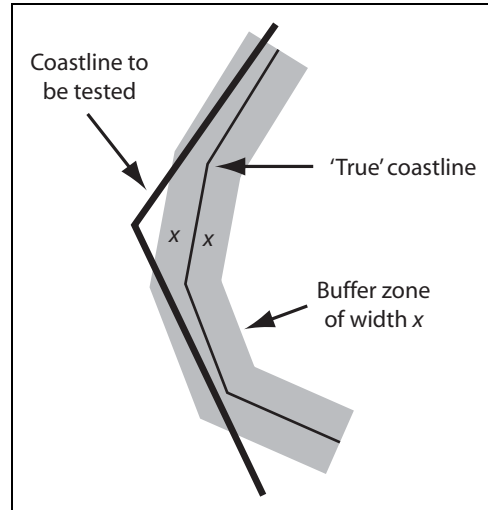


Abbildung 3.4: Ein Puffer der Breite  $x$  um die „echte“ Küstenlinie wird mit der Vergleichslinie verschnitten, um den Prozentsatz zu bestimmen, zu welchem die Vergleichslinie im Puffer liegt (Goodchild 1997:301).

Sowohl die normierte Zwischenfläche als auch das Mass von Goodchild haben den Nachteil, keine Distanzen im mathematischen Sinne zu sein. Insbesondere ist die Bedingung nicht erfüllt, dass die Distanz zwischen zwei Objekten nur dann null ist, wenn die Objekte strikt identisch sind. Das Mass von Goodchild ist beispielsweise auch dann null, wenn die Vergleichslinie eine Untermenge der Referenzlinie ist. Die *Hausdorff-Distanz* ist eine mathematische Distanz und eignet sich deshalb besser zum Matching. Sie ist ein Mass für die maximale Distanz zwischen zwei Linien. Sie ist das Maximum der beiden *Hausdorff-Komponenten* (Abbildung 3.5).

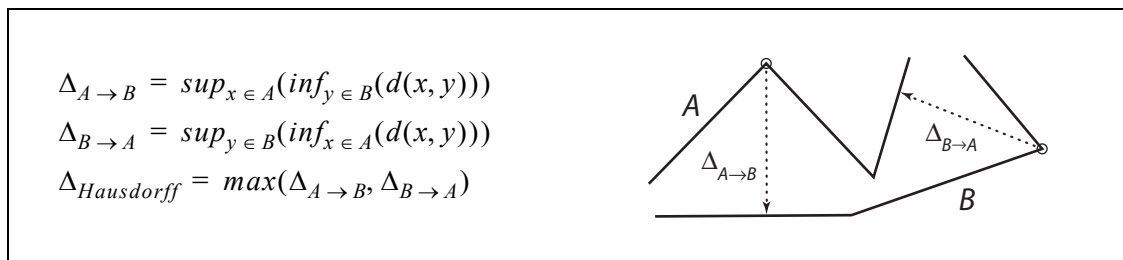


Abbildung 3.5: Berechnung der Hausdorff-Distanz (Hangouët 1995:2).

Die Berechnung der Hausdorff-Distanz kann man sich wie folgt vorstellen (Lemarié und Raynal 1996:407): Man fährt der Linie  $A$  entlang. An jedem Punkt von  $A$  setzt man Kreise mit grösser werdendem Radius, bis ein Kreis die Linie  $B$  berührt. Der Radius dieses Kreises ist die Distanz des aktuellen Punktes zu  $B$ . Das Maximum der Distanzen aller Punkte ist die Hausdorff-Komponente von  $A$  nach  $B$ . Die Komponente von  $B$  nach  $A$  wird analog berechnet.

Wenn die beiden Linien unterschiedlich lang sind, hängt die Hausdorff-Distanz nur von der Distanz der Endpunkte ab (Abbildung 3.6). Sie kann deshalb nicht verwendet werden, um Strassen von verschiedenem Massstab zu verknüpfen. In diesem Fall ist es besser, nur die Hausdorff-Komponente vom kleineren Massstab zum grösseren Massstab  $\Delta_{b_1 \rightarrow a}$  zu verwenden.

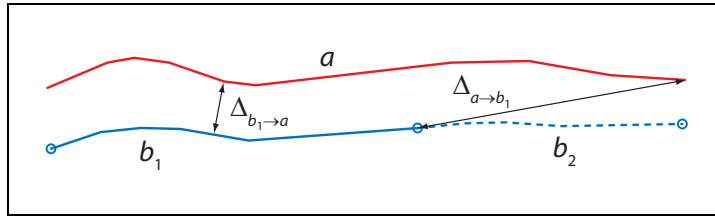


Abbildung 3.6: Hausdorffdistanz bei Linien verschiedener Länge.

Linien können auch durch ihre *Formähnlichkeit* miteinander verglichen werden. McMaster (1986) hat insgesamt 89 Linienmasse definiert. Einige davon, die zum Matching verwendet werden können, sind:

- Das Verhältnis der Linienlängen
- Das Verhältnis Sinuosität beider Linien. Die Sinuosität einer Linie ist definiert als die Länge der Linie geteilt durch die Länge der Basislinie (Abbildung 3.7a).
- Das Verhältnis der Winkel zwischen den beiden Basislinien
- Das Verhältnis der Winkelsummen zwischen den Segmenten (Abbildung 3.7b)

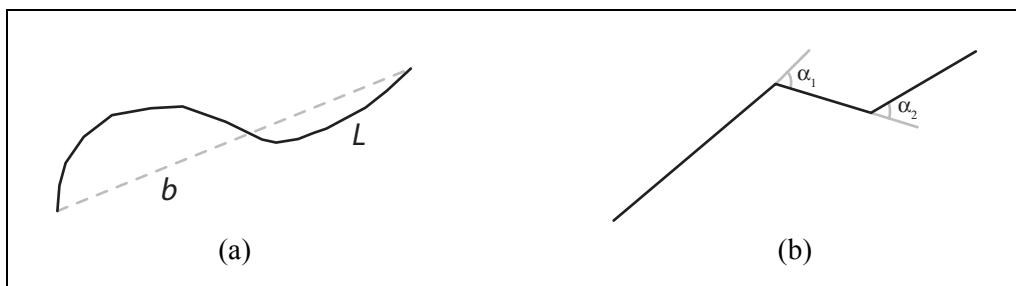


Abbildung 3.7: (a) Basislinie  $b$  einer Linie  $L$ . (b) Zwischenwinkel zwischen Segmenten.

### Ähnlichkeitsmasse für flächenhafte Geometrien

Geeignete Masse, um die Ähnlichkeit zweier Flächen zu erfassen, sind:

- Die gegenseitige Überlagerung:

$$m_1 = \frac{F(A \cap B)}{F(A)}, m_2 = \frac{F(A \cap B)}{F(B)} \quad (3.6)$$

Bard (2004:127) verwendete diese Masse, um Häuser(-blöcke) aus zwei verschiedenen Maßstäben zu verknüpfen. Dazu wurde ein Regelwerk verwendet, das angibt, bei welchen Schwellwerten zwei Flächen verknüpft werden sollen. Es ist auch denkbar, die beiden Masse zu mitteln um daraus ein symmetrisches Ähnlichkeitsmass abzuleiten.

- Die Assoziationswahrscheinlichkeit (Badard 2000:45):

$$\sigma_{faw}(A, B) = \frac{F(A \cap B)}{\min(F(A), F(B))} \quad (3.7)$$

- Die Flächendistanz (Badard 2000:46):

$$\sigma_{ffd}(A, B) = \frac{F(A\Delta B)}{F(A \cup B)} \quad (3.8)$$

wobei  $D$  die symmetrische Differenz ist:  $\Delta = A|B + B|A$

Daneben können auch Formmasse verglichen werden. Von diesen gibt es ebenso wie für Linien eine grosse Anzahl. Für das Matching eignen sich speziell (Devogele 1997:41):

- Verhältnis der Flächen
- Verhältnis des Umfangs der Flächen
- Verhältnis der Exzentrizitäten (Hauptachse / Nebenachse)
- Verhältnis der maximalen Distanzen zum Zentroid
- Verhältnis der Kompaktheit der Flächen. Die Kompaktheit kann definiert werden als

$$C = \frac{P}{\sqrt{F}} \quad \text{mit } P = \text{Umfang, } F = \text{Flächeninhalt.} \quad (3.9)$$

- Verhältnis des Forman-Godron-Formindex (Meng 2000:31) beider Flächen:

$$FI = \frac{P}{2 \times \sqrt{\pi \times F}} \quad \text{mit } P = \text{Umfang, } F = \text{Flächeninhalt} \quad (3.10)$$

### 3.2.3 Kontextabhängige Ähnlichkeitsmasse

Die bisher vorgestellten semantischen und geometrischen Ähnlichkeitsmasse vergleichen einzelne Objekte unabhängig von ihrer Umgebung. Durch das Betrachten des Kontexts kann wertvolle Zusatzinformation gewonnen werden. So wertvoll diese Information ist, so schwierig ist es, sie formal zu erfassen. In bestehenden Matchingansätzen fliesst der Kontext auf ganz unterschiedliche Weise mit ein. Anstatt einer Kategorisierung werden hier deshalb einige Beispiele aufgeführt:

- Das Strassen- und Gewässernetz bildet eine Partitionierung der Landkarte, in welche sich die anderen Objekte einpassen (Timpf 1998). Auch nach einer Generalisierung soll diese Ordnung erhalten bleiben, insbesondere sollten keine Gebäude ihre relative Lage zu Strassen ändern. Bei Ansätzen zur Zuordnung von Gebäuden aus verschiedenen Massstäben kann dies ausgenutzt werden (Stadler 2004): In einem ersten Schritt werden zuerst die Siedlungsblöcke einander zugeordnet. Der Suchbereich für Gebäude beschränkt sich dann auf diejenigen, welche innerhalb des korrespondierenden Siedlungsblockes des anderen Massstabes liegen.
- Samal et al. (2004) stellen einen Ansatz vor, um *Landmarks*, repräsentiert als Punktobjekte, zu verknüpfen. Zur Ergänzung der Positionsinformation wird für jedes zu verknüpfende Objekt ein *Nachbarschaftsgraph* (proximity graph) gebildet (Abbildung 3.8). Er besteht aus Kanten, die ein Objekt mit allen anderen Objekten seiner Repräsentation, die innerhalb einer bestimmten Distanz liegen, verbinden. Aus den Nachbarschaftsgraphen zweier potentieller Matchingpartner können Verschiebungsvektoren abgeleitet werden (Abbildung 3.9). Die Grösse der Summe dieser Verschiebungsvektoren ist ein Mass für die Ähnlichkeit der beiden Nachbarschaftsgraphen.



Abbildung 3.8: Der Nachbarschaftsgraph für das National Museum of Natural History (Samal et al. 2004:473).

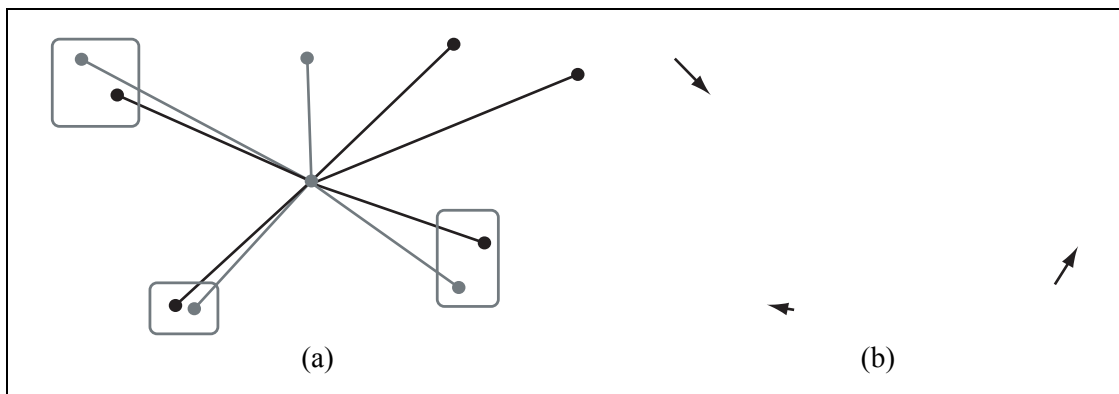


Abbildung 3.9: (a) Matching zweier Nachbarschaftsgraphen (b) Daraus abgeleitete Verschiebungsvektoren (Samal et al. 2004:474).

- Beim Matching von Strassen reicht die alleinige Betrachtung der Liniengeometrien oft nicht aus, um eine korrekte Zuordnung zu treffen. Dies trifft insbesondere zu, wenn die Datensätze einen unterschiedlichen Massstab haben. In Abbildung 3.10 würde aufgrund der geometrischen Ähnlichkeit die Referenzstrasse ( $O'O$ ) zur Kandidatenstrasse ( $N-2/2$ ) zugeordnet werden. Offensichtlich ist dies falsch, denn die richtige Kandidatenstrasse wäre ( $N-1/2$ ). Dieser Fehler kann vermieden werden, wenn zusätzlich zur Geometrie der Knoten-grad der Kreuzungen betrachtet wird: Der richtige Kandidat für  $O'$  muss ebenfalls Grad 3 haben.

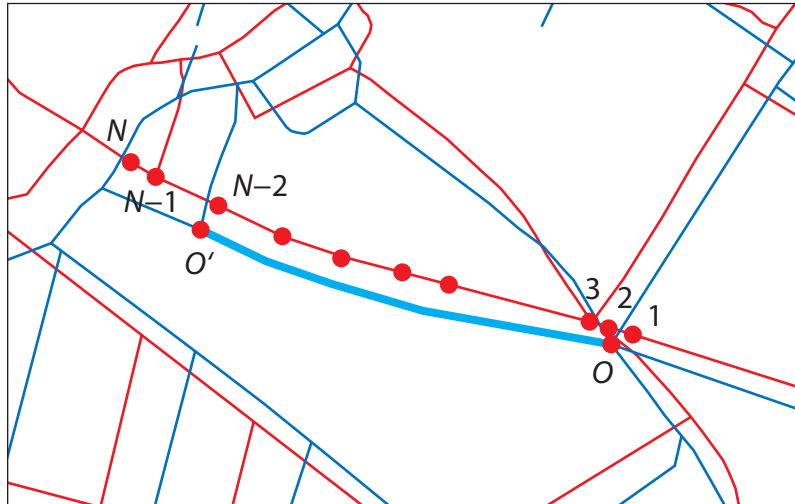


Abbildung 3.10: Beispiel für den Gebrauch des Knotengrades im Strassenmatching. Cyan: Referenzlinie. Rot: Kandidatenstrassen (Zhang et al. 2005:4).

### 3.2.4 Verknüpfung von Einzelmassen zu einem Gesamtmasse

Werden verschiedene der aufgeführten Ähnlichkeitsmasse verwendet, müssen sie zu einem Gesamtmasse verknüpft werden. Eine Möglichkeit ist, ein Regelwerk aus verschiedenen Schwellwerten für die Ähnlichkeitsmasse zu verwenden (Bard 2004:127).

Eine zweite Möglichkeit ist die gewichtete Mittelwertbildung. Das Mittel kann geometrisch (Dunkars 2003) oder arithmetisch gebildet werden (Zhang et al. 2005):

$$\Psi(A, B) = \sum w_i \times \sigma_i(A, B) \quad (3.11)$$

wobei  $w_i$  Gewichte für die Ähnlichkeitsmasse sind. In der Bestimmung dieser Gewichte liegt auch die Schwierigkeit. Zhang et al. (2005:8) schlagen vor, sie durch automatische Lernalgorithmen aus manuell zugeordneten Testgebieten zu gewinnen.

Eine dritte Möglichkeit, ein Gesamtmasse aus den Einzelmassen abzuleiten, findet man bei Shields et al. (1996): Es wurde eine logistische Regression verwendet, um Minenstandorte (als Punktobjekte) aus zwei Datenbanken einander zuzuordnen. Folgende Ähnlichkeitsmasse wurden berechnet:

- Eine Distanzkategorie:  $DMC = 1$  wenn die euklidische Distanz  $\leq 0.1$  km ist,  $DMC = 0$  sonst.
- Das Fördergut:  $CNMC = 1$ , wenn mindestens ein Fördergut (Gold, Kupfer, ...) übereinstimmt,  $CNMC = 0$  sonst
- Der Name der Mine:  $SNP$ . Die Anzahl der Buchstaben im kürzeren Namen, die mit denen im längeren Namen übereinstimmen, geteilt durch die Länge des kürzeren Namens (vgl. auch die Methode zur Bestimmung der Ähnlichkeit von Zeichenketten in Abschnitt 3.1.2.1).

Das logistische Modell eignet sich gut, wenn die abhängige Variable binäres Wertniveau besitzt. Auch das Matching-Problem kann auf diese Weise betrachtet werden:  $H$  bezeichne das Ereignis, dass zwei Minenstandorte homolog sind. Dann hat  $H$  die Ausprägungen  $H = 1$ , wenn zwei Standorte homolog sind, und  $H = 0$  andernfalls. Das logistische Modell ist nun wie folgt:

$$\log\left(\frac{P(H_i = 1|X_i)}{P(H_i = 0|X_i)}\right) = \log\left(\frac{p_i}{1-p_i}\right) = \sum b_k X_{ik} = Z_i \quad (3.12)$$

wobei:  $p_i$  die Wahrscheinlichkeit, dass es sich bei den zwei Objekten um homologe Objekte handelt

$b_k$  ein Gewichtungsfaktor

$X_{ik}$  die erklärende Variable (ein Einzelmass)

$k = 1$  bis  $m$  Einzelmasse

$i = 1$  bis  $n$  zu vergleichende Paare

$Z_i$  wird durch eine Regression von Vergleichsdaten geschätzt: Bekannten homologen Paaren wird  $Z_i = 1$ , bekannten nicht-homologen Paaren  $Z_i = 0$  zugewiesen. Mit der Lösung der Regressionsgleichung ergeben sich die gesuchten Gewichte  $b_k$ .

Schliesslich kann  $p_i$  abgeleitet werden, indem das logistische Modell danach aufgelöst wird:

$$p_i = \frac{e^{Z_i}}{1 + e^{Z_i}} \quad (3.13)$$

Shields et al. verwendeten Daten von 5000 nicht-homologen und 5000 homologen Minenpaaren. Sie erhielten als Lösung der Regressionsgleichung:

$$Z_i = -4.4297 + 0.5555 \times CNMC + 5.3538 \times SNP + 2.3510 \times DMC \quad (3.14)$$

Der Vorteil dieser Methode ist, dass die Gewichtungsfaktoren durch eine Regression von manuell verknüpften Testgebieten geschätzt werden. Es können zudem Standardabweichungen angegeben werden, so dass sie statistisch abgesichert sind. Der Nachteil ist, dass für die Regression in einem Testgebiet sinnvolle nicht-homologe Paare gebildet werden müssen.

### 3.3 Besprechung bestehender Ansätze zum Matching von Strassendaten

Im Abschnitt 3.1 wurde eine konzeptionelle Grundlage für das Matching von Geodaten aller Art erarbeitet. Speziell für das Matching von Strassendaten gibt es sehr viele Algorithmen, welche jeweils auf ganz bestimmte Zwecke und Datensätze zugeschnitten wurden. In diesem Abschnitt sollen einige davon vorgestellt werden, denn sie bilden die Grundlage für den Algorithmus, der für das Matching von VECTOR25 und VECTOR200 entworfen wurde.

#### A. Verfahren von Rosen und Saalfeld<sup>1</sup>

Gemäss Walter (1996:60) war die Motivation zur Entwicklung des Verfahrens, die topographischen Daten des United States Geological Survey (USGS) mit den Karten des Bureau of the Census zu verknüpfen. Das Bureau of the Census hatte von Hand Karten von über 5% des Landes digitalisiert, worauf 60% der Bevölkerung wohnten. Die beiden Datensätze unterscheiden sich

---

1. Die Ausführungen in diesem Abschnitt beruhen weitgehend auf Rosen und Saalfeld (1985) und Saalfeld (1988).

nicht hinsichtlich des Massstabes, sondern in der Geometrie. Ziel des Matchings war es, Fehler aufzudecken und gegenseitig Attribute und Daten zu ergänzen.

Es wurde ein iterativer Ansatz aus alternierendem Matching und Rubber-Sheeting gewählt. Im Matching-Teil werden Knoten miteinander verknüpft. Der Rubber-Sheeting-Prozess stützt sich auf die bereits verknüpften Knoten. Das Rubber-Sheeting soll die beiden Datensätze geometrisch aneinander angleichen und damit die Verknüpfung weiterer Knoten im nächsten Iterationsschritt erlauben. Der Prozess stützt sich auf verschiedene Matching-Kriterien, wobei die stärksten Kriterien zuerst zur Anwendung kommen, und die schwächeren erst, wenn mit den starken Kriterien keine Zuordnungspaare mehr gefunden werden können. Als Kriterien für das Matching der Knoten werden vorgeschlagen:

- die geographische Position
- die Anzahl von abgehenden Kanten
- die Richtungen der ausgehenden Kanten. Sie wurden platzsparend mit der *Spider-Funktion* modelliert. Diese teilt den Vollkreis in 8 Sektoren ein. Es wird nur festgehalten, ob ein Sektor durch mind. eine Kante belegt ist oder nicht. So entsteht ein Muster aus 8 Binärzahlen pro Knoten. Die *Spider-Funktion* hat sich auch als hilfreich zur statistischen Analyse von Situationen erwiesen.

Ist ein Knoten eindeutig zugeordnet worden, so können seine Nachbarknoten auf Grund von topologischen Kriterien einfacher zugeordnet werden (*Match precipitation*).

Lineare Zuordnungen werden basierend auf den zugeordneten Knoten erstellt. Weil sich die beiden Datensätze nicht im Massstab unterscheiden, werden 1:1-Zuordnungen zwischen den Elementen gebildet.

### B. Verfahren von Walter (1996)

Die Aufgabe war, Strassen aus dem Massstab 1:25'000 des deutschen Basisdatensatzes ATKIS (Amtliches Topographisch-Kartographisches Informationssystem) dem Datensatz GDF (Geographic Data File) zuzuordnen. GDF ist ein Standard für den Austausch von digitalen Strassenverkehrsdaten in Europa. In einigen Gebieten unterscheiden sich die Daten in der Erfassung kaum, jedoch können insbesondere in Kreuzungsbereichen starke Unterschiede auftreten (Abbildung 3.11).

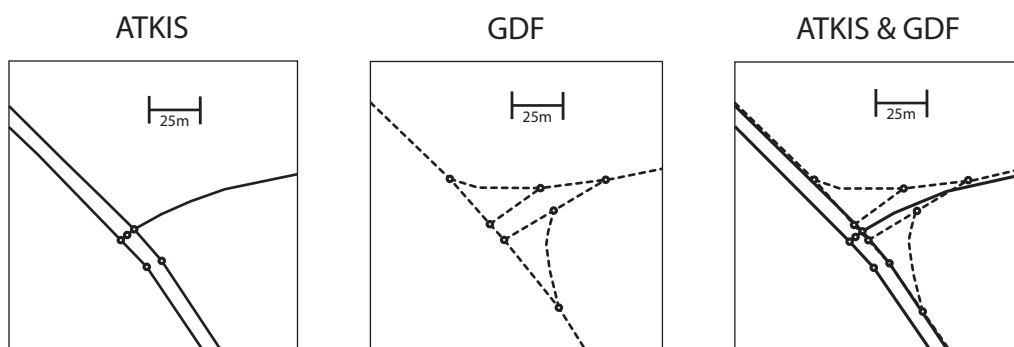


Abbildung 3.11: Gegenüberstellung ATKIS – GDF (Walter 1996:57).

Walter ordnet Strassenstücke direkt einander zu, ohne den Umweg über Knoten zu machen. Er betrachtet das Zuordnungsproblem als ein kombinatorisches Problem: Gegeben seien  $A = \{a_1, a_2, \dots, a_n\}$  die Menge der Elemente des Datensatzes  $A$  und  $B = \{b_1, b_2, \dots, b_n\}$  die Menge der Elemente des Datensatzes  $B$ . Gesucht wird eine Zuordnungsfunktion  $h(A \rightarrow B)$ , die die Menge der Elemente von  $A$  möglichst gut auf die Menge  $B$  abbildet. Um die Zuordnung zu bewerten, wird eine Leistungsfunktion  $l(a, b)$  eingeführt. Die beste Zuordnung ist diejenige, bei der die Summe aller Leistungsfunktionen der Einzelzuordnungen maximal wird:

$$L = \sum_{z=1}^Z l(a_{za}, b_{zb}) \quad (3.15)$$

Für die Berechnung der Leistungsfunktion zieht Walter einen Ansatz aus der Informationstheorie bei: Eine Zuordnung ist umso besser, je kleiner die gegenseitige Information ist. Die gegenseitige Information wird berechnet aus den Unterschieden der Form der Linie, des Winkels, des Längenunterschiedes, der Position und eines topologischen Masses *Verbundenheit*.

Da die Anzahl Kombinationsmöglichkeiten mit der Anzahl Elemente sehr schnell ansteigt, ist dies ein rechenintensiver Ansatz. Mit Heuristiken muss der Suchbaum so klein wie möglich gehalten werden. In einem ersten Schritt werden mit einem *Buffer Growing*-Verfahren mögliche N:M-Kandidatenpaare gebildet. Nur von diesen Kandidatenpaaren muss die beste Kombination gefunden werden. Mit geometrischen Beschränkungen wird die Anzahl Zuordnungspaare weiter verkleinert. Zuordnungspaare mit zu grossen Winkel- und Längenunterschieden werden als unwahrscheinlich betrachtet und scheiden aus.

Um die Rechenzeit weiter zu begrenzen, werden die Daten in überlappende Teilgebiete aufgeteilt, und für jedes Element Cluster gebildet, die alle Nachbarn bis zur Entfernung 2 beinhalten. Die Cluster werden anschliessend unabhängig voneinander mit einem *hill climbing*-Algorithmus optimiert.

Charakteristisch für dieses Verfahren ist, dass es keine manuelle Justierung von Parametern voraussetzt. Um die gegenseitige Information berechnen zu können, sind jedoch Statistiken von manuell verknüpften Testgebieten notwendig.

### C. Verfahren von Devogele (1997)

Im Rahmen seiner Dissertation hat Devogele einen Zuordnungsprozess für die beiden Datensätze BD CARTO und GEOROUTE des Institut Géographique National (IGN), Frankreich, entwickelt. BD CARTO wurde aus digitalisierten 1:50'000er Karten abgeleitet. GEOROUTE ist ein Datensatz für die Fahrzeugnavigation. Der Massstab von GEOROUTE ist unterschiedlich: In ländlichen Gebieten entsprechen die Daten denjenigen von BD CARTO, während in städtischen Gebieten detailliertere Daten vorhanden sind. Typische topologische Unterschiede sind kollabierte Kreisel und Kreuzungen. Die Lageunterschiede der Strassen sind mit maximal 30 m relativ klein.

Die Zuordnung gliedert sich in 5 Phasen:

1. Semantische Zuordnung aufgrund von Strassenbezeichnungen: Wo vorhanden, werden die Strassen nach ihrer Bezeichnung (z. B. „A31“) in Untermengen eingeteilt.

2. Bildung von Knotenkandidaten für BD CARTO-Knoten durch eine Puffer-Operation.
3. Provisorische Zuordnung der Strassen mittels der Hausdorff-Komponente GEOROUTE – BD CARTO. Ein Schwellenwert für die Hausdorff-Komponente wird für jede Strasse aus GEOROUTE stufenweise verkleinert, bis nur noch eine Strasse aus BD CARTO eine kleinere Hausdorff-Komponente hat. Devogele verwendete die Stufen {30 m, 20 m, 10 m}. Zuordnungen können mehrdeutig bleiben, wenn mehrere BD CARTO-Strassen ein Hausdorff-Komponente kleiner als 10 m zur betreffenden GEOROUTE-Strasse haben.
4. Evaluation der Knotenkandidaten durch die Strassenzuordnungen: Zwei Knoten können dann einander zugeordnet werden, wenn in Schritt 3 alle zu ihnen inzidenten Strassen miteinander verknüpft wurden. Gibt es mehrere Kandidatenknoten mit nur teilweisen Übereinstimmungen, tritt ein *prolongation* (Verlängerung) genannter Prozess in Kraft. Für alle übrigen Fälle ist eine manuelle Kontrolle notwendig.
5. Definitive Zuordnung der Strassen: Sind die Knoten einander eindeutig zugeordnet worden, werden die Strassensegmente definitiv zugeordnet. Dazu kommt ein kürzester-Weg-Algorithmus zum Einsatz: Geg. seien die Knoten  $A$  und  $B$  sowie ein Strassenstück  $s$  aus BD CARTO, welches  $A$  und  $B$  miteinander verbindet. Das entsprechende Strassenstück  $s'$  aus GEOROUTE muss die Knoten  $A'$  und  $B'$  aus GEOROUTE mit dem kürzesten Weg verbinden. Bei 1:N-Zuordnungen der Knoten, wie sie bei der Verlängerung entstehen, besteht hier natürlich die Schwierigkeit der richtigen Wahl der Knoten  $A'$  und  $B'$ .

Der Sinn der Verlängerung ist es, komplexe Kreuzungen aus GEOROUTE einem einzige Knoten aus BD CARTO zuzuweisen. Dazu werden die Kandidatenknoten in topologisch zusammenhängende Gruppen eingeteilt. Über die provisorische Zuordnung der Strassen kann dann die ähnlichste Gruppe bestimmt werden. Diese muss anschliessend noch gefiltert werden, um nicht dazu gehörende Strassenstücke zu entfernen.

Der Vorteil des Ansatzes ist es somit, dass Datensätze verschiedenen Massstabes direkt miteinander verknüpft werden können. Für Datensätze desselben Massstabes ist der Ansatz ungeeignet, da keine N:M-Verknüpfungen möglich sind.

#### **D. Verfahren von Cecconi (2003)**

Cecconi beschäftigte sich mit der Zuordnung der Datensätze VECTOR25 und VECTOR200 der Swisstopo (die Datensätze werden in Kapitel 4.1 beschrieben und miteinander verglichen). Der Prozess sieht wie folgt aus:

1. Verknüpfen der Knoten basierend auf den Kriterien i) nächster Knoten; ii) Richtung der inzidenten Strassensegmente; iii) Klassifikation der inzidenten Strassensegmente.
2. Verknüpfen der Strassen durch den kürzesten Weg zwischen verknüpften Knoten.

Es werden nur 1:1-Zuordnungen von Knoten behandelt, Spezialfälle wie Kreisel, welche in VECTOR200 auf einen einzigen Knoten reduziert wurden, werden nicht abgedeckt.



## Kapitel 4

# Ansatz zum Matching von Strassendaten stark unterschiedlicher Massstäbe

In diesem Kapitel wird ein eigener Matching-Ansatz erarbeitet. Die Datensätze VECTOR25 und VECTOR200 werden beschrieben und miteinander verglichen, um Konflikte im Datenmodell und in der Geometrie herauszuarbeiten. Die algorithmischen Komponenten zur Lösung dieser Konflikte werden vorgestellt und deren Zusammenspiel im Matching-Prozess anhand eines Beispiels erläutert.

### 4.1 Gegenüberstellung der Datensätze VECTOR25 und VECTOR200

Für die Entwicklung und die Erprobung des eigenen Matching-Ansatzes wurden die Datensätze VECTOR25 und VECTOR200 des Bundesamtes für Landestopografie der Schweiz (Swisstopo) eingesetzt. Sie entsprechen geometrisch und inhaltlich den Schweizer Landeskarten 1:25'000 resp. 1:200'000. Die Aktualisierung beider Datensätze geschieht vollständig getrennt: Ab Luftbildern werden die Elemente, welche in die Massstäbe 1:25'000 bzw. 1:200'000 aufgenommen werden sollen, identifiziert und separat für die beiden Datensätze aufbereitet.

#### 4.1.1 Der Datensatz VECTOR25<sup>1</sup>

VECTOR25 basiert auf der Schweizer Landeskarte 1:25'000. Der Datensatz beschreibt 8.5 Millionen Objekte in den neun thematischen Ebenen Strassennetz, Eisenbahnnetz, übriger Verkehr, Gewässernetz, Primärflächen (Wald, See usw.), Gebäude, Hecken- und Bäume, Anlagen, Einzelobjekte. Pro Ebene ist ein Attributsatz definiert, der unter anderem eine eindeutige und stabile Kennung (*ObjectId*) und eine Objektart (*ObjectVal*) enthält.

---

1. Die Ausführungen in diesem Abschnitt beruhen weitgehend auf Swisstopo (2004a).

Die Objektart entspricht der Kartenlegende<sup>1</sup>. Die Lagegenauigkeit wird entsprechend der Genauigkeit der Landeskarte mit 3 – 8 m angegeben.

Für die vorliegende Arbeit wurde nur die Ebene Strassennetz verwendet. Es sind folgende Attribute vorhanden:

Name	Wertebereich	NULL	Beschreibung
ObjectId	Integer(4 Bytes)	n	Eindeutiger und stabiler Identifikations-schlüssel
ObjectOrigin	Text(20)	n	Herkunft der Daten
ObjectVal	Text(20): siehe Objektarten	n	Objektart
YearOfChange	Integer:1900...9999	n	Nachführungsjahr des Objektes (Bildflug)
BridgeType	Text(10): Bruecke, GedBrue, Steg	j	Brückentyp (Brücke, gedeckte Brücke, Steg)
TunnelType	Text(10): Galerie, Tunnel	j	Tunneltyp
Strada_id	Text(24)	j	Strada-DB Attribut

Tabelle 4.1: Attribute der Ebene Strassennetz aus VECTOR25 (Swisstopo 2004a:8).

Das Attribut *ObjectVal* bezeichnet die Strassenklasse gemäss der Legende der Landeskarte. In deutschen und französischen Gebieten wurde die ausländische Bezeichnung verwendet. Da die betrachteten Testgebiete innerhalb der Schweiz liegen, werden hier nur die Schweizer Bezeichnungen aufgeführt:

ObjectVal	Beschreibung (Objektart)	gerichtet
Autobahn	Autobahn	j
Autob_Ri	Autobahn richtungsgetreunt	j
Autostr	Autostrasse	n
Ein_Ausf	Ein-/Ausfahrt (Autobahn / Strasse)	j
A_Zufahrt	Autobahnzufahrt	n
1_Klass – 6_Klass	1. Klass-Strasse – 6. Klass-Strasse	n
Q_Klass	Quartierweg	n
HistWeg	Historischer Weg	n

Tabelle 4.2: Objektarten der Ebene Strassennetz aus VECTOR25 (Swisstopo 2004a:9).

1. Eine Zeichenerklärung der Schweizer Landeskarte 1:25'000 ist auf dem Internet erhältlich. [http://www.swisstopo.ch/pub/down/products/analog/maps/symbols\\_de.pdf](http://www.swisstopo.ch/pub/down/products/analog/maps/symbols_de.pdf), Stand 20.12.2005

ObjectVal	Beschreibung (Objektart)	gerichtet
PzPiste	Panzerpiste	n
Parkweg	Parkweg	n
BrueckLe	Alleinstehende Brücke	n
GedBruLe	Alleinstehende Brücke gedeckt	n
StegLe	Alleinstehender Steg	n

Table 4.2: Objektarten der Ebene Strassennetz aus VECTOR25 (Swisstopo 2004a:9).

Die Einteilungskriterien in die Klassen 1. – 6. Klass-Strasse und Quartierstrasse beinhalten vor allem Strassenbreite, Art des Belags, und maximale Steigung<sup>1</sup>. Für Quartierstrassen ist zusätzlich vorgeschrieben, dass sie keine Bedeutung für den Durchgangsverkehr haben dürfen.

### 4.1.2 Der Datensatz VECTOR200<sup>2</sup>

VECTOR200 basiert auf der Landeskarte 1:200'000. Der Datensatz beschreibt 426'000 Objekte in den thematischen Ebenen Verkehrsnetz, Gewässernetz, Primärflächen, Gebäude, Einzelobjekte und Grenzen. Die Lagegenauigkeit wird mit 20 – 60 m angegeben, grössere Abweichungen ergeben sich an Stellen, wo aus kartographischen Gründen generalisiert wurde.

Die Ebene Verkehrsnetz umfasst neben dem Strassen- und Wegnetz der Landeskarte 1:200'000 auch das Eisenbahnnetz sowie weitere Objekte (z. B. Pass oder Bergbahn) zum Thema Transport. Die Netze der verschiedenen Verkehrsträger werden mit dem fiktiven Linienelement „Zugang“ verknüpft.

Die Ebene umfasst sowohl Knoten- wie auch Linienobjekte. Für die vorliegende Arbeit wurden nur Linienobjekte benutzt. Deren Attribute sind:

Name	Wertebereich	NULL	Beschreibung
ObjectId	Integer(4 Bytes)	n	Eindeutiger und stabiler Identifikations-schlüssel
ObjectOrigin	Text(20)	n	Herkunft der Daten
ObjectVal	Text(20): siehe Objektarten	n	Objektart
YearOfChange	Integer:1900...9999	n	Gesamtnachführung Karte
Construction	Text(10): Bruecke, Tunnel	j	Kunstabauten (Brücke oder Tunnel)
ObjectName	Text(30)	j	Bedeutende Tunnelnamen oder wichtige Strassenabschnitte

Table 4.3: Attribute der Ebene Verkehrsnetz aus VECTOR200 (Swisstopo 2004b:9).

1. [http://www.swisstopo.ch/pub/down/products/analog/maps/signs\\_de.pdf](http://www.swisstopo.ch/pub/down/products/analog/maps/signs_de.pdf), Stand 20.12.2005
2. Die Ausführungen in diesem Abschnitt beruhen weitgehend auf Swisstopo (2004b).

Von den Objektarten sind wiederum nur diejenigen aufgelistet, welche für das Strassen-Matching von Bedeutung sind. Objekte anderer Art wurden in der Vorverarbeitung aus den Datensätzen gelöscht.

ObjectVal	Beschreibung (Objektart)	gerichtet
Autobahn	Autobahn	n
Autob_Ri	Autobahn richtungsgetrennt	j
Autostr	Autostrasse	n
DurchgStr6	Hauptstrasse als Durchgangsstrasse 6 m (Minimalbreite)	j
DurchgStr4	Hauptstrasse als Durchgangsstrasse 4 m (Minimalbreite)	n
VerbindStr6	Hauptstrasse als Verbindungsstrasse 6 m (Minimalbreite)	n
VerbindStr4	Hauptstrasse als Verbindungsstrasse 4 m (Minimalbreite)	n
NebenStr6	Nebenstrasse 6 m (Minimalbreite)	n
NebenStr4	Nebenstrasse 4 m (Minimalbreite)	n
Fahrstraes	Fahrsträsschen	n

Tabelle 4.4: Objektarten der Ebene Verkehrsnetz aus VECTOR200 (Swisstopo 2004b:10).

Die Erfassungskriterien für die Strassenklassen der Landeskarte 1:200'000 (und damit VECTOR200) richten sich primär nach dem Verkehrsvolumen. Die Basis dafür bilden die Durchgangsstrassenverordnung (StrV) und die Signalisationsverordnung (SSV) des Bundes. Zusätzlich hat die Swisstopo eigene Erweiterungen definiert, um den Verkehrsfluss realistisch abzubilden.

### 4.1.3 Vergleich der Datenmodelle

Die beiden Datensätze VECTOR25 und VECTOR200 lagen im Shapefile-Datenformat vor und wurden in die Applikation importiert. Shapefile repräsentiert die Geometrie im Spaghetti-Datenformat als einzelne Einträge von Punkten, Linien, oder Flächen. Die Topologie wird nicht repräsentiert. Attributdaten werden in separaten dBase-Tabellen gehalten und über einen eindeutigen Schlüssel mit den Geometrie-Einträgen verbunden<sup>1</sup>.

Bei beiden Datensätzen werden Beziehungen zwischen Strassen (Kreuzung, Unterführung) nicht explizit modelliert. Kreuzen sich zwei Strassen, so haben sie einen gemeinsamen Knoten. Bei einer Unterführung trägt die eine Strasse das Attribut „Brücke“, am Punkt der Überschneidung liegt jedoch kein gemeinsamer Knoten – es liegt also keine planare Topologie vor, sondern eine Netzwerk-Topologie.

Aus Tabelle 4.2 respektive Tabelle 4.4 ist ersichtlich, dass die Klassierung der Strassen in den beiden Datensätzen unterschiedlich ist. Trotzdem können die Strassenklassen das Matching erleichtern, da sich gewisse Kategorien ausschliessen. Durch eine Analyse der manuell verknüpf-

1. <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>, Stand 20.12.2005

ten Testgebiete und einer vorgängigen Untersuchung von Cecconi (2003:62) konnte folgende Liste erstellt werden:

- Autobahn = {Autobahn}
- Autostr = {Autostr}
- DurchgStr6 = {1\_Klass, 2\_Klass}
- VerbindStr6 = {1\_Klass, 2\_Klass}
- VerbindStr4 = {1\_Klass}
- Nebenstr3 = {2\_Klass, 1\_Klass, Q\_Klass, 3\_Klass}
- Fahrstraes = {3\_Klass, 2\_Klass, 4\_Klass, Q\_Klass}

Sie ist so zu lesen: Als Kandidaten für eine als „DurchgStr6“ klassierte Strasse in VECTOR200 kommen nur als „1\_Klass“, „2\_Klass“ oder „Q\_Klass“ klassierte Strassen aus VECTOR25 in Frage. Die Reihenfolge in der Liste bezeichnet die Häufigkeit, mit der ein solcher Match in den Testdaten vorkam.

#### 4.1.4 Vergleich der geometrischen Erfassung

Selbst wenn zwei Datensätze dasselbe Datenmodell haben und dieselben Erfassungskriterien zugrunde liegen, so werden sie sich aufgrund unterschiedlicher Interpretationen der Erfasser oder anderer Digitalisierung doch geometrisch unterscheiden. Im Fall von VECTOR25 und VECTOR200 sind die Unterschiede nicht nur in der getrennten Digitalisierung, sondern auch in einem anderen Massstab und eventuell in einem anderen Aktualitätsstand zu suchen. Globale Lageunterschiede treten hingegen nicht auf. Wo Lageunterschiede zu verzeichnen sind, so haben diese ihren Ursprung in der Generalisierung und sind damit lokal begrenzt. Daher kann auf eine geometrische Entzerrung verzichtet werden.

Das VECTOR200-Strassennetz ist im Wesentlichen eine Untermenge des VECTOR25-Strassennetzes: Von den Strassen in VECTOR25 sind in VECTOR200 hauptsächlich die für die nationale und regionale Verkehrsführung bedeutenden Strassen enthalten, während Quartier- und Nebenstrassen, Landwirtschaftswege etc. wegfallen. Die Regel sind N:1-Beziehungen zwischen VECTOR25/200-Objekten, wenn sich auch die Geometrie wegen der Generalisierung stark unterscheiden kann. Ausnahmen davon sind vor allem in Kreuzungsbereich zu finden:

- *Kollabierende Strassenkreisel* (Abbildung 4.1): Kreisel aus VECTOR25 kollabieren systematisch zu einem Knoten in VECTOR200.
- *Kollabierende Segmente* (Abbildung 4.2): Diese Art von Kollaps tritt sporadisch auf. Kurze Strassenstücke in VECTOR25 fallen weg, wobei die beiden Endknoten in den VECTOR200-Knoten aggregiert werden.

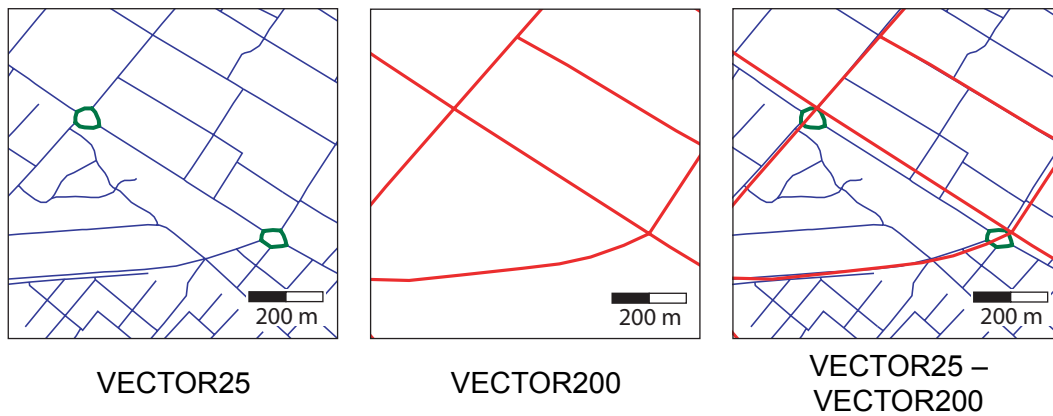


Abbildung 4.1: Die beiden Kreisel (grün markiert) in VECTOR25 kollabieren zu einem Knoten in VECTOR200.

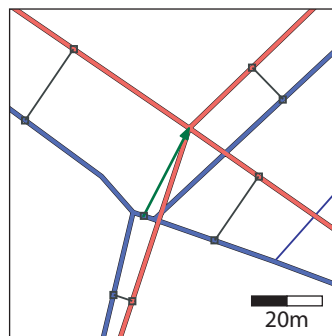


Abbildung 4.2: VECTOR25-Segment, das in VECTOR200 zu einem Knoten kollabiert ist (grüner Pfeil).

Bei Autobahnen sind in VECTOR25 die Fahrspuren teilweise als getrennte Fahrspuren modelliert, während in VECTOR200 nur Mittelachsen dargestellt werden (Abbildung 4.3).

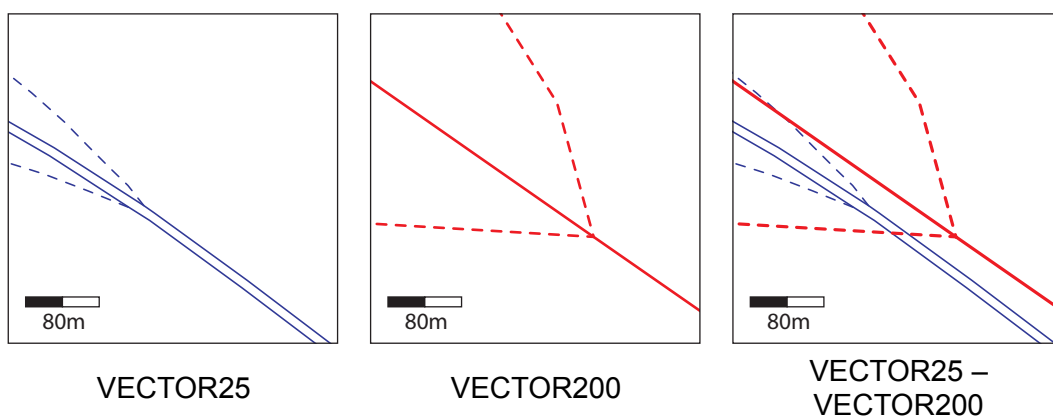


Abbildung 4.3: Modellierung der Autobahnen mit getrennten Fahrspuren (VECTOR25) bzw. mit einer Mittelachse (VECTOR200). Durchgezogen: Als Autobahn klassierte Straßen. Gestrichelt: Autobahneinfahrten bzw. -ausfahrten.

## 4.2 Matching-Prozess

### 4.2.1 Motivation zur Entwicklung des Matching-Ansatzes

Abbildung 4.4 zeigt eine VECTOR200-Strasse und die mit ihr korrespondierenden VECTOR25-Strassen. Die Knotenpunkte sind Kreuzungen mit weiteren, nicht korrespondierenden VECTOR25-Strassen, die in der Abbildung nicht dargestellt sind. In der Regel korrespondieren wie in der Abbildung gezeigt sehr viele VECTOR25-Strassen mit einer einzigen VECTOR200-Strasse. Die vielen kurzen VECTOR25-Strassen sind nicht direkt mit der VECTOR200-Strasse vergleichbar, beispielsweise können die Richtungen lokal sehr unterschiedlich sein. Auch Bildung und Vergleich von N:1-Strassenkandidaten, wie es etwa beim *Buffer Growing* geschieht, ist nicht vielversprechend, weil sich Referenzstrasse und Kandidatenstrassen bezüglich ihrer Linienmasse stark unterscheiden können: Die VECTOR200-Strasse wird i. A. stark geglättet wiedergegeben. Würde man die Strassenlängen als Matching-Kriterium verwenden, so würde man in der Regel zu wenige VECTOR25-Strassen selektieren, weil die VECTOR200-Strasse aufgrund der Glättung kürzer ist. Ebenso wenig aufschlussreich ist der Vergleich der Basislinien.

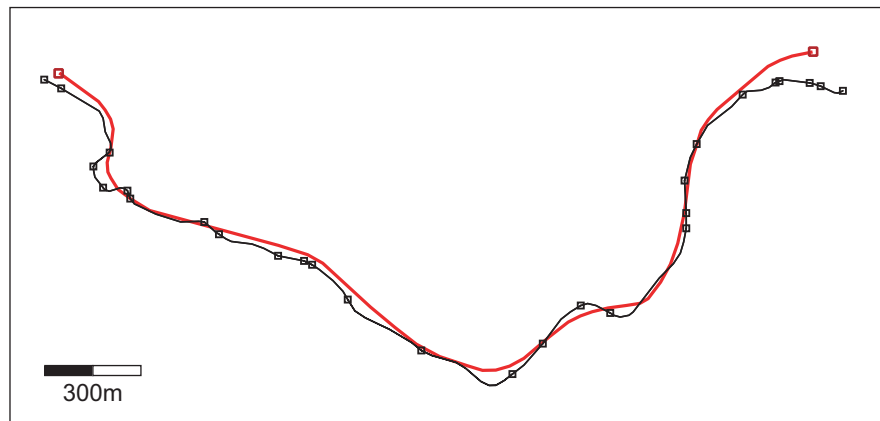


Abbildung 4.4: Vergleich der Längen von Strassenstücken.

Ein Ansatz, der zuerst Knoten verknüpft und die Knotenverknüpfungen in einem weiteren Schritt zu Strassenverknüpfungen umsetzt, ist deshalb sinnvoller als ein Ansatz, der auf dem direkten Vergleich von Strassenkandidaten beruht. In der vorliegenden Arbeit wurde der knotenbasierte Ansatz von Devogele (1997) implementiert und getestet. Für die Erkennung von korrespondierenden Knoten stützt sich das Verfahren auf eine temporäre Zuordnung von Strassen ab (siehe Abschnitt 3.3, Punkt C). Ein VECTOR25-Knoten ist dann ein gültiger Kandidatenknoten für einen VECTOR200-Knoten, wenn für jede zum VECTOR200-Knoten inzidente Strasse eine verknüpfte VECTOR25-Strasse existiert, die zum VECTOR25-Kandidaten inzident ist (sie werden dann als *vollständige Kandidaten* bezeichnet). Die von Devogele verwendeten Schwellenwerte für die Hausdorff-Komponenten waren zu klein für die Zuordnung von VECTOR25 und VECTOR200, so dass viele der korrespondierenden VECTOR25-Strassen irrtümlicherweise nicht zugeordnet wurden. Bei grösseren Schwellenwerten wurden oft viele nahe beieinander liegende VECTOR25-Strassen derselben VECTOR200-Strasse zugeordnet. Dann

sind aber auch viele VECTOR25-Knoten gleichzeitig *vollständige Kandidaten*, und in der Folge kann der Algorithmus auch keinen der Kandidatenknoten als korrespondierenden Knoten identifizieren. Die Methode der temporären Strassenzuordnungen eignet sich also nicht für Datensätze mit starken Unterschieden in der Dichte der Strassennetze, wie sie bei Datensätze mit stark unterschiedlichen Massstäben auftreten.

Aufgrund der Schwächen der bestehenden Matching-Verfahren bezüglich stark unterschiedlicher Massstäbe wurde ein eigener Ansatz erarbeitet, der VECTOR200 und VECTOR25 erfolgreich verknüpfen kann.

### 4.2.2 Abgrenzung

Der entwickelte Matching-Prozess wählt den VECTOR200-Datensatz als Referenz, zu welchem aus VECTOR25 Kandidatenmengen gebildet und wahrscheinliche Matches erzeugt werden sollen. Es werden zuerst Knoten verknüpft und diese anschliessend zu Strassenverknüpfungen erweitert.

In der vorliegenden Arbeit werden nur 1:1-Beziehungen zwischen Knoten betrachtet. Die beiden in Abschnitt 4.1.4 erläuterten Fälle der N:1-Beziehungen zwischen VECTOR25- und VECTOR200-Knoten müssen gesondert behandelt werden:

- *Kollabierende Strassenkreisel*: Sie lassen sich aufgrund der charakteristischen Form und Fläche leicht in einer Vorverarbeitungsstufe erkennen. Sie können zu einem *Superobjekt* zusammengefasst werden, wobei der Kreiselmittelpunkt der stellvertretende Knoten wird.
- *Kollabierende Segmente*: Sie können nicht in der Vorverarbeitung erkannt werden wie Kreisel. Devogele (1997:126–129) stellt ein Verfahren zur Behandlung solcher N:1-Beziehungen zwischen Knoten vor. Die Anwendung dieses Verfahrens auf VECTOR25 und VECTOR200 ergab jedoch keine befriedigenden Ergebnisse. Da solche Kollapse relativ selten auftreten, werden sie dem Benutzer zur Nachbearbeitung überlassen.

Autobahnen und Autostrassen wurden wegen der Problematik der getrennten Fahrspuren vom Matching ausgeschlossen. Wie Thom (2005) zeigt, können getrennte Fahrspuren in der Vorverarbeitung zu einer Mittelachsendarstellung reduziert werden. Die Implementation des Verfahrens hätte jedoch den Zeitrahmen gesprengt.

In dichten Stadtgebieten wie in Zürich sind die Unterschiede in der Erfassung teilweise so gross, dass nicht einmal mehr manuell eine eindeutige Zuordnung möglich ist – in diesem Gebiet kann ein automatisches Verfahren nur schlechte Ergebnisse liefern. Das in der vorliegenden Arbeit entwickelte Verfahren konzentriert sich auf ländliche Gebiete und Siedlungen bis zu einer Grösse von etwa 100'000 Einwohnern – womit flächenmässig der grösste Teil der Schweiz behandelt werden kann.

### 4.2.3 Übersicht über den Ablauf

Im Folgenden wird der entwickelte Matching-Prozess vorgestellt. In diesem Abschnitt soll ein Überblick gegeben werden. Der Matching-Prozess lässt sich in vier verschiedene Phasen gliedern:

### 1. **Bildung von Kandidatenmengen**

Zu den Knoten und Strassen aus VECTOR200 werden Kandidaten aus VECTOR25 erzeugt. Dazu werden um die Elemente aus VECTOR200 Puffer gebildet und mit VECTOR25 verschnitten. Diejenigen VECTOR25-Elemente, die innerhalb eines Puffers liegen, sind Kandidaten für das entsprechende VECTOR200-Element. Es handelt sich jedoch erst um eine Grobauswahl. Sie kann durch die Anwendung von Beschränkungen und zweier algorithmischer Komponenten noch verfeinert werden. Dieser Prozess ergibt die definitiven Kandidaten für das Matching.

### 2. **Matching der Knoten**

Es werden automatisch 1:1-Verknüpfungen zwischen Knoten gebildet. Bei Knoten, bei denen der Matching-Algorithmus keine eindeutige Zuordnung findet, weil keiner der Kandidatenknoten sich signifikant von den anderen Kandidaten unterscheidet, oder weil keine 1:1-Beziehung vorliegt, muss der Benutzer eingreifen und den Knoten von Hand zuweisen.

### 3. **Matching der Strassen**

Die gebildeten Knotenverknüpfungen werden automatisch zu Strassenverknüpfungen umgesetzt.

### 4. **Nachbearbeitung**

Die erstellten Matches müssen durch den Benutzer kontrolliert, eventuell korrigiert und ergänzt werden. Hier kommen die Werkzeuge zum Einsatz, wie sie auch für das vollständig manuelle Matching verwendet werden.

Abbildung 4.5 zeigt den Ablauf des Prozesses. Die grau umrandeten Boxen entsprechen den genannten Phasen. Die letzte Phase (die Nachbearbeitung) wird nicht dargestellt, da sie vom Benutzer manuell durchgeführt werden muss. Die einzelnen Funktionsblöcke (Module), die im Gesamtprozess teilweise mehrfach oder iterativ zur Anwendung kommen, werden in Abschnitt 4.2.4 vorgestellt. In den Abschnitten 4.2.5–4.2.8 werden die vier Phasen anhand eines Beispiels in voller Tiefe behandelt.

## 4.2.4 Module des Matching-Prozesses

### 4.2.4.1 Modul *Beschränkung: Strassenklasse*

Wie in Abschnitt 4.1.3 erläutert wurde, sind die Strassenklassen von VECTOR200 und VECTOR25 semantisch nicht äquivalent, die Definitionen überlappen sich aber doch teilweise, so dass eine gewisse Filterung der VECTOR25-Kandidaten für eine VECTOR200-Strasse aufgrund der Strassenklassen vorgenommen werden kann. Die durch eine manuelle Verknüpfung hergeleitete Liste aus Abschnitt 4.1.3 kann in eine binäre Kreuztabelle umgesetzt werden (Tabelle 4.5). Das Modul prüft für jede Kandidatenstrasse, ob eine Zuordnung gemäss der Tabelle möglich ist und verwirft allenfalls den Kandidaten. Hat beispielsweise die VECTOR200-Strasse die Objektklasse „Durchgangsstrasse 6 m“, so können alle VECTOR25-Kandidaten, welche keine „1. Klass-Strasse“ oder „2. Klass-Strasse“ sind, eliminiert werden. Im Sinne der Klassifikation der Ähnlichkeitsmasse in Abschnitt 3.2 handelt es sich um ein semantisches Mass.

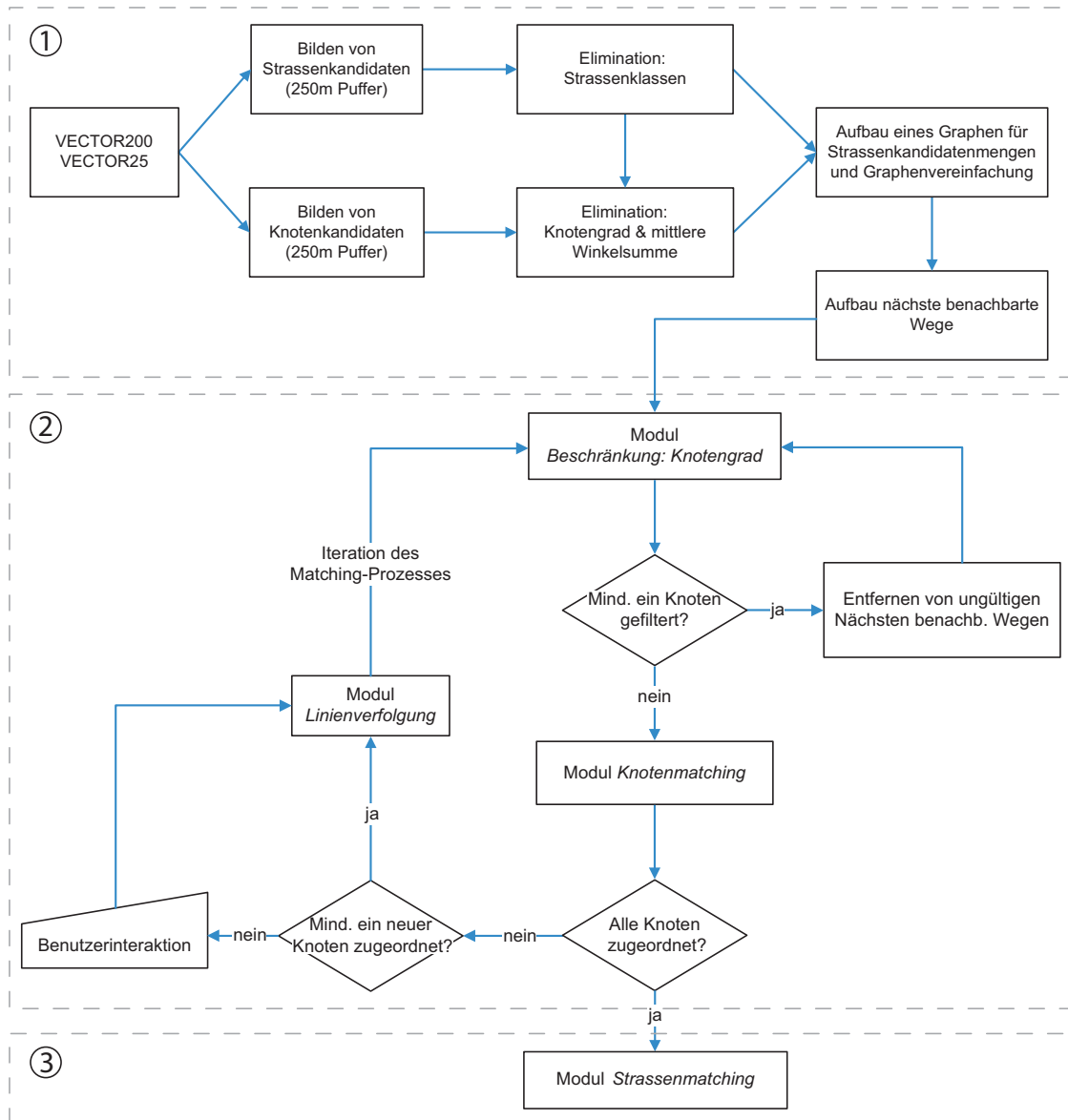


Abbildung 4.5: Ablauf des Matching-Prozesses.

	1_Klass	2_Klass	3_Klass	4_Klass	5_Klass	6_Klass	Parkweg	Q_Klass
DurchgStr6	1	1	0	0	0	0	0	0
VerbindStr6	1	1	0	0	0	0	0	0
VerbindStr4	1	0	0	0	0	0	0	0
NebenStr3	1	1	1	0	0	0	0	1
Fahrstraess	0	1	1	1	0	0	0	1

Tabelle 4.5: Vereinbarkeit der Objektklassen (0 = nicht vereinbar, 1 = vereinbar). Zeilen: VECTOR200-Objektklassen. Spalten: VECTOR25-Objektklassen.

#### 4.2.4.2 Modul *Beschränkung: Knotengrad*

Dieses Modul wendet ein topologisches Ähnlichkeitsmass an. Es wirkt auf Kandidatenknoten. Es vergleicht die Anzahl Strassen, die mit einem Knoten verbunden sind, zwischen einem Referenzknoten und einem Kandidatenknoten. Die mit dem Kandidatenknoten verbundenen Strassen müssen natürlich in einer Kandidatenmenge der mit dem Referenzknoten verbundenen VECTOR200-Strassen sein. Daher läuft dieses Modul erst, nachdem Strassen mit ungültigen Strassenklassen entfernt worden sind.

Die Bedingung ist, dass der VECTOR25-Kandidat mindestens denselben Knotengrad hat wie der VECTOR200-Referenzknoten. Ansonsten wird er aus der Kandidatenmenge entfernt.

#### 4.2.4.3 Modul *Mittlere Zwischenwinkelsumme*

Die alleinige Betrachtung der Knotengrade genügt oftmals nicht, um eine eindeutige Entscheidung zu treffen. Daher kommt ein metrisches Mass zur Anwendung. Bei homologen Strassenkreuzungen führen die Strassen ungefähr in derselben Richtung von den Kreuzungen weg. Ist der Zwischenwinkel zwischen einer VECTOR200-Strasse und einer VECTOR25-Strasse klein, so handelt es sich wahrscheinlich um dasselbe Realwelt-Objekt und die beiden Strassen können einander zugeordnet werden. Diese Zuordnung kann für alle einfallenden Strassen ausgeführt und die Zwischenwinkel können aufsummiert werden. Je kleiner die Zwischenwinkelsumme, desto ähnlicher ist der Kandidatenknoten zum Referenzknoten.

Die zu Kandidatenknoten und Referenzknoten inzidenten Strassen werden einander so zugeordnet, dass die Summe der Zwischenwinkel zwischen den zugeordneten Strassen minimal wird (siehe Abbildung 4.6). Jede zum Referenzknoten einfallende Strasse erhält einen Partner aus der Menge der zum Kandidatenknoten einfallenden Strassen. Falls der Kandidatenknoten einen höheren Grad hat als der Referenzknoten, bleiben die übrigen VECTOR25-Strassen ohne Zuordnungspartner. Die so erhaltenen Zwischenwinkel je inzidenter Strasse  $i$  werden summiert und mit dem Grad des Referenzknotens  $\kappa$  normiert:

$$\Delta_{ZWS} = \left( \sum_{i=1}^{\kappa} \gamma_i \right) / \kappa \quad (4.1)$$

Die mittlere Zwischenwinkelsumme  $\Delta_{ZWS}$  wird im Matching-Prozess sowohl als Beschränkung als auch als Vergleichsmass zwischen Kandidatenknoten eingesetzt.

In Abbildung 4.6 hat der Referenzknoten aus VECTOR200 beispielsweise zwei Kandidaten. Kandidat 1 ist der zum Referenzknoten korrespondierende Knoten. Wie man in Abbildung 4.6b und Abbildung 4.6c sieht, ist die Zwischenwinkelsumme von Kandidat 1 tatsächlich kleiner als diejenige von Kandidat 2.

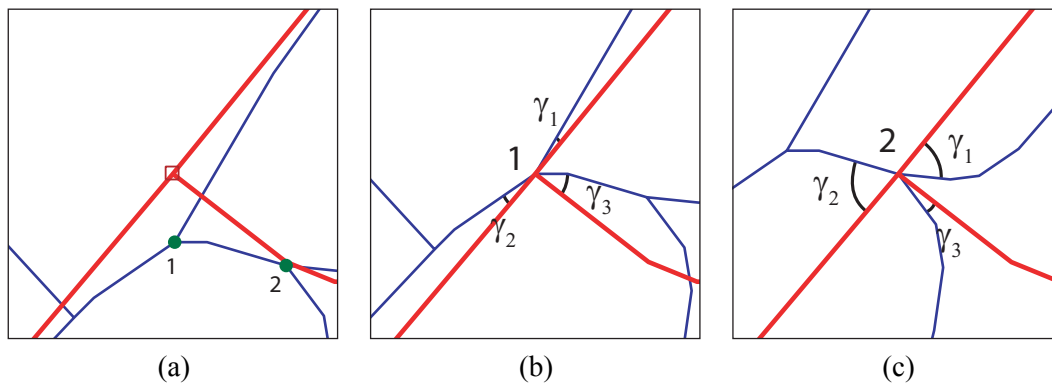


Abbildung 4.6: Vergleich zweier Knotenkandidaten mittels Zwischenwinkelsumme. (a) Situation. Grün: VECTOR25-Knotenkandidaten. (b) Zwischenwinkel für Knotenkandidat 1. (c) Zwischenwinkel für Knotenkandidat 2.

#### 4.2.4.4 Modul *Nächste benachbarte Wege*

Oft bleiben nach der Filterung durch Strassenklassen viele Kandidatenstrassen übrig, welche nicht zum Matching gehören und deshalb herausgefiltert werden müssen. Abbildung 4.9 zeigt eine solche Situation. Zur Lösung dieses Problems schlägt Devogele vor, den kürzesten Weg zwischen den zugeordneten Endknoten zu berechnen (Devogele 1997:138). Im Vergleichsdatensatz wird der kürzeste Weg zwischen den beiden dem Endknoten zugeordneten Knoten gesucht. Diejenigen Kandidatenstrassen, die Bestandteil des kürzesten Wegs sind, stellen die korrekten Matching-Partner dar.

In einigen Situationen kann die Methode des kürzesten Wegs versagen (Abbildung 4.7b). Es wäre besser, einen nächsten Weg anstatt eines kürzesten Wegs zu berechnen (Abbildung 4.7c).

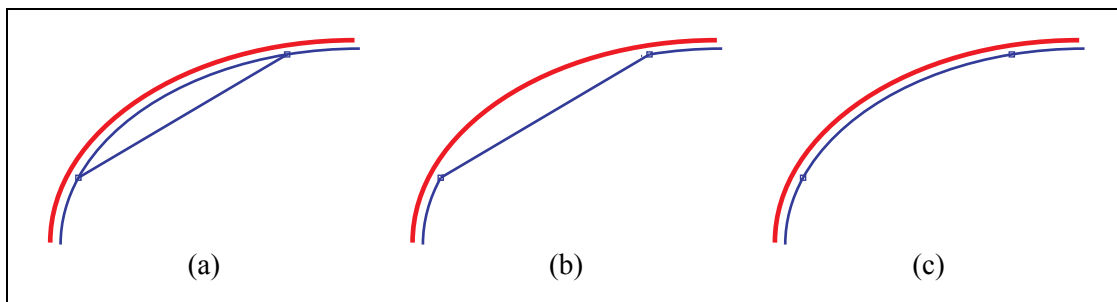


Abbildung 4.7: Filterung nach dem kürzesten und nach dem nächsten Weg (Devogele 1997:138).

In der vorliegenden Arbeit wurde ein Algorithmus für die Berechnung eines nächsten Wegs entwickelt. Die Methode basiert auf dem *shortest path*-Algorithmus von Dijkstra. *Shortest path*-Algorithmen berechnen in einem Graphen einen Weg zwischen zwei Knoten mit der Eigenschaft, dass dieser Weg bezüglich eines Kantengewichts minimal ist. Dem Namen dieser Algorithmengruppe zum Trotz muss das Kantengewicht nicht der Länge der Kante entsprechen, sondern kann eine beliebige Eigenschaft sein<sup>1</sup>.

1. In Routenplanern werden Shortest Path-Algorithmen beispielsweise auch genutzt, um den Weg zwischen zwei Orten zu berechnen, welcher die kürzeste Fahrtzeit hat.

Der Algorithmus läuft wie folgt ab (Sedgewick 2003:293–302): Gegeben sei ein Graph  $G$  mit einer Menge von Knoten und einer Menge von Kanten, welche Gewichte tragen (grün in Abbildung 4.8a). Zwei der Knoten sind als Startknoten ( $S$ ) resp. Zielknoten ( $Z$ ) ausgezeichnet. Der Algorithmus fügt ausgehend vom Startknoten pro Schritt eine Kante zum *shortest path* hinzu:

1. Wähle  $S$  als aktuellen Knoten.
2. Verfolge alle vom aktuellen Knoten ausgehende Kanten. Für die damit verbundenen Knoten, setze  $M_{neu} = M_{aktueller\ Knoten} + Kantengewicht$ . Falls der Knoten noch keine Markierung  $M$  hat oder seine Markierung  $M$  grösser als  $M_{neu}$  ist, setze  $M = M_{neu}$ . Ansonsten behält er  $M$ .
3. Markiere den aktuellen Knoten als besucht. Wähle denjenigen Knoten mit der kleinsten Markierung  $M$ , der noch nicht besucht worden ist, als neuen aktuellen Knoten. Füge die Kante zum *shortest path* hinzu.
4. Ist der neue aktuelle Knoten der Zielknoten, so ist der Algorithmus fertig. Ansonsten fahre mit Punkt 2 fort.

Zur Illustration des Algorithmus dient Abbildung 4.7. Im ersten Schritt sind die Knoten 1 und 2 durch Kanten mit dem Startknoten verbunden. Sie erhalten deshalb eine Markierung (rot in der Abbildung), die gerade dem Kantengewicht entspricht, da der Startknoten mit  $M = 0$  vorbelegt ist (Abbildung 4.7b). Knoten 2 hat von den beiden die kleinere Markierung, also wird er zum neuen aktuellen Knoten (Abbildung 4.7c). Da  $S$  schon besucht wurde, gibt es nur noch eine ausgehende Kante, welche verfolgt werden muss. Sie führt zu Knoten 1. Dieser hat schon eine Markierung  $M = 6$ , diese ist aber grösser als  $M_{neu} = 4$  und wird deshalb ersetzt. Da Knoten 1 der einzige Knoten ist, wird dieser zum aktuellen Knoten. Der Prozess setzt sich fort, bis der Zielknoten erreicht ist und somit der *shortest path* konstruiert wurde (Abbildung 4.7e).

Die Zeitkomplexität der implementierten Lösung ist  $O(V^2)$ . Es gibt auch Implementierungen, die das Problem in  $O(E \log V)$  lösen ( $V = \text{Anzahl Knoten}$ ,  $E = \text{Anzahl Kanten}$ ), jedoch einen grösseren Implementierungsaufwand erfordern (Sedgewick 2003:301).

Um einen nächsten Weg anstatt des kürzesten Wegs zu erhalten, wird für die Berechnung der *shortest paths* nicht die Strassenlänge als Kantengewicht verwendet, sondern die Hausdorff-Komponenten der Kandidatenstrassen zu der Referenzstrasse. Ein besuchter Knoten wird in Schritt 2 des Algorithmus mit dem Mittel der Summe der Hausdorff-Komponenten markiert:

$$M_{neu} = \frac{\sum_{i=1}^n HK_i}{n} \quad (4.2)$$

mit  $HK_i = \text{Hausdorff-Komponente der bisher am } shortest\ path\ teilnehmenden\ Strassen$ ,  $n = \text{Anzahl der bisher am } shortest\ path\ teilnehmenden\ Strassen$ .

Die Mittelbildung ist notwendig, weil sonst  $M_{neu}$  mit jedem Segment mehr im *shortest path* grösser würde und der Algorithmus eher die Anzahl der Segmente minimieren würde anstatt die Distanz zur Referenzstrasse.

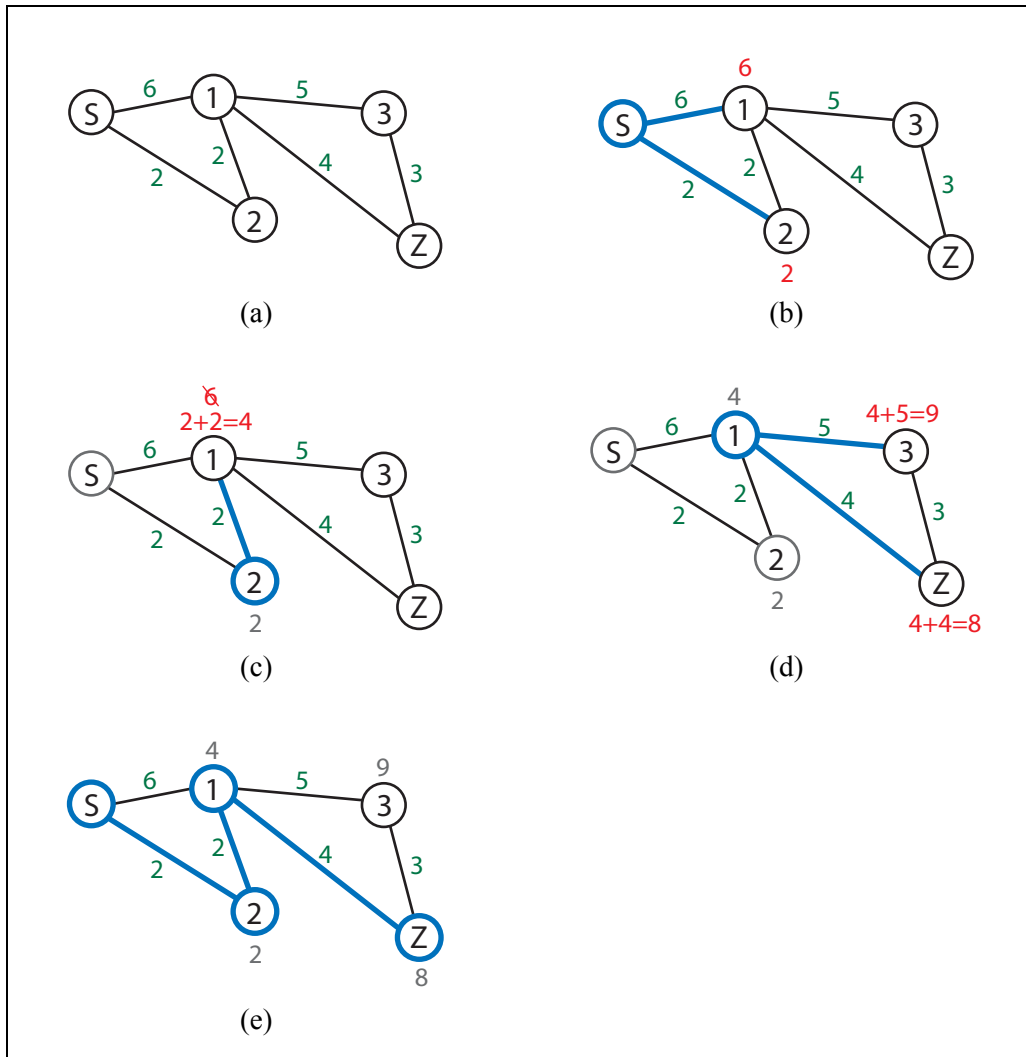


Abbildung 4.8: Ablauf des Shortest Path-Algorithmus nach Dijkstra.

Ein so erhaltener Weg wurde *nächster benachbarter Weg* genannt. Situationen wie in Abbildung 4.7 werden dadurch korrekt aufgelöst.

Das Modul baut aus der Kandidatenmenge einen Graphen auf und berechnet für jede Referenzstrasse alle möglichen *shortest paths* zwischen den Knotenkandidaten der beiden Endknoten. Es wird also nicht vorausgesetzt, dass Endknoten schon eindeutig zugeordnet worden sind. Die Zahl der Wege steigt so zwar schnell mit der Anzahl Kandidatenknoten (beispielsweise ergibt eine Strasse mit zwei Kandidaten für den einen und drei Kandidaten für den anderen Endknoten  $2 \times 3 = 6$  Wege), die meisten Knoten haben jedoch zwischen einem und drei Kandidaten und somit bleibt der Rechenaufwand vertretbar.

Schliesslich entfernt das Modul alle Kandidatenstrassen, die nicht an einem nächsten benachbarten Weg teilnehmen. In der Nähe der Endknoten können noch unpassende Kandidatenstrassen vorhanden sein, weil deren eindeutige Zuordnung zu einem Kandidatenknoten noch nicht erfolgt ist. Dazwischen werden jedoch alle unpassenden Kandidaten weggefiltert (Abbildung 4.9).



Abbildung 4.9: Filterung durch das Modul *Nächste benachbarte Wege*. Rot: VECTOR200-Strasse. Grün: Knotenkandidaten. Grau/Schwarz: Alle Kandidatenstrassen. Schwarz: Kandidatenstrassen, welche Teil eines nächsten benachbarten Wegs sind.

#### 4.2.4.5 Modul *Linienverfolgung*

Nach der Anwendung des Moduls *Nächste benachbarte Wege* bleibt in Abbildung 4.9 rechts eine Gabelung in den Kandidatenstrassen, da der Referenzknoten dort zwei Kandidatenknoten hat. Falls sich die Kandidatenknoten bezüglich ihrer Zwischenwinkelsumme und dem Abstand zum Endknoten nicht signifikant unterscheiden, können sie nicht automatisch zugeordnet werden und der Benutzer müsste interaktiv eingreifen. Der linke Endknoten hat nur einen Kandidaten und kann eindeutig zugeordnet werden.

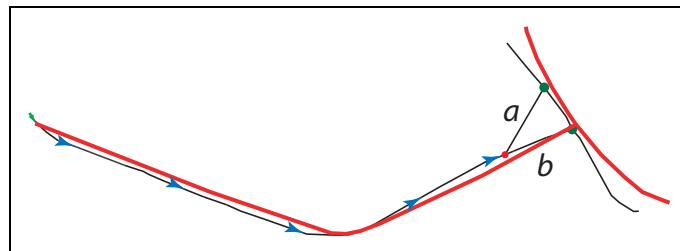


Abbildung 4.10: Die Linienverfolgung trifft auf eine Gabelung.

Diese Situation ergibt sich häufig und kann mit dem Modul *Linienverfolgung* automatisch aufgelöst werden. Der Prozess behandelt alle Referenzstrassen, bei denen der eine Endknoten zugeordnet werden kann, der andere jedoch mehrere Kandidatenknoten hat, zwischen denen keine Entscheidung möglich ist. Er beginnt beim zugeordneten Endknoten und folgt den Strassen der Kandidatenmenge entlang. Trifft er auf eine Gabelung (d. h. einen Knoten mit Grad  $> 2$ ), so vergleicht es für die fortsetzenden Strassen:

- die Winkelveränderung, d. h. den Winkel, den die fortsetzende Strasse mit der aktuellen Strasse einschliesst;
- die Hausdorff-Komponente zur Referenzstrasse.

In der Regel entstehen bei der Strassenführung Geraden und glatte Kurven; abrupte Richtungswechsel wie das Abbiegen an einer Kreuzung sind selten („Good Continuation“-Prinzip). Daher sollte die Strasse mit der kleineren Winkelveränderungen die richtige Fortführung sein. In wenigen Fällen trifft das „Good Continuation“-Prinzip nicht zu. Um dann keine Fehlentscheidung zu

treffen, wird zusätzlich die Hausdorff-Komponente einbezogen: Nur wenn beide Indikatoren die kleinsten Werte für dieselbe fortsetzende Strassen aufweisen, wird diese Strasse gewählt und die anderen fortsetzenden Strassen aus der Kandidatenmenge entfernt. Ansonsten bricht der Algorithmus ab, weil aufgrund der Kriterien keine Entscheidung getroffen werden kann.

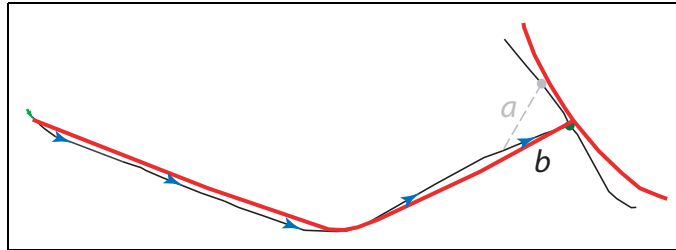


Abbildung 4.11: Situation nach der Auflösung der Gabelung durch die Linienverfolgung. Strassenkandidat *a* fällt weg, dadurch verbleibt nur noch ein Knotenkandidat, der eindeutig zugeordnet werden kann.

#### 4.2.4.6 Modul *Knotenmatching*

Aus der Menge der verbleibenden Kandidatenknoten muss schliesslich einer als korrespondierender Knoten identifiziert und dem Referenzknoten zugewiesen werden. Dies ist die Aufgabe des Moduls *Knotenmatching*.

Hat ein Referenzknoten keinen oder nur einen Kandidaten, so ist die Zuordnung einfach: Im erstgenannten Fall wird angenommen, dass es keine 1:1-Zuordnung gibt und der Benutzer muss später interaktiv eingreifen. Im zweitgenannten Fall ist der eine Kandidat mit grosser Wahrscheinlichkeit der homologe Knoten und kann zugeordnet werden.

Wenn mehr als ein Kandidat vorliegt, vergleicht das Modul die beiden Kriterien *Abstand zum Referenzknoten* und *mittlere Zwischenwinkelsumme*. Die Absolutwerte werden durch die Schwellwerte der Beschränkungen geteilt und so normiert. Auf die Berechnung der Schwellwerte wird in Abschnitt 4.2.5 eingegangen. Sie betragen 250 m für den Abstand und 45 Grad für die mittlere Zwischenwinkelsumme. Für jedes der beiden Kriterien wird eine Rangliste der Kandidaten erstellt. Damit ein Knoten eindeutig als homolog erkannt wird, muss gelten:

- Der Knotenkandidat steht in beiden Ranglisten an erster Stelle.
- Der Abstand zum zweitplatzierten Knoten ist beim Abstand zum Referenzknoten mindestens 0.3 und bei der mittleren Zwischenwinkelsumme mindestens 0.04. Die beiden Werte wurden empirisch ermittelt.

Gelten die beiden Bedingungen, wird der betreffende „Siegerkandidat“ dem Referenzknoten zugeordnet. Ansonsten wird das Problem als zum gegebenen Zeitpunkt unlösbar betrachtet und einem späteren Iterationsschritt des Matching-Ablaufes überlassen.

#### 4.2.4.7 Modul *Strassenmatching*

Die Voraussetzung für das Matching der Strassen ist, dass die Knoten vollständig zugeordnet wurden (entweder automatisch oder interaktiv). Dann gibt es für eine Referenzstrasse nur noch einen nächsten benachbarten Weg zwischen den beiden zugeordneten Endknoten. Der nächste

benachbarte Weg enthält alle Kandidatenstrassen, die zusammen das homologe Gegenstück der Referenzstrasse ergeben.

Das Modul Strassenmatching braucht also nur noch diese Kandidatenstrassen zu extrahieren und die Zuordnungsobjekte zu erstellen.

### 4.2.5 Bildung von Kandidatenmengen

In der ersten Phase wird für jede VECTOR200-Strasse und jeden VECTOR200-Knoten eine Menge von Kandidaten gebildet. Zwar liegt das Zwischenziel bei der Zuordnung der Knoten, jedoch müssen Strassenkandidaten trotzdem immer mitgeführt werden, da topologische Zuordnungskriterien verwendet werden. Das Flussdiagramm in Abbildung 4.12 zeigt noch einmal den Ablauf dieser Phase. Die Wirkungsweise des Prozesses soll an einem Beispiel illustriert werden. Abbildung 4.13 zeigt die VECTOR200-Strasse, die dafür verwendet wird, und ihr Umfeld. Die Strasse ist als Nebenstrasse 3 m klassiert.

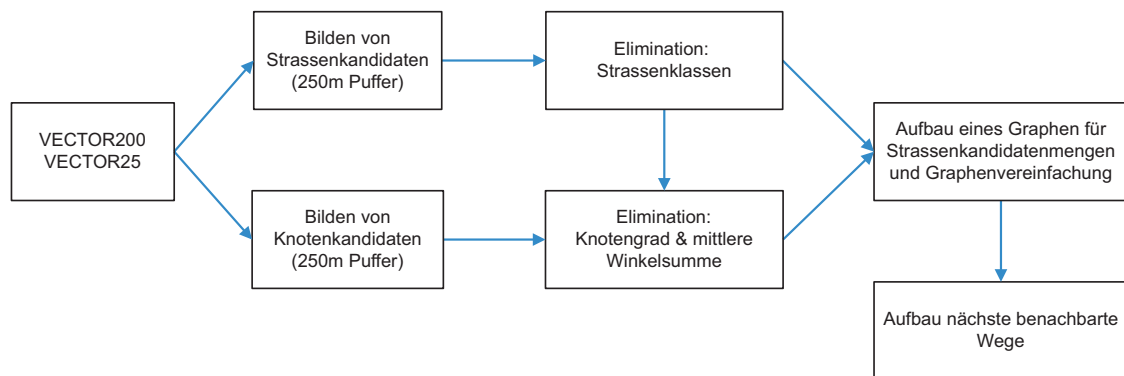


Abbildung 4.12: Flussdiagramm der Phase *Bildung von Kandidatenmengen für Strassen und Knoten*.

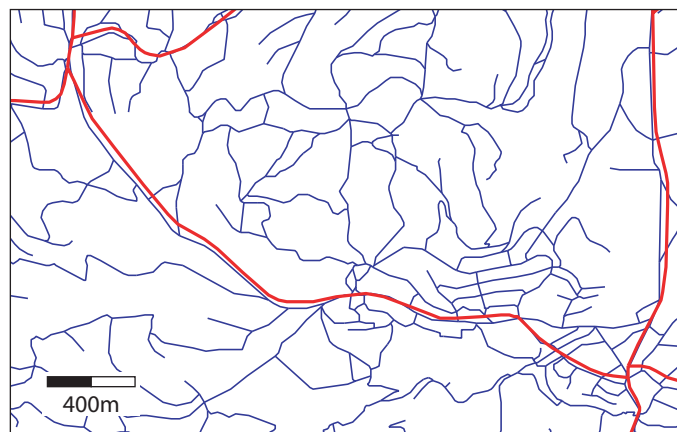


Abbildung 4.13: Beispiel zur Illustration der Kandidatenbildung. Rot: VECTOR200. Blau: VECTOR25.

### Bilden von Strassenkandidaten

Der erste Schritt ist der Aufbau von Kandidatenmengen für die VECTOR200-Strassensegmente. Kandidatenmengen für Knoten werden erst später gebildet, da dafür die Strassenkandidaten vorhanden sein müssen.

Zur Bestimmung der Puffergrösse wurde das Gebiet Pfäffikon ZH von Hand verknüpft und die Matches statistisch ausgewertet. Die Prozedur zur Ermittlung der Puffergrösse testet, ob alle VECTOR25-Strassen, die einer VECTOR200-Strasse zugeordnet wurden, innerhalb eines Puffers der Grösse  $X$  um die VECTOR200-Strasse liegen. Die Puffergrösse  $X$  wird iterativ so lange vergrössert, bis die Bedingung erfüllt wird. Für Pfäffikon konnte so eine Grösse von  $X = 200$  m ermittelt werden. Da angenommen wurde, dass der Puffer bei anderen Datensätzen höher sein könnte, wurde pauschal eine Puffergrösse von 250 m um Strassensegmente als auch um Knoten gesetzt. Die Puffergrösse stellt eine implizite Beschränkung des Abstands auf 250 m dar.

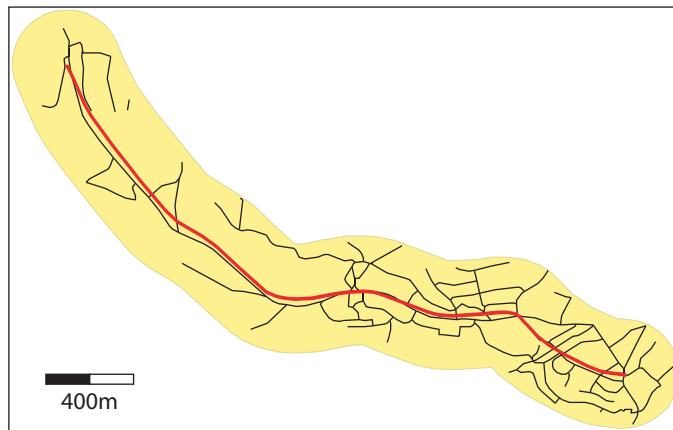


Abbildung 4.14: 250 m-Puffer um das VECTOR200-Strassensegment (gelb) und VECTOR25-Segmente, die innerhalb dieses Puffers liegen (schwarz).

### Elimination: Strassenklasse

Die Anwendung der Beschränkung *Strassenklasse* auf die Strassenkandidaten ist je nach Gebietscharakter unterschiedlich hilfreich: In schwach besiedelten Gebieten bleiben meist nur die korrekten homologen VECTOR25-Strassen übrig, da dort eine einzelne gut ausgebaute Strasse zwischen vielen 5. und 6. Klass-Strassen durchführt. Die kleineren Strassen werden aber generell in VECTOR200 nicht repräsentiert und fallen weg. In Ortschaften ist der Effekt wegen des dort gut ausgebauten Strassennetzes relativ gering. In dem hier angeführten Beispiel fallen durch die Anwendung der Strassenklassen kaum VECTOR25-Kandidaten weg, weil die östliche Hälfte der Strasse in der Ortschaft Adetswil liegt. Dort sind viele Strassen als Quartierstrassen oder 2. Klass-Strassen klassiert und sind damit potentielle Matching-Partner für eine Nebenstrasse 3 m.

### Bilden von Knotenkandidaten

Nach diesem Schritt werden Knotenkandidaten ermittelt. Kandidaten sind alle VECTOR25-Knoten, die innerhalb eines 250m-Puffers um den jeweiligen VECTOR200-Knoten liegen. Abbildung 4.16 zeigt das Ergebnis für die Endknoten der Beispiel-Strasse. Die dargestellten Stras-

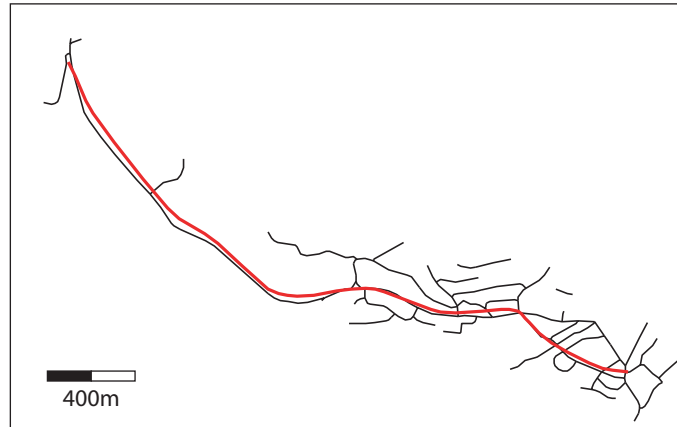


Abbildung 4.15: Kandidatenmenge nach der Anwendung der Beschränkung *Strassenklasse*.

senkandidaten (schwarz) stammen aus den Kandidatenmengen aller zu den VECTOR200-Knoten inzidenten VECTOR200-Strassen.

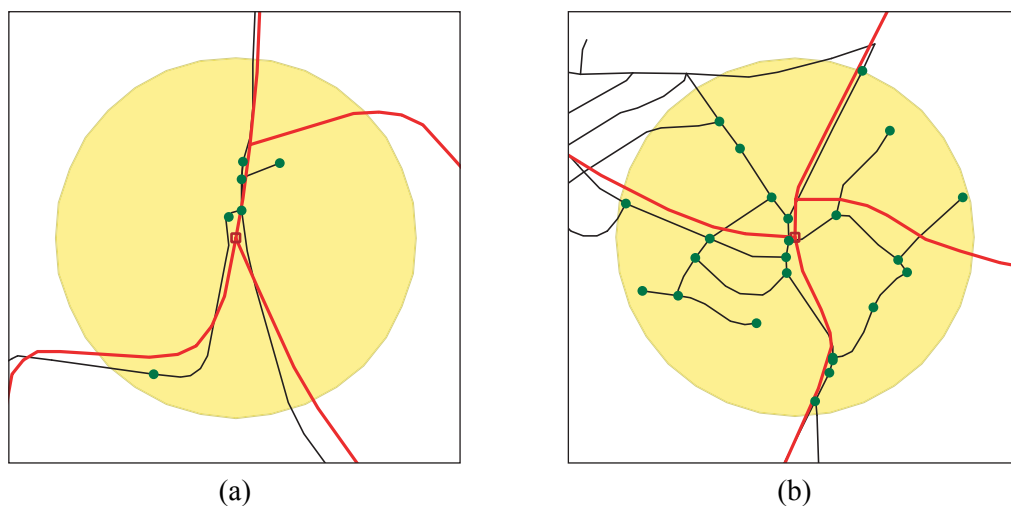


Abbildung 4.16: Umgebung der beiden Endknoten (rote Rechtecke) der Beispiels-Strasse und deren Knotenkandidaten. (a) linker Endknoten (b) rechter Endknoten.

### Elimination: Knotengrad & mittlere Zwischenwinkelsumme

Viele der Kandidatenknoten passen nicht zum VECTOR200-Knoten, weil sie einen zu kleinen Grad haben. Diese werden im nächsten Schritt aus der Kandidatenmenge entfernt. Die Kandidatenknoten werden auch durch eine Beschränkung der im Abschnitt 4.2.4.3 erläuterten mittleren Zwischenwinkelsumme gefiltert. Wie man im Histogramm von Abbildung 4.17 sieht, nimmt die Anzahl der homologen Knotenpaare mit grösseren mittleren Zwischenwinkelsummen rasch ab; der höchste Wert liegt bei 35 Grad. Die mittleren Zwischenwinkelsummen für alle anderen Paarungen bestehend aus Referenzknoten – Kandidatenknoten sind breiter gestreut. Erst bei über 65 Grad reduziert sich die Anzahl der Knotenpaare deutlich.

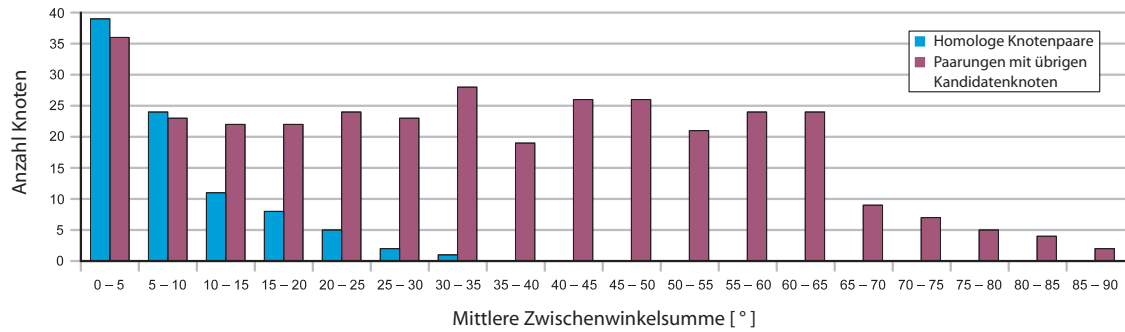


Abbildung 4.17: Histogramm der mittleren Zwischenwinkel für das von Hand zugeordnete Gebiet Pfäffikon.

Der Schwellwert für die mittlere Zwischenwinkelsumme wurde daher auf 45 Grad gesetzt: Alle Kandidatenknoten mit einer grösseren Zwischenwinkelsumme werden aus der jeweiligen Kandidatenmenge entfernt. Abbildung 4.18 zeigt das Ergebnis dieser beiden Filterungen auf die beiden Endknoten des Beispiels.

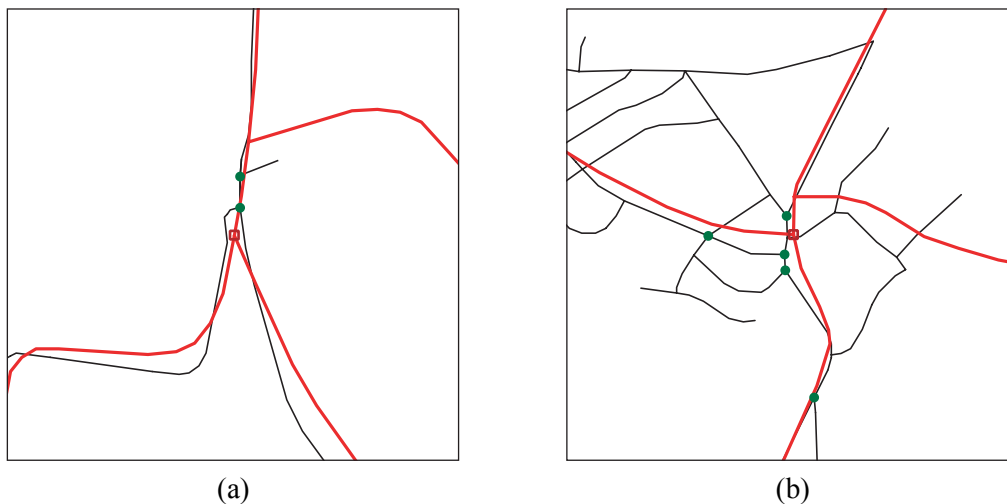


Abbildung 4.18: (a) Der linke Endknoten und (b) der rechte Endknoten nach der Anwendung der Beschränkungen *Knotengrad* und *mittlere Zwischenwinkelsumme*.

### Aufbau eines Graphen für Strassenkandidatenmengen

Mit den so erhaltenen Strassen- und Knotenkandidaten wird ein Graph aufgebaut. Dieser wird für das nachfolgende Modul *Nächste benachbarte Wege* benötigt. Dabei werden auch gleich Sackgassen und kurze, vom Rest des Graphen abgetrennte Teilgraphen entfernt, sofern sie nicht mehr als 25% der Länge der Referenzstrasse aufweisen. Sackgassen und abgetrennte Teilstücke würden zwar auch durch das nachfolgende Modul entfernt werden – jedoch ist die Berechnung einer Hausdorff-Distanz eine rechnerisch aufwändige Operation, weshalb man die Anzahl dieser Operationen möglichst klein halten möchte.

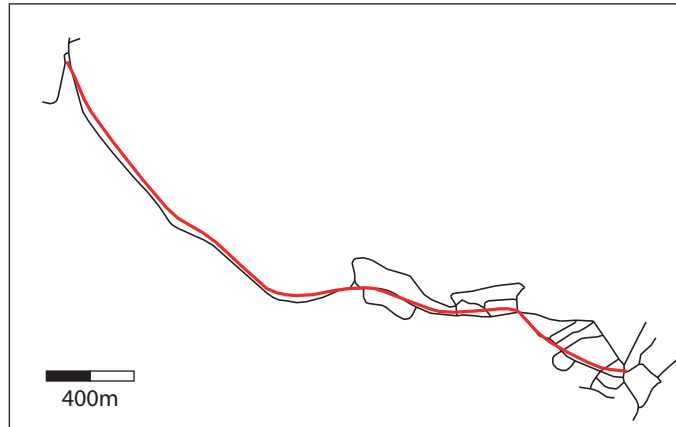


Abbildung 4.19: Kandidatenmenge nach der Bildung des Graphen und Entfernen von Sackgassen und der separaten Teilgraphen.

### Aufbau nächste benachbarte Wege

Mit dem Graphen und den Kandidaten für die beiden Endknoten kann nun das Modul *Nächste benachbarte Wege* ablaufen. Die Knotenkandidaten aus Abbildung 4.18 sind in Abbildung 4.20 als grüne Kreise dargestellt. Es werden alle nächsten benachbarten Wege zwischen den Kandidaten des linken und den Kandidaten des rechten Endknotens der VECTOR200-Strasse berechnet. Alle Kandidatenstrassen, die nicht an einem Weg teilnehmen, werden eliminiert, so dass nur noch die in Abbildung 4.20 dargestellten Strassenkandidaten übrig bleiben. Man beachte, dass durch die Reduktion der Kandidatenstrassen auch mehrere Kandidatenknoten ungültig geworden sind, da sie nun einen zu kleinen Knotengrad haben. Sie können daher durch eine wiederholte Anwendung des Moduls *Beschränkung: Knotengrad* aus der Kandidatenmenge entfernt werden. Das Modul wird aber erst in der Phase „Matching der Knoten“ als Teil des iterativen Prozesses aufgerufen werden (siehe Abschnitt 4.2.6). Abbildung 4.21 zeigt das Ergebnis der wiederholten Knotenelimination. Im angeführten Beispiel hat jeder Endknoten nur noch einen Knotenkandidaten.

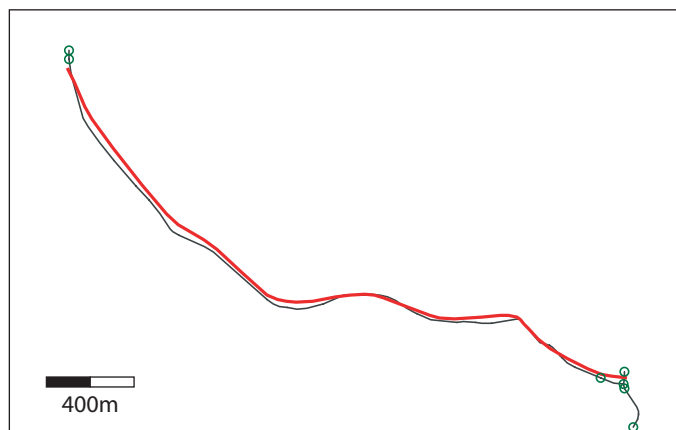


Abbildung 4.20: Kandidatenmenge nach der Bildung der nächsten benachbarten Wege. Die grünen Kreise bezeichnen die Knoten, für welche nächste benachbarte Wege berechnet wurden.

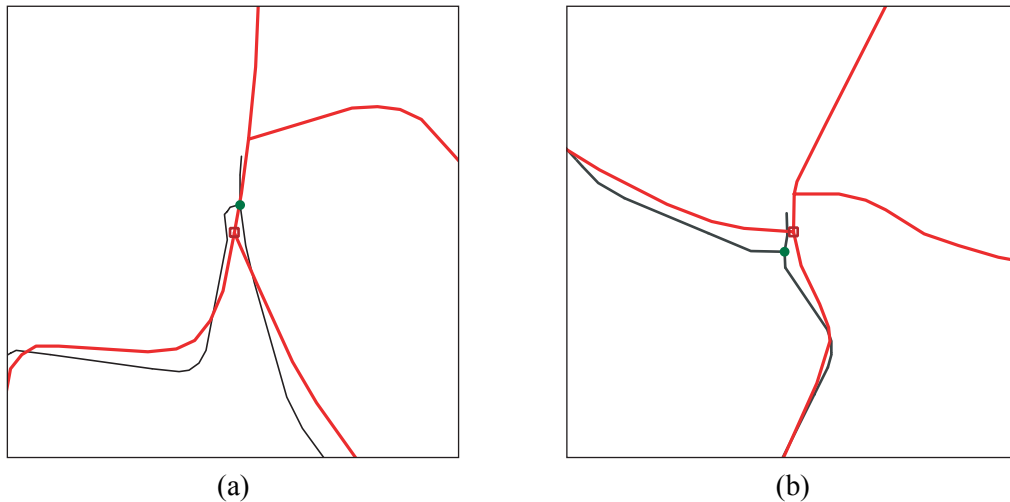


Abbildung 4.21: Nächste benachbarte Wege für die zu den Endknoten angrenzenden Strassen und die verbleibenden Knotenkandidaten.

## 4.2.6 Matching der Knoten

Im Fall des in Abschnitt 4.2.5 angeführten Beispiels verbleibt nur noch ein Knotenkandidat pro Endknoten. Sie können somit eindeutig zugeordnet werden. In der Regel hat jedoch ein Referenzknoten auch nach der Filterung mehrere Kandidatenknoten. In der Matching-Phase werden die verbliebenen Kandidatenknoten in einem iterativen Prozess ausgewertet und zugeordnet:

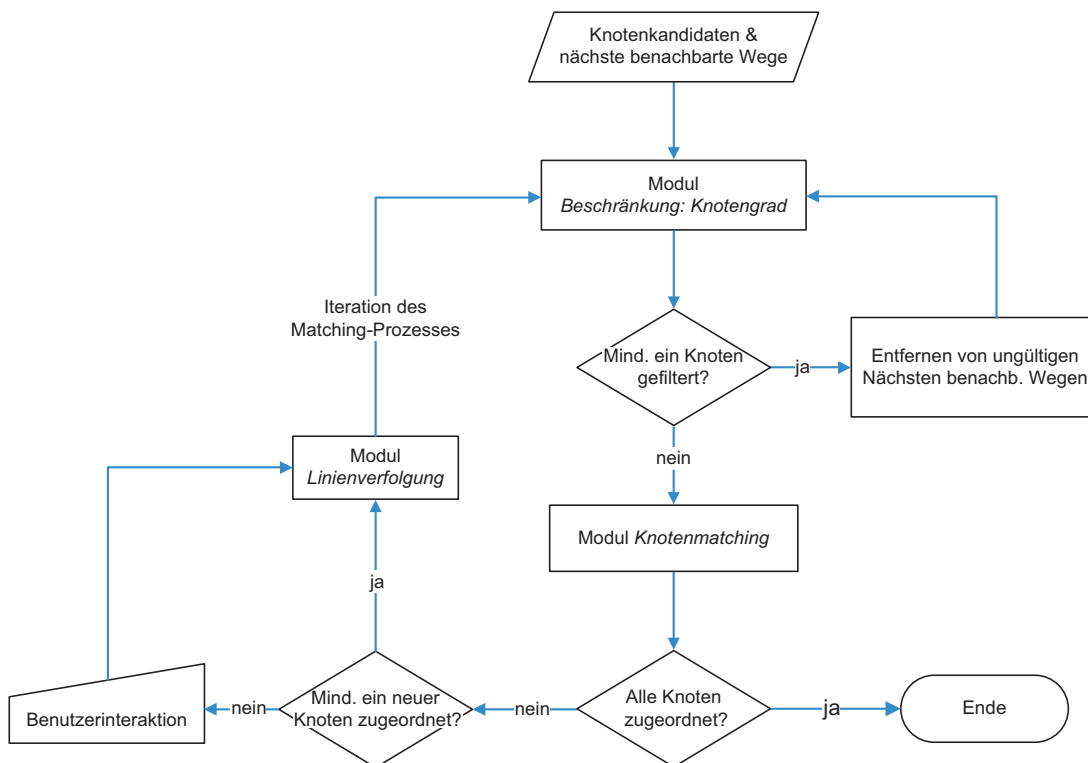


Abbildung 4.22: Flussdiagramm der Phase Matching der Knoten.

Abbildung 4.22 zeigt den Ablauf dieser Phase. Zentral ist das Modul *Knotenmatching*, das Zuordnungen aufgrund der Kriterien *Distanz zum Referenzknoten* und *mittlere Zwischenwinkelsumme* vornimmt. Im ersten Iterationsschritt können meist nicht alle Knoten eindeutig zugewiesen werden. Mit jedem zugeordneten Knoten vereinfacht sich aber die Situation, weil die nächsten benachbarten Wege, die zu einem im Matching-Modul unterlegenen Knotenkandidaten führen, entfernt werden, und weil durch die Linienverfolgung inkorrekte Strassensegmente eliminiert werden können. Dadurch, dass bei beiden Vorgängen Strassensegmente wegfallen, können weitere Kandidatenknoten entfernt werden, weil ihr Knotengrad zu klein wird.

Konnte in einem Iterationsschritt kein neuer Knoten zugeordnet werden, so muss der Benutzer eingreifen. Die Applikation zeigt dann dem Benutzer einen noch nicht verknüpften Knoten mit seinen Kandidatenknoten. Der Benutzer wählt aus den Vorschlägen den korrespondierenden Kandidatenknoten aus und setzt den Prozess fort.

Die Prozedur endet, wenn alle Referenzknoten entweder einem Kandidatenknoten zugeordnet oder als Knoten ohne Korrespondenten klassifiziert worden sind.

### 4.2.7 Matching der Strassen

Auf Anstoss des Benutzers werden die Knotenzuordnungen zu Strassenzuordnungen umgesetzt, wie es in Abschnitt 4.2.4.7 beschrieben wird. Strassen, welche an einem Knoten ohne Korrespondenten enden, können nicht behandelt werden – sie müssen in der Nachbearbeitungsphase von Hand zugeordnet werden.

### 4.2.8 Nachbearbeitung

Nicht alle Strassen können durch die beschriebene Methode automatisch zugeordnet werden. Knoten, bei denen eine 1:N-Beziehung vorliegt, bleiben ohne Korrespondenz und somit können auch die damit verbundenen Strassen nicht zugeordnet werden. Obwohl bei der Entwicklung der Methode darauf geachtet wurde, dass Falschzuordnungen unwahrscheinlich sind, können diese nicht ganz ausgeschlossen werden.

In der Nachbearbeitungs-Phase müssen somit die Verknüpfungen vom Benutzer ergänzt, kontrolliert, und eventuell verbessert werden. Im Vergleich zum vollständig manuellen Matching ist der Zeitaufwand für die Nachbearbeitung aber gering.

Der Benutzer kann während der Nachbearbeitung durch das Matching-System mit verschiedenen Massnahmen unterstützt werden. Wichtig ist eine klare Darstellung der Zuordnungen, so dass der Benutzer nicht bzw. falsch zugeordnete Strassen einfach erkennt. Für die Verknüpfungen kann zudem ein Vertrauensmass berechnet werden, das die Wahrscheinlichkeit einer korrekten Zuordnung angibt (siehe Abschnitt 7.3.1.1). Es kann entsprechend visualisiert werden und so dem Benutzer helfen, unsichere Situationen zu erkennen. Eine weitere Möglichkeit ist, dass der Benutzer vom System auf unverknüpfte Objekte und Verknüpfungen mit kleinem Vertrauensmasszahlen aktiv hingewiesen wird. In Abschnitt 5.5.5 wird gezeigt, wie dies realisiert werden kann.



## Kapitel 5

# Implementation des Prototyps

Ein Prototyp diene als Versuchsplattform für den in Kapitel 4 vorgestellten Matching-Ansatz. Dieses Kapitel geht auf die Implementation des Prototyps ein. In einem der Schwerpunkte des Kapitels werden Möglichkeiten untersucht, Verknüpfungen zwischen korrespondierenden Linienelementen zu visualisieren.

### 5.1 Anforderungen

Für den Prototyp der Matching-Applikation ergaben sich folgende Anforderungen:

- *Visualisierung*: Die mit dem Matching-Algorithmus erzeugten Verknüpfungen müssen so visualisiert werden, dass der Benutzer gut erkennen kann, welche Objekte miteinander verknüpft sind, und welche Art von Beziehung zwischen den Objekten vorliegt. Diese Frage ist insbesondere im Hinblick auf den künftigen praktischen Einsatz von MRDB interessant.
- *Benutzerinteraktion*: Wie in den Kapiteln 3 und 4 gezeigt wurde, können einzelne Teilschritte von Matching-Prozessen ein Eingreifen des Benutzers erfordern. In der Nachbearbeitung und für die manuelle Verknüpfung von Datensätzen muss der Benutzer Verknüpfungen von Hand erzeugen und manipulieren können. Dazu müssen geeignete Werkzeuge vorhanden sein.
- *Persistenz*: Die Erzeugung einer MRDB ist zu zeitaufwändig für eine ad hoc-Lösung. Die erzeugten Verknüpfungen müssen daher für eine spätere Verwendung permanent gesichert werden.

### 5.2 Software-Komponenten

Für die Erstellung von räumlichen Applikationen existiert heute eine Anzahl von verschiedenen Lösungen, die gewisse Basis-Operationen (geometrische Datentypen und Operationen, Visualisierung, etc.) bereitstellen. Ein Beispiel ist die kommerzielle Software ESRI ArcGIS, die mit verschiedenen Programmiersprachen flexibel zum Aufbau fachspezifischer Applikationen benutzt werden kann. Der Autor hat sich für eine Lösung von Vivid Solutions entschieden, die frei als Open Source erhältlich ist. Dadurch konnte der Autor eine MRDB mitbenutzen, die am Institut in Zusammenarbeit mit der Firma Axes Systems AG entstand.

In diesem Abschnitt werden die Software-Komponenten besprochen, die im Prototyp verwendet wurden. Abbildung 5.1 zeigt das Zusammenspiel der Komponenten. Die Komponente *RoadMatcher* beinhaltet diejenigen Elemente, die der Autor selbst entwickelt hat. Als Programmiersprache wurde Java 1.4 verwendet.

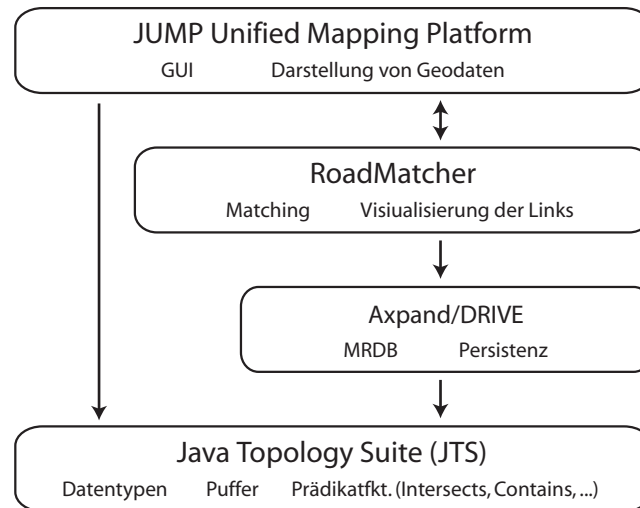


Abbildung 5.1: Software-Komponenten des Prototyps.

### 5.2.1 Java Topology Suite (JTS)

Die *Java Topology Suite (JTS)* ist eine Implementation eines räumlichen Datenmodells nach der *OpenGIS Simple Features Specification For SQL*<sup>1</sup> und von einigen wichtigen Algorithmen für geometrische Objekte. JTS wurde von Vivid Solutions<sup>2</sup> entwickelt und als Open Source unter der GNU LGPL<sup>3</sup> publiziert. Wichtige Funktionalitäten, die von der JTS bereitgestellt werden, sind:

- Räumliche Datentypen: *Point*, *LineString*, *Polygon*, *GeometryCollection*, etc.
- Pufferzonen
- Overlay-Funktionen: Schnittmenge/Differenz der Geometrien zweier Objekte
- Räumliche Prädikatfunktionen: Diese überprüfen topologische Relationen zwischen zwei geometrischen Objekten. Die theoretische Grundlage bildet das Modell der *9-intersection matrix* (Egenhofer 1991). Von den möglichen Relationen werden folgende unterstützt: *equals*, *disjoint*, *intersects*, *touches*, *crosses*, *within*, *contains*, *overlaps*.
- Räumliche Indizierung mittels Quadtree oder einer (nur lesbaren) R-tree-Version (*STR packed R-tree*) (siehe Rigaux et al. 2002). Im Matching-Prototyp *RoadMatcher* wird allerdings eine am Institut entwickelte Version eines R-trees verwendet.

1. <http://www.opengeospatial.org/specs/>, Stand 20.12.2005

2. <http://www.vividsolutions.com>, Stand 20.12.2005

3. <http://www.gnu.org/licenses/licenses.html>, Stand 20.12.2005

## 5.2.2 JUMP Unified Mapping Platform

JUMP stammt ebenfalls von Vivid Solutions und ist als Open Source unter der GNU GPL<sup>1</sup> publiziert. JUMP bietet ein Framework für Applikationen mit räumlichem Bezug. JUMP nutzt dabei die Funktionen von JTS. Die Geo-Objekte in JUMP bestehen aus der Geometrie und einer Liste von Attributen. Die Benutzeroberfläche erlaubt eine intuitive Interaktion mit den Geodaten. Sie ermöglicht eine flexible Konfiguration von Menüs, Werkzeugleisten und Dialogboxen. Das Framework kann beliebig erweitert werden: Einzelne Programmkomponenten (*Plug-ins*) können beim Programmstart geladen und dem Benutzer als Menüpunkt oder als Schaltflächen zugänglich gemacht werden. Zudem können die bestehenden Klassen in ihrer Funktionalität durch objektorientierte Vererbung geändert und erweitert werden.

## 5.2.3 Axpand/DRIVE<sup>2</sup>

Das Projekt DRIVE (*Derivation of Vector Models*) ist ein Kooperationsprojekt zwischen dem Geographischen Institut der Universität Zürich und Axes Systems AG. Der Schwerpunkt des Projektes liegt in der Erweiterung des *Axpand*-Systems. Die Verbesserungen betreffen die Verknüpfung der digitalen Vektormodelle in verschiedenen Massstäben mittels Multirepräsentationsdatenbanken (MRDB), aber auch die automatische Generalisierung durch Einbezug von Topologie, Nachbarschaftsrelationen, und eines Workflow-Managements, welches den Ablauf von Generalisierungsoperatoren steuert.

Den Kern dieser Neuentwicklungen bildet eine MRDB. Sie ist eine von *Axpand* unabhängige Java-Applikation, es können jedoch Daten über eine XML-Schnittstelle mit *Axpand* ausgetauscht werden. Auf die Modellierung der MRDB wird in Abschnitt 5.3 näher eingegangen.

## 5.2.4 Java Matrix Package (JAMA)

JAMA stellt einige fundamentale Funktionen der linearen Algebra bereit. Mit JAMA können u.a. lineare Gleichungssysteme numerisch stabil gelöst werden. Beim verwendeten Algorithmus zur Berechnung der Hausdorff-Distanz (Hangouët 1995) müssen Linienschnitte berechnet werden. Das Problem lässt sich als lineares Gleichungssystem formulieren und mit JAMA lösen.

# 5.3 Modellierung der MRDB

## 5.3.1 Modellierung von Repräsentationen

Das Datenmodell von Axpand/DRIVE entspricht einer schwachen Integration: Die Repräsentationen existieren unabhängig voneinander, Objekte von verschiedenen Repräsentationen werden durch Verknüpfungsobjekte miteinander verbunden. Abbildung 5.2 zeigt das Datenmodell einer Repräsentation. Sie wird durch die Klasse *GenResolution* modelliert. Kartenobjekte sind je nach räumlicher Ausprägung vom Typ *GenPointObject*, *GenLineObject*, oder *GenAreaObject*. Die Objekte tragen eine Objektgeometrie (in der Abbildung Rechtecke mit transparentem Hinter-

1. <http://www.gnu.org/licenses/licenses.html>, Stand 20.12.2005

2. Die Ausführungen in diesem Abschnitt beruhen weitgehend auf Bobzien et al. (2005).

grund) und eine beliebige Anzahl von Attributen. Bei punkt- und linienförmigen Objekten kann zusätzlich auch eine flächenhafte Signaturgeometrie (in der Abbildung grau schattierte Rechtecke) erzeugt werden. Diese ist wichtig für die automatische Generalisierung, insbesondere für die Verdrängung benachbarter Objekte. Für das Matching spielt die Signaturgeometrie hingegen keine Rolle und wurde im Prototyp deshalb nicht verwendet.

Die Topologie wird für jede Repräsentation in einem Graphen (Klasse *PlanarGraph*) gespeichert. Die Elemente der Topologie sind mit den korrespondierenden Kartenelementen verknüpft: Beispielsweise hat jede Instanz von *GenLineObject* als Partner eine oder mehrere *Edge*-Instanzen aus *PlanarGraph*. Die korrespondierenden Instanzen von *Edge* repräsentieren dieselbe Geometrie (als *JTSTopoLineString*), enthalten aber zusätzlich topologische Informationen. So wird gespeichert, von welchen beiden Knoten (*Node*-Instanzen) die Linie begrenzt wird, und welche beiden Flächen (*Face*-Instanzen) links und rechts der Linie liegen. Eine Instanz von *GenLineObject* wird mit mehr als einem *Edge*-Objekt verknüpft, wenn sich zwei Linien kreuzen. Die Linie wird dann am Kreuzungspunkt aufgetrennt und für jede der beiden Teillinien wird ein *Edge*-Objekt erzeugt.

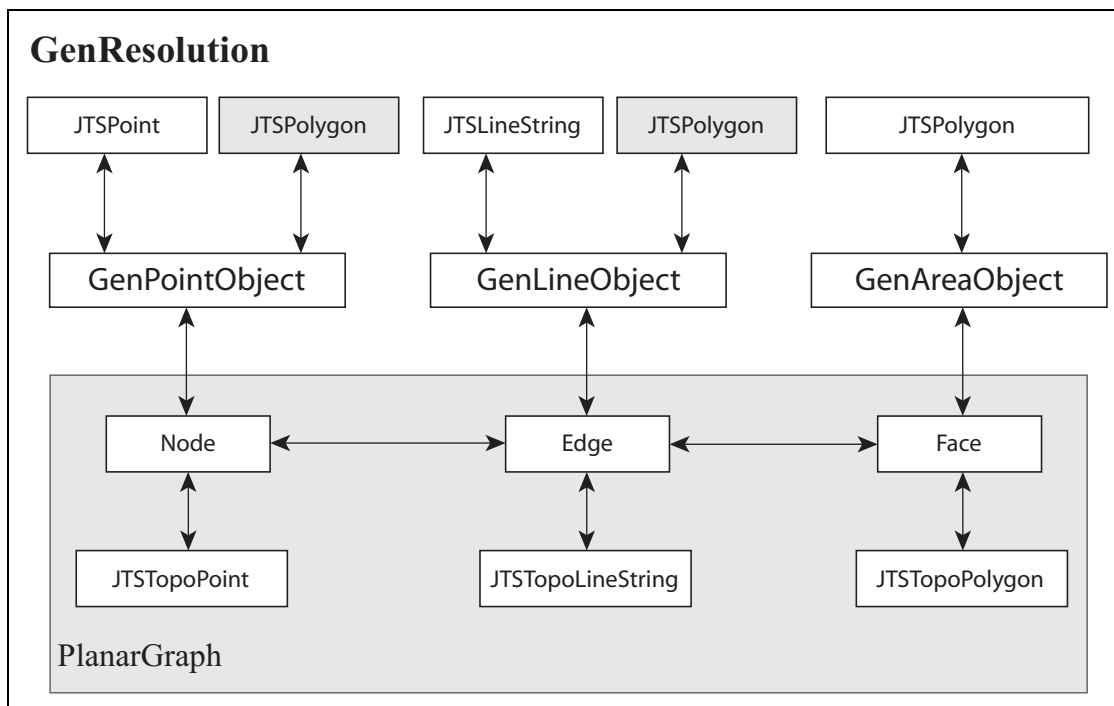


Abbildung 5.2: Axpand/DRIVE Datenmodell einer Repräsentation.

Die drei Klassen *GenPointObject*, *GenLineObject* und *GenAreaObject* haben eine gemeinsame Superklasse *GenObject*. Abbildung 5.3 zeigt die wichtigsten Eigenschaften von *GenObject* in UML-Notation. Sie enthält erstens eine Liste von beliebigen Attributen im Vector *attributes*. Zweitens trägt sie auch die Menge der Verknüpfungen des Kartenobjekts zu korrespondierenden Objekten der nächsthöher und nächsttiefer aufgelösten Repräsentationen. Auf die Verknüpfungen wird in Abschnitt 5.3.3 näher eingegangen. Im Klassendiagramm von *GenLineObject* ist ne-

ben der Objektgeometrie eine Eigenschaft vom Typ *jumpExtension.VisFeature* eingetragen. *VisFeature* ist eine Adapterklasse für die Visualisierung und Interaktion mit JUMP.

### 5.3.2 Modellierung der Strassen-Objekte

VECTOR25- und VECTOR200-Strassen werden durch die Klasse *GenObjectStreet* modelliert, die von *GenLineObject* abgeleitet ist (Abbildung 5.3). Sie ergänzt *GenLineObject* um die in der Matching-Applikation benötigten Attribute *SwisstopoId*, *Strassenklasse* und *Konstruktionsart*.

Kreisel in VECTOR25 kollabieren in der Regel zu einzelnen Knoten in VECTOR200 (siehe Kapitel 4.1.4). Dies entspricht einer 1:N-Beziehung zwischen VECTOR25-Strassen und einem VECTOR200-Knoten. In diesem Fall wird ein Aggregations-Objekt (Klasse *GenObjectRoundabout*) erzeugt und in der Eigenschaft *genParentObject* der beteiligten Strassen eingetragen. Aber auch *GenObjectRoundabout* hält eine Liste von allen Strassen, die Teil des Kreisverkehrs sind (Attribut *streetSegments*). Für die Visualisierung werden in der Klasse *GenObjectRoundabout* zudem noch die Position des Zentroids, die während der Erzeugung im Konstruktor berechnet wird, und die Kreiselfläche als Polygon mitgeführt.

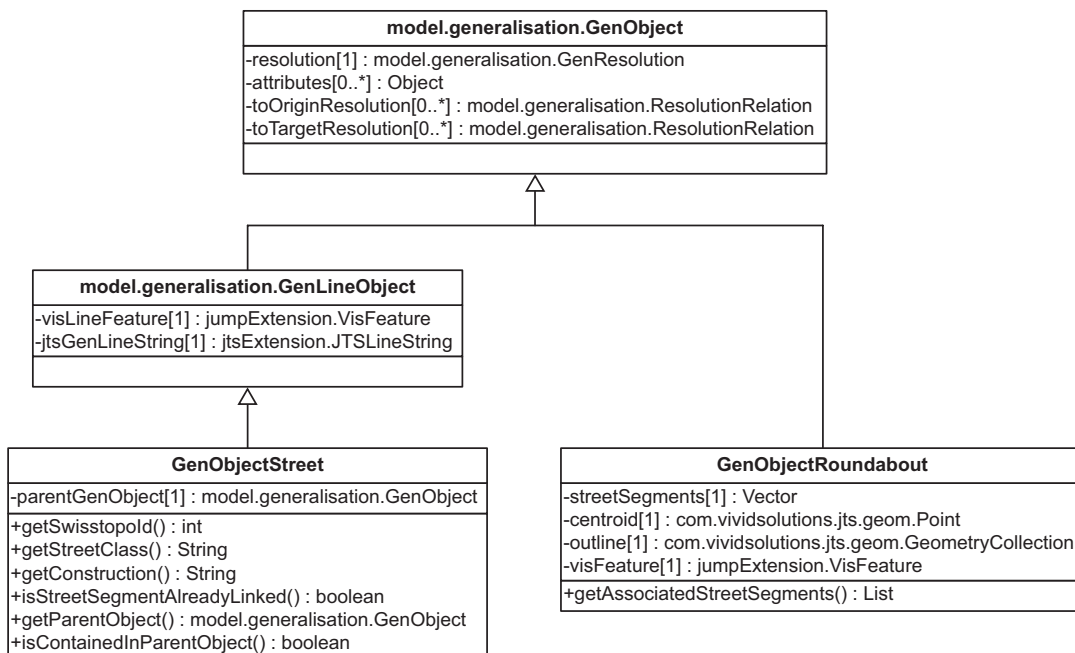


Abbildung 5.3: UML-Klassendiagramm von *GenObjectStreet* und *GenObjectRoundabout*.

### 5.3.3 Modellierung von Verknüpfungen

Eine Verknüpfung wird durch eine eigene Klasse *ResolutionRelation* modelliert. Dadurch kann die Verknüpfung zusätzliche Information tragen. Bei durch Generalisierung erzeugten Verknüpfungen sind dies beispielsweise die verwendeten Generalisierungsoperatoren und ihre Parameter; im Fall von durch Matching erzeugten Verknüpfungen macht die Sicherheit, mit welcher auf einen Match geschlossen wird, Sinn.

Eine Instanz von *ResolutionRelation* modelliert nur eine 1:1-Verknüpfung zwischen zwei Kartenobjekten aus verschiedenen Repräsentationen. N:1-Beziehungen und N:M-Beziehungen lassen sich prinzipiell auf zwei verschiedene Arten erzeugen:

- *Implizit*, indem  $n \times m$  Instanzen von *ResolutionRelation* erzeugt werden, welche die  $n$  Objekte der einen mit den  $m$  Objekten der anderen Repräsentation einzeln verknüpfen.
- *Explizit*, indem die  $n$  Objekte der einen und die  $m$  Objekte der anderen Repräsentation zu Aggregations-Objekten verknüpft werden. Die Aggregations-Objekte können mit einem einzigen Verknüpfungsobjekt mit dem Objekt der anderen Repräsentation verbunden werden.

Als Beispiel soll der Strassenkreisel aus Abbildung 5.20a auf beide Arten verknüpft werden. Der Kreisell besteht aus vier VECTOR25-Strassen, die mit einem einzigen VECTOR200-Knoten verknüpft sind. Es handelt sich somit um eine N:1-Beziehung zwischen den VECTOR25-Strassen und dem VECTOR200-Knoten.

Abbildung 5.4 zeigt ein UML-Objektdiagramm der implizit modellierten Verknüpfung. Für jede der vier Strassen des Kreisells muss eine Instanz von *ResolutionRelation* erzeugt und mit dem VECTOR200-Knoten (eine Instanz von *GenPointObject*) verknüpft werden. Der Nachteil dieser Art der Modellierung ist, dass nirgends festgehalten wird, dass die vier VECTOR25-Strassen an derselben N:1-Beziehung teilnehmen (d. h. dass sie demselben Kreisverkehr angehören). Für eine Rekonstruktion des Kreisells müssen die beteiligten Strassen später wieder gesammelt werden. Zudem kann ein Kreisell nur dann existieren, wenn er mit einer Instanz von *GenPointObject* verknüpft ist. Dies ist ungünstig, wenn Kreisell vor dem eigentlichen Matching durch eine entsprechende automatische Prozedur erkannt werden sollen.

Deshalb wurde im Prototyp eine implizit modellierte Verknüpfung der Kreisell umgesetzt. Sie wird in Abbildung 5.5 dargestellt. Das Aggregationsobjekt wird durch eine Instanz der Klasse *GenObjectRoundabout* repräsentiert. Da sie eine Unterklasse von *GenObject* ist, lässt sie sich mit einer beliebigen anderen Instanz vom Typ *GenObject* verknüpfen. Diese Lösung ist vorzuziehen, da das Aggregationsobjekt auch ohne Verknüpfung existieren und zusätzliche Informationen über das aggregierte Objekt speichern kann.

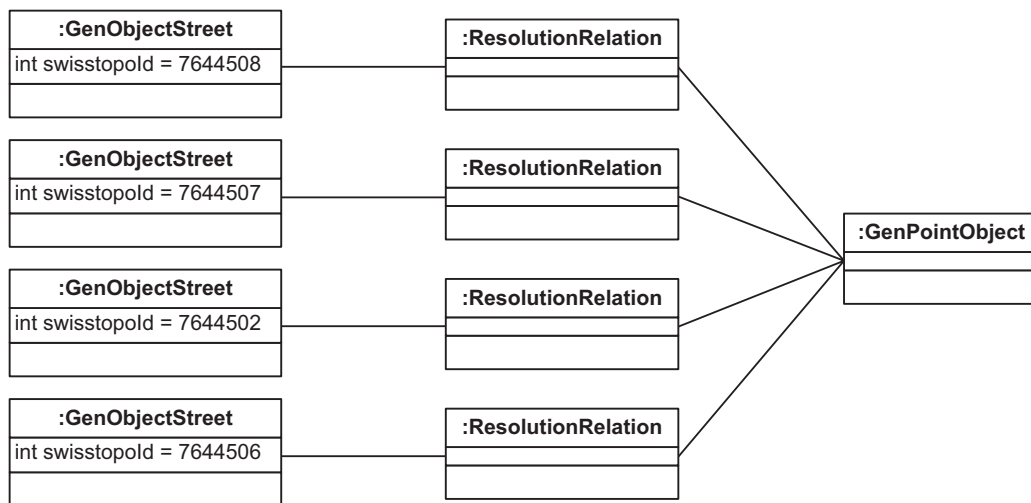


Abbildung 5.4: Implizite Modellierung einer 1:N-Verknüpfung.

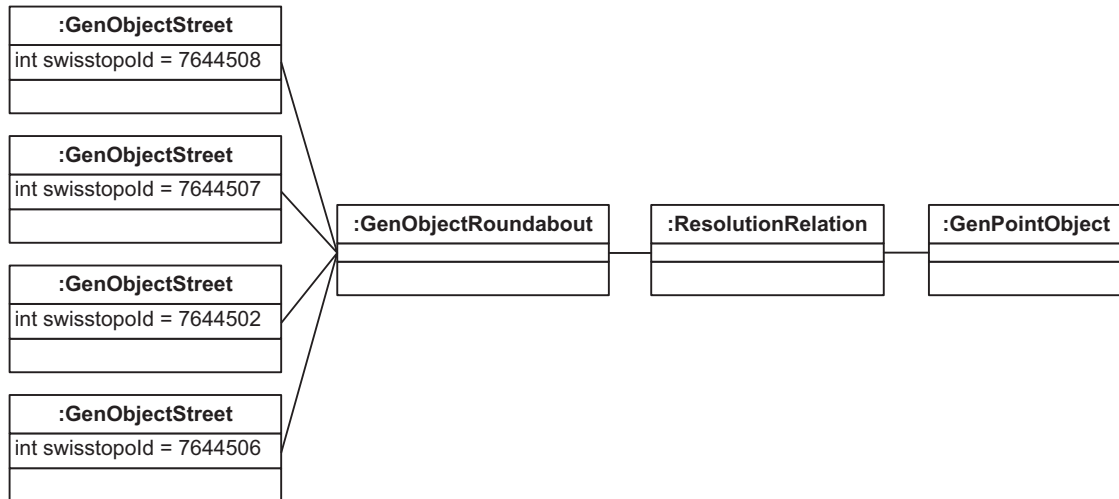


Abbildung 5.5: Explizite Modellierung einer 1:N-Verknüpfung.

## 5.4 Persistenz

Die Funktion zur permanenten Speicherung der MRDB war in Axpand/DRIVE noch nicht vorhanden und musste neu implementiert werden. Es standen zwei alternative Ansätze zur Diskussion:

- *Dateien*: Die MRDB kann in Dateien exportiert werden. Die Logik für Organisation und Zugriff auf einzelne Datensätze muss selbst implementiert werden.
- *Datenbankverwaltungssystem (DBMS)*: Ein DBMS ist eine Sammlung von Programmen, die die anwendungsunabhängige, dauerhafte Speicherung von Daten in einer Datenbank ermöglicht und die damit verbundene Verwaltung übernimmt<sup>1</sup>. Die Anwendungen kommunizieren mit dem DBMS durch eine standardisierte Abfragesprache (meist SQL). Vorteile dieser Lösung sind ein kleinerer Implementations-Aufwand, erhöhte Effizienz, Mehrbenutzerfähigkeit, und die Möglichkeit, durch Abfragen nur Teildatenbestände zu bearbeiten. Der Nachteil ist, dass die meisten DBMS einen gewissen Installationsaufwand erfordern und während des Betriebs gewartet werden müssen. Das DBMS sollte räumliche Datentypen unterstützen. Ein Beispiel für ein räumliches DBMS ist die Kombination der Open Source Software PostGIS und PostgreSQL<sup>2</sup>. PostgreSQL ist ein relationales DBMS. PostGIS fügt die Unterstützung von räumlichen Datentypen nach der *OpenGIS Simple Features Specification For SQL* hinzu.

Welche der beiden Ansätze besser geeignet ist, hängt von der Anwendung (Grösse der Datensätze, Anzahl Benutzer) ab. Für den Matching-Prototyp genügt die dateibasierte Persistenz. Weil eine allgemeine Lösung für Axpand/DRIVE geschaffen werden sollte, wurde ein zweistufiger Import-/Export-Prozess entwickelt, der sowohl dateibasierte als auch datenbankbasierte Persistenz realisiert:

1. <http://de.wikipedia.org/wiki/DBMS>, Stand 20.12.2005

2. PostgreSQL: <http://www.postgresql.org/>. PostGIS: <http://postgis.refractory.net/>. Stand 20.12.2005

1. Die Java-Klassen der MRDB werden in eine relational normalisierte Datenstruktur übersetzt. Jede MRDB-Klasse enthält ein Persistenz-Pendant, welches die Eigenschaften der MRDB-Klasse öffentlich zugreifbar enthält. Die Persistenz-Klasse entspricht den Schemas von relationalen Tabellen, und deren Instanzen entsprechen relationalen Tupeln.
2. Die Persistenz-Objekte können entweder in eine relationale Datenbank oder in ein ZIP-komprimiertes Archiv geschrieben werden. Die relationale Tabellen sind im ZIP-Archiv als Tabulator-getrennte Textdateien gespeichert. Objektgeometrien werden in ein Textformat konvertiert.

## 5.5 Matching-Prototyp

In diesem Kapitel soll näher auf den Aufbau und die Funktionalität des erstellten Matching-Prototyps *RoadMatcher* eingegangen werden.

### 5.5.1 Benutzeroberfläche

Abbildung 5.6 zeigt die Benutzeroberfläche von *RoadMatcher*. Die Funktionalität wird in Form eines Menüs und einer Werkzeugleiste bereitgestellt. Das Menü enthält Funktionen zum Laden und Speichern der Datensätze, zur Auswahl verschiedener Visualisierungsarten, zum Aufruf von Statistiken über die Verknüpfungen, und zum Anstossen des automatischen Matchingprozesses. Auf die Werkzeugleiste wird in Abschnitt 5.5.3 näher eingegangen.

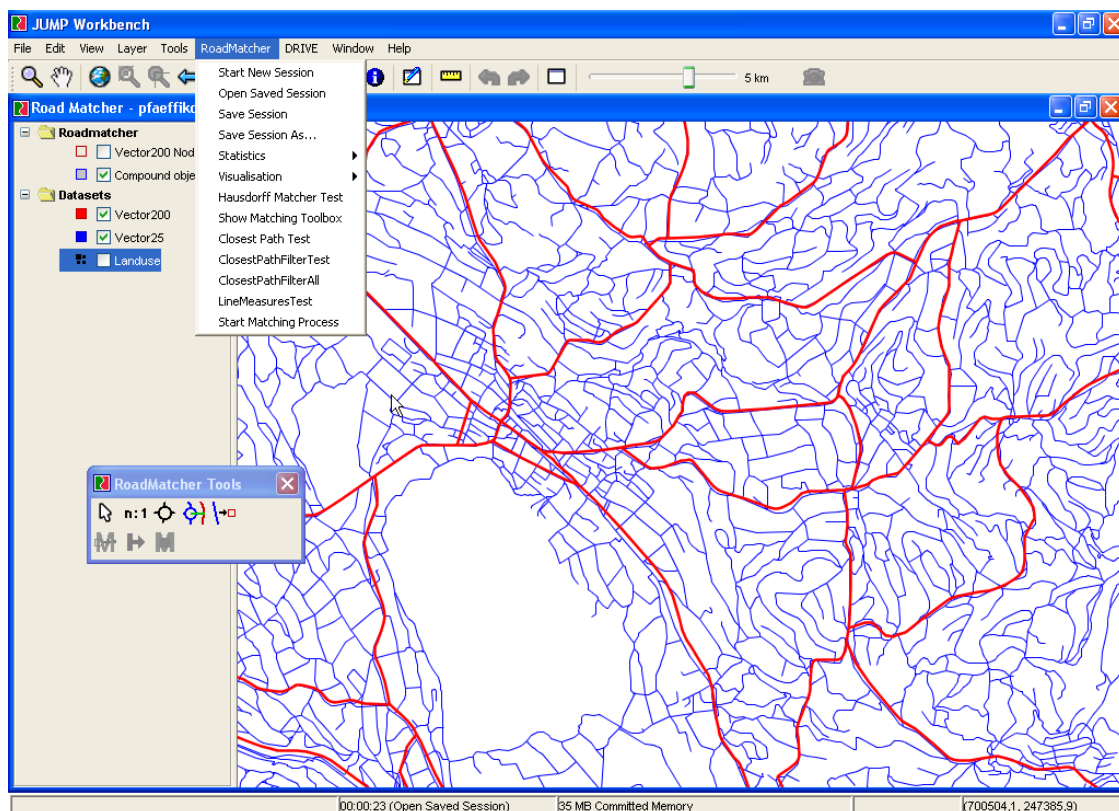


Abbildung 5.6: Oberfläche des Matching-Prototyps.

### 5.5.2 Erzeugung der MRDB

JUMP stellt Funktionen zum Laden von ESRI Shapefiles bereit. Nachdem der Benutzer Shapefiles von VECTOR25/VECTOR200 geladen hat, kann er mit dem Menüpunkt *Start New Session* die Dialogbox zum Import der Daten in die *RoadMatcher*-MRDB öffnen (Abbildung 5.7). Optional kann die Ebene *Primärfläche* aus VECTOR25 oder VECTOR200 als Hintergrundkarte hinzugefügt werden. Ist die MRDB einmal erzeugt worden, so kann sie in ZIP-Archiven gesichert werden (Menüpunkte *Save Session* und *Open Saved Session* in Abbildung 5.6). Die Shapefiles werden nicht mehr benötigt.

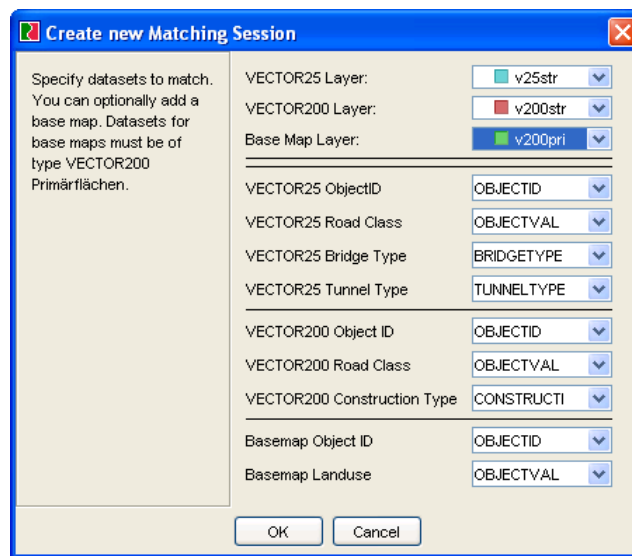


Abbildung 5.7: Dialogbox zum Import von neuen Daten in den *RoadMatcher*.

### 5.5.3 Manuelle Erzeugung und Nachbearbeitung von Verknüpfungen

Die Werkzeuge zur Benutzerinteraktion wurden in einer Werkzeuggestreife zusammengefasst (Abbildung 5.8), die sich als Fenster frei auf der Oberfläche verschieben lässt. Die Werkzeuge in der oberen Reihe dienen der manuellen Erzeugung und Änderung von Verknüpfungen, während die unteren drei Werkzeuge für das automatische Matching verwendet werden.

Die Manipulation von Verknüpfungen folgt einem Workflow: Ein bestimmtes Mauswerkzeug wird eingeschaltet, der Benutzer selektiert eines oder mehrere Objekte, und schliesslich wird über das Kontextmenü der Workflow in die nächste Phase geschaltet oder abgeschlossen. Die Klasse *MatchingToolsWorkflowManager* regelt solche Workflows für Benutzerinteraktionen. *MatchingToolsWorkflowManager* ist selbst ein sog. Mauswerkzeug, delegiert aber alle ankommenden Maus-Events an das Mauswerkzeug *MatchingSelectTool*. *MatchingSelectTool* ist grösstenteils identisch mit dem originalen JUMP-Selektionswerkzeug, es erweitert die Funktionalität aber um die Möglichkeit, Selektionen nur für bestimmte Ebenen zu erlauben und in der Kardinalität zu begrenzen.

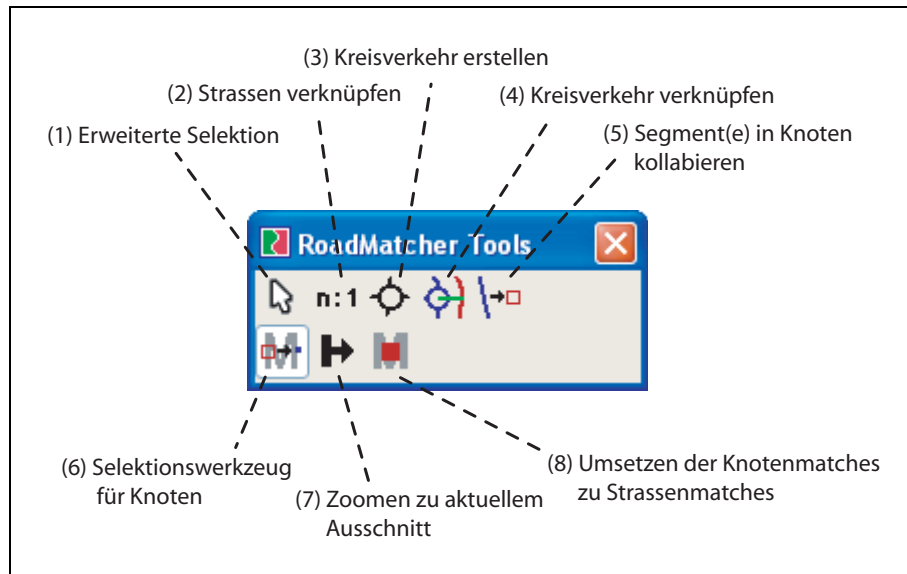


Abbildung 5.8: Werkzeugleiste des Matching-Prototyps.

In *MatchingToolsWorkflowManager* sind manuelle Abläufe für die folgenden Szenarien definiert (die Nummerierung entspricht den Werkzeugnummern in Abbildung 5.8):

1. Erweiterter Selektion: Klickt der Benutzer eine verknüpfte Strasse an, so wird die Selektion auf alle verknüpften Objekte ausgedehnt. Der Benutzer kann über das Kontextmenü die Verknüpfungen löschen.
2. Verknüpfen von mehreren VECTOR25-Strassen zu einer VECTOR200-Strasse. Im ersten Schritt Auswahl einer oder mehrerer Strassen aus VECTOR25, im zweiten Schritt Auswahl einer Strasse aus VECTOR200, schliesslich werden Verknüpfungen für die gewählten Strassen erzeugt und der Workflow wird wieder beim ersten Schritt gestartet.
3. Erzeugen eines VECTOR25-Kreisverkehrs: Selektiert mehrere VECTOR25-Strassen und erzeugt ein Exemplar von *GenObjectRoundabout*.
4. Verknüpfen eines VECTOR25-Kreisverkehrs mit einem VECTOR200-Knoten.
5. Kollaps von einzelnen VECTOR25-Segmenten in einen VECTOR200-Knoten.

### 5.5.4 Kandidatenbildung im automatischen Matching-Prozess

Der automatische Matching-Prozess wird mit dem Menüpunkt *Start Matching Process* (vgl. Abbildung 5.6) aufgerufen. Für die Erstellung der Kandidatenmengen ist die Klasse *CandidateBuilder* zuständig. Die Knoten- und Strassenkandidatenmengen sind dort in einem Zugriffsbaum (Java TreeMap) mit dem VECTOR200-Knoten bzw. der VECTOR200-Strasse als Schlüssel eingetragen.

Eine Knotenkandidatenmenge wird durch die Klasse *NodeCandidateSet* modelliert, eine Strassenkandidatenmenge durch *StreetCandidateSet* (siehe Abbildungen 5.9 und 5.10).

Ein Knotenkandidatenpaar (VECTOR200-Knoten – VECTOR25-Kandidat) wird durch *NodeCandidate* repräsentiert. Es werden dort auch die zum Matching notwendigen Masse Distanz und mittlere Zwischenwinkelsumme mitgeführt. Das *NodeCandidateSet*-Objekt eines

VECTOR200-Knotens enthält in einem Vector Referenzen auf die *NodeCandidate*-Objekte. Mit der Methode *rankNodeCandidates()* werden die *NodeCandidate*-Objekte nach Distanz und mittlerer Zwischenwinkelsumme in eine Rangliste gebracht. Die Methode *getWinningCandidate()* überprüft, ob es eindeutig einen am besten passenden Kandidaten gibt.

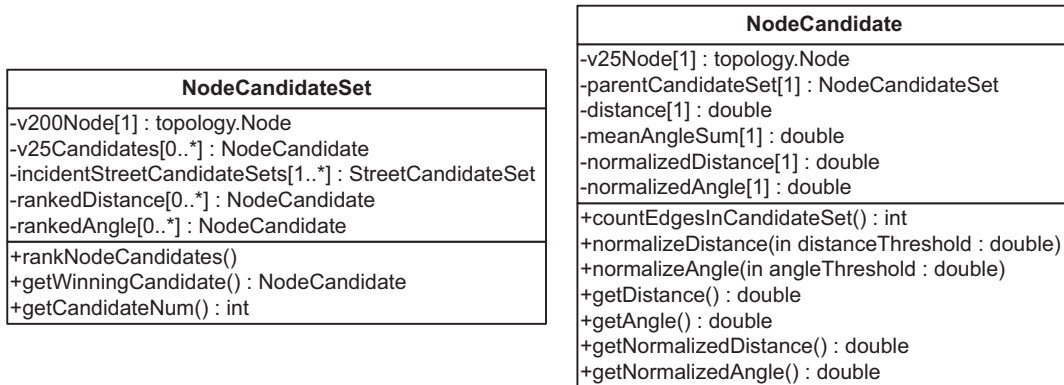


Abbildung 5.9: UML-Klassendiagramme von *NodeCandidateSet* und *NodeCandidate*. Es sind nur die wichtigsten Attribute und Methoden aufgeführt.

In der Klasse *StreetCandidateSet* werden die Kandidatenstrassen direkt in einem Vector gespeichert (Abbildung 5.10). Darüber hinaus enthält *StreetCandidateSet* auch den lokalen Graphen der Strassenkandidatenmenge und die nächsten benachbarten Wege (*closest paths*). Der lokale Graph wird durch Aufruf von *buildGraph()* erzeugt. *removeSingleEdges()* entfernt kurze, mit dem Rest unverbundene Teilstücke, *removeDanglingEdges()* entfernt Sackgassen. Ein nächster benachbarter Weg ist ein Subgraph des lokalen Graphen; seine Kanten werden in einem Vector gespeichert. *removeInvalidClosestPaths()* baut die Menge der nächsten benachbarten Wege für die Strassenkandidatenmenge auf. Da Strassenkandidaten ab diesem Zeitpunkt doppelt geführt werden (nämlich im Vector *v25StreetCandidates* und als Teil von nächsten benachbarten Wegen), müssen sie nach Änderungen synchronisiert werden – dies geschieht mit *removeInvalidClosestPaths()* und *filterStreetCandidateSetByClosestPaths()*. Die Methode *traceLineFilter()* implementiert den in Kapitel 4 besprochenen Algorithmus zur Linienverfolgung.

Der Aufbau der Kandidatenmengen, sowie die Anwendung der in Kapitel 4 erläuterten Beschränkungen *Strassenklassen*, *mittlere Zwischenwinkelsumme* und *Knotengrad* geschieht in der Klasse *CandidateBuilder* in eigenen Methoden. *buildStreetCandidateSet()* baut die Strassenkandidatenmengen für eine Menge von VECTOR200-Strassen auf. *filterNodeCandidateSetsByAngle()* wendet die Beschränkung *mittlere Zwischenwinkelsumme* auf die Knotenkandidatenmengen an. Die einzelnen Funktionen werden in der Methode *executeMatchingProcess()* zu einem Workflow kombiniert. Diese Methode wird durch den Menüpunkt *Start Matching Process* aufgerufen.

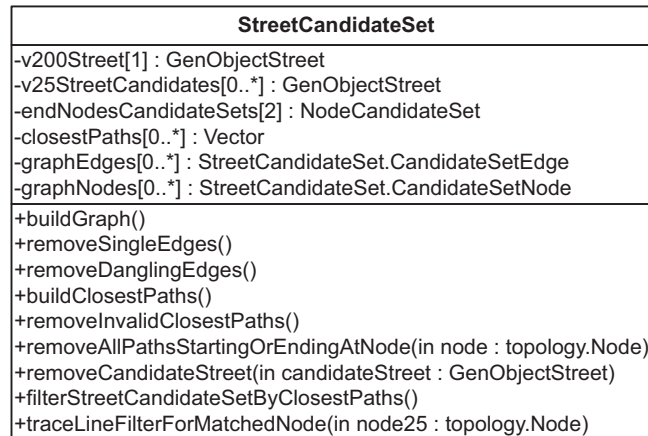


Abbildung 5.10: UML-Klassendiagramm von *StreetCandidateSet* mit den wichtigsten Attributen und Methoden.

### 5.5.5 Benutzerinteraktion im automatischen Matching-Prozess

Das eigentliche Matching ist in der Klasse *InteractiveMatchingProcess* implementiert. Bei der Erzeugung muss ihr ein Exemplar von *CandidateBuilder* übergeben werden, da diese Klasse die Kandidatenmengen kapselt. Wie in Kapitel 4 erläutert wurde, werden zuerst Knoten zugeordnet. Dazu wird für jede Kandidatenmenge die Methode *NodeCandidateSet.getWinningCandidate()* aufgerufen. Sie gibt den zum Referenzknoten ähnlichsten Kandidatenknoten zurück, sofern er existiert.

Konnte in einem Iterationsschritt auf diese Weise kein Knoten zugeordnet werden, so wird die Benutzerinteraktion aktiv. Die Applikation zeigt dem Benutzer die Umgebung des VECTOR200-Knotens mit den meisten Kandidaten. In Abbildung 5.11 hat der gelb markierte VECTOR200-Knoten zwei Kandidaten (rote Punkte auf den VECTOR25-Strassen), die bezüglich Distanz und mittlerer Zwischenwinkelsumme fast identisch sind. Der Benutzer kann entweder einen der Kandidatenknoten als Match-Partner auswählen oder er entscheidet, dass der Knoten gar keinen Match-Partner hat.

Falls der Benutzer auf ein anderes Mauswerkzeug wechselt (z.B. das Lupenwerkzeug) oder den Ausschnitt verliert, kann er mit Werkzeug (6) in Abbildung 5.8 wieder das Knotenmatching-Werkzeug einstellen bzw. mit Werkzeug (7) zum ursprünglich angezeigten Ausschnitt zurück.

Wurden aus Benutzersicht genügend Knoten zugeordnet, so kann der Benutzer den Matching-Prozess abschliessen, indem er die Schaltfläche (8) in Abbildung 5.8 aktiviert. VECTOR200-Strassen, bei welchen beide Endknoten zugeordnet werden konnten, haben nur noch einen nächsten benachbarten Weg. Die VECTOR25-Strassen, aus denen der Weg besteht, werden mit der VECTOR200-Strasse verknüpft. Damit ist das automatische Matching beendet.

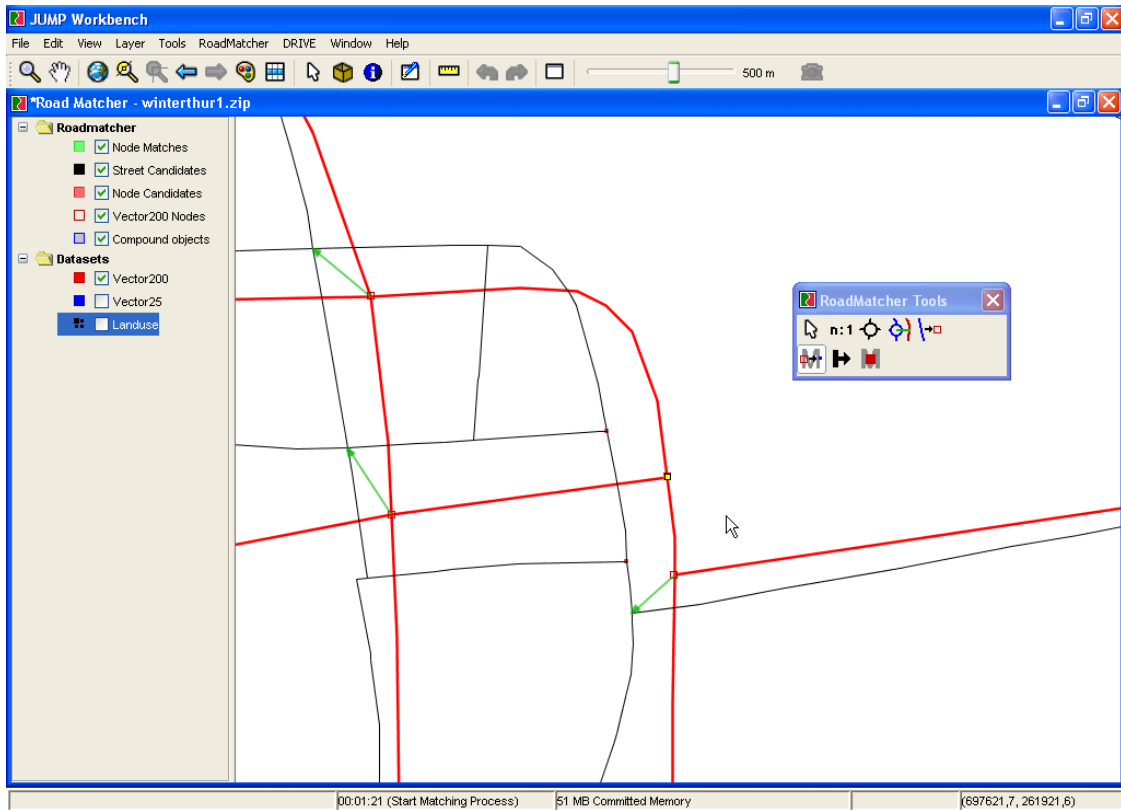


Abbildung 5.11: Benutzerinteraktion im automatischen Matching-Prozess.

## 5.6 Visualisierung der MRDB

Ein Operateur, der die Aufgabe hat, die Verknüpfungen zu erstellen, zu überprüfen und zu aktualisieren, muss möglichst einfach und schnell sowohl den Inhalt einzelner Repräsentationen als auch die Verknüpfungen zwischen den Objekten verschiedener Repräsentationen erkennen können.

Zur Visualisierung von zwei Repräsentationen gibt es prinzipiell zwei verschiedene Möglichkeiten: Entweder werden die beiden Repräsentationen nebeneinander in zwei separaten Kartenfeldern angezeigt (Abbildung 5.12) oder beide werden in einem Feld überlagert. In der vorliegenden Arbeit wurde der zweite Ansatz verfolgt. Er hat gegenüber der Darstellung in getrennten Kartenfeldern den Vorteil, dass korrespondierende Objekte einfacher zu identifizieren sind und so die manuelle Bearbeitung von Verknüpfungen einfacher ist.

### 5.6.1 Visualisierung von 1:N-Verknüpfungen zwischen Straßen

Im Rahmen der Arbeit wurden verschiedene Visualisierungsmethoden geprüft. Hier sollen zwei davon miteinander verglichen werden.

#### 5.6.1.1 Schraffuren zwischen verknüpften Objekten

Bei dieser Visualisierungsmethode wird die Fläche zwischen zwei verknüpften Strassenstücken mit einer linienhaften Schraffur gefüllt. Für jede VECTOR200-Strasse werden alle mit ihr ver-

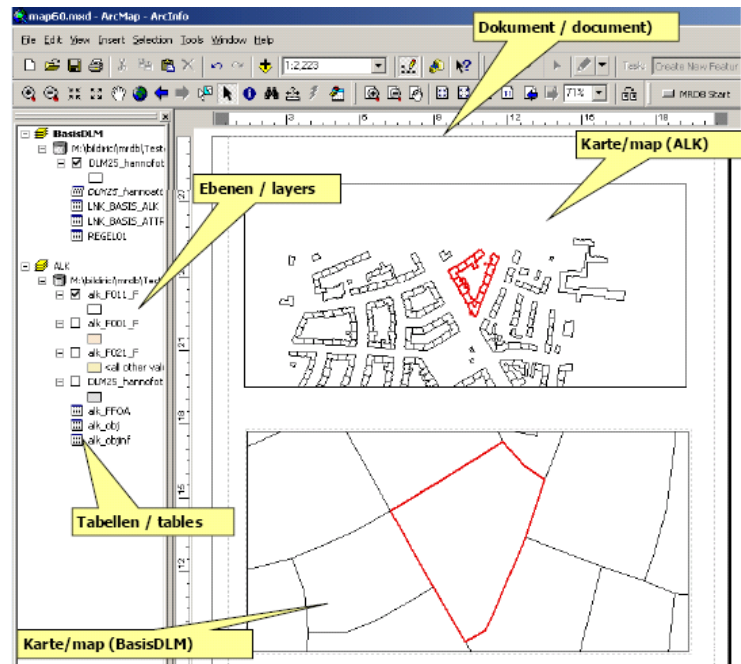


Abbildung 5.12: Repräsentationen in separaten Kartenfeldern (Anders et al. 2003).

knüpften VECTOR25-Strassen in eine einzige Linie fusioniert. Die Endpunkte der Verknüpfungslinien liegen auf der fusionierten Linie jeweils 20 m auseinander. Zur Berechnung der Endpunkte auf der VECTOR200-Strasse werden die 20 m proportional zum Verhältnis der Linielängen skaliert:

$$d_{200} = d_{25} \cdot \frac{l_{200}}{l_{25}} \quad (5.1)$$

$d_{25}$  ist der Abstand des aktuellen Punktes des VECTOR25-Strassenstückes vom Anfang,  $d_{200}$  der zu berechnende Abstand des korrespondierenden Punktes auf dem VECTOR200-Strassenstück und  $l_{25}$  bzw.  $l_{200}$  die Länge der VECTOR25/VECTOR200-Strassen.

Bei ähnlichen Massstäben ergeben sich mit dieser Visualisierung schöne Resultate (Abbildung 5.13). Die Richtung und Länge der Verbindungslinien zeigen die geometrische Verschiebung an. Probleme ergeben sich jedoch bei grösseren lateralen Verschiebungen und in Kreuzungsbereichen. Dort lässt sich teilweise schlecht erkennen, welche Objekte miteinander verknüpft sind.

### 5.6.1.2 Senkrechte Verbindungslinien

Es wurde eine weitere Visualisierungsmethode entwickelt. Sie sollte in allen Fällen eine klare Darstellung der Verknüpfungen erlauben, ein Abbild der internen Datenstruktur sein und es dem Operator erlauben, die Attribute, die einer Verknüpfung zugeordnet sind (Generalisierungsoperator, Sicherheit der Verbindung bei automatischer Zuordnung, etc.), mittels Mausklick abzufragen. Jedes Verknüpfungsobjekt der Datenbank wird durch eine eigene Verknüpfungslinie dargestellt. Sie steht senkrecht auf der VECTOR200-Strasse und berührt die VECTOR25-Strasse in ihrem Mittelpunkt (Abbildung 5.17).



Abbildung 5.13: Visualisierung mit flächenhaften Schraffuren. Rot: VECTOR200. Blau: VECTOR25.

Es gibt Situationen, wo mehrere solche senkrechten Verbindungen möglich sind und der Algorithmus eine sinnvolle Entscheidung treffen muss. In anderen Situationen ist keine senkrechte Abbildung möglich und es muss ein Ersatz gefunden werden. Der Algorithmus zur Bildung der Verknüpfungsgeraden läuft wie folgt ab:

#### A. Vorverarbeitung

1. Sortiere die Menge der zugeordneten VECTOR25-Strassenstücke.

#### B. Kandidatenbildung und Bewertung

1. Suche für jede VECTOR25-Strasse die Menge der möglichen senkrechten Abbildungen auf die verknüpfte VECTOR200-Strasse.
2. Eliminiere ungültige Abbildungen.
3. Wähle aus den verbleibenden die beste aus.

#### C. Nachbearbeitung

1. Behandle diejenigen VECTOR25-Strassen, für die keine senkrechte Abbildung gefunden werden konnte.

#### A. Vorverarbeitung

Im Nachbearbeitungs-Schritt müssen die Nachbarn der VECTOR25-Strassen bekannt sein. Deshalb werden zuerst die an einer Verknüpfung beteiligten VECTOR25-Strassen sortiert und gegebenenfalls die Reihenfolge der Stützpunkte umgedreht, so dass sich eine zusammenhängende Linie ergibt. Diese muss die gleiche Richtung haben wie das verknüpfte VECTOR200-Strassenstück, was ebenfalls überprüft und gegebenenfalls korrigiert wird.

#### B. Kandidatenbildung und Bewertung

Hier wird versucht, für jede verknüpfte VECTOR25-Strasse eine Verknüpfungslinie mit der VECTOR200-Strasse zu bilden. Diese Verknüpfungslinie steht senkrecht auf der VECTOR200-Strasse und soll die VECTOR25-Strasse möglichst nah beim Mittelpunkt schneiden. Abbildung

5.14 zeigt das Prinzip dieses Schrittes. Ausgehend vom Mittelpunkt der VECTOR200-Strasse wird nach links und nach rechts in 5 m-Schritten ausgeschwenkt und geprüft, ob sich eine Senkrechte zur VECTOR200-Strasse finden lässt, die diesen Punkt schneidet. Wird eine solche Senkrechte gefunden, so gilt sie als Kandidat.

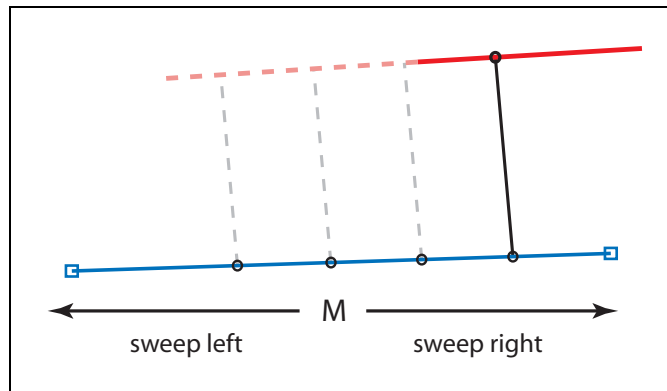


Abbildung 5.14: Suche nach einer Geraden, die senkrecht auf dem VECTOR200-Strassenstück (rot) steht.

Besonders in Berggebieten, wo es längere, kurvenreiche Strassenstücke gibt, können Linien generiert werden, die ungünstig sind, weil sie die VECTOR25-Strassen überschneiden (Abbildung 5.15a). Diese Linien müssen verworfen werden. Es ist ebenfalls möglich, dass am Schluss dieses Verfahrens mehrere Kandidatenlinien übrig bleiben. In diesem Fall wird die kürzeste der generierten Linien gewählt.

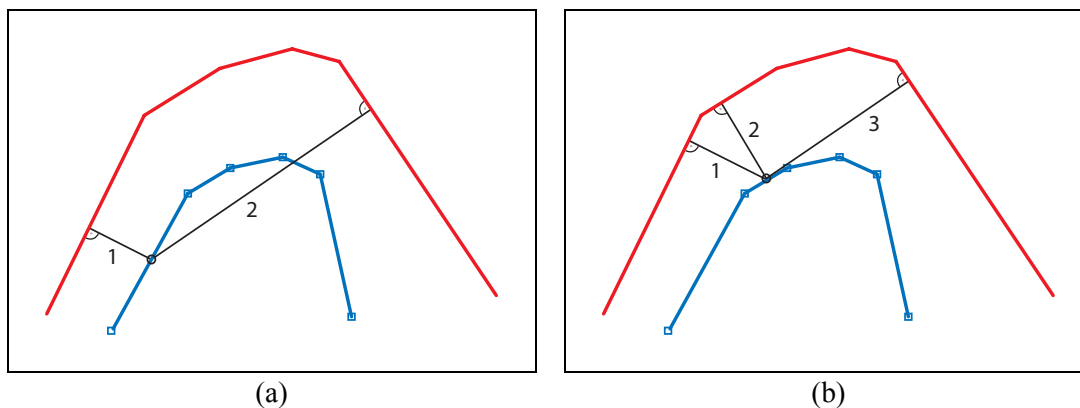


Abbildung 5.15: (a) Verbindungslinie 2 ist inkorrekt, da sie ein zugeordnetes VECTOR25-Segment überschneidet. (b) Fall mit 3 gültigen Verbindungslinien.

### C. Nachbearbeitung

Es ist aber auch möglich, dass sich gar kein Kandidat finden lässt, weil die laterale Verschiebung so gross ist, dass keine senkrechte Abbildung auf die VECTOR200-Strasse mehr möglich ist. Für diese VECTOR25-Strassen wird in der Nachbearbeitung ermittelt, wo die Verknüpfungsendpunkte der linken und rechten Nachbarn auf der VECTOR200-Strasse liegen. Der Verknüpfungspunkte der linken und rechten Nachbarn auf der VECTOR200-Strasse liegen. Der Verknüpfungspunkte der linken und rechten Nachbarn auf der VECTOR200-Strasse liegen.

fungsendpunkt wird so bestimmt, dass er in der Mitte zwischen den Endpunkten der Nachbarn liegt. Lässt sich zu mehreren benachbarten VECTOR25-Strassen keine senkrechte Verknüpfungslinie finden, wird der Raum zwischen den nächsten Nachbarn mit gültiger senkrechter Verknüpfungslinie auf diese Strassen aufgeteilt.

In Abbildung 5.16 kann Verbindungslinie 1 senkrecht gesetzt werden, Verbindungslinie 2 jedoch nicht. Ihr Endpunkt wird deshalb in die Mitte zwischen den Endpunkt von Linie 1 und dem VECTOR200-Knoten gesetzt, da es rechts keinen Nachbarn gibt.

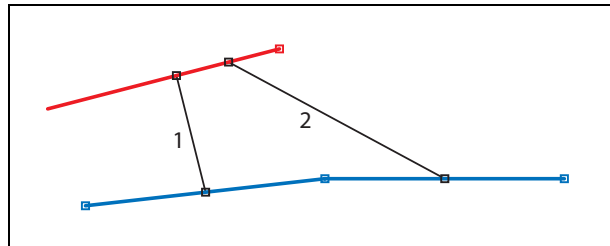


Abbildung 5.16: Situation, wo keine senkrechte Verbindung möglich ist.

Abbildung 5.17 zeigt das Resultat der besprochenen Prozedur.



Abbildung 5.17: Visualisierung mit senkrechten Verbindungslinien. Rot: VECTOR200. Blau: VECTOR25.

### 5.6.2 Visualisierung von Kreiseln und kollabierten Strassensegmenten

Kreisel in VECTOR25 werden in der Datenbank zu einem Aggregationsobjekt zusammengefasst. Dies kommt auch in der Darstellung (Abbildung 5.18a) zum Ausdruck. Die Kreiselfläche

wird grau eingefärbt, und der Mittelpunkt des Kreisels wird als Ankerpunkt mit dem verknüpften VECTOR200-Knoten durch einen Pfeil verbunden.

Bei einzelnen VECTOR25-Strassen, die in VECTOR200 kollabiert sind, wird der Mittelpunkt mit dem VECTOR200-Knoten verbunden (Abbildung 5.18b).

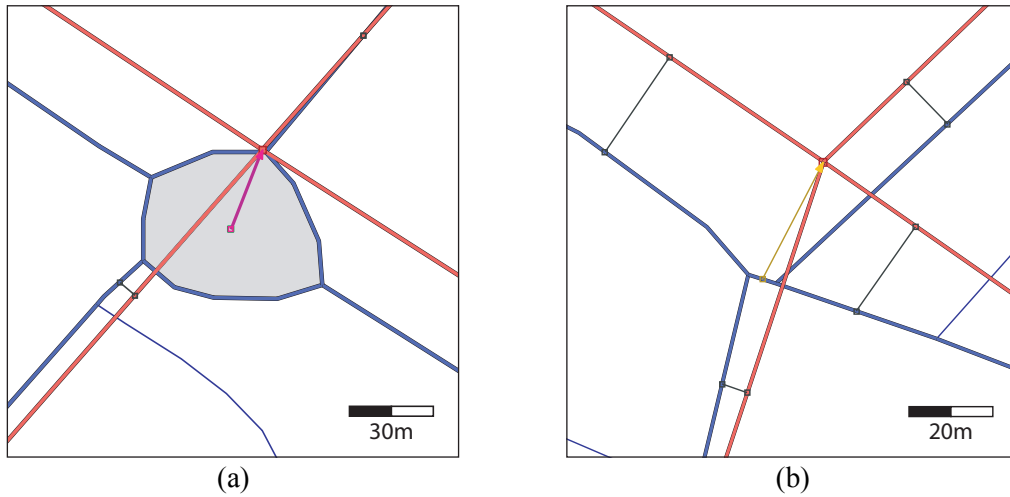


Abbildung 5.18: (a) Kollabierter VECTOR25-Kreisel (b) Kollabierte VECTOR25-Strasse.

## Kapitel 6

# Evaluation

In diesem Kapitel soll der entwickelte Algorithmus auf seine Eignung zum Matching von VECTOR200 und VECTOR25 untersucht werden. Laufzeitmessungen zeigen das Potential des Algorithmus für das Matching grösserer Gebiete auf.

### 6.1 Übersicht

Der Matching-Prozess wurde in zwei Gebieten von je 10x10 km Grösse getestet.

- Der Ausschnitt „Pfäffikon“ (698750–708750 / 241240–251240) enthält ein dichtes Netz von kleineren und mittleren Gemeinden und entspricht damit der typischen Struktur des Schweizer Mittellandes. Die beiden grössten Gemeinden sind Wetzikon mit 18'000 Einwohnern und Pfäffikon mit 9'500 Einwohnern (nach Volkszählung 2000<sup>1</sup>). Weitere Gemeinden sind Fehraltendorf, Bäretswil, Russikon und Hittnau, alle mit Einwohnerzahlen zwischen 3000 und 5000. Der Ausschnitt ist in Abbildung 6.1a dargestellt.
- Der Ausschnitt „Winterthur“ (696250–706250 / 254100–264100) deckt im Nordwesten die Stadt Winterthur mit 90'000 Einwohnern ab. Ansonsten ist einzig Zell (4600 Einwohner) als grössere Gemeinde zu vermerken. Dieser Ausschnitt wurde gewählt, weil die Strassensituation in Winterthur komplexer ist – er soll die Grenzen des Matching-Algorithmus aufzeigen. Abbildung 6.1b zeigt den Ausschnitt „Winterthur“.
- Für die Herleitung von Statistiken für Strassenklassen wurde zudem ein Teil der Innenstadt von Zürich in einem 7x7 km grossen Ausschnitt von Hand zugeordnet.

---

1. <http://www.statistik.zh.ch/>, Stand 20.12.2005

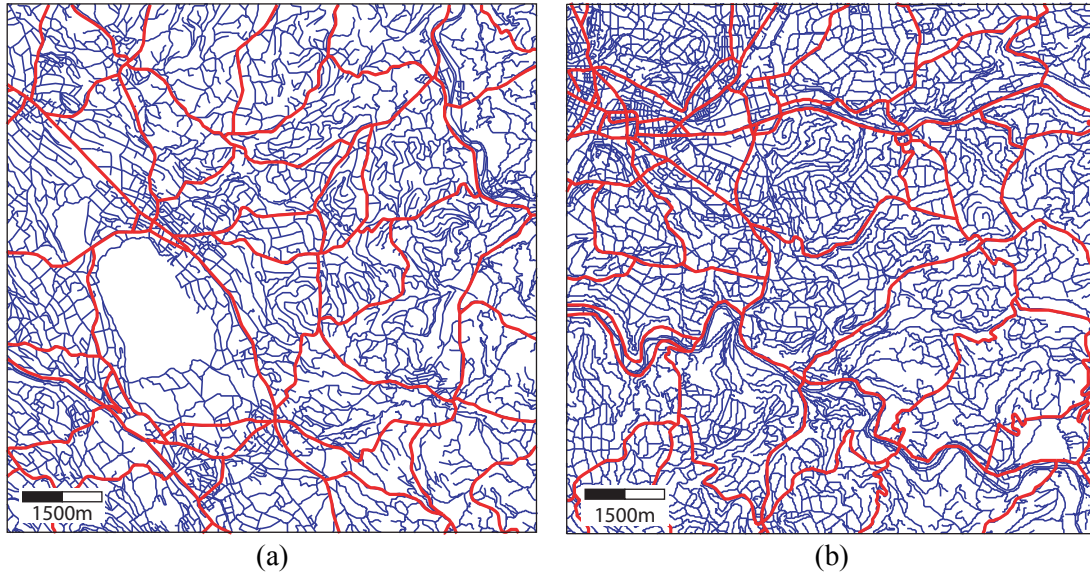


Abbildung 6.1: (a) Testgebiet „Pfäffikon“ (b) Testgebiet „Winterthur“.

Abbildung 6.2 zeigt einen Ausschnitt aus dem Gebiet „Pfäffikon“. Im Ausschnitt rechts sind die zu VECTOR200 korrespondierenden Strassen aus VECTOR25 zu sehen, wie sie automatisch vom Matching-Algorithmus extrahiert wurden. In Abbildung 6.3 wird ein Ausschnitt aus dem Gebiet „Winterthur“ und die extrahierten VECTOR25-Strassen gezeigt.

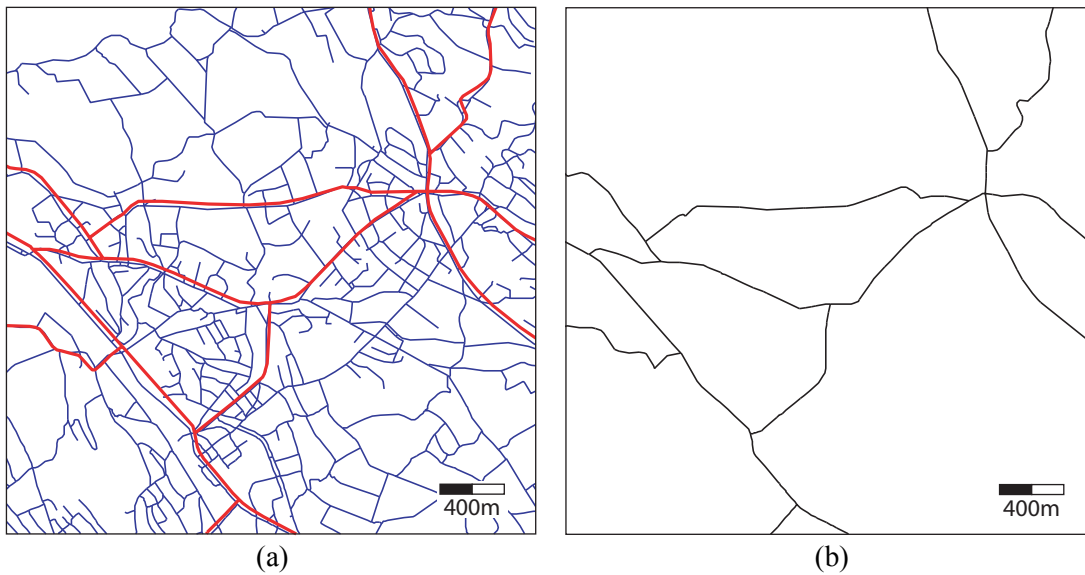


Abbildung 6.2: Ausschnitt aus dem Gebiet „Pfäffikon“. Links: VECTOR25 (blau) überlagert mit VECTOR200 (rot). Rechts: Extrahierte VECTOR25-Strassen.

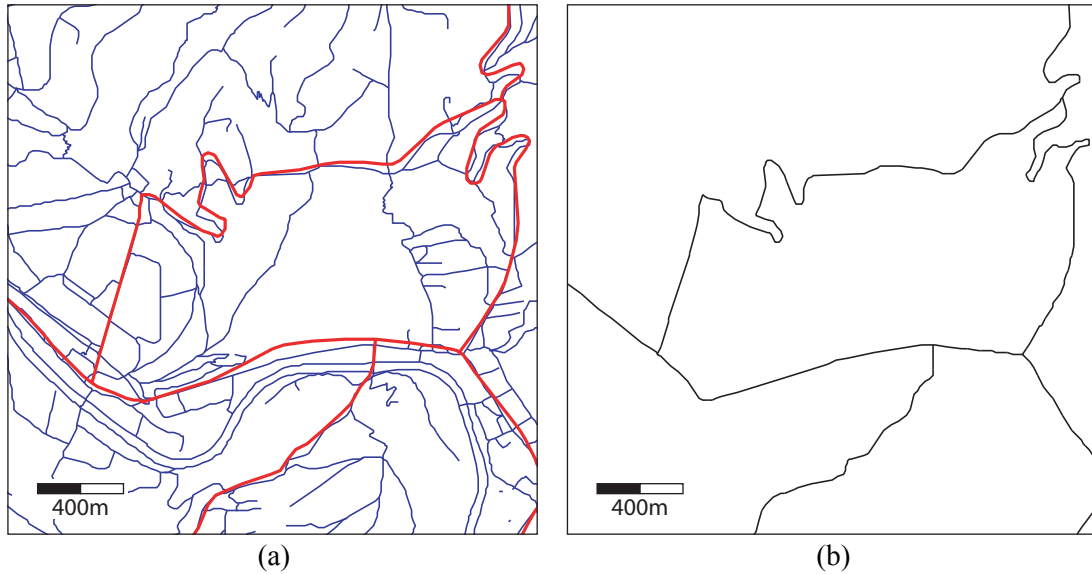


Abbildung 6.3: Ausschnitt aus dem Gebiet „Winterthur“. Links: VECTOR25 (blau) überlagert mit VECTOR200 (rot). Rechts: Extrahierte VECTOR25-Strassen.

## 6.2 Evaluation des automatischen Matching-Algorithmus

Die Leistung eines Zuordnungsalgorithmus kann mit zwei Kennzahlen erfasst werden:

1. Der Anteil der automatisch zugeordneten Objekte an der Gesamtzahl der Objekte (die *Zuordnungsrate*)
2. Der Anteil der falsch zugeordneten Objekte an den automatisch zugeordneten Objekten (die *Fehlerrate*)

Die Zuordnungsrate alleine sagt wenig aus. Wenn viele Objekte automatisch zugeordnet werden konnten, jedoch die meisten der Matches falsch sind, ist der Nachbearbeitungsaufwand ebenso gross wie wenn nur ein kleiner Teil automatisch zugeordnet werden konnte, diese Matches dafür aber korrekt sind.

### 6.2.1 Zuordnungsrate

Tabelle 6.1 zeigt die Zuordnungsrate des Matching-Algorithmus für die beiden Gebiete „Pfäffikon“ und „Winterthur“ getrennt nach Knotenzuordnung und Strassenzuordnung. Die Knotenmatches wurden vor der Benutzerinteraktion erfasst (nach Schritt 2 in Kapitel 4.2.3). Die Quotienten für die Strassenmatches wurden nach dem automatischen Umsetzen der Knotenmatches zu Kantenmatches erfasst (nach Schritt 3 in Kapitel 4.2.3); in der Benutzerinteraktion wurden zuvor fehlende Knotenmatches ergänzt. Die Zuordnungsrate der Strassenmatches ist etwas niedriger als diejenige der Knotenmatches, weil ein nicht zugeordneter Knoten dazu führt, dass alle mit ihm verbundenen Strassen auch nicht zugeordnet werden können. Im letzten Teil der Tabelle (Strassen in VECTOR25) wird aufgeführt, wieviele der VECTOR25-Strassen, die zu den VECTOR200-Strassen korrespondieren, automatisch verknüpft werden konnten.

Im Gebiet „Pfäffikon“ sind die Resultate sowohl für Knotenmatches als auch für Strassenmatches sehr gut. Von den 5 Knoten, die interaktiv gematcht werden mussten, liegen 3 am Rand des Ausschnittes. Für diese Ranknoten mit Grad 1 gibt es oft sehr viele Kandidaten. Es handelt sich jedoch um einen Randeffect, der mit grösser werdendem Gebiet an Bedeutung verliert.

Entsprechend der komplexeren Situation im Gebiet „Winterthur“ ist die Zuordnungsrate dort niedriger. Von den 13 interaktiv gematchten Knoten liegen 5 am Rand des Ausschnittes. Die Zahl der Knoten, bei denen es keine 1:1-Korrespondenz gibt, ist höher als in Pfäffikon. Dies führt dann auch zu den schlechteren Zahlen bei den Strassenzuordnungen, da die an diese Knoten angrenzenden Strassen nicht automatisch zugeordnet werden können.

	<b>Pfäffikon</b>	<b>Winterthur</b>
<b>Knoten in VECTOR200</b>	97 (100%)	119 (100%)
automatisch zugeordnet	89 (91.7%)	100 (84.0%)
interaktiv zugeordnet	5 (5.2%)	13 (10.9%)
keine 1:1-Korrespondenz	2 (2.1%)	5 (4.2%)
keine Korrespondenz (Inkonsistenz)	1 (1.0%)	1 (0.9%)
<b>Strassen in VECTOR200</b>	112 (100%)	144 (100%)
automatisch zugeordnet	100 (89.3%)	115 (79.9%)
interaktiv zugeordnet	11 (9.8%)	28 (19.4%)
keine Korrespondenz (Inkonsistenz)	1 (0.9%)	1 (0.7%)
<b>Strassen in VECTOR25 (homolog zu VECTOR200)</b>	963 (100%)	1149 (100%)
automatisch zugeordnet	858 (89.1%)	871 (75.8%)
interaktiv zugeordnet	105 (10.9%)	278 (24.2%)

*Tabelle 6.1:* Zuordnungsraten des automatischen Matching-Prozesses für die Testgebiete Pfäffikon und Winterthur.

In der Regel konnten die Strassen auch dort zugeordnet werden, wo es zwischen korrespondierenden Daten grössere Lageunterschiede gab. Die Strassenschleufe in Abbildung 6.4a wurde beispielsweise korrekt automatisch zugeordnet. Abbildungen 6.4b bis 6.4d zeigen einige Situationen, wo der Matching-Algorithmus zum fraglichen Referenzknoten  $N$  den homologen Knoten aus VECTOR25 nicht automatisch ermitteln kann und deshalb der Benutzer eingreifen muss. Allen Situationen ist gemeinsam, dass mehrere Kandidatenknoten (in den Abbildungen nummeriert) in ähnlicher Entfernung vom VECTOR200-Knoten liegen und die Strassen in ähnlichen Richtungen zu den Kandidatenknoten einfallen. Mit diesen beiden Massen allein kann der Algorithmus deshalb nicht eindeutig einen der Kandidaten als homologen Knoten bestimmen.

Bei Abbildung 6.4b ist denkbar, dass der Einbezug von Linienmassen der einfallenden Strassen helfen könnte, weil die Referenzstrasse ( $N_B-N$ ) gerade verläuft, die Kandidatenstrasse ( $N_B'-2$ ) hingegen einen rechten Winkel bildet und deshalb Kandidat 2 ausgeschlossen werden kann. Bei Abbildung 6.4c unterscheiden sich potentielle Kandidatenstrassen zumindest bezüglich Sinuosität und Linienlänge nicht genügend voneinander, um eine Entscheidung zwischen den beiden Knotenkandidaten treffen zu können.

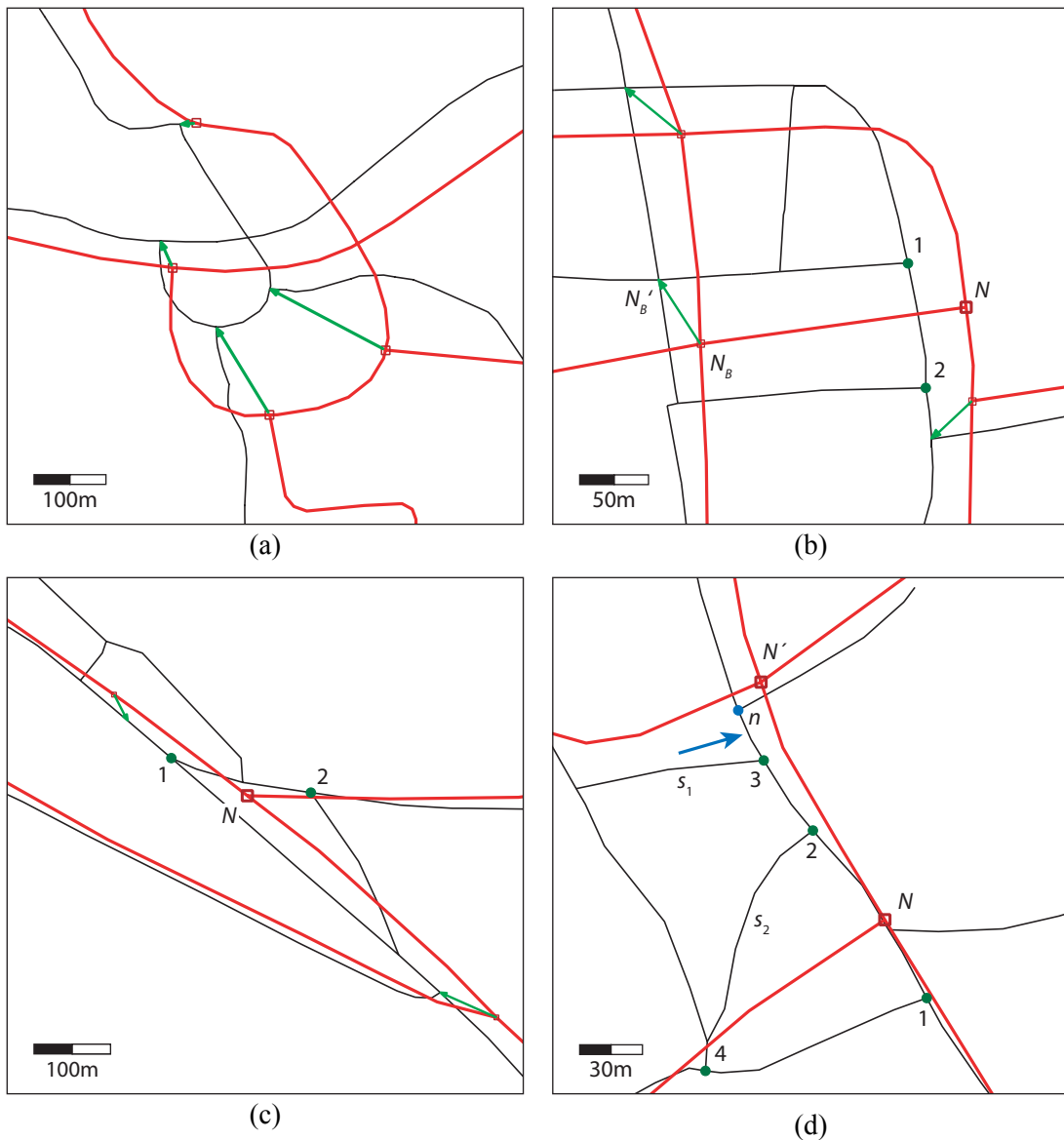


Abbildung 6.4: (a) Erfolgreiches Knotenmatching. (b) – (d) Situationen, wo der korrespondierende Knoten nicht automatisch gefunden wurde.

Eine andere Problematik soll Abbildung 6.4d verdeutlichen. Die mit dem blauen Pfeil markierte VECTOR25-Strasse kollabiert, so dass die Knoten 3 und  $n$  zusammen dem Knoten  $N'$  entsprechen (N:1-Beziehung zwischen Knoten). Somit existiert in der Umgebung von  $N'$  kein VECTOR25-Knoten mit Grad vier und der Knoten  $N'$  bleibt ohne Kandidatenknoten. Deshalb

kann für die einfallenden Strassen von  $N'$ , insbesondere für das Segment  $(N'-N)$ , das Modul *nächste benachbarte Wege* nicht ausgeführt werden. Ohne diesen Filter bleiben in der Umgebung von  $N$  zu viele Knotenkandidaten übrig, so dass der Algorithmus keine Entscheidung für einen der Kandidaten mehr treffen kann. Könnte das Modul *nächste benachbarte Wege* ausgeführt werden, so würden die Strassensegmente  $s_1$  und  $s_2$  entfernt werden, da sie die zu  $N$  und  $N'$  korrespondierenden VECTOR25-Knoten nur über grosse Umwege miteinander verbinden.

In Abbildung 6.5 korrespondieren die blau markierten Strassen  $s_{25}$  mit der VECTOR200-Strasse  $s_{200}$ . Die als Nebenstrasse 3 m klassierte VECTOR200-Strasse ist von einer parallel dazu verlaufenden Strasse (Verbindungsstrasse 6 m) etwa 100 m nach Norden verdrängt worden. Der Kurvenverlauf aus VECTOR25 mit seinen rechten Winkeln wird zudem in VECTOR200 stark geglättet wiedergegeben. Die Linienverfolgung vom benachbarten, zugeordneten Knoten  $N_B$  wurde im Punkt  $C_1$  abgebrochen, weil die Linie  $(N_B-2)$  eine kleinere durchschnittliche Hausdorffdistanz zu  $s_{200}$  hat als  $(N_B-1)$  oder  $(N_B-3)$ , bei einer Wahl des Segments  $(C_1-2)$  aber das „Good Continuation“-Prinzip verletzt werden würde. Ebenso wurde die Linienverfolgung vom nördlichen benachbarten Knoten her abgebrochen, weil im Punkt  $C_2$  das „Good Continuation“-Prinzip verletzt wird. Der dritte benachbarte Knoten im Süden blieb unverknüpft. Für den Knoten  $N$  verbleiben somit die vier Kandidaten 1–4, von denen der Benutzer manuell den korrespondierenden Knoten wählen muss.

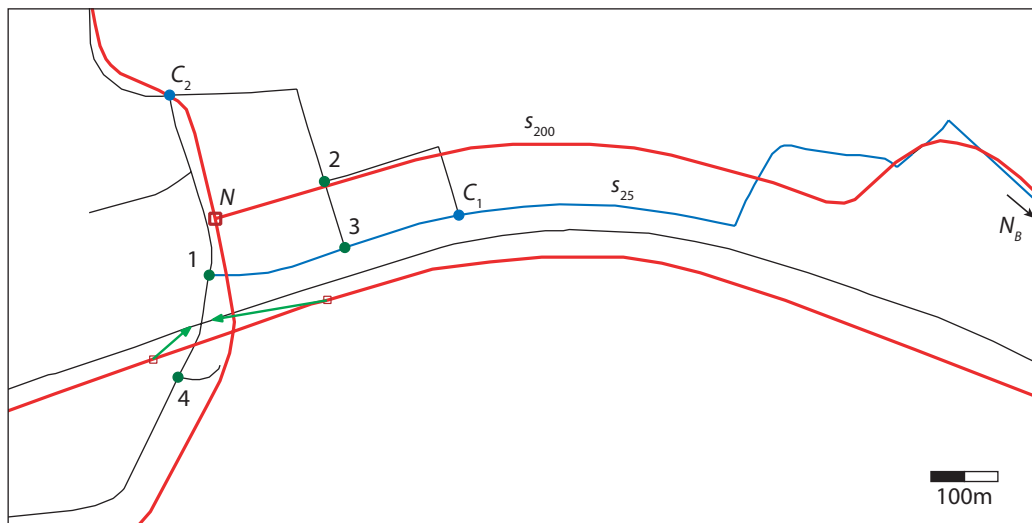


Abbildung 6.5: Situation mit stark unterschiedlichen korrespondierenden Strassen.

## 6.2.2 Fehlerrate

Der in dieser Arbeit entwickelte Matching-Algorithmus verhält sich konservativ: Es wird nur dann eine Verknüpfung automatisch erstellt, wenn sie mit hoher Wahrscheinlichkeit korrekt ist. Andernfalls wird der Benutzer zur Interaktion aufgefordert. Deshalb ist die Zahl der falschen Zuordnungen in den untersuchten Testgebieten klein. Im Gebiet Pfäffikon trat keine Falschzuordnung auf. Im Gebiet Winterthur wurde keine falsche Knotenzuordnung, aber eine falsche Strassenzuordnung festgestellt. Sie ist in Abbildung 6.6 dargestellt. Die in dunkelrot dargestellte Kurve in VECTOR200 wurde der dunkelblauen VECTOR25-Strasse zugeordnet. Korrekt wäre

jedoch das graue Strassenstück. Ausgelöst wurde die Fehlzuzuweisung dadurch, dass die Strassenschleife im Norden im Massstab 1:200'000 stark vergrössert werden musste, um sie noch darstellen zu können. Dadurch verschob sich auch die Kurve nach Süden. In dieser Situation hat das dunkelblau gefärbte Strassenstück die kleinere mittlere Hausdorff-Distanz zur VECTOR200-Kurve als das eigentlich korrekte graue Strassenstück, und es wird eine fehlerhafte Verknüpfung produziert. Die übrigen Objekte der Situation, insbesondere die Strassenschleife im Norden, wurden vom Algorithmus jedoch korrekt verknüpft.

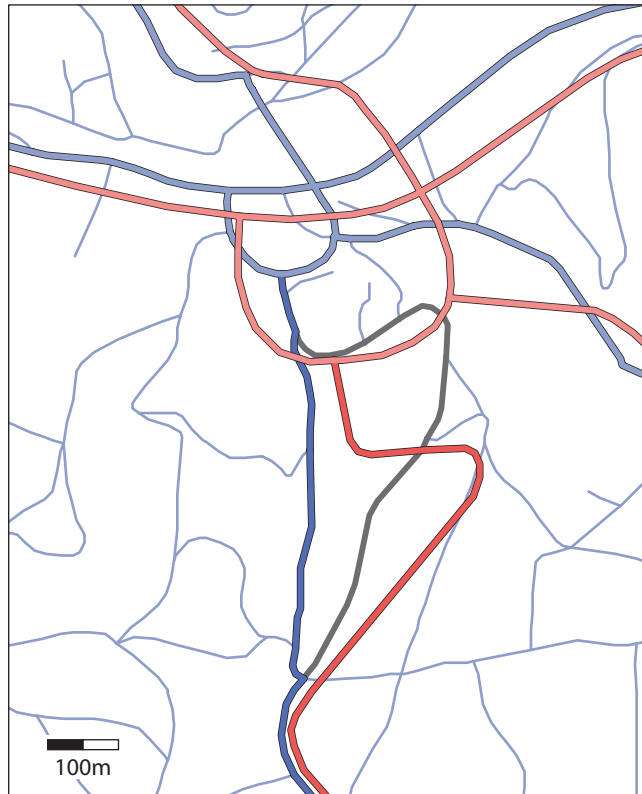


Abbildung 6.6: Falschzuordnung von Strassen im Gebiet Winterthur.

## 6.3 Zeitverhalten

Es muss unterschieden werden zwischen der Phase der Kandidatenbildung und der interaktiven Phase. Da die Kandidatenbildung autonom abläuft, sind längere Laufzeiten in dieser Phase nicht problematisch. In der interaktiven Phase muss der Benutzer aber ohne Wartezeiten zu den strittigen Knoten geführt werden.

Für eine empirische Untersuchung wurden 8 verschieden grosse Gebiete automatisch zugeordnet und der Zeitaufwand dafür gemessen. Ein Ausschnitt um Wetzikon wurde dafür sukzessive vergrössert; die 8 Gebiete überlappen sich also. Die Zeitmessungen wurden auf einem Pentium M 1.6 GHz mit 512MB RAM durchgeführt. Die Laufzeiten für jedes Gebiet wurden dreimal gemessen und davon das Mittel gebildet.

### 6.3.1 Abhängigkeit der Laufzeit von der Dateigrösse

Da nur lokal innerhalb von Kandidatenmengen eines Referenzobjektes optimiert wird, ist die Zeitkomplexität linear zu der Dateigrösse. Die Laufzeit kann aber von der Situation abhängen: Bei dichterem VECTOR25-Strassennetzen innerhalb von Siedlungen müssen durchschnittlich mehr Kandidaten und nächste benachbarte Wege berechnet werden. Es ist deshalb zu erwarten, dass die Laufzeit in städtischen Gebieten höher ist als in ländlichen Gebieten.

Abbildung 6.7 zeigt die Gesamtlaufzeit der Kandidatenbildung für die verschiedenen Datensatzgrössen. Zwischen den beiden Datensätzen mit 186 und 235 Strassen gibt es einen auffälligen Sprung, während die Entwicklung ausserhalb davon linear verläuft. Beim Datensatz mit 235 Strassen kommt ein östlich zur Stadt Winterthur angrenzendes Gebiet hinzu. Einige komplexe Strassensituationen mit vielen VECTOR25-Kandidatenstrassen in diesem Gebiet führen dazu, dass viele nächste benachbarte Wege berechnet werden müssen. Deshalb steigt dort die Laufzeit überproportional stark an. Das Diagramm zeigt somit, dass die Laufzeit von der Komplexität der Situation abhängt und mit der Dateigrösse nicht übermässig stark ansteigt.

Der grösste der getesteten Datensätze mit 438 VECTOR200-Strassen bedeckt eine Fläche von 330 km<sup>2</sup>, was 19% der Kantonsfläche von Zürich entspricht. Die Laufzeit für die Kandidatenbildung beträgt dort 2 Minuten 46 Sekunden. In der interaktiven Phase blieb die Wartezeit zwischen zwei manuell zu verknüpfenden Knoten unter einer Sekunde. Damit eignet sich die Methode auch, um grössere Flächen zu verknüpfen.

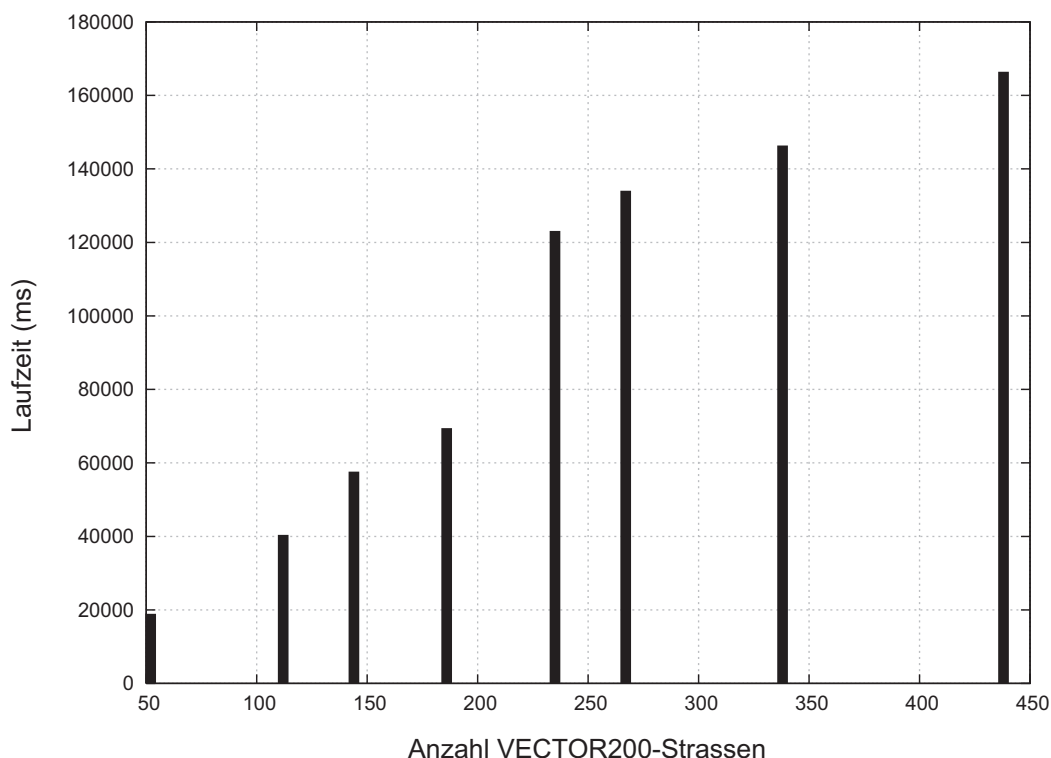


Abbildung 6.7: Laufzeit der Kandidatenbildung in Abhängigkeit von der Datensatzgrösse.

### 6.3.2 Anteil der einzelnen Module

Interessant ist auch, welchen Anteil die in Kapitel 4.2.3 erläuterten Module an der Gesamtlaufzeit haben. Dazu wurde der Quotient aus Modullaufzeit und Gesamtlaufzeit gebildet. Der Quotient zeigte sich unabhängig von der Gebietsgrösse und wurde deshalb über alle 8 Gebietsgrössen gemittelt. Mit fast 94% der Gesamtlaufzeit macht der Aufbau der nächsten benachbarten Wege den grössten Teil aus. Grund dafür ist die aufwändige Berechnung der Hausdorff-Distanz, die im *shortest path*-Algorithmus häufig ausgeführt wird. Daneben fällt nur noch die Bildung der Strassenkandidaten ins Gewicht. Die Bildung der Strassenkandidaten verläuft in zwei Schritten:

1. In einem R-tree-Index werden alle VECTOR25-Strassen gesucht, die innerhalb der *minimum bounding box* (mbb) der VECTOR200-Strassen liegen. Diese grobe Suche ist sehr schnell.
2. Die in Schritt 1 gebildeten Kandidatenmengen werden verfeinert, indem eine geometrische Verschneidung zwischen den VECTOR25-Strassen und dem Puffer der VECTOR200-Strasse gebildet wird.

Die geometrische Verschneidung ist eine rechnerisch aufwändige Funktion. Je nach Länge und Lage der VECTOR200-Strasse kann die mbb zu gross sein und die grobe Suche somit viele Kandidaten liefern, für welche die geometrische Verschneidung berechnet werden muss.

Die Anwendung von Beschränkungen und der Aufbau des Graphen benötigen wenig Rechenzeit. Bei Beschränkungen wird mit einfachen Attributdaten oder Skalaren gearbeitet. Der Aufbau des Graphen geschieht durch eine Selektion der Untermenge des Topologiegraphen für die Repräsentation.

Modul	Anteil an der Gesamtlaufzeit
Bilden der Strassenkandidaten	5.26%
Bilden der Knotenkandidaten	0.46%
Beschränkung Strassenklasse	0.07%
Beschränkungen Knotenrad & mittlere Zwischenwinkelsumme	0.15%
Aufbau & Vereinfachung des Graphen	0.2%
Aufbau der nächsten benachbarten Wege	93.85%

Tabelle 6.2: Anteil der verschiedenen Module an der Gesamtlaufzeit in der Phase Kandidatenbildung.



## Kapitel 7

# Schlussfolgerungen und Ausblick

In dieser Arbeit wurden Verfahren untersucht, um verschiedene Datensätze durch Matching in eine Multirepräsentationsdatenbank (MRDB) zu integrieren. Ziel der Arbeit war es, ein Verfahren zum Matching von Strassendaten zu finden, die in stark unterschiedlichen Massstäben vorliegen – so haben die beiden verwendeten Datensätze VECTOR25 und VECTOR200 die Massstäbe 1:25'000 bzw. 1:200'000. Im ersten Abschnitt dieses Kapitels wird zusammengefasst, was in der vorliegenden Arbeit erreicht worden ist. Im zweiten Abschnitt werden der neu entwickelte Matching-Algorithmus und die Aspekte der Implementation diskutiert. Im dritten Abschnitt werden schliesslich Vorschläge für weitere Forschungen gegeben.

### 7.1 Erreichtes

Multiple Repräsentationen und MRDBs wurden definiert. Die beiden grundsätzlichen Möglichkeiten für die Datenmodellierung von MRDBs, die starke und die schwache Integration, wurden miteinander verglichen. Die Anwendungen von Multirepräsentationsdatenbanken wurden durch verschiedene Beispiele aufgezeigt.

Es wurde ein konzeptioneller Rahmen für Matching-Prozesse erarbeitet. Zentral bei Matching-Prozessen ist die Bestimmung der Ähnlichkeit von räumlichen Objekten. Es wurden verschiedene Ähnlichkeitsmasse vorgestellt, die sich auf semantische, geometrische und topologische Eigenschaften abstützen. Bestehende Ansätze zum Matching von Strassendaten wurden zusammengefasst.

Die beiden zu integrierenden Datensätze VECTOR25 und VECTOR200 wurden bezüglich ihrer Datenmodelle und der Erfassung der Geometrie miteinander verglichen. Der Vergleich zeigte einige Schwächen der bestehenden Algorithmen beim Matching von stark unterschiedlichen Massstäben auf.

Daher wurde ein neuer Matching-Ansatz für Strassendaten stark unterschiedlicher Massstäbe erarbeitet. Das Verfahren arbeitet in zwei Schritten: Zuerst werden VECTOR200-Knoten mit VECTOR25-Knoten verknüpft; die Knotenverknüpfungen werden im zweiten Schritt zu Strassenverknüpfungen umgesetzt.

Der Matching-Ansatz wurde in einer prototypischen MRDB-Applikation in Java implementiert. Es wurde ein neuer Visualisierungsalgorithmus für Verknüpfungen von Liniendaten erarbeitet. Zudem ermöglichen verschiedene Werkzeuge eine einfache Manipulation der MRDB.

Schliesslich wurde die Praxistauglichkeit des Matching-Ansatzes mit realen Kartendaten geprüft.

## 7.2 Diskussion

### Vergleich mit bestehenden Matching-Verfahren

Bestehende Matching-Verfahren wurden meist für Datensätze geschaffen, die in gleichem oder zumindest ähnlichem Detaillierungsgrad vorliegen. In solchen Fällen lassen sich Korrespondenzen gut durch die Berechnung der Ähnlichkeit individueller Objekte – Vergleich der Linielänge, Linienrichtung, der Grade der Endknoten, etc. – erkennen. Im Fall von stark unterschiedlichen Massstäben können die Verläufe korrespondierender Strassen aufgrund der Generalisierung lokal unterschiedlich sein: Im kleineren Massstab werden kurvenreiche Strassen geglättet wiedergegeben; bei Strassenschlaufen findet eine starke Verdrängung statt; einzelne, markante Kurven bleiben erhalten, werden aber vergrössert; schliesslich korrespondieren oft viele kurze Strassenstücke aus dem grösseren Massstab mit einer einzigen Strassen aus dem kleineren Massstab. Daher ist der Vergleich von individuellen Ähnlichkeitsmassen hier wenig sinnvoll und es musste ein neuer Ansatz gefunden werden. Eines der grössten Probleme war die Reduktion des dichten VECTOR25-Strassennetzes auf die wenigen Elemente, die mit dem VECTOR200-Strassennetz korrespondieren. Mit der Bildung von *nächsten benachbarten Wegen* konnte dafür eine gute Lösung gefunden werden.

### Behandlung von 1:N-Beziehungen zwischen Knoten

Der grösste Nachbearbeitungsaufwand entsteht bei N:1-Beziehungen zwischen Knoten, wie sie beim Kollaps eines Kreisverkehrs oder eines Strassenstücks zu einem Knoten vorkommen. Diese Fälle bleiben derzeit unbehandelt und der Benutzer muss die einfallenden Strassen von Hand verknüpfen. Der Nachbearbeitungsaufwand könnte einfach verkleinert werden, indem der Benutzer in der interaktiven Phase auch Beziehungsobjekte für Kollapse erstellen kann statt nur 1:1-Verknüpfungen. Die optimale Lösung läge in einer vollständig automatischen Erkennung von Segmentkollapsen. Da sie jedoch relativ selten vorkommen, schien der Entwicklungsaufwand dafür nicht gerechtfertigt.

### Eignung des Verfahrens für verschiedene Gebietstypen

In Gebieten mit niedriger bis mittlerer Besiedlungsdichte liefert das entwickelte Matching-Verfahren sehr gute Resultate. In komplexen urbanen Gebieten wie der Innenstadt von Zürich waren die Ergebnisse hingegen nicht zufrieden stellend. Teilweise waren die Situationen inkonsistent oder unterschieden sich so stark, dass auch manuell nur schwer eine Zuordnung gefunden werden konnte. Die mittlere Zwischenwinkelsumme eignet sich ausserdem in Stadtgebieten weniger gut als Vergleichsmass, weil sich dort Strassen in der Regel rechthöckrig schneiden. Da viele der Quartierstrassen gerade sind, würde ein mit Linienmassen angereicherter Algorithmus in diesen

Gebieten bessere Resultate erzielen. Der Einbezug von Linienmassen wurde in weniger dicht besiedelten Gebieten geprüft und hatte dort einen eher negativen Einfluss auf die Ergebnisse.

### Anwendung des Verfahrens für grossflächige Gebiete

Der grösste der getesteten Datensätze umfasste eine Fläche von  $330 \text{ km}^2$ , was etwa dem 1.5 fachen der Fläche eines Kartenblattes der LK 1:25'000 entspricht ( $17.5 \text{ km} \times 12 \text{ km} = 210 \text{ km}^2$ ). Datensätze dieser Grösse lassen sich problemlos mit dem Matching-Verfahren behandeln. Liegen zwei Kartenobjekte genügend weit voneinander entfernt, so lassen sie sich unabhängig voneinander verknüpfen, weil ihre Kandidatenmengen nicht miteinander in Konflikt stehen. Die Bearbeitung grösserer Ausschnitte als der eines Kartenblattes macht also wenig Sinn. Sollen beispielsweise Datensätze der gesamten Schweiz verknüpft werden, ist es sinnvoller, diese in einzelne Kacheln zu unterteilen, die unabhängig voneinander und von mehreren Benutzern gleichzeitig bearbeitet werden können. Auf dieses sog. *Tiling* wird im Ausblick eingegangen.

### Implementation

JTS und JUMP sind stabile, weitgehend fehlerfreie Plattformen für räumliche Applikationen. Sie erlauben es, auf einem hohen Niveau zu programmieren und ersparen dem Applikationsentwickler dadurch viel Zeit. Trotzdem sind sie einfach erweiterbar und dadurch sehr flexibel einsetzbar. Leider ist vor allem JUMP schlecht dokumentiert. Wie Mauswerkzeuge oder *Renderer* zur Visualisierung von Ebenen funktionieren, konnte nur anhand des Quellcodes nachvollzogen werden.

Dank der Mitbenutzung der DRIVE-MRDB mussten nur die Matching-Funktionalität selbst implementiert werden. Die Arbeit zeigt, wie die DRIVE-MRDB für andere Zwecke als Generalisierung verwendet und wie Benutzerinteraktion mit der DRIVE-MRDB realisiert werden kann.

## 7.3 Ausblick

### 7.3.1 Ausbau des Prototyps

#### 7.3.1.1 Entwicklung eines Vertrauensmasses

Der in der vorliegenden Arbeit entwickelte Matching-Ansatz verwendet nur Beschränkungen und algorithmische Komponenten. Die Bildung eines Gesamtmasses wie in Abschnitt 3.2.4 besprochen wurde nicht vorgenommen. Wie in Abschnitt 6.2.1 gezeigt wurde, kann die Zuordnungsrate in einigen Fällen verbessert werden, indem anstatt der absoluten Kriterien „Knoten ist der nächste und Knoten hat die kleinste Zwischenwinkelsumme“ ein Ähnlichkeitsmass gebildet wird:

$$\Psi(K_1, K_2) = \sum w_i \times \sigma_i(K_1, K_2) \quad (7.1)$$

Die Masse  $\sigma_i$  sind die Knotendistanz und die mittlere Zwischenwinkelsumme. Wenn Nachbarknoten schon zugeordnet wurden, können auch Linienmasse einfallender Strassen berücksichtigt

werden, beispielsweise die Differenz der Sinuosität zwischen Linienkandidat und VECTOR200-Referenzstrasse oder die Winkeldifferenz der Basislinien.

Neben der Bestimmung der Gewichte ist ein weiteres Problem bei dieser Art von Vertrauensmass, dass leicht Fehlzuordnungen geschehen können. Eine Fehlzuordnung zu korrigieren ist aufwändiger, als eine fehlende Zuordnung zu ergänzen. Aus diesem Grund wurde auf die Verwendung eines Gesamtmasses in der vorliegenden Arbeit verzichtet. Ein Test mit Verwendung von Vertrauensmassen kann Aufschluss darüber geben, ob das Matching von stark unterschiedlichen Massstäben dadurch verbessert werden kann.

### 7.3.1.2 Aufteilen grosser Datensätze (*Tiling*)

Soll der Prototyp zu einem Produktionssystem ausgebaut werden, müssen sehr viel grössere Datenbestände unterstützt werden als die in dieser Arbeit untersuchten Testgebiete, beispielsweise der gesamte VECTOR25/VECTOR200-Datensatz der Schweiz. Es ist nicht effizient und oft auch nicht möglich, den gesamten Datensatz im Computerspeicher zu halten. Es werden nur noch Ausschnitte davon bearbeitet und oft arbeiten verschiedene Benutzer an mehreren Ausschnitten gleichzeitig. Dafür eignen sich ZIP-Archive zur persistenten Haltung der Daten nicht. Es muss ein räumliches Datenbanksystem eingesetzt werden, das mit *langen Transaktionen* für den konsistenten Mehrbenutzerbetrieb sorgt. Die Funktionalität des Matching-Prototyps muss so erweitert werden, dass neue Matches in Ausschnitten konsistent in den bestehenden Datenbestand eingespielt werden können. Effekte am Rand des Gebietes müssen dabei berücksichtigt werden, indem z. B. Strassen, die den Rand des Gebietes schneiden, nicht zugeordnet werden.

## 7.3.2 Entwicklung einer generellen Matching-Plattform

Es existiert heute eine grosse Anzahl von verschiedenen Matching-Algorithmen, von denen jeder für bestimmte Datensätze geschaffen wurde und die für die verwendeten Testgebiete gut funktionieren. Da sie als Prototypen auf verschiedenen, oft proprietären Plattformen geschaffen wurden, lässt sich nicht überprüfen, wie gut sie für das Matching grösserer Datenbestände oder anderer Datensätze geeignet sind. So entstehen parallel viele eigenständige, nicht wiederverwendbare Lösungen.

Sowohl in der Industrie wie in der Forschung besteht eine Nachfrage nach Matching-Lösungen. Daher wären generelle Matching-Werkzeuge, die als Modul in einem Standard-GIS verfügbar sind, von grossem Nutzen. Den heutigen Matching-Prototypen fehlt die Flexibilität, um sich den Eigenschaften verschiedener Datensätze anzupassen. Es fehlt auch an Erfahrung, wie gut sich die bekannten Ansätze für verschiedene Daten eignen. Zukünftige Forschung sollte sich mit diesen beiden Problemkreisen befassen.

Um eine Matching-Plattform flexibel einzusetzen, muss der Matching-Prozess in einem bestimmten Rahmen konfigurierbar sein. Beispielsweise werden immer wieder Attributdaten verwendet, wobei die Attributsemantik aber verschieden sein kann. Dem kann begegnet werden, indem der Matching-Prozess aus einzelnen Modulen aufgebaut ist. Eine Kategorisierung nach dem in Kapitel 3.1 präsentierten Rahmen eignet sich dafür gut. Das Modul *SimilarityMeasure\_StringCategoryMatrix* berechnet beispielsweise die Ähnlichkeit zweier

String-Attribute nach einer vom Benutzer definierbaren Ähnlichkeitsmatrix. Das Modul *Matching\_SimilaritySumWeighted* nimmt einen Match vor, wenn der gewichtete Durchschnitt von definierbaren Ähnlichkeitsmassen eine bestimmte Sicherheitsschwelle übertrifft. So kann für jede Matching-Phase eine Menge von Alternativen angeboten werden. Es ist eine graphische Oberfläche denkbar, wo der Benutzer aus den einzelnen Modulen den auf seine Datensätze abgestimmten Matching-Workflow zusammensetzen kann.

Die Matching-Plattform muss offengelegte Schnittstellen haben, damit selbst programmierte Speziallösungen in Form von Modulen eingebunden werden können. Der Autor schlägt eine auf *Web Services* aufbauende Architektur vor. Web Services sind Software-Applikationen, die mit anderen Software-Agenten XML-basierte Nachrichten über Internet-basierte Protokolle austauschen können (Alonso et al. 2004:124). Web Services sind plattformunabhängig, weil spezifische, weit verbreitete Standards für den Austausch von Nachrichten verwendet werden. Die Interaktion zwischen Web Services geschieht beispielsweise mit dem *Simple Object Access Protocol* (SOAP). SOAP-Nachrichten können zwischen verschiedenen Endpunkten durch HTTP-Anfragen übertragen werden (Alonso et al. 2004:155–165). Die Art der Dienstleistung, benötigte Parameter und Funktionsrückgabe der beteiligten Web Services müssen dazu bekannt sein. Die Beschreibung der Web Services wird deshalb in der Beschreibungssprache *Web Services Description Language* (WSDL) (Alonso et al. 2004:165–174) formuliert und in einem Verzeichnis veröffentlicht.

Burghardt et al. (2005) erörtern die Bereitstellung von Generalisierungsfunktionen im Internet durch Web Services. Analog können auch Matching-Dienste angeboten werden. Ein Modul, das die Ähnlichkeit zweier Objekte misst, könnte in einer SOAP-Nachricht beispielsweise die beiden Objekte im GML-Format<sup>1</sup> erhalten und das Resultat als double zurückgeben. Eigene Komponenten können so einfach eingebunden werden.

Auf dieser offenen, erweiterbaren Plattform sollten wichtige bestehende Matching-Ansätze implementiert und an verschiedenen Datensätzen verglichen werden. Daraus kann ein Leitfaden erarbeitet werden, der dem Benutzer beim Zusammenstellen eines Matching-Prozesses hilft. Es können auch Standard-Szenarios festgelegt werden, die dem Benutzer im Sinne eines Expertensystems angepasste Matching-Prozesse vorgeben.

---

1. Geography Markup Language (GML) ist eine XML-basierte Sprache zur Beschreibung von räumlichen Daten. Siehe <http://www.opengeospatial.org/specs/>. Stand 9.1.2006



# Literaturverzeichnis

- ALONSO, G., CASATI, F., KUNO, H. UND MACHIRAJU, V. (2004): *Web Services. Concepts, Architectures and Applications*. Springer-Verlag, Berlin.
- ANDERS, K.-H., BILDIRICI, Ö. UND SESTER, M. (2003): Erzeugung und Visualisierung von MRDB-Daten mit ArcGIS. Vortrag an der 40. Sitzung der Arbeitsgruppe Automation in der Kartographie (AgA), 23.–24. September, Erfurt.
- BADARD, T. (2000): *Propagation des mises à jour dans les bases de données géographiques multi-représentations par analyse des changements géographiques*. Dissertation, Université de Marne-la-Vallée.
- BARD, S. (2004): *Méthode d'évaluation de la qualité de données géographiques généralisées. Application aux données urbaines*. Dissertation, Université de Paris 6.
- BALLEY, S., PARENT, C. UND SPACCAPIETRA, S. (2004): Modelling geographic data with multiple representations. In: *International Journal of Geographical Information Science*, **18** (4), 327–352.
- BERNIER, E., BÉDARD, Y. UND HUBERT, F. (2005): UMapIT: An On-Demand Web Mapping Tool Based On A Multiple Representation Database. 8th ICA Workshop on Generalisation and Multiple Representation, A Coruña, July 7–8th, 2005.
- BOBZIEN, M., BURGHARDT, D. UND PETZOLD, I. (2005): Automatische Ableitung Digitaler Vektormodelle – Projekt Drive. Erscheint in: *Mitteilungen des Bundesamtes für Kartographie und Geodäsie*.
- BURGHARDT, D., NEUN, M., WEIBEL, R. (2005): Generalization Services on the Web – Classification and an Initial Prototype Implementation. In: *Cartography and Geographic Information Science*, **32** (4), 257–268.
- CECCONI, A. (2003): *Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping*. Dissertation, Universität Zürich.

- COBB, M.A., CHUNG, M.J., FOLEY, H., PETRY, F.E. UND SHAW, K.B. (1998): A Rule-based Approach for the Conflation of Attributed Vector Data. In: *GeoInformatica*, **2** (1), 7–35.
- DEVOGELE, T., TREVISAN, J. UND RAYNAL, L. (1996): Building a multi-scale database with scale-transition relationships. In: *Advances in GIS Research II: Proceedings 7th International Symposium on Spatial Data Handling*, Delft, August 12–16th, 1996, Session 6, 19–33.
- DEVOGELE, T. (1997): *Processus d'intégration et d'appariement de Bases de Données Géographiques. Application à une base de données routières multi-échelles*. Dissertation, Université de Versailles.
- DUNKARS, M. (2003): Matching of Datasets. In: *ScanGIS'2003 – The 9th Scandinavian Research Conference on Geographical Information Science*, 4–6 June 2003, Espoo, Finland – Proceedings, 67–78.
- EGENHOFER, M.J. (1991): Reasoning about Binary Topological Relations. In: *Advances in Spatial Databases. Proceedings of the Second Symposium on Large Spatial Databases*, Zurich, August 20–28th, 1991, 143–160.
- ELMASRI, R. UND NAVATHE, S.B. (2004): *Fundamentals of Database Systems*. Fourth Edition, International Edition, Pearson Addison-Wesley, Boston.
- GLINZ M. (2001): *Modellierung*. Skript zur Vorlesung Informatik II, Universität Zürich.
- GOODCHILD, M.F. (1997): A simple positional accuracy measure for linear features. In: *International Journal of Geographical Information Science*, **11** (3), 299–306.
- GULBINS, J. UND OBERMAYR, K. (1999): *Desktop Publishing mit FrameMaker*. 3. Auflage, Springer-Verlag, Berlin.
- HAKE, G. UND GRÜNREICH, D. (1994): *Kartographie*. 7. Auflage, de Gruyter, Berlin.
- HAMPE, M. UND SESTER, M. (2004): Generating and using a Multi-representation Database (MRDB) for mobile applications. ICA Workshop on Generalisation and Multiple representation, Leicester, August 20–21th, 2004.
- HANGOÛËT, J.F. (1995): Computation of the Hausdorff Distance Between plane Vector Polylines. In: *Auto-Carto 12. ACSM/ASPRS Annual Convention & Exposition Technical Papers*, Bethesda, **4**, 1–10.
- HANGOÛËT, J.F. (2004): Geographical multi-representation: striving for the hyphenation. In: *International Journal of Geographical Information Science*, **18** (4), 309–326.

- HARRIE, L. UND HELLSTRÖM, A.-K. (1999): A Prototype System for Propagating Updates Between Cartographic Data Sets. In: *The Cartographic Journal*, **36** (2), 133–140.
- HARVEY, F. UND VAUGLIN, F. (1996): Geometric match processing: applying multiple tolerances. In: *Advances in GIS Research II: Proceedings 7th International Symposium on Spatial Data Handling*, Delft, August 12–16th, 1996, Session 4A, 13–29.
- KILPELÄINEN, T. UND SARJAKOSKI, T. (1995): Incremental generalization for multiple representations of geographical objects. In: Müller, J.C., Lagrange, J.P. und Weibel, R. (Hrsg.) (1995): *GIS and generalization: Methodology and Practice*. Taylor & Francis, London, 209–218.
- KREITER, N. (2003): Projekt OPTINA-LK: Neuaufbau der Landeskarten. In: *Jahresbericht 2003 des Bundesamtes für Landestopographie (swisstopo)*, Wabern, S. 31.
- JONES, C.B. UND ABRAHAM, I.M. (1986): Design considerations for a scale-independent cartographic database. In: *Proceedings 2nd International Symposium on Spatial Data Handling*, Seattle, July 5–10th, 1986, 384–398.
- LEMARIÉ, C. UND RAYNAL, L. (1996): Geographic data matching: first investigations for a generic tool. In: *GIS/LIS '96 Annual Conference and Exposition: Proceedings*, Denver, November 19–21th, 1996, 405–420.
- LONGLEY, P.A., GOODCHILD, M.F., MAGUIRE, D.J. UND RHIND, D.W. (2001): *Geographic Information Systems and Science*. John Wiley & Sons, Chichester.
- MANTEL, D. UND LIPECK, U. (2004): Matching Cartographic Objects in Spatial Databases. In: *Proceedings of the XXth Congress of the ISPRS*, Istanbul, July 12–23th, 2004, Int. Archives of Photogrammetry, Remote Sensing and Spatial Inf. Sciences Vol. XXXV, Commission IV Papers, Part B4, 172–176.
- MATOUSEK, J. UND NESETRIL, J. (1998): *Invitation to Discrete Mathematics*. Oxford University Press, New York.
- MCMASTER, R.B. (1986): A Statistical Analysis of Mathematical Measures for Linear Simplification. In: *The American Cartographer*, **13** (2), 103–116.
- MENG, L. (2000): ATKIS: Modell- und kartographische Generalisierung. Vorstudien zum AdV-Forschungs- und Entwicklungsvorhaben.
- MENG, L. UND TÖLLNER, D. (2004): Ein Reverse-Engineering-Ansatz zur Generalisierung topographischer Daten. In: *Kartographische Nachrichten*, Nr. 4, 2004, 159–163.

- PETZOLD, I., BURGHARDT, D. UND BOBZIEN, M. (2005): Automated derivation of town plans from large scale data – on an example of area to line simplification. *Proceedings of the 22nd ICA International Cartographic Conference*, A Coruña, July 9–16th, 2005.
- RIGAUX, P., SCHOLL, M. UND VOISARD, A. (2002): *Spatial Databases. With Application to GIS*. Morgan Kaufmann, San Francisco.
- ROSEN, B. UND SAALFELD, A. (1985): Match criteria for automatic alignment. In: *Digital representations of spatial knowledge: Auto-Carto 7 proceedings*, Washington, March 11–14th, 1985, 456–462.
- SAALFELD, A. (1988): Automated map compilation. In: *International Journal of Geographical Information Systems*, **2** (3), 217–228.
- SAMAL, A., SETH, S. UND CUETO, K. (2004): A feature-based approach to conflation of geospatial sources. In: *International Journal of Geographical Information Science*, **18** (5), 459–489.
- SEDGEWICK, R. (2003): *Algorithms in Java. Part 5: Graph algorithms*. 3. Auflage, Addison-Wesley, Boston.
- SESTER, M., ANDERS, K.-H. UND WALTER, V. (1998): Linking Objects of Different Spatial Data Sets by Integration and Aggregation. In: *GeoInformatica*, **2** (4), 335–358.
- SHIELDS, D.J., TODD, S.W. UND BROWN, D.D. (1996): Determining Match Probabilities Between Two Mineral Location Databases Using Logistic Regression. In: *GIS/LIS '96 Annual Conference and Exposition: Proceedings*, Denver, November 19–21th, 1996, 421–434.
- STADLER, A. (2004): *Verknüpfung korrespondierender Kartenelemente im Hinblick auf automatisierte Fortführung*. Diplomarbeit, Technische Universität Wien.
- STIGMAR, H. (2004): Merging Route Data and Cartographic Data. ICA Workshop on Generalisation and Multiple representation, Leicester, August 20–21th, 2004.
- SWISSTOPO (2004a): *VECTOR25. Das digitale Landschaftsmodell der Schweiz. Produkteinformation*. Bundesamt für Landestopografie, Wabern, 2004.
- SWISSTOPO (2004b): *VECTOR200. Das kleinmassstäbliche digitale Landschaftsmodell der Schweiz. Produkteinformation*. Bundesamt für Landestopografie, Wabern, 2004.
- THOM, S. (2005): A Strategy for Collapsing OS Integrated Transport Network™ dual carriageways. 8th ICA Workshop on Generalisation and Multiple Representation, A Coruña, July 7–8th, 2005.

- TIMPF, S. UND DEVOGELE, T. (1997): New Tools for Multiple Representations. In: *Proceedings of the 18th ICA International Cartographic Conference*, Stockholm, July 23–27th, 1997, 1381–1386.
- TIMPF, S. (1998): Map Cube Model – a model for multi-scale data. In: *Proceedings of 8th International Symposium on Spatial Data Handling*, Vancouver, July 11–15th, 190–201.
- TIMPF, S. UND KUHN, W. (2003): Granularity Transformations in Wayfinding. In: Freksa, C., Brauer, W., Habel, C. und Wender, K.F. (Hrsg.) (2003): *Spatial Cognition III*. Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 77–88.
- THOMSON, R.C. UND RICHARDSON, D.E. (1999). The ‚Good Continuation‘ Principle of Perceptual Organization applied to the Generalization of Road Networks. In: *Proceedings of the 19th ICA International Cartographic Conference*, Ottawa, August 14–21th, 1999, 1215–1223.
- WALTER, V. (1996): *Zuordnung von raumbezogenen Daten – am Beispiel der Datenmodelle ATKIS und GDF*. Dissertation, Universität Stuttgart.
- WALTER, V. UND FRITSCH, D. (1999): Matching spatial data sets: a statistical approach. In: *International Journal of Geographical Information Science*, **13** (5), 445–473.
- WEIBEL, R. (1997): Generalization of Spatial Data: Principles and Selected Algorithms. In: Van Kreveld, M., Nievergelt, J., Roos, T. und Widmayer, P. (Hrsg.) (1997): *Algorithmic Foundations of Geographic Information Systems*. Springer-Verlag, Berlin, 99–152.
- YUAN, S. UND TAO, C. (1999): Development of conflation components. In: *Geoinformatics and Socioinformatics. The Proceedings of Geoinformatics '99 Conference*, Ann Arbor, June 19–21th, 1999, 1–13.
- ZEILER, M. (1999): *Modeling Our World. The ESRI Guide to Geodatabase Design*. ESRI Press, Redlands.
- ZHANG, M., SHI, W. UND MENG, L. (2005): A generic matching algorithm for line networks of different resolutions. 8th ICA Workshop on Generalisation and Multiple Representation, A Coruña, July 7–8th, 2005.

