TECHNISCHE UNIVERSITÄT MÜNCHEN

Institut für Photogrammetrie und Kartographie
Lehrstuhl für Kartographie

Methods and Implementations of Road-Network Matching

Meng Zhang

Vollständiger Abdruck der von der Fakultät für Bauingenieur- und Vermessungswesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.- Prof. Dr.-Ing. Matthäus Schilcher

Prüfer der Dissertation:

1. Univ.- Prof. Dr.-Ing. Liqiu Meng
2. Univ.- Prof. Dr. rer. nat. Udo Lipeck,
   Leibniz Universität Hannover

Die Dissertation wurde am 14.10.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Bauingenieur- und Vermessungswesen am 07.12.2009 angenommen.

To my parents, brother and especially my wife; without their love and support, any of my achievements would have not been possible.

吾听吾忘
吾见吾记
吾做吾悟
−孔子

I hear and I forget,
I see and I remember,
I do and I understand.
- Confucius

Abstract

The growing demand on geospatial services requires an emphasized study on geo-information from various sources covering the same geographic space. Data matching, which aims at establishing logical connections (linkages) between corresponding objects or object parts in two comparable datasets, is one of the fundamental measures that helps make different datasets interoperable. This thesis is devoted to an automatic matching approach for road networks.

A road network serves in many cases as the geometric and functional backbone in a comprehensive digital landscape model. Street matching has been intensively and extensively researched during the last decade. Buffer Growing and Iterative Closest Point are two popular matching algorithms that have been most frequently cited in literature so far. A majority of the existing matching approaches based on these two algorithms or their combination reveal a high matching rate and efficiency on certain data types of selected test areas. However, the problem of uncertain matching remains either in areas where the context is too complex or when one of the datasets contains little or no meaningful semantic information. Under the assumption that if more context information could be involved, the matching result would be better, the author has developed and implemented a new contextual matching approach based on the Delimited-Stroke-Oriented (DSO) algorithm. Extensive experiments with various road networks have convincingly confirmed the generic nature and high performance of this contextual matching approach.

The DSO algorithm consists of five processes: (1) data pre-processing to reduce the noise of irrelevant details and eliminate topologic ambiguities in the datasets to be matched; (2) construction of the 'graph' to record the relationships between conjoint objects; (3) connection of the Delimited Strokes; (4) matching of the Delimited Strokes; and (5) treatment of fragmental matching areas. As compared to Buffer Growing and Iterative Closest Point algorithm, the DSO algorithm is able to make use of topologic information more conveniently and sufficiently, which allows a context-related topologic analysis, thus helps to improve the results of geometric or semantic matching. In principle, the DSO algorithm can be applied to match all kinds of road networks from various data resources, as it does not rely on any semantic information at all.

Furthermore, the automatic matching process is flanked by three assisting methodologies:

- Matching guided by *'structure'*: to a certain extent, a street network can be regarded as a unit constituted by various road structures. Since various road structures take on different geometric or topologic characteristics, it is hardly possible to match all of them efficiently by the same criteria or methods. An efficient way to circumvent this problem is to classify the road network into several structure categories; then develop a category-sensitive matching strategy with proper parameters or criteria.

- Matching guided by *'semantics'*: besides spatial knowledge, some semantic attributes can also be utilized to guide the matching calculation. Two topics are discussed in depth: (a) how to calculate the semantic similarity*;* and (b) how to take advantage of the semantic similarity in different matching scenarios.

- Matching guided by *'spatial index'*: - a good matching algorithm should be able to create high-quality results at a high speed. To this end, two grid-based spatial indexes have been established - one is for point data and the other for linear data. With spatial indexes, the process of road-network matching has been dramatically accelerated, especially when the involved datasets are very large.

Being supported by the extendable Delimited Strokes*,* network-based matching and the three assisting methodologies, the contextual matching approach is able to handle geometric, topologic and semantic information in a large matching environment and provide a considerably improved matching performance in terms of 'automatic matching rate and certainty', 'high computing speed', and ‘robustness and generic nature’. Its underlying theoretical and methodological concepts have been successfully applied in three real-world projects, viz.: (i) Postal data integration, (ii) Integration of the routing-relevant information from different datasets, and (iii) Conflation of pedestrian ways between different datasets. The projects (i) and (ii) were sponsored by the German Federal Agency for Cartography (BKG) while the project (iii) was funded by the Corp. United Maps.

The prototypical experiments with test beds representing extensive road networks of real world have not only proved the high performance of the new contextual approach, but brought in a lot of practical experiences. Due to its large potentials of enriching mega datasets, the contextual matching approach is being commercialized.

Zusammenfassung

Die wachsende Nachfrage nach raumbezogenen Dienstleistungen erfordert eine intensive Untersuchung der Geo-Informationen aus verschiedenen Quellen, die denselben geografischen Raum beschreiben. Daten-Matching, das sich auf die logischen Verbindungen zwischen den entsprechenden Objekten oder Objektteilen in zwei vergleichbaren Datensätzen bezieht, ist eine der grundlegenden Maßnahmen, die dazu beitragen, unterschiedliche Datensätze interoperativ zu bearbeiten. Die vorliegende Arbeit widmet sich einem automatischen Matching-Ansatz für Straßennetze.

Ein Straßennetz dient in vielen Fällen als geometrisches und funktionales Gerüst eines umfassenden digitalen Geländemodells. In den letzten zehn Jahren wurde das Thema Straßen-Matching intensiv und ausgiebig untersucht. Buffer-Growing und Iterative-Closest-Point sind zwei meist zitierte Algorithmen. Eine Mehrheit der entwickelten Matching-Ansätze, die auf diesen beiden Algorithmen oder deren Kombination basieren, zeigen eine hohe Matching-Rate und -Effizienz auf bestimmte Datentypen der ausgewählten Testsbereiche. Allerdings bleibt das Problem des unsicheren Matchings entweder in Bereichen, wo der Kontext zu komplex ist, oder wenn einer der Datensätze wenig bzw. keine nützlichen semantischen Informationen enthält. Unter der allgemeinen Annahme, dass bessere Matching-Ergebnisse zu erwarten wären, wenn mehr Kontextinformationen einbezogen werden könnten, wurde in der vorliegenden Arbeit ein neuer kontextueller Matching-Ansatz entwickelt, der auf dem Delimited-Stroke-Oriented (DSO)-Algorithmus basiert und in der Lage ist, ein robusteres und generisches Daten-Matching zwischen verschiedene Straßennetzen zu realisieren.

Der DSO-Algorithmus besteht aus fünf Prozessen: (1) Datenvorverarbeitung zur Reduzierung der irrelevanten Informationen und zur Eliminierung der topologischen Unklarheiten in den Datensätzen; (2) Aufbau des „Graphen“ um die Beziehungen zwischen Conjoint-Objekte explizit aufzuzeichnen; (3) Verbindung der Delimited-Strokes, (4) Matching der Delimited-Strokes; und (5) Bearbeitung von fragmentalen Matchingsbereichen. Im Vergleich zu Buffer-Growing und Iterative-Closest-Point ist der DSO-Algorithmus in der Lage, topologische Informationen zweckmäßig und adäquat auszunutzen, welches eine kontextbezogene topologische Analyse ermöglicht und somit die Ergebnisse des geometrischen oder semantischen Matching verbessert. Im Prinzip lässt sich der DSO-Algorithmus wegen seiner Unabhängigkeit von semantischen Informationen allgemein implementieren, um alle Arten von Straßennetzen aus verschiedenen Datenquellen zusammenzubringen.

Darüber hinaus kommen drei Ergänzungsmethoden dem automatischen Matching-Prozess zugute:

- Struktur-gestütztes Matching - ein Straßennetz lässt sich als eine aus verschiedenen Straßenstrukturen bestehende Einheit betrachten. Da den verschiedenen Straßenstrukturen unterschiedlichen geometrische oder topologische Eigenschaften zuzuordnen sind, ist es kaum möglich, alle von ihnen nach denselben Kriterien und Methoden effizient zu behandeln. Ein pragmatischer Lösungsweg besteht darin, das Straßennetz in mehrere Kategorien zu klassifizieren und anschließend eine Kategorie-spezifische Matching-Strategie mit entsprechenden Parametern anzuwenden.
- Semantik-gestütztes Matching - Neben dem räumlichen Wissen kommen auch einige semantische Attribute zum sinnvollen Einsatz. Dabei geht es um zwei wesentliche Fragestellungen (a) wie wird die semantische Ähnlichkeit berechnet? und (b) wie wird die semantische Ähnlichkeit in verschiedene Matching-Szenarien einbezogen?
- Index-gestütztes Matching - ein leistungsfähiger Matching-Algorithmus soll in der Lage sein, hochqualitative Ergebnisse in hoher Geschwindigkeit zu erzeugen. Zur Erhöhung der Rechengeschwindigkeit wurden zwei Grid-basierte räumliche Indizes jeweils für punkthafte und

linienhafte Daten gebildet. Mittels dieser Indizes wurde der Prozess des Straßennetz-Matching dramatisch beschleunigt, insbesondere bei sehr großen Datensätzen.

Unter Verwendung von erweiterbaren Delimited-Strokes, des netzwerkbezogenen Matchings und drei Ergänzungsmethoden ist der kontextuelle Matching-Ansatz in der Lage, die geometrischen, topologischen und semantischen Informationen zusammenhängend zu betrachten. Daraus entsteht eine deutlich verbesserte Matching-Leistung bezüglich der automatischen Matching-Rate und -Sicherheit, der hohen Rechengeschwindigkeit, der Robustheit sowie der generischen Natur. Der Matching-Ansatz hat ein hohes Potential zur Anreicherung von Mega-Daten. Er wurde bereits in drei Projekten erfolgreich implementiert: (i) Integration von Postdaten, (ii) Integration von Routing-relevanten Informationen aus verschiedenen Datensätzen, und (iii) Verschmelzung von Fußgängerwegen aus verschiedenen Datensätzen. Projekte (i) und (ii) wurden vom Bundesamt für Kartographie (BKG) finanziert. Projekt (iii) wurde vom Unternehmen United Maps unterstützt.

Die prototypischen Experimente haben nicht nur die Leistungsfähigkeit des kontextuellen Matching-Ansatzes überzeugend bestätigt, sondern auch umfassende praktische Erfahrungen mit sich gebracht. Im Hinblick auf das Potential zur Anreicherung von großen Datenmengen wird zurzeit der kontextuelle Matching-Ansatz kommerzialisiert.

Table of Contents

Chapter 1

Introduction

With the rapid improvement of geospatial data acquisition and processing techniques, large amount of geospatial data from various public and private organizations has become readily available. Apart from the thematic diversities, these datasets may cover the same geographic space and differ in geometry, accuracy, actuality and resolution. Often one dataset may be superior to other datasets in one, but not all aspects. Therefore, various datasets have to work in concert so that various maps and analytical functions can be generated for various applications. Their efficient use depends strongly on how far they can be made interoperable. Depending on the contents and resolutions of datasets to be interoperated, three ways of data sharing are possible: (i) objects in both datasets are separately stored but linked with each other so that information incl. updates of one dataset can be propagated to the other, (ii) one dataset is entirely integrated into the other, and (iii) both datasets are merged into a new one. The interoperability of sharable datasets is generally regarded as a prerequisite for the realization as well as power enhancement of query and analytical functions in a geographic information system. One of the fundamental measures for the interoperability is to establish logical connections between corresponding features in comparable datasets by means of data matching. The objectives of data matching include eliminating data discrepancy and thus increasing spatial accuracy and consistency, updating or adding new spatial features into datasets, updating or adding more attributes that associate with the spatial features of the datasets, etc. (Yuan and Tao 1999). This thesis is devoted to a highly automatic, generic and efficient matching approach to correlate the corresponding road networks.

1.1 Motivation

The process to bring together diverse data sources that reveal a strong geospatial and/or semantic similarity is termed as data matching. With the aim to make different geospatial datasets interoperable, data matching serves a number of useful purposes in the context of cartography and geographic information science (GIS):

- Data matching can add the value of existing data by transferring attributes or object classes from one dataset to another. For example, different matching techniques have been developed in United States of America for the attribute exchange of US Census Bureau TIGER data with other datasets. The TIGER database contains a large number of different attributes but it is spatially very often inaccurate. Therefore, many users want to integrate the highly attributed TIGER data with their own captured data with a more accurate geometry (Tomaselli 1994; Brown et al. 1995; Kang 2001; Song et al. 2006), which requires, as a first step, the matching between different datasets.
- Matching techniques can be extended to support automatic updating processes. On this occasion, the objects of different datasets are matched and the desired data transferred from one dataset to the other (Stigmar 2006). Important is, as pointed out by Jones et al. (1996), that the updating procedure is able to identify when original data should be replaced and when new data should be simply appended. For instance, with the help of the matching approach developed by Kang (2001), administrators in Delaware County of Ohio, USA successfully updated the county's 2000 collection blocks, corrected inaccurate addresses and identified missing housing units and their locations. This research allows the local governments to

correlate their in-house detailed parcel data with demographic data at the block level, which permits very interesting and intricate statistical, sociological, and spatial analysis on growth and change patterns.

- Data matching can help to evaluate and improve the data quality as well. In order to check the currency and correctness, one dataset can be compared to a second or third dataset, e.g. Gösseln and Sester (2004) investigated the geometric differences among different datasets of ATKIS (Official Topographic Cartographic Information System in Germany), GK25 (geological map from NLfB - Lower Saxony Agency of Soil Research in Germany), and BK25 (soil map from NLfB) so as to adjust the shape and location of the objects. Safra and Doytsher (2006) reported another example for the quality control of the existing geospatial datasets, where the location accuracy in cadastral maps has been improved through data comparison.
- Data matching plays a fundamental role in multiple representation databases (MRDBs). An MRDB can be described as a spatial database, which is used to store the same real-world phenomenon at different levels of thematic and geometric detail (Hampe and Sester 2004). When creating a new or maintain an existing MRDB, data matching is necessary so as to define the linkages between different representations of the corresponding objects. For example, in the German ATKIS-model federal datasets for the resolutions 1:25 000, 1:250 000 and 1:1 000 000 exist independently. Updating these datasets requires extensive workload because every single dataset has to be manually adjusted. In order to update only the finest resolution, and then propagate the changes to all other resolutions, Mantel and Lipeck (2004) built up a MRDB architecture based on a framework for the computation of object matching from different federal datasets. Other approaches on how to define the linkages between various datasets in a MRDB can be found in Devogele et al. (1996), Sester et al. (1998), Dunkars (2003), Anders and Bobrich (2004), Tiedge et al. (2004) and Lüscher et al. (2007).
- More recently, data matching begins to play a key role in providing navigation solutions for many Location Based Services (LBS), which can be regarded as the activities to trace the actual positions of the users and provide the associated information on a mobile computing device (Jong et al. 2001; Pyo et al. 2001; Yang et al. 2005; Neuland and Kürner 2007; Wu et al. 2007; Lv et al. 2008). Ochieng et al. (2003) implemented a map-matching algorithm for Global Navigation Satellite Systems (GNSS). Being supported by the information about error sources associated with the positioning sensors, the historical trajectory of the vehicle, topologic information of the road network and the heading and speed information of the vehicle, the algorithm can precisely identify the road on which the vehicle is travelling. This algorithm has been elaborated by Quddus et al. (2006). On the basis of fuzzy logic, the elaborated algorithm provides a significant melioration over existing map-matching algorithms in terms of identifying correct links and estimating the vehicle position on the links, especially in high density areas where the average distance between neighbouring roads is less than 100 meters. Zhou (2005) proposed a general map-matching approach in the context of travel/activity studies. By transferring the attributes of road network to the travel route derived from recorded GPS points, this approach serves the purpose of inferring travel behaviour and conducting the corresponding analysis.
- Data matching also supports image processing for purposes of image registration and recognition (Stilla 1995; Wang 1998; Dowman 1998). Raw images obtained from remote sensors usually contain various kinds of distortions. A distorted image can be overlaid with an existing map in order to get geometrically rectified (Novak 1992; Xiong 2000; Pendyala 2002). On the other way round, the vector map can also be transformed back to a raster image after the data matching (Chen et al. 2003). Moreover, since image data only implicitly contain geospatial information, the first step is to make this information explicit, i.e. to extract the objects of interest from the images. For reasons of speed and cost, this step should be automated as much as possible using image analysis methods. In many cases existing vector data are used as prior knowledge to support object extraction (Butenuth et al. 2007; Löcherbach 1994; Baltsavias 2004; Zhang et al. 2006).

Due to its importance for various applications, data matching becomes an issue of growing interest since a decade or so and it is getting more and more complex with the increasing availability of diverse geospatial databases (Badard 1999; Devogle et al. 1996; Dunkars 2003; Meng and Töllner 2004; Raimond and Mustière 2008). The reported work so far has doubtlessly paved way to the development of comprehensive approaches for the data matching. Although the currently available methods have revealed satisfactory matching performances on some selected test areas, the problems of inadequate matching still exist in terms of matching accuracy, computing speed, degree of automation and generic nature, especially in areas where (a) the geometries of the map objects reveal unsystematic deviations from those of the underlying objects and there is no automatic mechanism to predict the extent of individual deviations; or (b) the topologic conditions are too complicated to allow a reliable identification of matching pairs. Besides, the lack of meaningful semantic attributes or the disparate object structure between different geospatial datasets also increases the difficulties to realize an automatic matching process with qualified computing results.

Being aware of the fact that a road network with geospatial objects representing real-world roads serves in many cases as the geometric and functional backbone of a comprehensive digital landscape model, the author is focused on the development of operational matching approach for road networks. Although correspondences between different road networks can be visually recognized, matching network counterparts remains a very difficult task. A fundamental problem is the reconciliation of disparate network representations when these networks are obtained at different times or come from different application domains. Matching different networks manually is possible, but such operations are tedious, inefficient and error-prone, and above all time-consuming (Xiong and Sperling 2004; Nystuen et al. 1997; Chen 2005). Therefore, this dissertation addresses an automatic matching algorithm for road networks, termed as *Delimited-Stroke-Oriented (DSO) algorithm*. Comparing with the existing well-known matching algorithms of BG (Buffer Growing) and ICP (Iterative Closest Point), the DSO algorithm has its strength in dealing with geometric information and topologic relationships in an extensive context; hence provide a high matching rate and certainty. Furthermore, the dissertation also outlines the advantages of the use of several assisting technologies like ‘pattern recognition’, ‘data generalization’, ‘spatial indexing’ etc. within the field of data matching and as a result proposes a *contextual matching approach based on the DSO algorithm.* This contextual matching approach shows a highly automatic, generic and efficient matching performance in different experiments to match the various road networks from ATKIS, Tele Atlas, NAVTEQ, OpenStreetMap, etc. Taking the advantage of its generality and high matching capability, this approach has been utilized for the real-world navigational data enrichment in the region of whole Germany. And it is being extended to a commercial matching software which is intended to deal with other large road networks, such as those from France, Spain, East Europe, etc. (Ltd. Corp. Untied Maps 2009).

1.2 Structure of the thesis

This thesis consists of seven chapters. Chapter 1 outlines the major applications of data matching in various fields of enrichment of existing datasets, maintenance of a MRDB, quality assurance of the geospatial data, navigation support in Location Based Services, image registration and recognition, etc. Then it elucidates the objectives and the associated research tasks of the dissertation - developing a highly automatic, generic and efficient matching approach for the correlation of corresponding road networks.

Chapter 2 gives an overview on methods of data matching and the state of the art. Section 2.1 reviews the origin and evolution of the ‘geospatial data matching’. Section 2.2 categorizes the existing matching strategies from different points of view and highlights a few significant matching criteria which have been widely utilized. Section 2.3 investigates and compares several well-known matching algorithms and brings forward the requirements of a new contextual matching approach based on the DSO (Delimited-Stroke-Oriented) algorithm. Section 2.4 discusses some significant terminologies in the environment of road-network matching.

Chapter 3 presents the DSO algorithm in details. This brand-new matching algorithm goes through five stages iteratively, incl.: (1) data pre-processing to reduce the noise of irrelevant details and eliminate topologic ambiguities in the datasets to be matched; (2) construction of the ‘graph’ to record the relationships between conjoint objects; (3) connection of the Delimited Strokes; (4) matching of the Delimited Strokes; and (5) dealing with fragmental matching areas. With the help of ‘graphs’, the conjoint road objects to a Delimited Stroke can be easily brought together. The corresponding network is then treated as an integral unit in the matching process, i.e. the DSO algorithm will lead to a network-based matching which allows the consideration of more topologic information in a larger context environment.

To go further, Chapter 4 proposes three assisting methodologies to benefit the DSO matching process. Section 4.1 presents a structure-guided matching strategy to deal with special matching cases concerned with looping crosses and dual carriageways. Section 4.2 is dedicated to a semantic-guided matching strategy to make the optimal use of the available semantic attributes, where the following two topics are deeply discussed: (a) how to calculate the semantic similarity; and (b) how to take advantage of the semantic similarity in different matching scenarios. Section 4.3 develops two grid-based spatial indexes to organize the point data and linear data respectively, which have been embedded into the matching approach.

In Chapter 5, the performance of the proposed matching approach is evaluated with respect to three measures - matching rate, matching correctness and computing speed. The conducted matching experiments involve four datasets, viz. ATKIS, Tele Atlas, NAVTEQ and OpenStreetMap, and cover various urban, rural and mountain areas which are distributed in different federal states of Germany. In addition, this chapter assesses the matching quality by classifying the matching results into various certainty levels and conducts a number of investigations on the geometric differences between the corresponding counterparts from different datasets.

As analyzed and illustrated in Chapter 6, the contextual matching approach based on the DSO algorithm has yielded a substantial capability for the real-world mega data enrichment. Up to date, this matching approach has been successfully implemented in three practical projects: (i) Postal data integration (funded by Federal Agency of Cartography and Geodesy, BKG); (ii) Integration of the routing-relevant information from different datasets (funded by BKG); and (iii) Conflation of the pedestrian ways between different datasets (funded by United Maps Corp.),

As the synthesis part of the dissertation, Chapter 7 summarizes the main achievements of this work and gives an outlook for the upcoming research tasks.

Chapter 2

The State of the Art and Methodological Background

Geo-spatial data matching has been a topic of intensive research since two decades and it is getting more and more complex with the increasing availability of diverse geospatial databases (Badard 1999; Devogle et al. 1998; Dunkars 2003; Ochieng et al. 2003; Meng and Töllner 2004; Sehgal et al. 2006; Olteanu et al. 2006; Mustière 2006; Moosavi and Alesheikh 2008). In a short review of the previous work, Section 2.1 illustrates the evolution of the geospatial data matching. Section 2.2 outlines different matching strategies and criteria which have been intensively reported in literature. Section 2.3 attends to the research emphasis of this dissertation - road-network matching: it firstly defines the various relationships and cardinalities between corresponding road objects; then brings forward the necessity of a contextual matching approach after the investigation and comparison of two current well-known matching algorithms for the road-networks matching, viz. Buffer Growing and Iterative Closest Point. Section 2.4 discusses several significant terminologies in the domain of road-network matching.

2.1 Evolution of data matching

Data matching is well known under the name Conflation, which originates from the Latin words *'con flare'* and means *'blow together'*. Although it was contemplated several years earlier, matching of different geospatial datasets firstly became a reality in the mid-1980s through a project initiated by the United States Geological Survey (USGS) and the Bureau of Census to consolidate the digital vector maps of both organizations (Rosen and Saalfeld 1985). The initial focus of data matching was to remove geometric inconsistency between heterogeneous overlapped geospatial datasets. From then on a lot of new ideas and technologies haven been promoted in this area.

Lupien and Moreland (1987) proposed a conflation approach made up of two stages, namely coverage alignment and feature matching. Coverage alignment is achieved by triangle-based Rubber-Sheeting. The Rubber-Sheeting makes use of user-defined links between matching locations in the source coverages. These links are used to construct a pair of distortion surfaces that are then used to transform features. Feature matching is based on distance measures involving points and arcs. The basic idea is that two points are matched if they lie within a specified tolerance of each other, while two arcs are matched if all points on one arc are within tolerance of the other.

Saalfeld (1988) presented an application case to match different digital datasets provided by the Bureau of Census and the United States Geological Survey (USGS). In this work, an iterative matching process has been developed, where feature matching is performed first according to the strongest set of criteria and progressing to the weakest. These matching pairs can be either interactively defined on a screen or automatically detected by comparing topologic and geometric characteristics. Based on the matched features, the remaining (unmatched) features are aligned via a Rubber-Sheet Transformation after iteration. The feature-matching process is then repeated. When this process for a given set of criteria finds no matches, the next weaker set of criteria is used, etc. The approach makes the assumption that both maps are nearly isomorphic and handles therefore only one-to-one matchings. One-to-one matchings are not sufficient for datasets which differ in topology.

Deretsky and Rodny (1993) described a method for combining geometric and attribute data from two digital map sources into one ultimate map. It requires the identification of the matching elements from different road maps. The presented matching process relies on the chains of arcs to determine the corresponding intersections between them. The chains are formed a priori by using both attributes and geometric properties of the arcs. The intersections are treated as relations between chains and are stored in an external relational database. Thus, the matching process becomes a set of the standard operations on RDBMS (Relational Database Management System) as well as the rejection of the possible ambiguous identifications according to some configurable rules with respect to attributes proximity, geometric similarity and topologic consistency.

Gabay and Doytsher (1994) demonstrated a two-stage procedure to match maps which are slightly different in terms of geometric properties but may considerably differ from each other concerning the topologic properties. In this matching procedure, arcs that have matched end nodes are matched first. The remaining arcs are then further evaluated. The decision whether they can be matched together is dependent on geometric and topologic characteristics of the data. This procedure can not only identify common elements between maps, but also reveal unique elements that appear in only one of the maps. This capability allows geometric inconsistency and differences of topologic characteristics to be recognized and handled during the map-matching process. Under the assumption that one dataset is captured with a higher quality, Gabay and Doytsher (1995) presented an automatic approach to improve the geometry of a dataset with a lower quality. This approach is able to handle many-to-many matchings.

Brown et al. (1995) discussed the importance of *m:n* matching and then described a conflation system in a GIS environment. This system explores existing GIS functions such as Rubber-Sheeting and dynamic segmentation to adjust network geometry and establish node and arc correspondence between matched networks. With the identified correspondences, linear mappings along network edges are possible, and network attributes, such as direction flags, can be transferred from one network to the other.

Walter (1997) presented a geometric matching strategy with the purpose of mutually exchanging attributes between vehicle navigation data and German topographic map data. These two datasets are created for different purposes and have distinct data schemas. To achieve the correct matching result, he combined different methods such as *Buffer Growing*, geometric comparison of angles, lengths and shapes and optimized them in his work. Walter and Fritsch (1999) conducted an Affine Transformation before the matching process, where the control points were assigned interactively. As the global errors are probably eliminated by the affine transformation, a smaller buffer (searching scope) could be used to find the potential matching pairs, which increases the matching accuracy and reduces the computing time. For each potential matching pair, many different criteria are implemented to check whether it is a proper match. In order to determine proper values for the involved parameters, a statistical study on discrepancies between different datasets has been conducted according to the existing previously matched data. Volz (2006) extended this work by applying an *ICP* (*Iterative Closest Point*) approach. By identifying seeds which show a high likelihood of correspondence, the matching process is initialized. A combined edge and node matching algorithm is then used to detect *1:1* correspondences. In case no *1:1* match could be found, an enhanced edge matching approach is triggered to recognize *1:2* matches. The whole process runs in multiple iterations with stepwise relaxed constraints.

Cobb et al. (1998) performed a complete matching approach for the conflation of attributed vector data such as the Vector Product Format (VPF) datasets produced and disseminated by the National Imagery and Mapping Agency (NIMA). While some precious work in the field of data matching has used statistical techniques based on proximity of features, the matching approach presented here utilizes both spatial and non-spatial information associated with data, including attribute information such as feature codes from standardized set, associated data quality information of varying levels, and topology, as well as more traditional measures of geometry and proximity. Based on semantic similarities of attribute values and shape similarity of linear features, Cobb et al. developed a hierarchical rule-based approach for feature matching, which considers the strongest criteria first and progresses to the weakest criteria to improve the likelihood of a match. Similarly, members of a candidate set must be considered iteratively to increase the likelihood of finding correctly matched features. An implementation based on an expert system and the OVPF (Object Vector Product Format) prototype and utilizing NIMA's VPF data has been performed which demonstrates the effectiveness and practicality of the matching method.

Pendyala (2002) combined the bottom-up and top-down computations. The bottom-up computation starts with node matching, proceeds to segment matching, and ends up with edge matching. The top-down procedure works in the reverse direction. At the initial stage of the top-down computation, potential matching pairs are first hypothesized using screening criteria such as distance and angle difference. Then segment matching proceeds. Through segment matching, matching measures for those hypothesized matches are computed, like edges are confirmed, and unlike edges are rejected. The main advantage of this approach is that the bottom-up computation leads to the identification of matches where node matches can be quickly established, while the top-down computation tends to find matches where node matches fail or network structures differ. Another advantage is reflected in its overall computational procedure that consists of node matching, segment matching and edge matching. Node matching establishes node correspondences between two networks using Euclidean distance and angel patterns formed by incident edges. Segment matching tracks segment pairs along potential matching edges. The result of segment matching can be used to locate node locations on the edges of the counterpart network when no node correspondences are established. In precious research, node matching and edge matching have been frequently used, but not segment matching. Segment matching is important because it can be used to evaluate correspondences between each pair of segments on potentially matched edges. Through segment matching, matching measures between each pair of segments are first computed; then, overall matching measures at the edge level are obtained. Due to these detailed considerations, edge-matching measures derived from segment matching can be more sensitive in recognizing differences and similarities among different edge pairs. Using these measures, there is a better chance to find the best matches.

A semi-automated method for network matching was described in Xiong and Sperling (2004) with the purpose to match linear road features extracted from aerial photographs and existing street network in vector format. This method involves an automated matching algorithm and an interactive procedure to form an effective and reliable network matching solution. The automated algorithm combines node matching, segment matching and edge matching as well (similar to Pendyala 2002) and incorporates a cluster-based matching mechanism. The automated matching is then capable of generating reliable matching measures, resolving difficult matches, and in both cases, correctly identifying the overwhelming majority of the matching counterparts between different networks. The interactive procedure allows a human operator to visually check and correct the results generated by the automated algorithm. This reduces the incidence of errors in the subsequent matching and provides improved performance and reliability that may not be achieved with the automated algorithm alone.

Sheeren et al. (2004) pointed out that the ability to analyse and understand the differences among various datasets is one of the most important issues for data conflation. Therefore, they proposed a process to interpret the differences between multiple representations and embedded machine learning technologies in an expert-system.

Samal et al. (2004) introduced a three-step strategy to match features from multiple sources. An initial matching step compares the common attributes of the features. The similarity between features from different sources is measured based on these individual attributes regardless of their relationships with other features. This context-independent similarity is computed as a weighted average of the similarities between the attribute values of the features. However, if any one of the attribute values differs significantly, the two features can not be matched, irrespective of the similarity between other attribute values. In the second step, the features are examined in relation to their neighbouring features, i.e. the geographic context, which is central to any geographic information. Since the context-dependent process can become computationally very expensive, only the more significant features - the landmarks, are used instead of all the neighbouring features in the context. In the final step, both the context-independent and context-dependent similarities are integrated into one similarity score. The matching procedure then groups features into sets based on their similarity. Each similarity set includes at most one feature from each source and represents one physical entity.

JCS Conflation Suite (JUMP 2004) is an open source Java library containing a set of interactive tools to perform conflation on spatial datasets (Vivid Solutions 2005). JCS provides an algorithm to match different versions of the same road network. The matching algorithm starts from a node matching. Based on the distance between the nodes and the angular difference between the edges, the corresponding nodes from different datasets are computed. The matching is then continued with focus on the edges. By stepping through the matched edges, the algorithm is able to find differences between segment lengths in the two datasets and to split the segments where necessary in order to create more similar geometries in the two datasets. In this way the new segments can be matched more easily. Where the automatic algorithm is unable to determine a match, manual tools are provided to create new matches as well as modify or delete wrong matches. The matching result can be output for further processing: e.g. attributes can be transferred between matched road sections, and missing road sections can be added from one road network to another. The JCS is designed to provide easy access to conflation algorithms both interactively and programmatically, which operates independently of commercial software offerings. The JCS does not support the native formats, but through transformation software available through GIS vendors or third parties, it will be realistic to use the JCS tools with typical geospatial data. In particular, JCS will make use of a constrained form of GML (Geography Markup Language) for all input and output. JCS uses the JUMP Workbench and API to provide visualization, support for interactive workflow, and spatial data processing, and uses the JTS Topology Suite to provide basic geometric functionality. Based upon the JCS, Stigmar (2005) developed a matching approach to correlate together navigational data and topological, which has been integrated into a system of real-time map services. To enable a more complete match, Stigmar suggested a topologic search to find potential matching following the JCS approach.

Mustière (2006) conducted several experiments with the automated matching of two datasets at different scales. He suggested the simultaneous use of semantic, geometric and topologic information. This work attempts to answer the questions how data from IGN-France databases can be integrated, how far this integration can be automated, and how the matching results can be interactively assessed and modified. Different networks that are formed by roads, electric power lines, hydrographical features, railways and hiking routes were used as test datasets.

More recently, Mustière and Devogele (2008) developed a process for matching networks at different levels of details. This process is composed of some steps for a rough node matching, some steps for a rough arc matching, and then some final steps that combine the precious matching results as the final decisions. The strategy proves efficient for various themes. However, it has only been applied to networks from the same producer with limited heterogeneity. An important insight from experiments of this process is that the non-spatial properties, such as name, type of width of roads, also need to be taken into account, to get the best results for different networks like railways, roads, rivers or electric lines. Another insight is that the efficiency of the process depends on its ability to manage imperfection in the data. Indeed, attribute values may be imprecise, erroneous or even missing, and these imperfections make the matching process difficult. In order to overcome these limitations, a data matching approach based on the belief theory (Dempster 1968; Olteanu 2007) was proposed by Raimond and Mustière (2008). This approach can efficiently model imperfect knowledge according to (Shafer 1976). In addition, it provides techniques to efficiently combine sources of knowledge for decision making and is able to handle conflicts between knowledge sources. The matching approach based on the *belief theory* has been proved to be efficient on heterogeneous geographic networks with different scales. Raimond and Mustière believe that the explicit representation of imperfection, like imprecision, uncertainty and incompleteness or ignorance is very promising for studying geographic data.

2.2 Categorization of data matching

2.2.1 Horizontal, vertical and internal matching

Based on the relations of the datasets to be processed, the matching problems can be classified into three groups: horizontal, vertical and internal (Blasby et al. 2003).

- Horizontal Matching

Horizontal matching is referred to the data matching with the objective to eliminate discrepancies in terms of spatial feature positions and attributes which exist in the common areas of two neighbouring datasets (Yuan and Tao 1999). Typically horizontal matching can be utilized to align common boundaries between two different adjacent networks.

- Vertical Matching

Vertical matching is intended to identify corresponding counterparts from different datasets that cover same regions in space, where the input datasets consist of different versions of the same features. After the process of vertical matching, attributes can be transferred between matched features, and unmatched features may be transferred in their entirety (Blasby et al. 2003). This dissertation is devoted to vertical matching.

- Internal Matching

An internal matching problem involves conflating or resolving features within a single dataset. For example, in coverage cleaning we may need to fix gaps or remove overlaps in coverages (Blasby et al. 2003).

2.2.2 Manual matching vs. automatic matching

Before the end of 1980s, many matching tasks were carried out in a manual process, with a human operator identifying corresponding counterparts distributed in different datasets and thus manually correcting geometry or attribute. Although some of the matching processes were operated with the assistance of computer tools, scripts and macros, the overall process would be generally characterized as 'computer-assisted' manual matching (Blasby et al. 2003). Such 'computer-assisted' manual approaches are tedious, inefficient, time-consuming and error-prone (Nystuen et al. 1997).

In the past two decades, the manual matching approaches have been progressively evolved to a more automated data matching paradigm which requires minimum human judgment and intervention within an essentially automated process. Obviously the ultimate goal would be the entire automation of the matching process. However, this might not be possible for the two reasons as follows:

Algorithmic limitations: many matching methods reported in literature hitherto have revealed high matching rate and efficiency on certain data types of selected test areas (Walter 1997; Meng and Töllner 2004; Ochieng et al. 2003). However, these methods were mostly developed as specialized solutions with limited reusability. Their matching performance drops dramatically as soon as the data type or the test area changes, i.e. there is not yet a single algorithm which is able to correctly identify all the matches (Blasby et al. 2003; Xiong and Sperling 2004).

Data ambiguity: the homologous objects from different datasets are not exactly identical in geometries (e.g. shape and location) or topologies. On certain occasions, it may not even be possible for a human expert to determine a correct course of action (Blasby et al. 2003; Mustière 2006).

The algorithmic limitations may be compensated by human interventions at certain stages of the data matching approach. For instance, with help of some interactive tools the human operator is able to create new matches for the unmatched features as well as remove the wrongly identified matches by the automatic routine (Zhang and Meng 2007). In this way, the matching completeness as well as accuracy can be increased. Ideally the human interaction can be carried out in conjunction with automatic matching in an iterative process. In this situation, the effectiveness of human assistance is directly proportional to the quality and richness of the user interface which is characterized by factors such as execution speed, ease-of-use, variety of feedback, visualization ability, and the presence of tools such as transaction recorders, etc. For practical uses, the design of an effective user interface is as challenging as the development of automatic algorithms. To reduce the data ambiguity, additional attributes need to be involved in the matching computation. As a matter of fact, a well-developed 'human-assisted' automatic matching approach can greatly improve the productivity and reliability of the real-world data matching implementations (Blasby et al. 2003).

2.2.3 Strategies of geometric, topologic and semantic data matching

The development of matching strategies is inherently dependent on the properties of the datasets to be matched. Three different properties of the objects in the datasets can be used in order to find correspondences (Stigmar 2006). As introduced in previous sections, many approaches have been developed during the past two decades to solve different matching problems in which different matching criteria were used. The predominant criterion used to match corresponding features can be categorized into geometric, topologic and semantic matching respectively.

2.2.3.1 Geometric matching

Geometric matching is based on the detection of corresponding objects in different datasets by comparing their geometric characteristics (Cecconi 2003).

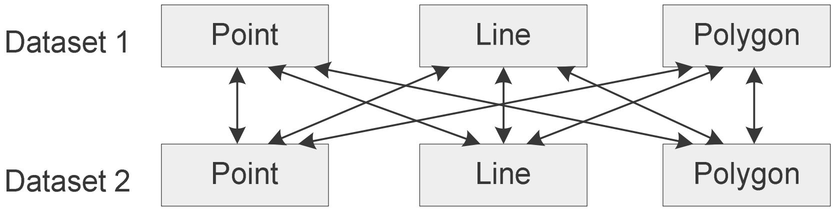


Figure 2.1 Spatial data matching types (Yuan and Tao 1999)

In 2D geospatial databases, there are three types of features - points, lines and polygons which are handled by different matching operations (see Figure 2.1). For example, for point-to-point matching, Euclidean distance is suitable, but for line-to-line matching, Hausdorff distance is better (Yuan and Tao 1999). To realize such a geometric matching, a wide variety of geometric criteria can be used. It has been generally assumed that geometric matching requires the two datasets share similarity in distance, location, size, orientation, shape, etc.

- **Distance**

A real-world object should have similar location in different maps. Comparing the distance between two objects from two different datasets is the most straightforward way to match homologous spatial objects: an object within a certain range of another object has a large probability to represent the same real-world entity (Yuan and Tao 1999). There are three popular distance measures: Euclidean distance, Hausdorff distance and Frechét distance.

Euclidean distance is used to calculate the distance from point to point, point to line segment, etc. For instance, the Euclidean distance from point $p_i = (x_i, y_i)$ to point $p_j = (x_j, y_j)$:

$$d_e(p_i, p_j) = [(p_i - p_j)\cdot(p_i - p_j)^{\mathrm{T}}]^{1/2}; \qquad \text{...[2-1]}$$

To measure the distance between a point and a line segment, the distance introduced by White et al. (1985) and Marchal et al. (2004) can be used. Let P_0' be the projection of P_0 on the line segment $l(p_i, p_j)$ $(i \neq j)$ where p_i and p_j are the end points of l, the distance between P_0 and $l(p_i, p_j)$ is defined as Equation [2-2], where d_e denotes the Euclidean distance illustrated in Equation [2-1].

$$d_{PtoL}(P_0, l(p_i, p_j)) = \begin{cases} d_e(P_0, P_0') & \text{if } P_0' \in l(p_i, p_j) \\ \min\{d_e(P_0, p_i), \quad d_e(P_0, p_j)\} & \text{elsewhere} \end{cases} \qquad \text{...[2-2]}$$

Hausdorff distance indicates the maximum distance from any point of one polyline to another polyline (Hangouët 1995; Rucklidge 1996; Min et al. 2007). It is often used to calculate distance between two polylines to search for line-to-line matches (Yuan and Tao 1999; Volz and Walter 2004; Deng et al. 2005; Kampshoff 2005; Mustière and Devogele 2008). Let $L1 =< p_{1,1} p_{1,2} \cdots p_{1,m-1} p_{1,m} >$ and $L2 =< p_{2,1} p_{2,2} \cdots p_{2,m-1} p_{2,m} >$ represent two polylines that are compared, the Hausdoff distance between $L1$ and $L2$ can be calculated by D_H defined by Equation [2-3].

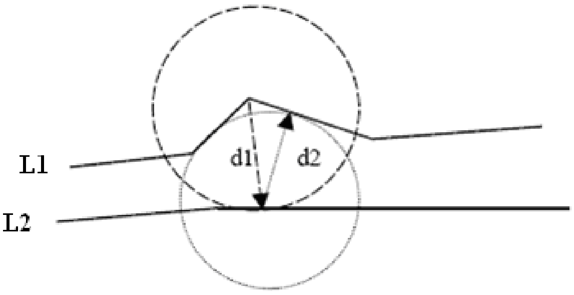


Figure 2.2 Hausdorff distance between $L1$ and $L2$ (Yuan and Tao 1999)

$$D_H = \max(d1, d2) \qquad \text{...[2-3]}$$

Where, $d1 = \max\lfloor \min[d_{PtoL}(p_{1,i}, l(p_{2,j}, p_{2,j+1})] \rfloor$ ($p_{1,i} \in L1$, $p_{2,j}, p_{2,j+1} \in L2$);

$d2 = \max\lfloor \min[d_{PtoL}(p_{2,j}, l(p_{1,i}, p_{1,i+1})] \rfloor$ ($p_{2,j} \in L2$, $p_{1,i}, p_{1,i+1} \in L1$);

$d_{PtoL}(p, l)$ represents the distance from point p to line segment l (ref. Equation [2-2]).

In this equation, $d1$ denotes the largest minimum distance from $L1$ to $L2$; and $d2$ denotes the largest minimum distance from $L2$ to $L1$, where $d1$ and $d2$ are obtained by moving a dynamic circle along one line so that it always tangent to the other (Yuan and Tao 1999), see an example in Figure 2.2.

Frechét distance is a measure of similarity between two oriented curves. It can be intuitively defined as follows: *"A man is walking a dog on a leash: the man moves on one curve, the dog on the other; both may vary their speed, but backtracking is not allowed. The Frechét distance refers to the length of the shortest leash that is sufficient for traversing both curves. Given two curves that are equivalent to two continuous functions: $f[a,a'] \rightarrow V$ and $g[b,b'] \rightarrow V$, where $a, a', b, b' \in \Re, a < a', b < b'$ and (V, d) is a metric space, their Frechét distance d_F is defined as:"* (Eiter and Mannila 1994)

$$d_F(f, g) = \inf_{\substack{\alpha[0.1] \rightarrow [a,a'] \\ \beta[0.1] \rightarrow [b,b']}} \max_{t \in [0,1]} d(f(\alpha(t)), g(\beta(t))) \qquad \text{...[2-4]}$$

The Frechét distance takes into account the location and ordering of the points along the curves and therefore it is often more suitable than the Hausdorff distance. Measures based on Frechét distance are for example particularly pertinent to compare very sinuous linear or polygonal curves such as coastlines (Eiter and Mannila 1994; Mascret et al. 2006; Mustière and Devogele 2008).

- **Orientation**

If several matches are found by distance criteria, angular criteria can be used to determine which one would be most probably the correct match (Yuan and Tao 1999). In Raimond and Mustière (2008), for example, the orientation criterion are used to compare angles or directions between two arcs denoted as *Arc1* and *Arc2*. It measures the differences between the orientations of tangents to *Arc1* and *Arc2*, respectively at the point of *Arc1* nearest to *Arc2*, and at the point of *Arc2* nearest to *Arc1*. If the angle between two arcs is about *0* radians, arcs are nearly parallel and have the same orientation; if the value of the angle is close to π, arcs are parallel but have opposite directions; if the angle approximates $\pi/2$, then arcs are perpendicular. The orientation of polygonal objects, however, can be represented by the angle of its longest and shortest chord.

- **Location**

Based on relative locations, it is possible to judge whether a point is inside or outside of a polygon, whether a polygon is entirely or partially overlapped with another polygon etc. Between two polygons *A* and *B*, *symmetric difference*, also called *template metric*, can be measured. It is defined as the areas found in one polygon only, that is $((A-B)U(B-A))$. Translating convex polygons so that their centroids coincide also gives an approximate solution for the symmetric difference (Veltkamp 2001). The more the two polygons overlap, the lower the symmetric difference, and the higher the matching probability. Sometimes the location relationship can be the main criterion to find point-to-polygon or polygon-to-polygon matches (Yuan and Tao 1999).

- **Shape**

Lines or polygons can differ from each other in their shapes. As a well-known shape descriptor for open or closed polylines, the *cumulative angle function* $\Theta_A(s)$, also denoted as *turning function*, gives the angle between the counter-clockwise tangent and the x-axis as a function of arc lengths. $\Theta_A(s)$ keeps track of the turning that takes place, increasing with left hand turns, and decreasing with right hand turns, see Figure 2.3.

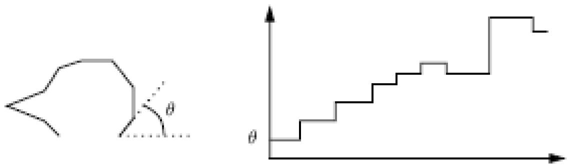


Figure 2.3 A polyline and its turning function (Veltkamp 2001)

This function is invariant under translation of the polyline. Rotating a polyline over an angle results in a vertical shift of the function with an amount θ. The turning function is a piecewise constant function, increasing or decreasing at the vertices, and constant between two consecutive vertices (Veltkamp 2001). Without losing any generality, each polygon or polyline could be rescaled so that the total perimeter length is 1, i.e. $\Theta_A(s)$ becomes a function from [0, 1] to $\Re$ (Arkin et al. 1991). Since the turning function can reflect the shape character, it can be utilized for matching purposes. E.g. Arkin et al. (1991) employs the turning function to match polygons. Consider two polygons *A* and *B* and their associated turning functions $\Theta_A(s)$ and $\Theta_B(s)$. The degree to which *A* and *B* are similar can be indicated by the distance between the turning functions $\Theta_A(s)$ and $\Theta_B(s)$ according to the L_P metrics defined by Equation [2-5].

$$L_P(A,B)=\left\|\Theta_A-\Theta_B\right\|_P=(\int\left|\Theta_A-\Theta_B\right|^P\cdot ds)^{\frac{1}{P}} \qquad \ldots[2\text{-}5]$$

The larger the L_P is, the less the similar shape of the polygons to be matched. In practice, the parameter *P* is often assigned to the value ‘1’, thus $L_P(A,B)$ can represent the area enclosed by the curve $\Theta_A(s)$ and $\Theta_B(s)$, see the shadow part of Figure 2.4.

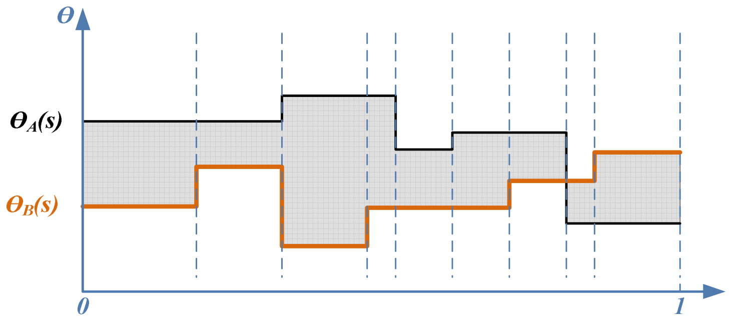


Figure 2.4 The rectangular strips formed by the functions $\Theta_A(s)$ and $\Theta_B(s)$ (Arkin et al. 1991)

- **Size**

A polyline has a length and distance between its two endpoints; a polygon has perimeter, area, length of chords etc. Often, these various size descriptors are utilized as different criteria during the process of matching polygon to polygon or line to line.

2.2.3.2 Topologic matching

Topologic matching uses composition or topologic relationships between different objects to match a given object. If two relationships correspond, then this correspondence can be used to find homologous objects linked by this relationship (Devogele 2002). The concept of topology referred to the branch of mathematics dealing with proprieties which do not change under geometric transformations (Cichociski 2008). Taking advantage of these invariant properties, the United States Census Bureau applied the topology to maps to reduce errors in tabulating large amount of census data (Theobald 2001).

In geoinformatics, topology is generally defined as the spatial relationships between adjacent or neighbouring objects. A two-dimensional plane typically includes topologic relationships such as connectivity between edges, adjacency between polygons and composition relationships such as outlets of a node, edges that form a polygon, and so on. In context of line network matching, for example, emphasis is usually laid on the topologic organisations of the edges and nodes: nodes may be roughly characterized by valence, i.e. the number of edges connected to them (Walter and Fritsch 1999; Zhang et al. 2005), or more accurately by *spider function* suggested by Saalfeld (1988).

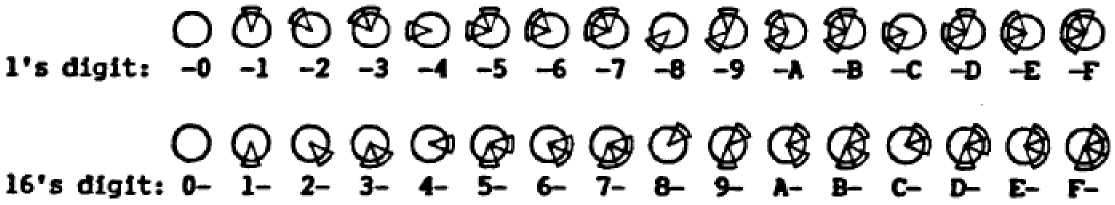
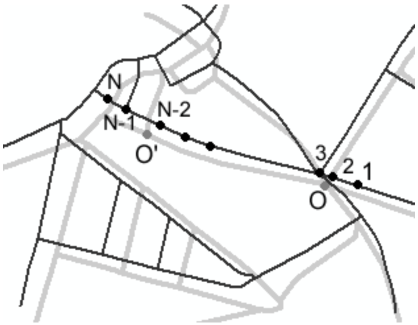

1's digit:
-0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -A -B -C -D -E -F
16's digit:
0- 1- 2- 3- 4- 5- 6- 7- 8- 9- A- B- C- D- E- F-

Figure 2.5 Hexadecimal and sector patterns for *spider function* (Saalfeld 1988)

“The representation of the value of the spider function as a hexadecimal integer has additional desirable properties. It is a two-digit number and each digit describes the street directional behaviour in a four-sector band constituting a semi-circular region. A digit in the second position describes the same configuration as the same digit would in the first position except for a rotation of 180°” (Saalfeld 1988). According to the hexadecimal and sector patterns for *spider function* (see Figure 2.5), the hexadecimal number 55 can represent the north-west-south-east intersections.

It has been common sense that an accurate match can be hardly reached if only the geometric characteristics are compared (Zhang et al. 2005). For example, in Figure 2.6, in terms of shape and location, the polyline *1→N-2* is identified as the perfect match for the polyline *O→O'*. In reality, however, the correct match should be the polyline *3→N-1* if topologic relationship is considered.



grey: dataset 1 black: dataset 2

Figure 2.6 An example showing the limitation of geometric matching strategy

The topologic characteristics of the nodes can be implemented as the key information guiding the matching process for line networks. For example, nodes could be matched first, subsequently the edges connected to these nodes are matched together (Xiong 2000; Xiong and Sperling 2004; Volz 2004; Safra et al. 2006). In this sense, topologic matching is a higher level matching strategy than geometric matching (Mustière and Devogele 2008). Often, the topologic matching is used to reduce the search scope for the geometric matching, it is seldom used alone. Sometimes it can be used as a main criterion, based on some known matched objects, or combined with geometric matching processes. The topologic matching triggers the analysis of neighbouring or connected objects, thus has the potentiality to spread the discrete matching to a larger context (Cobb et al. 1998; Yuan and Tao 1999; Stigmar 2006).

2.2.3.3 Semantic matching

Semantic matching puts objects in correspondence according to their semantic similarity which characterizes the proximity degree of the semantic attributes between two objects (or objects clusters) from different datasets (Devogele 2002; Cohen 2000). The semantic matching can be used to find corresponding objects from different datasets that share some common or comparable attributes (Yuan and Tao 1999). The simplest case is that two datasets have the same attributes whose meanings or value ranges are defined in the same way. However, the semantic similarity can be also identified even when the objects from various datasets have significant representational differences (Sheth 1991). According to Kashyap and Sheth (1996), the semantic proximity of two attributes from different databases can be classified into four levels:

- *Level 1 - Semantic Equivalence:* Two attributes are defined to be semantically equivalent when they represent the same real-world entity or concept, i.e. a total 1-1 value mapping between the domains of these two attributes in any known and coherent context.
- *Level 2 - Semantic relationship:* Two attributes are said to be semantically related when there exists a partial value mapping, a generalization, or aggregation abstraction between the domains of the two attributes.
- *Level 3 - Semantic relevance:* Two attributes are considered to be semantically relevant if they can be related to each other using some abstraction in some context. Thus the notion of semantic relevance between two attributes is context-dependent, i.e., two attributes may be semantically relevant in one context, but not so in another.
- *Level 4 - Semantic resemblance:* whenever two attributes can not be related to each other by any abstraction in any context, but they are associated with contexts in which they have the

> same role and their definition contexts are coherent with respect to each other, they can be said to semantically resemble to each other.

The attributes from different datasets can be semantically related in one of the above four levels. Obviously, the best case in matching processes is that two datasets have semantically equivalent attributes (level 1) and these fields are fully filled by values. In this case the correspondence is clear and thereby the semantic matching can be performed very efficiently by comparing specific attributes so that the homologous objects between different datasets can be identified. Moreover, the attributes which are related in level 2 and level 3 between different datasets can be also utilized in the matching process, acting as a *‘filter’* to eliminate all object types that should not be considered as potential matches by comparing specific attribute values. To do this, the datasets need to be analyzed beforehand in order to identify the semantically proximate object types distributed in different databases (Stigmar 2006; Gösseln and Sester 2003). Little research, however, has been done in this aspect so far, especially in the context of line network matching.

Worthwhile to mention is that the semantic matching is not a generic method since in practice one or both datasets to be matched seldom bear sufficiently semantic information beyond the class definition and design specifications (Walter and Fritch 1999; Zhang et al. 2005; Volz 2006). For this reason, the key components of a generic matching model are usually constituted by geometric and topologic matching, while the semantic matching often acts as an accessorial tool to enhance the matching performance.

2.3 Concerns of road-network matching

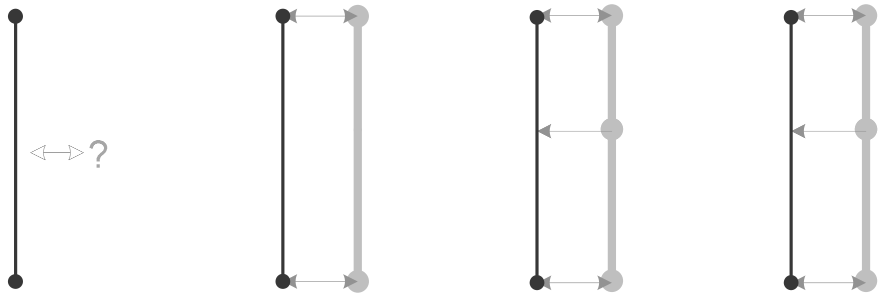
A road network represents a set of real-world roads, using nodes and polylines. A node, also called topologic node, is either an intersection where two or more roads meet or a dead end where the road terminates without intersecting another road. A road object is usually represented by a polyline, which is a continuous line composed of one or more line segments, such that every two consecutive segments only intersect in their common endpoint while non consecutive segments do not intersect (Safra et al. 2006). A polyline will be termed as ‘polygon’ if its two endpoints have the same geometric position.

In a road database, several objects may represent different parts of the same real-world road, e.g., each lane in a highway could be represented by a different object, and vice versa an object may also represent more than one real-world road. A road object has associated spatial and non-spatial attributes. Spatial attributes describe the location, length, shape and topology of a road while the non-spatial attributes consist of identifier (ID), road name, direction of the traffic flow, number of lane, functional road class, speed limit, etc. (Safra et al. 2006). As a road network serves in many cases as the geometric and functional backbone of a comprehensive digital landscape model, this dissertation is dedicated to road-network matching.

2.3.1 Relationships between the corresponding road objects

Specific conflicts may appear between diverse geospatial datasets. The representation specifications of the objects in different databases can differ from one another, especially when the datasets were captured by different organizations for different applications and/or at different LoDs (level of details). For instance, (a) an object can be represented in one dataset but be absent in the other; or (b) an object in the first dataset can correspond to a group of objects in the second. *“These conflicts must be identified during the declaration of correspondences so that the matching techniques can be extended to solve them”* (Sheeren et al. 2004). Therefore it is important to understand all possible relationships between different corresponding objects or object groups before starting the matching computation.

In the context of road-network matching, the relationships between the corresponding counterparts from different datasets are usually classified into four groups according to literature reported so far which are one-to-null, one-to-one, one-to-many, and many-to-many correspondences respectively (Walter 1997; Parent and Spaccapietra 2000; Zhang et al. 2005; Hu et al. 2008). Taking a matching example between two road networks NW_A and $NW_{B,}$ where NW_A consists of the road objects {a_1, a_2, ... , a_I} and NW_B is {b_1, b_2, ..., b_J}, these four corresponding relationships can be described as follows (see Figure 2.7):



(a) one-to-null (b) one-to-one (c) one-to-many (d) many-to many

black lines: road objects from dataset *A*; grey lines: road objects from dataset *B*

Figure 2.7 Cardinality of the *m:n (m≥0, n≥0, m·n≠0)* matching pairs

One-to-null correspondence, viz. *1:0* or *0:1* relationship (non-correspondence): some objects may appear in one database but have no homologous counterpart in the other. These cases may originate from different levels of details, errors between updates, differences between specifications, and diverse emphasises or purposes for dataset usability (Mustière and Devogele 2008).

One-to-one correspondence, viz. *1:1* relationship: one object from NW_A is corresponding to one single object of NW_B, and vice versa. Mathematically the *1:1* relationship can be defined as:

$$a_i \Leftrightarrow b_j\,,\ (a_i \in NW_A, b_j \in NW_B) \qquad \ldots[2\text{-}6]$$

One-to-many correspondence, viz. *1:N* or *M:1 (M>1, N>1)* relationship: one road object of NW_A is corresponding to two or more objects of NW_B. These *(1:N, N>1)* cases may occur if NW_A is much less detailed than NW_B; however, the reverse cases, i.e. the *M:1 (M>1)* relationship, may also appear due to the fact that different segmentation rules are adopted in various datasets (Mustière and Devogele 2008). The *1:N* and *M:1 (M>1, N>1)* correspondences between the road networks NW_A and NW_B can be represented by the Expression [2-7] and [2-8] respectively:

$$a_i \Leftrightarrow \{b_{j(1)}, b_{j(2)}, ..., b_{j(N)}\}\,,\ (a_i \in NW_A, b_{j(\cdot)} \in NW_B) \qquad \ldots[2\text{-}7]$$

$$\{a_{i(1)}, a_{i(2)}, ..., a_{i(M)}\} \Leftrightarrow b_i\,,\quad (a_{i(\cdot)} \in NW_A, b_j \in NW_B) \qquad \ldots[2\text{-}8]$$

Many-to-many correspondence, viz. *M:N (M>1, N>1)* relationship: two road object chains in NW_A and NW_B correspond to each other, see the mathematical representations in [2-9]. Comparing to *1:1*, *1:N* and *M:1 (M>1, N>1)* relationships, the *M:N* correspondences may occur regardless of LoDs of the datasets.

$$\{a_{i(1)}, a_{i(2)}, ..., a_{i(M)}\} \Leftrightarrow \{b_{j(1)}, b_{j(2)}, ..., b_{j(N)}\}\,,\ (a_{i(\cdot)} \in NW_A,\ b_{j(\cdot)} \in NW_B) \qquad \ldots[2\text{-}9]$$

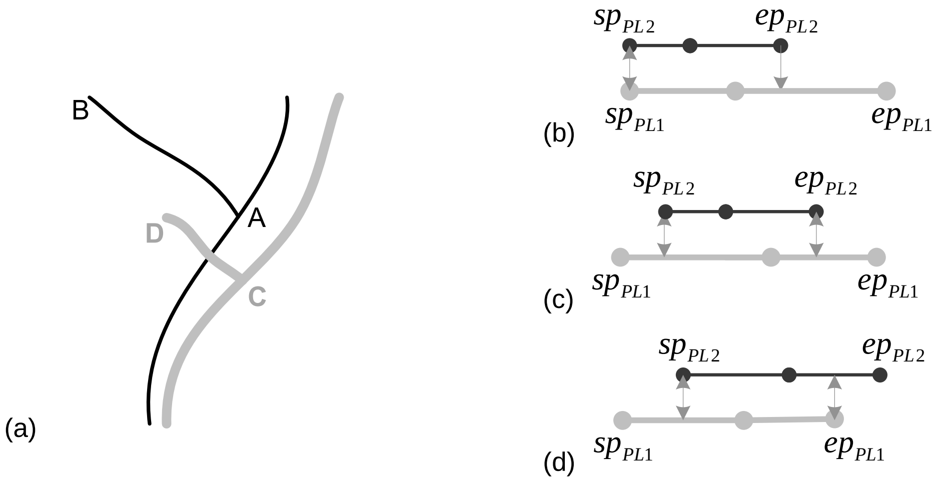
The one-to-null, one-to-one, one-to-many and many-to-many correspondences can be generally concluded as *m:n (m≥0, n≥0, m·n≠0)* cardinality. Although the *m:n* cardinality is often regarded as the most generic relationships that the corresponding counterparts from different datasets can have, it can not cover all of the matching cases that occurred in real-world applications (Mustière 2006). In order to deal with more generic matching cases, two external corresponding relationships are defined in this dissertation: one is partial correspondence, the other is equivalent correspondence.

- **Partial correspondence**

As illustrated by Figure 2.8-a, the polyline *A→B* is chained by the road objects from the dataset *A* and the polyline *C→D* is constituted by road objects from dataset *B*. The *A→B* and *C→D* can not be identified as *m:n (m≥1, n≥1)* matching pair because they do not completely correspond to each other. It is not correct either, however, if the relationship between *A→B* and *C→D* is identified as non-correspondence at all. To solve this conflict, the cardinality of partial correspondence between two object chains has to be investigated.

Consider two polylines (viz. object chains) denoted as PL_1 and PL_2, each from a different dataset; Let sp_{PL1} and ep_{PL1} represent the endpoints of PL_1; sp_{PL2} and ep_{PL2} be the endpoints of PL_2, then three possible relationships of the partial correspondences can be identified:

- *Extended correspondence:* PL_1 will be considered as an 'extension' to PL_2 when (i) PL_1 and PL_2 have a pair of corresponding endpoints, and (ii) the other endpoint of PL_2 has the projection at an intermediate point in PL_1. In Figure 2.8-b, sp_{PL1}, sp_{PL2} are corresponding nodes and the projection of ep_{PL2} is an intermediate point in PL_1.
- *Contained correspondence:* PL_1 contains PL_2 when the projections of both endpoints of PL_2 are intermediate point of PL_1, see Figure 2.8-c.
- *Lapped correspondence:* PL_1 and PL_2 are 'lapped corresponding pair' if each of these two polylines has the projection at an intermediate point in the other, see Figure 2.8-d, where the projection of sp_{PL1} is an intermediate point of PL_2 and on the contrary side the projection of ep_{PL2} is located on PL_1.



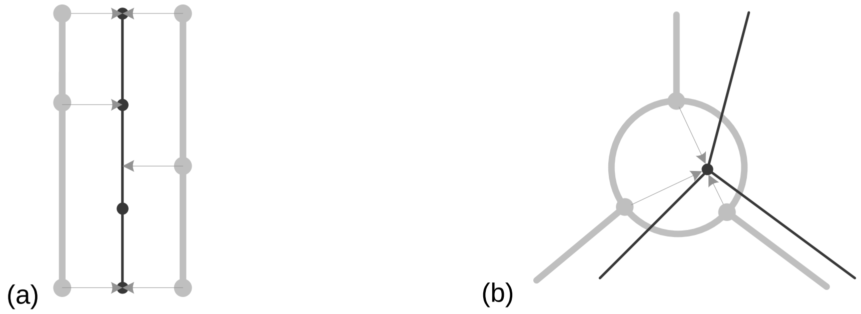
black lines: road objects from dataset *A*; grey lines: road objects from dataset *B*

Figure 2.8 Cardinality of the partial correspondences

The partial correspondence often occurs in the matching area which contains a lot of dead-ends of roads or where one (or both) of the road networks to be matched is quite fragmental. Little research has been conducted to solve the matching problems of partial correspondences in the field of road-network matching.

- **Equivalent correspondence**

As illustrated in Figure 2.9-a, the equivalent correspondences exist when one single road object chain of the less detailed dataset is corresponding to two or more object chains of the other dataset which are not topologically connected. For example, in one detailed dataset a dual carriageway is represented by two parallel lines which have different directions of traffic flow whereas in the other less detailed dataset it is abstracted by the middle axis (Sheeren et al. 2004; Volz and Walter 2004; Mustière and Devogele 2008). In addition, a node in one dataset might correspond to a polygon of the other dataset since they represent the same real-world loop crossing in different LoDs, see Figure 2.9-b.



black lines: road objects from dataset *A*; grey lines: road objects from dataset *B*

Figure 2.9 Instances of the equivalent correspondences

Let NW_A and NW_B represent the road networks to be matched. Suppose that NW_A is less detailed and consists of the road objects $\{a_1, a_2, \ldots , a_I\}$ and NW_B is $\{b_1, b_2, \ldots, b_J\}$, the equivalent correspondence can be mathematically defined by the following two expressions:

$$\{a_{i(1)}, a_{i(2)}, \ldots, a_{i(M)}\} \Leftrightarrow \{b_{j(1)}, b_{j(2)}, \ldots, b_{j(N1)}\} \cup \{b_{k(1)}, b_{k(2)}, \ldots, b_{k(N2)}\}\,, \quad (a_{i(\cdot)} \in NW_A,\ b_{j(\cdot)}, b_{k(\cdot)} \in NW_B) \qquad \ldots[2\text{-}10]$$

$$a_{i(1)} \cap a_{i(2)} \cap \ldots \cap a_{i(M)} \Leftrightarrow \{b_{j(1)}, b_{j(2)}, \ldots, b_{j(N1)}\}\,,\ (a_{i(\cdot)} \in NW_A,\ b_{j(\cdot)}, b_{k(\cdot)} \in NW_B) \qquad \ldots[2\text{-}11]$$

[2-10] defines the correspondence between a single line and a pair of parallel lines where the parallel lines are represented by a union of two object chains; [2-11] defines the relationship between a node and a polygon where the node is represented by the crossing point of many road objects.

2.3.2 Current matching algorithms for road networks

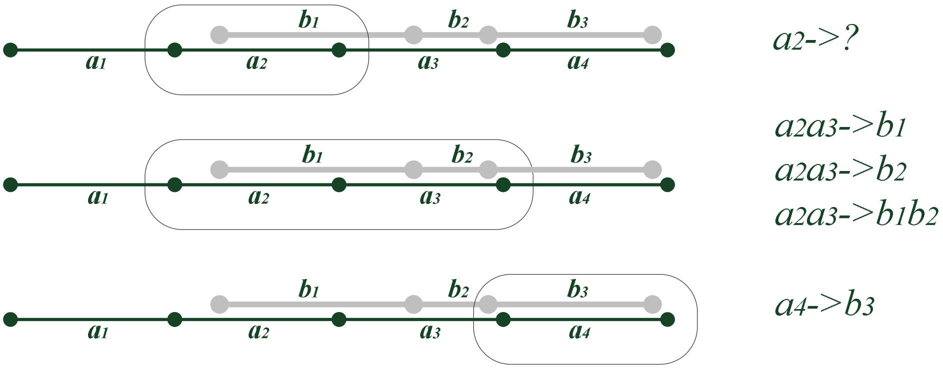
As the road network has a significant role in geospatial databases, street matching has been intensively and extensively researched during the recent decades. Despite other approaches, two of the most popular and well-known matching algorithms reported in literature hitherto are Buffer Growing (Walter 1997; Mantel and Lipeck 2004; Zhang et al. 2005; Zhang and Meng 2007) and Iterative Closest Point (ICP) (Besl and McKay 1992; Gösseln and Sester 2004; Gösseln 2005; Volz. 2006).

- **Buffer Growing**

Buffer Growing (BG) along with the necessary parameters is an efficient algorithm for the general task of line matching.

As depicted by Walter (1997), the *1:1* and *1:N (N>1)* matching pairs can be identified by buffers. A buffer is created around an object a_i from the dataset *A*. Then the road objects b_i or topologically connected object chain $\{b_{j(1)}, b_{j(2)}, \ldots, b_{j(N)}\}$ from the dataset *B* which are entirely inside this buffer will be confirmed as a possible matching candidate of a_i. To compute the *M:1* and *M:N (M>1, N>1)* matching pairs, however, a Buffer Growing process is necessary. Figure 2.10 illustrates the Buffer Growing process. In the first step a buffer is created around the road object a_2 from the dataset *A*.

The road object a_2 can not be matched to its corresponding counterpart because there is no object from dataset *B* which completely falls inside this buffer. Then the buffer is extended to the next road object a_3. Thus, the road object a_2 and a_3 are combined into one logical integrity a_2a_3 and can be matched to the road object b_1 because it is located in the buffer completely. In the next step, the road objects b_1 and b_2 can be combined in the same way to the integrity b_1b_2 and then matched to the a_2a_3. In this process, it is not necessary to consider the matching pair of $a_2a_3a_4 \Leftrightarrow b_1b_2b_3$ because this is already done by the combination of the matching pairs of $a_2a_3 \Leftrightarrow b_1b_2$ (*M:N* relationship, *M=2, N=2*) and $a_4 \Leftrightarrow b_3$ (*1:1* relationship) (Walter and Fritch 1999).



black lines: road objects in dataset *A*; grey lines: road objects in dataset *B*

Figure 2.10 The BG process (Buffer Growing) (Walter and Fritch 1999)

After the Buffer Growing process, a list of potential candidates for the matching reference is computed. The list may be ambiguous and typically contains a large number of matching candidates. By computing the geometric and topologic similarity between each matched pair, the best matching candidate can be confirmed as the final solution (Zhang and Meng 2007).

- **Iterative Closest Point Algorithm**

Iterative Closest Point (ICP) is an algorithm employed to match two point clouds. It was initially developed to align two-dimensional or three-dimensional objects using a rigid transformation (Besl and McKay 1992; Gösseln 2005).

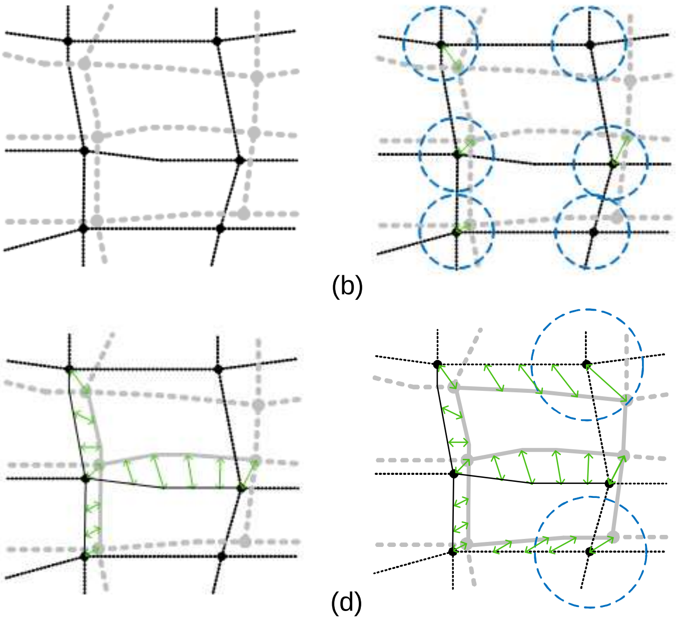
In the field of line network matching, the ICP algorithm is constituted by four steps:

Step 1: Extract all nodes of the line networks, in this way, the line network is represented as a matrix of nodes (see the nodes in Figure 2.11-a).

Step 2: Associate the corresponding nodes between different datasets by combining the measures of the Euclidian distance, the number of incident edges (viz. *valence*) and the angle differences between emanating edges (see the matched nodes in Figure 2.11-b).

Step 3: Based on the associated node pairs, the corresponding line objects between different datasets are then matched (see three matched lines in Figure 2.11-c). The basic philosophy behind this approach is that if two roads from different datasets have homologous starting and ending nodes, then there is a high likelihood that the roads themselves are corresponding counterparts (Volz 2006).

Step 4: Perform iterations so as to re-associate the weak correspondences of nodes with larger tolerant values and then identify new matching pairs of line objects. Thus, the network of matched line objects constantly grows (Gösseln and Sester 2004; Volz 2006) (see Figure 2.11-d).



(a) (b)

(c) (d)

black lines: road objects in dataset *A*; grey lines: road objects in dataset *B*
dashed lines: unmatched road objects; solid lines: matched road objects; arrows: linkages

Figure 2.11 The ICP algorithm (Iterative Closest Point)

The Buffer Growing is often regarded as a line-based matching while the ICP algorithm is based on the points (nodes). In general, the ICP algorithm has a higher matching speed than Buffer Growing.

2.3.3 Necessity for a contextual matching approach

The discussions of related works indicate the growing significance and indispensability of the data matching technique. So far a majority of the developed matching approaches based on Buffer Growing, Iterative Closest Point or the combination and evolution of them reveals high matching rate and efficiency on certain data types of selected test areas (Walter 1997; Xiong 2000; Meng and Töllner 2004; Xiong and Sperling 2004; Mantel and Lipeck 2004; Kolahdouzan et al. 2005; Zhang and Meng 2007). However, the problem of uncertain matching remains either in areas where the context is too complex or when one of the datasets contains little or no meaningful semantic information at all. For instance, in the related work the line matching has usually been approached by matching end nodes of lines and/or by searching corresponding lines in a given buffer. In cases where multiple lines are close to each other, identifying best matches become difficult and error-prone. In addition, with the current matching algorithms it is hardly possible to identify the counterparts which have partial or equivalent correspondences between different datasets. Indeed the matching of the line networks with dissimilar LoDs has been considered in some work (Mustière and Devogele 2008). However, these matching processes often require semantic attributes and can be only applied to the networks coming from the same data supplier. Therefore, they have limited value for more general network matching (Raimond and Mustière 2008).

An important insight from the precious research on line matching is that beside the geometry the topologic characteristics have to be taken into account to get more accurate matching pairs. Another insight is that a generic matching process should not depend on the semantic attributes (e.g. non-spatial properties) because the useful semantic information seldom exists in both datasets to be matched. However, the non-spatial attributes, like the road name, direction of the traffic flow, number of lane, width of the road, functional road class, speed limit, etc., need to be taken into account to enhance the matching performance in case that such valuable semantic attributes are available. Moreover, it has been generally assumed that better matching result can be reached with more context information (Stigmar 2006; Mustière 2006; Zhang and Meng 2008). Based on this common sense assumption, a new contextual matching approach based on the Delimited-Stroke-Oriented (DSO) algorithm has been developed to achieve the more robust and generic matching between different datasets.

According to the DSO algorithm, the conjoint edges to a Delimited Stroke can be easily brought together with the help of the ‘graph’ that records the conjoint objects. The corresponding network is then treated as an integral unit in the matching process, i.e. the DSO algorithm will lead to a network-based matching. As compared with point- or line-based matching, such as Buffer Growing and ICP algorithm, the network-based matching is able to implement topologic information more thoroughly, which allows a context-related topologic analysis and thus helps to improve the results of geometric or semantic matching. In principle, the DSO algorithm can be implemented to match all kinds of road networks from various data resources, as it does not rely on any semantic information. To be more exactly, with the DSO algorithm, not only the matching pairs with *m:n (m≥1, n≥1)* relationship, i.e. a chain of the road objects from the dataset *A* is entirely corresponding to one object chain from the dataset *B* (c.f. Figure 2.7), but also the matching pairs which are partially corresponded to each can be identified (c.f. Figure 2.8). Chapter 3 is dedicated to the more detailed descriptions of the DSO algorithm.

Furthermore, the proposed contextual matching approach utilizes three assisting methodologies described in Chapter 4 to benefit the automatic matching process as well. They can be briefly summarized as follows:

Methodology 1 - matching guided by *‘structure’:* to certain extent, a street network can be regarded as a unit constituted by various road structures, such as dual carriageways (parallel lines), roundabouts, narrow passages, navigation stubbles, slip roads around cloverleaf junctions and normal single carriageways. Since various road structures reveal different geometric or topologic characteristics, it is hardly possible to match them efficiently by the same criteria or methods. To circumvent this problem, a structure-guided matching strategy with proper parameters or criteria is developed. As the generalization technique has been integrated to the matching process, the structure-guided strategy acquires the capability to deal with the cases of equivalent correspondences, namely to identify the corresponding counterparts which reveal dissimilar LoDs in different datasets.

Methodology 2 - matching guided by *‘semantics’*: the semantic attributes should be considered to benefit the matching calculation if they are available in both datasets to be matched. This methodology deals with two essential questions (a) how to calculate the *semantic similarity;* and (b) how to take advantage the *semantic similarity* in different matching scenarios.

Methodology 3 - matching guided by *‘spatial index’:* a good matching approach should produce results with both high quality and at high speed. In order to increase the matching speed, especially when very large datasets are concerned, grid-based spatial indexes can be established for point objects and linear objects.

The contextual matching approach based on the DSO algorithm can be applied to automatically correlate different versions of the road networks under unfavourable conditions such as (a) some corresponding roads are represented at different LoDs; (b) in one of the datasets the road attributes are not completely covered with values, especially the street names which are essential clues for the matching are only sporadically available.

2.4 Terminology

In order to avoid confusions or misunderstandings, some significant terminologies which are closely related to the topic of road-network matching are discussed below.

- **Road object**: also refers to road feature in an image or a map, is the ‘basic item’ of a road network. In geospatial databases, a road network can be regarded as a unit formed by a number of connected road objects. Usually, a road object consists of an edge (line body, also termed as 'arc') and two nodes (viz. starting node and ending node, also denoted as 'from and to -node'), see an example in Figure 2.12.

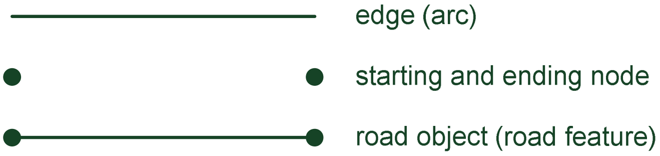
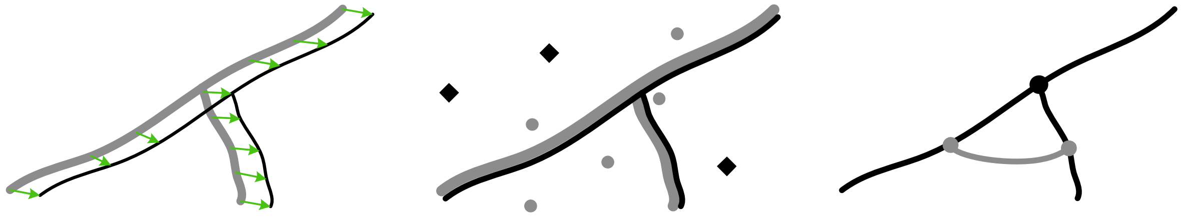


Figure 2.12 Decomposition of a road object

- **Road property**: is any physical or intangible entity that is owned by a road object. In general, the properties of a road object can be categorized into three groups:
  - **Geometry**: is a chain of coordinate pairs (x, y) which describes the location of a road object.
  - **Topology**: is introduced to distinguish qualitative relations from the ordinary geometry. In the context of road-network matching, the topology can represent the connectivity among various road objects.
  - **Semantic attributes**: in addition to the geometry and topology, semantic attributes denote other properties of a road object. Some of them, e.g. the street name, functional class, built material, maximum load, speed limit etc., represent non-spatial characteristics of a road whereas the others, e.g. street length, street width, number of lanes, direction of traffic flow, reflect certain spatial characteristics. Considering that these properties are often stored in the same way (e.g. in extra tables) in a geospatial database, they are uniformly treated as semantic attributes throughout this thesis.



(a) road-network matching (b) road-network integration (c) road-network conflation

black: reference dataset; grey: target dataset; green arrows: linkages

Figure 2.13 Diagrammatic sketches of road-network matching, integration and conflation

- **Road-network matching**: refers to the task of finding the object chains that refer to the same road entity between different road networks. In the matching process, one of the road networks is termed as 'reference' and the other is 'target', see the diagrammatic sketch in Figure 2.13-a.
- **Road-network integration**: aims at combining various kinds of geospatial data/attributes residing in different road networks so as to provide users with a new unified network, see the diagrammatic sketch in Figure 2.13-b.
- **Road-network conflation**: is a specific type of road-network integration. Conflation occurs when different road networks need to be jointed together, which involves splitting and connecting the associated roads or road parts from different data sources, both geometrically and topologically, see the diagrammatic sketch in Figure 2.13-c.

Chapter 3

Delimited-Stroke-Oriented Matching Algorithm

Matching of different road networks is essentially a process to identify the corresponding objects that represent the same real-world road in distinct datasets, i.e. the corresponding objects should be joined together during the data matching process. Nevertheless, some objects in one dataset may represent a real-world entity that is not represented in the other dataset. Such objects should not be joined with any object, and thus, should not appear in any pair of corresponding objects. As discussed in Chapter 2, the corresponding relationships of the counterparts distributed in different road networks can be generally classified into three groups:

(a) Entire correspondence, viz. $m:n\ (m\geq 1, n\geq 1, m, n\in N)$ matchings;

(b) Partial correspondence; and

(c) Equivalent correspondence.

This chapter deals with the Delimited-Stroke-Oriented (DSO) algorithm which is a new approach to solve the matching problems of group (a) and (b), that is, identify the matching pairs with $m:n\ (m\geq 1, n\geq 1, m, n\in N)$ or partial corresponding relationship between different road networks; whereas the equivalent corresponding counterparts are handled in Chapter 4.

Since there are no global identifiers that can tell whether two objects are corresponding objects, it is necessary to conduct the matching process based on the available properties of the geospatial objects. Three different groups of properties are often used to find the correspondences, viz. geometries, topologies and semantic attributes (ref. Section 2.3). Using geometries alone for computing matching pairs is not always sufficient. First, geometries are not accurate. The same road may have different geometries in different sources. Second, a polyline is represented by more than one point and two polylines that represent the same road may have different numbers of segments. Consequently, there is no straightforward way to compare all shape points along two polylines and determine whether the polylines are counterparts or not. The proposed DSO algorithm works with these difficulties by applying a hybrid matching approach based on geometries and topologies, where the semantic attributes are not yet considered. This is because the geometry and topology are always available or derivable for spatial objects whereas semantic attributes are not. As a result, the proposed algorithm can lead to an on-the-fly matching process based on the principle of *"What You Can See Is What You Can Match (WYCSIWYCM)"*, i.e. it is a generic matching algorithm which can be utilized for various kinds of road networks.

The matching rate and matching certainty are two significant criteria reflecting the quality of a matching approach. Most applications prefer a high matching certainty to a high matching rate, for example, a matching with 99% certainty and 70% matching rate is obviously more useful than one with 99% matching rate but 70% certainty. For this reason, the DSO algorithm is based on the following rationale: striving for a nearly perfect match for the majority of road objects, and then finding a hypothetic match for as many remaining line objects as possible.

Figure 3.1 depicts the key components of the DSO matching process, which goes through the following five stages iteratively:

1. Data preprocessing;
2. Graph construction;
3. Connection of the Delimited Strokes (DSs) at different levels;
4. Matching of the Delimited Strokes at each level; and
5. Treatment of fragmental matching areas.

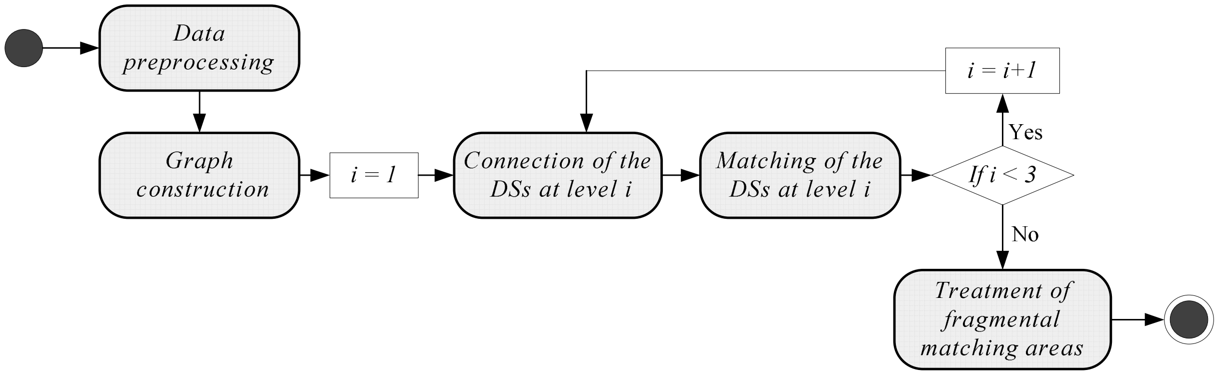

Data preprocessing
Graph construction
i = 1
Connection of the DSs at level i
Matching of the DSs at level i
If i < 3
Yes
i = i+1
No
Treatment of fragmental matching areas

Figure 3.1 The schematic flow diagram of the DSO matching algorithm

Data preprocessing aims at reducing the noise of irrelevant details, classifying the road intersections, describing topology with node geometry and eliminating topologic ambiguities in the datasets to be matched. The preprocessed geographic networks become *'graphs'* so that the conjoint objects can be directly identified and the separated road objects can be connected to Delimited Strokes at three different levels from integral to fragmental. The Delimited Strokes act as fundamental elements that undergo the iterative matching process. The matching stage itself is tuned for the worst case, that is only geometric and topologic information can be fully utilized due to incomplete values of essential semantic attributes in many practical cases. After the iterative matching process at different levels of Delimited Strokes, the final stage is triggered to deal with the matching areas where the road networks are fragmentized. In other words, this stage is dedicated to computing the matching pairs with partial correspondence between different datasets.

3.1 Data preprocessing

The process of data preparation is characterized by four procedures: reduction of noise or irrelevant details, topologic classification of road intersections, topologic description with node geometry and elimination of topologic ambiguity. The preprocessed data are stored in a temporary file and used for the matching, while the original geometries and attributes are preserved in the final matching results.

3.1.1 Reduction of noise or irrelevant details

If the datasets reveal very different LoDs, it is hard to match them directly (Mustière 2006; Volz and Bofinger 2002; Walter and Fritsch 1999). By comparing the number of object classes, the overall distribution density, the relative distribution density of each object class, and the number of attributes existing in the two datasets, i.e. the reference dataset and the target dataset for the matching, it is possible to exclude those roads entities or classes that exist only in one of the datasets. Further, geometric and topologic details of the dataset at higher resolution are simplified until its level of details approaches that of the other dataset (Zhang and Meng 2007).

3.1.2 Topologic typification of road intersections

Topology plays a significant role in the matching process. For example, the valence of node can be applied to roughly describe the topology (see Table 3.1).

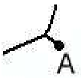
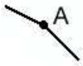
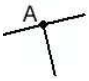
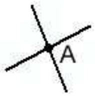
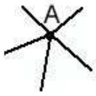
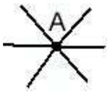
Example						
Valence	1	2	3	4	5	6

Table 3.1 The values of *Valence* (Zhang et al. 2005)

However, such a rough description is not adequate for the topologic matching. As shown in Figure 3.2, the nodes P_0 and P_1 have the same valence but they are apparently not homologous in fine-tuned topology. Identifying them as a matched pair would cause a topologic confusion.

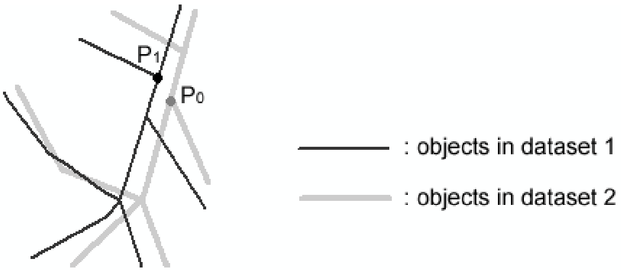


Figure 3.2 An example of equal valence but non-homologous object pair

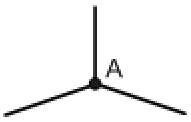
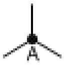
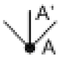
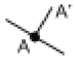
	Classification	$Typ_{TopoR=3}$	$Angle_{TopoR=3}$
 The nodes with the *Valence* equal to 3.		0	
		1	The direction of $\overrightarrow{AA'}$ $\in [0, 360°)$
		2	The direction of $\overrightarrow{AA'}$ $\in [0, 360°)$

Table 3.2 The values of $Typ_{TopoR=3}$ and $Angle_{TopoR=3}$

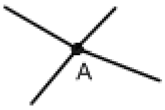
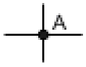
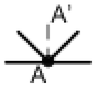
	Classification	$Typ_{TopoR=4}$	$Angle_{TopoR=4}$
 The nodes with the *Valence* equal to 4.		0	
		1	The direction of $\overrightarrow{AA'}$ $\in [0, 360°)$

Table 3.3 The values of $Typ_{TopoR=4}$ and $Angle_{TopoR=4}$

In order to enrich the topologic attributes, four further variables $Typ_{TopoR=3}$, $Angle_{TopoR=3}$, $Typ_{TopoR=4}$ and $Angle_{TopoR=4}$ are defined. According to the variable $Typ_{TopoR=3}$ (see Table 3.2), the nodes which have their valence equal to three can be classified into three groups and for the second and third groups ($Typ_{TopoR=3}$=1 and $Typ_{TopoR=3}$=2) the variable $Angle_{TopoR=3}$ is introduced to describe their further topologic characteristics - main directions. Correspondingly the variables $Typ_{TopoR=4}$ and $Angle_{TopoR=4}$ are employed to represent the characteristics of the nodes with the valence equal to four (see Table 3.3).

3.1.3 Topologic description of the endpoints

In Section 3.1.2, the five variables of valence, $Typ_{TopoR=3}$, $Angle_{TopoR=3}$, $Typ_{TopoR=4}$ and $Angle_{TopoR=4}$ are employed to distinguish different types of road intersections. Although such a typification is efficient to describe topologic characteristics of an intersection in most cases, it may still be insufficient for an exact road-network matching.

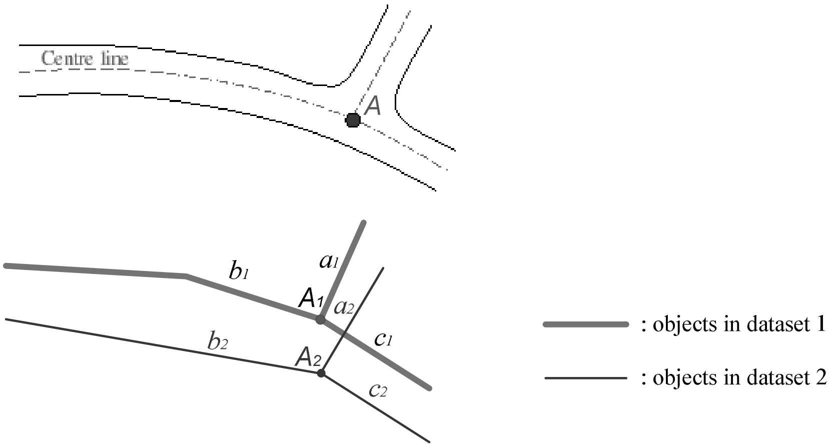


Figure 3.3 Similar representations of a road intersection in different datasets

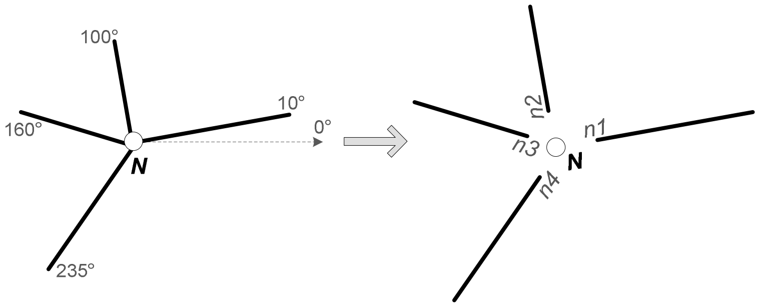


Figure 3.4 Decomposition of the node (N) into different endpoints (n1,n2,n3,n4)

Node	*Valence*	*Endpoints*	*Angle-Index*			
			Angle_1	*Angle_2*	*Angle_3*	*Angle_4*
N	*4*	*n1*	*10°*	*100°*	*160°*	*235°*
		n2	*100°*	*160°*	*235°*	*10°*
		n3	*160°*	*235°*	*10°*	*100°*
		n4	*235°*	*10°*	*100°*	*160°*

Table 3.4 Angle-Index of the node N

As depicted in Figure 3.3, the real-world road intersection *A* has quite similar topologic representations in dataset 1 (see node A_1) and dataset 2 (see node A_2). The similar topologies between *A1* and *A2* guarantee a high matching likelihood when the streets which are connected to this intersection are compared. This rule holds true for the streets a_1 and a_2 as they are corresponding branches from node *A*. *A* problem may occur, however, when the streets a_1 and c_2 are compared. Regardless of the considerable topologic proximities of the node A_1 and A_2, the down-left endpoint of a_1 reveals distinct topologic characteristics to the upper-left endpoint of c_2 because streets a_1 and c_2 represent different branches of node *A* in the reality. This example demonstrates that an accurate road-network matching process requires in addition to the topologic typification of road intersections also the information on how and in which sequence the branched streets are conjoint. To this end, an Angle-Index for the endpoints of street objects is added in the proposed DSO algorithm, see example in Figure 3.4 and Table 3.4.

The Figure 3.4 and Table 3.4 illustrate that one node with the valence larger than 1 is corresponding to several different endpoints and each of them has its own Angle-Index in the proposed matching model. Following this way, different endpoints of the same node, e.g. endpoints *n1*, *n2*, *n3* and *n4* of node *N*, can be distinguished from the topologic point of view although they are overlapped in geometry.

3.1.4 Elimination of topologic ambiguity

Though the noise and irrelevant details are reduced in the step 3.1.1, there are still topologic differences between the datasets to be matched. One of the common cases is that a node in one dataset may correspond to two or more adjacent nodes in the other. Figure 3.5-a shows a typical topologic ambiguity, which could result in erroneous matchings (e.g. in Volz 2006).

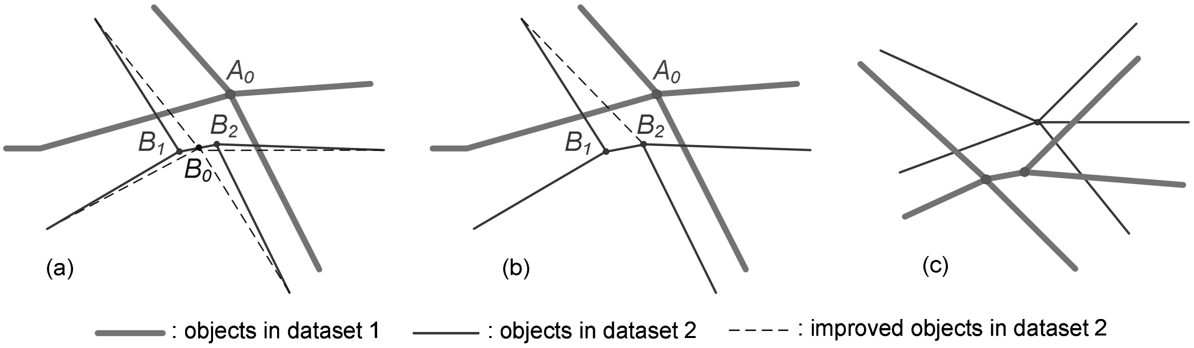


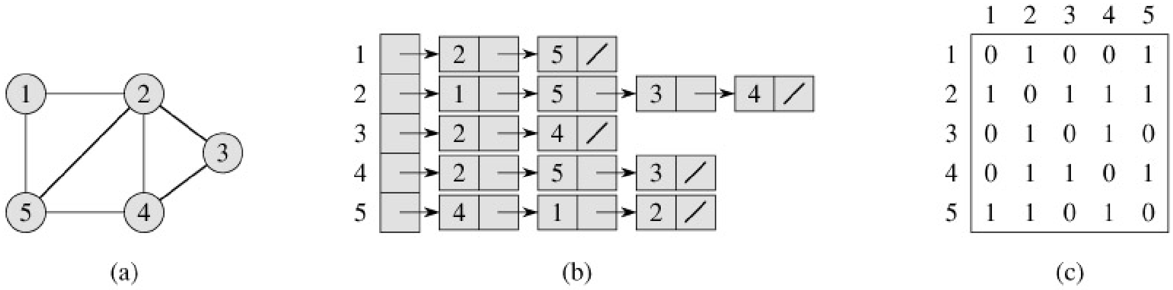
Figure 3.5 Further reduction of topologic differences through node aggregation

Such topologic ambiguities should be eliminated to build the unequivocal connection between the two datasets. In (Zhang and Meng 2007), the ambiguous nodes B_1 and B_2 are aggregated to their central point B_0 and at the same time the line segment $B_1 \rightarrow B_2$ is deleted (see Figure 3.5-a). So that more initial geometries and topologies can be kept, the proposed DSO algorithm implements another way to solve such problems by displacing one of the ambiguous nodes to another one. In Figure 3.5-b, the node B_1 has been moved to B_2 which is closer to A_0. This improvement process is triggered under the conditions (1) the Valences of both B_1 and B_2 are equal to 3; (2) the $Typ_{TopoR=3}$ of B_1 and B_2 are not 0; (3) the difference between the $Angle_{TopoR=3}$ of B_1 and B_2 is approximately to ±180°; and (4) neither B_1 nor B_2 has a corresponding node in the other dataset. The method can be easily extended to deal with other topologic ambiguities as shown in Figure 3.5-c where a node aggregation can be conducted in dataset 1.

3.2 Graph construction

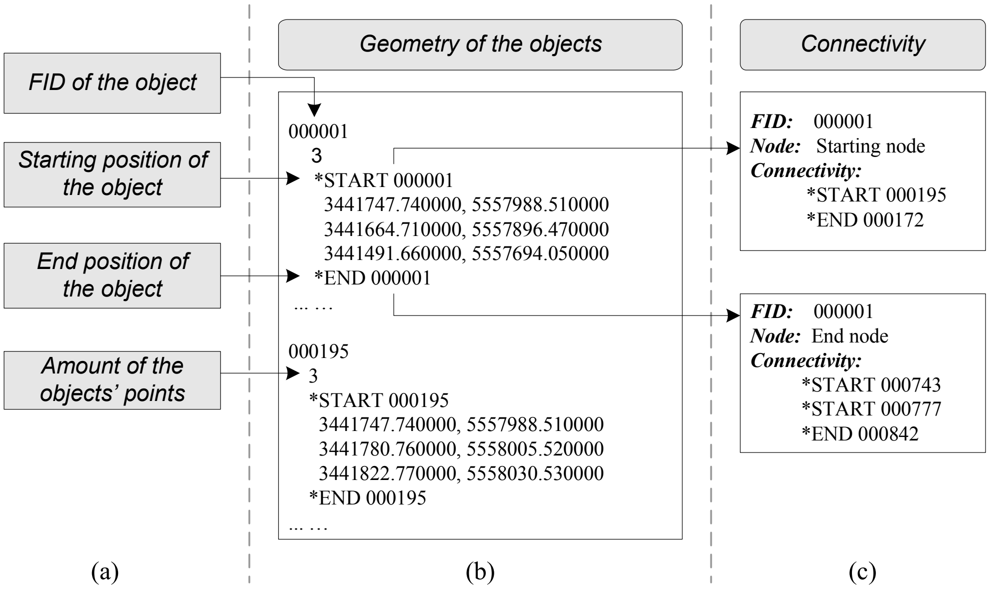
A robust matching process requires the connection of some conjoint objects into more extensive objects. However, the connection relationships between these objects are not clear in the beginning. The initial road network is often represented by many individual and fragmental objects. The geometry of each object is an oriented chain of turning points. In order to create and record the connection information between conjoint objects, the initial geographic networks need to be transformed into a graph structure.

The term '*graph'* is an abstract notion used to represent the idea of some kind of connection between pairs of objects (Rossum 1998). Mathematically, the graph of a line network can be described by an ordered pair $G=(N,E)$ comprising a set N of nodes together with a set E of edges, which are 2-element subsets of N (see Figure 3.6-a). Typically, there are two standard ways to represent a graph $G=(N,E)$ in computer science, as a collection of adjacency lists (see Figure 3.6-b) or as an adjacency matrix (see Figure 3.6-c). The adjacency-list representation is usually preferred because it provides a compact way to represent sparse graphs. The adjacency-list representation of a graph $G=(N,E)$ consists of an array Adj of $|N|$ lists, one for each node in N. For each $u\in N$, the adjacency list *Adj_N[u]* contains all the nodes *v* such that there is an edge $(u,v)\in E$. That is, *Adj_N[u]* consists of all the nodes adjacent to u in G. The nodes in each adjacency list are typically stored in an arbitrary order, for example, Figure 3.6-b uses a hash table with an array of adjacent nodes to represent the graph of the network in Figure 3.6-a (Cormen et al. 2001).



$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & 1 & 1 & 1 \\ 3 & 0 & 1 & 0 & 1 & 0 \\ 4 & 0 & 1 & 1 & 0 & 1 \\ 5 & 1 & 1 & 0 & 1 & 0 \end{array}$$

Figure 3.6 Two representations of a graph (Cormen et al. 2001): (a) A graph *G*; (b) An adjacency-list representation of *G*; (c) The adjacency-matrix representation of *G*.


Geometry of the objects
Connectivity
FID of the object
Starting position of the object
End position of the object
Amount of the objects' points
(a)
(b)
(c)

```
000001
  3
  *START 000001
    3441747.740000, 5557988.510000
    3441664.710000, 5557896.470000
    3441491.660000, 5557694.050000
  *END 000001
 ... ...
000195
  3
  *START 000195
    3441747.740000, 5557988.510000
    3441780.760000, 5558005.520000
    3441822.770000, 5558030.530000
  *END 000195
 ... ...
```

```
FID:   000001
Node:  Starting node
Connectivity:
        *START 000195
        *END 000172
```

```
FID:   000001
Node:  End node
Connectivity:
        *START 000743
        *START 000777
        *END 000842
```

Figure 3.7 Adjacency list for recording the conjoint objects

As an evolution to the adjacency-list representation, a graph can be also described by an array indexed by the positions points to a singly-linked list of the neighbours of each node. In the process of road-network matching, the adjacency list *Adj_N[u]* are better to be replaced by the list of adjacency edges to *u* in *G*, that is *Adj_E[u]* which consists of all the edges *(u, v)* where *v*∈ *Adj_N[u]*. Thus, the connection relationship between conjoint objects can be more clearly described. For instance, with the adjacency list illustrated in Figure 3.7, it can be easily recognized that five objects are conjoint with the object 000001: (a) the starting node of the object 000001 is conjoint with the starting node of object 000195 and the ending node of the object 000172; (b) the ending node of object 000001 is conjoint with the starting node of the object 000743, the starting node of the object 000777 and the ending node of the object 000842.

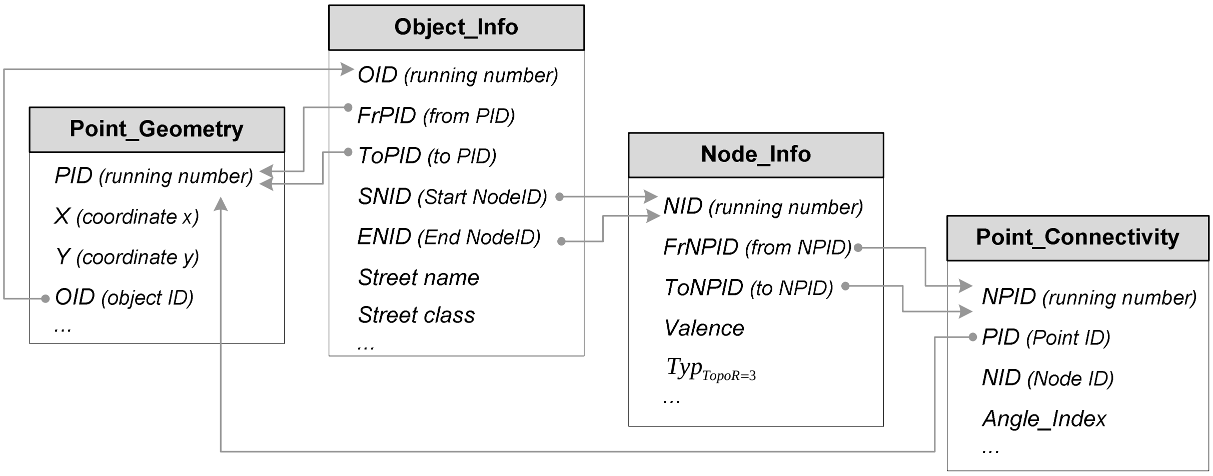
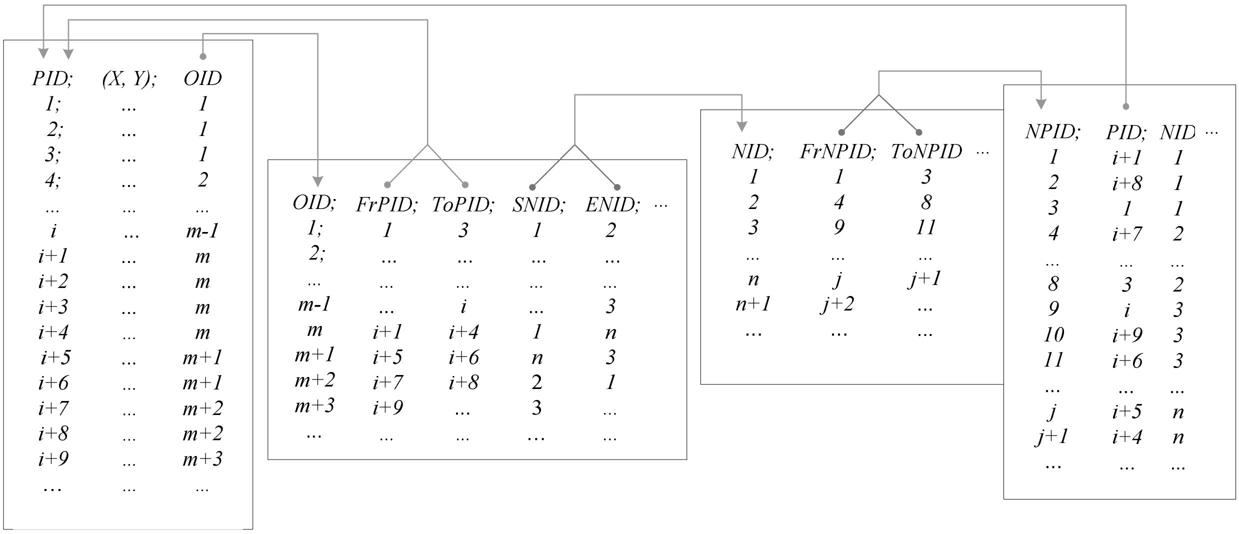


Figure 3.8 Data structure of the DSO matching algorithm



PID;	(X, Y);	OID
1;	...	1
2;	...	1
3;	...	1
4;	...	2
...	...	...
i	...	m-1
i+1	...	m
i+2	...	m
i+3	...	m
i+4	...	m
i+5	...	m+1
i+6	...	m+1
i+7	...	m+2
i+8	...	m+2
i+9	...	m+3
...	...	...

OID;	FrPID;	ToPID;	SNID;	ENID;	...
1;	1	3	1	2	
2;	...	...	...	...	
...	...	...	...	...	
m-1	...	i	...	3	
m	i+1	i+4	1	n	
m+1	i+5	i+6	n	3	
m+2	i+7	i+8	2	1	
m+3	i+9	...	3	...	
...	...	...	...	...	

NID;	FrNPID;	ToNPID	...
1	1	3	
2	4	8	
3	9	11	
...	...	...	
n	j	j+1	
n+1	j+2	...	
...	...	...	

NPID;	PID;	NID	...
1	i+1	1	
2	i+8	1	
3	1	1	
4	i+7	2	
...	...	...	
8	3	2	
9	i	3	
10	i+9	3	
11	i+6	3	
...	...	...	
j	i+5	n	
j+1	i+4	n	
...	...	...	

Figure 3.9 An example of applying the proposed data structure

A difficulty of the adjacency list depicted in Figure 3.7 is that it has no obvious place to store data associated with the objects of a graph, such as street name, street class etc. Moreover, since the objects may have different numbers of conjoint neighbours to each other, it is difficult to assign the length of the arrays to record the connection information under such a data structure. A remedy to these two disadvantages can be a more object-oriented data structure where the road objects act as the keys of the graph, and for each node of a key, the corresponding value is a list containing the edges that are connected to this node. To complete the structure, each object must point back to the positions of its endpoints; see details of the data structure in Figure 3.8.

Figure 3.9 shows an example of applying the proposed data structure, where the object with the *OID* equal to *1* termed as object *1*; has three turning points; (i) the starting node of the object *1* (i.e. *OID=1*) is conjoint with two objects, one is the starting node of object *m* (*NPID=1*) and the other is the ending node of object *m+2* (*NPID=2*); while its ending node (*NPID=8*) is conjoint with four objects with the *NPID* from 4 to 7; (ii) the object *m* is chained by four turning points and has three conjoint objects all together, which are the object *m+2* (*NPID=2*), object *1* (*NPID=1*), object *m+1* (*NPID=j*) respectively; and so on.

The original road networks will be firstly transformed into the graphs depicted by Figure 3.8 and 3.9. Following the graph structure, the geometry and semantic attributes of the objects, the connection relationships between conjoint objects, as well as other required information relevant to data matching, e.g. the valence, Angle-Index, etc., can be efficiently and compactly organized.

The standardized graphs are stored in the physical memory of computer and treated as a global variable in the whole matching approach. With such graphs, the conjoint objects can be easily identified, which makes it possible to involve more context information in the subsequent matching process.

3.3 Connection of the Delimited Strokes at different levels

In the process of data modelling, a new road object is often created when: (1) a street changes the name or other significant properties; (2) a street lies on the border of federal states; (3) the object type changes (e.g. a street is modelled as line-shaped or complex); (4) area-shaped objects of different classes adjoin to each other; (5) there is one entrance to an important POI (point of interest), etc. Therefore, one real-world road could be divided into a number of much shorter objects which may be too fragmental to reveal sufficient geometric and topologic characteristics for an accurate lines matching. Moreover, too many unnecessary divisions can also reduce the matching efficiency since in the process of testing all possible matches it has to consume some computing resources. Therefore, an efficient and accurate matching algorithm requires the *'fundamental elements'* at more abstracted levels, i.e. a series of conjoint road objects could be chained together and then act as the fundamental element in the matching process.

On the other hand, however, using the fundamental elements at more abstract levels may not always benefit the matching result as it may lead to many partial correspondences between different road networks and thus reduce the ratio of successful matching. The more abstract or longer fundamental element may lead to a higher matching accuracy and computing speed, but a lower matching rate.

To overcome this dilemma, the author introduced an iterative matching algorithm – Delimited-Stroke-Oriented algorithm or DSO algorithm. The Delimited Strokes are progressively constructed at three different levels from abstract to fragmental (see definitions in Table 3.5) and act as the fundamental elements in each iterative matching process.

	Definition of Delimited Strokes
level 1	It represents a series of connected objects which have *good continuity* to each other and are delimited by *efficient terminating nodes*.
level 2	It represents an object chain which is delimited by *crossing*(s) or *dead-end*(s).
level 3	It is equal to an object.

Table 3.5 Definition of the Delimited Strokes at different levels

The terms *crossing*, *dead-end*, *good continuity* and *efficient terminating node* used in Table 3.5 are explained as follows:

- ***Crossing***

In the context of road networks, the crossing, also called intersection, refers to a node with the valence (ref. Table 3.1) larger than 2 (≥3), see examples of points A, B, C and D in Figure 3.10.

- ***Dead-end***

A node with the valence equal to 1 can be regarded as a dead-end, see the example of point D in Figure 3.11-b.

- ***Good continuity***

The line segment B→C follows another line segment A→B in almost in the same direction. This characteristic is termed as good continuity (Figure 3.10-a), whilst B→D reveals a sharp turning at a sharp angle ß from A→B, therefore, it is disqualified as a good continuity (Figure 3.10-b).

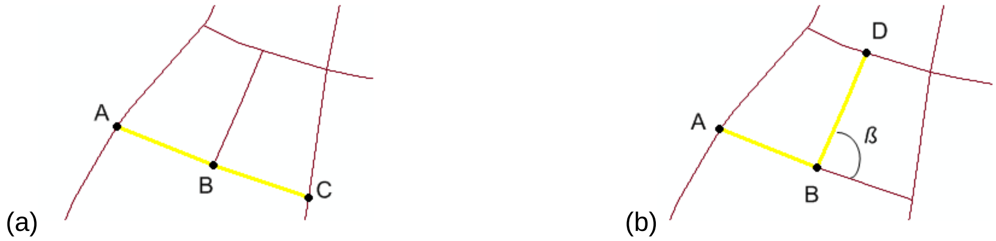


Figure 3.10 Examples of (a) good continuity and (b) bad continuity

- ***Efficient terminating nodes***

The black nodes in Figure 3.11 are several examples of efficient terminating nodes. They follow a number of rules. The node with a valence larger than three (≥4) (e.g. node A and B in Figure 3.11-a) are regarded as efficient terminating nodes because at such a node the road objects cross each other. The node with the valence equal to 1 (e.g. the node D in Figure 3.11-b) is also an efficient terminating node since it is the dead end of a road.

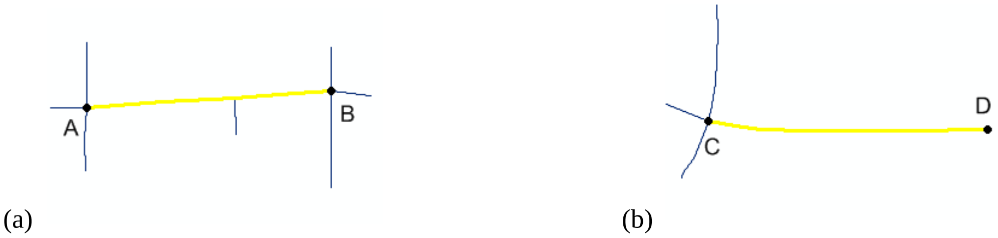


Figure 3.11 Examples of ’efficient terminating nodes’ with the *valence* either larger than 3 or equal 1

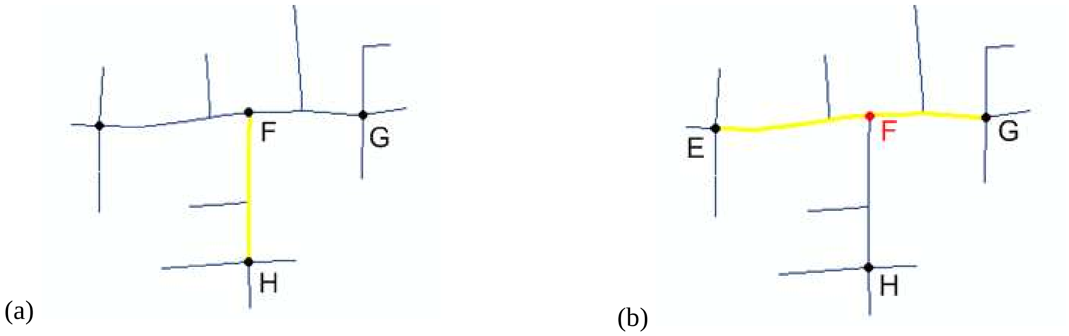


Figure 3.12 Examples of ‘efficient terminating nodes’ with the *valence* equal to 3

If the valence of a node is equal to three, it is not straightforward to judge whether this node belongs to efficient terminating nodes or not. On this occasion more context information has to be considered, e.g. the node F acts as a terminating point to delimit the stroke F→H in Figure 3.12-a.

However in Figure 3.12-b, F can not be treated as efficient terminating node because the roads E→F and F→G have a smooth connectivity at this position, i.e. in our proposed matching approach neither E→F nor F→G, but the E→G could be regarded as the Delimited Stroke at level 1.

3.4 Matching of the Delimited Strokes

As illustrated in Figure 3.13, the stage of matching corresponding Delimited Strokes between different datasets starts with an exploring process to compute the list of the potential matching pairs of Delimited Strokes between the datasets to be matched (step 3.4.1). In step 3.4.2, some of the incorrect matching pairs can be identified and eliminated after the comparison of their geometric similarities. For explicit representation, the matching pairs which passed the exclusion criteria of step 3.4.2 are denoted as promising matches. Since the promising matching pairs tend to be inaccurate, a further test on each promising pair is conducted in step 3.4.3, which is followed by the process of network-based selection to confirm the unique matching pairs (step 3.4.4). The nodes on twigs of the matched networks can be further treated as seeds for the matching-growing in step 3.4.5. In the following sections these individual steps are described in more detail.

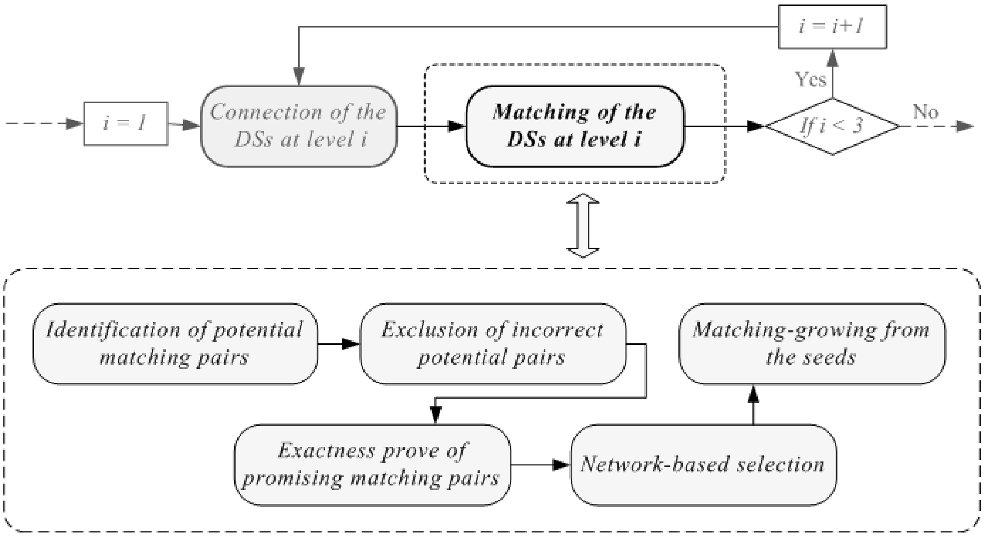

i = i+1
Yes
i = 1
Connection of the DSs at level i
Matching of the DSs at level i
If i < 3
No
Identification of potential matching pairs
Exclusion of incorrect potential pairs
Matching-growing from the seeds
Exactness prove of promising matching pairs
Network-based selection

Figure 3.13 The stage of matching corresponding Delimited Strokes (DSs) between different datasets

3.4.1 Identification of the potential matching pairs

If the datasets to be matched have similar resolutions, most of the corresponding Delimited Strokes (*DSs*) will have *1:1* relationship in the first matching iteration, that is one Delimited Stroke from the reference dataset is corresponding to one Delimited Stroke from the target dataset at the abstraction level 1 of Table 3.5. However, *M:1*, *1:N* or *M:N (M>1, N>1)* relationships are common when the datasets reveal dissimilar levels of detail. In this latter case, an exploring process is necessary which can be elucidated by the example in Figure 3.14.

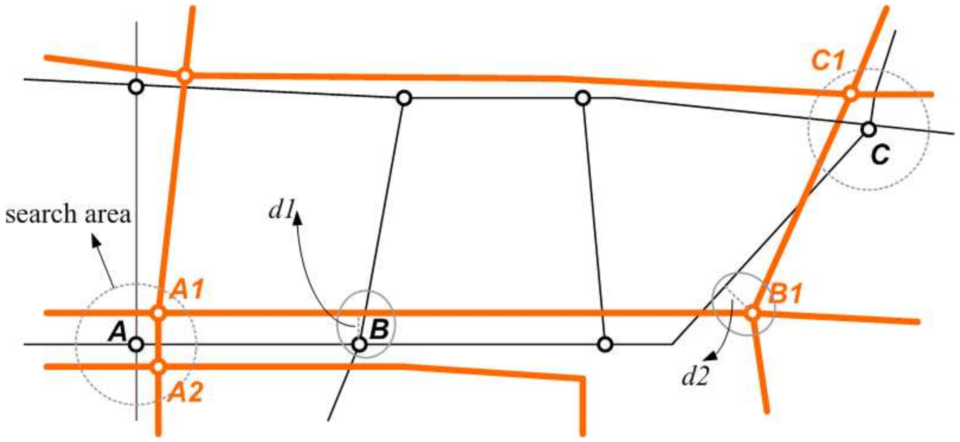
The Delimited Stroke *A→B* from the reference dataset is termed as the *'seed polyline (sPL)'*. At first, a user-defined searching area is built around point *A* - the starting node of the *'sPL'*. All nodes from the target dataset which fall inside this area and reveal sufficient topologic similarity to point *A* (e.g. *A1* and *A2* in Figure 3.14) are selected as initial spots of the potential matching candidates. The set of initial spots *U(A)* can be represented as $U(A) = \{p_i | \Delta d(p_i, A) < T_d, \Pr oximity(p_i, A) > T_{pro}\}$ mathematically, where *A* is the centroid of the searching area in Figure 3.14; $\Delta d(p_i, A)$ is the Euclidian distance between p_i and *A*; $\Pr oximity(p_i, A)$ represent the proximity of the two nodes p_i and *A*, which is a function of their Euclidian distance $\Delta d(p_i, A)$ and topologic difference

$\Delta Topo\_N(p_i, A)$; T_d is the radius of the user-defined area and T_{pro} is an empirical threshold. Since there is no linear object to be referenced at present, the topologic difference $\Delta Topo\_N$ can be only be roughly calculated by Expression [3-1].

$$\Delta Topo\_N(N1, N2) = \qquad \ldots[3\text{-}1]$$

$$\begin{cases} 0 & \text{if } Valence(N1) = Valence(N2) \neq 3 \text{ and } Valence(N1) = Valence(N2) \neq 4 \\ 0 & \text{if } Valence(N1) = Valence(N2) = 3 \text{ and } Typ_{TopoR=3}(N1) = Typ_{TopoR=3}(N2) = 0 \\ \left\| \left| \dfrac{Angle_{TopoR=3}(N1) - Angle_{TopoR=3}(N2)}{180^\circ} \right| \right\|_{(\leq 1)} & \text{if } Valence(N1) = Valence(N2) = 3 \text{ and } Typ_{TopoR=3}(N1) = Typ_{TopoR=3}(N2) \neq 0 \\ \xi_{\Delta Topo,1} & \text{if } Valence(N1) = Valence(N2) = 3 \text{ and } Typ_{TopoR=3}(N1) \neq Typ_{TopoR=3}(N2) \\ 0 & \text{if } Valence(N1) = Valence(N2) = 4 \text{ and } Typ_{TopoR=4}(N1) = Typ_{TopoR=4}(N2) = 0 \\ \left\| \left| \dfrac{Angle_{TopoR=4}(N1) - Angle_{TopoR=4}(N2)}{180^\circ} \right| \right\|_{(\leq 1)} & \text{if } Valence(N1) = Valence(N2) = 4 \text{ and } Typ_{TopoR=4}(N1) = Typ_{TopoR=4}(N2) \neq 0 \\ \xi_{\Delta Topo,2} & \text{if } Valence(N1) = Valence(N2) = 4 \text{ and } Typ_{TopoR=4}(N1) \neq Typ_{TopoR=4}(N2) \\ \xi_{\Delta Topo,3} & \text{if } Valence(N1) \neq Valence(N2) \end{cases}$$

In this expression, *N1* and *N2* represent two nodes; $Valence$, $Typ_{TopoR=3}$, $Angle_{TopoR=3}$, $Typ_{TopoR=4}$ and $Angle_{TopoR=4}$, are variables defined in Section 3.1.2; $\|\mu\|_{(\leq 1)}$ is a norm which can be defined as: if $|\mu| \leq 1$ then $\|\mu\|_{(\leq 1)}$ = $|\mu|$, otherwise $\|\mu\|_{(\leq 1)}$ =2- $|\mu|$; $\xi_{\Delta Topo,1}$, $\xi_{\Delta Topo,2}$ and $\xi_{\Delta Topo,3}$ are three empirical coefficients which are usually settled following the rules of $0.5 < \xi_{\Delta Topo,1} \leq \xi_{\Delta Topo,2} < \xi_{\Delta Topo,3} < 1$.



black: lines from the reference dataset orange: lines from the target dataset

Figure 3.14 Identification of the potential Delimited Stroke matching pairs

The *U(A)* consists of various set of initial spots from different datasets which show a high likelihood of correspondences. Subsequently, the exploring matching process goes on with these nodes from set *U(A)* one after another. In the stage of matching the Delimited Strokes at abstraction level 1 (ref. Table 3.5), for instance, it firstly picks up the Delimited Stroke *A_1→B_1* from the target dataset and treats it as a potential matching candidate since the distance *d_1* between the reference node *B* and *A_1→B_1* is small enough. However, *A→B* and *A_1→B_1* can not be regarded as matching pairs in the end due to their too large length discrepancy. As the current '*sPL (seed polyline)' A→B* is much shorter than *A_1→B_1*, the reference dataset is further explored with the attempt to extend the *'sPL'* with one more Delimited Stroke *B→C* so that the overall length is close to *A_1→B_1*. As a result, *A→C* becomes the new *'sPL'* and a new searching in the target dataset will be triggered considering that the new *'sPL'* is much longer than the current potential matching candidate *A_1→B_1* and the distance *d_2* is small enough. The exploring process is iteratively executed until the terminating spots of the matching reference and candidate are sufficiently near to each other, i.e. the terminating spot of the matching candidate (see point C_1 in Figure 3.14) falls inside the searching area of the terminating spot of the reference polyline (see point *C)*. On this occasion, the matching reference and candidate is considered to be potentially corresponding to each other. During the iteration, the Delimited Stroke in reference and target may successively grow to include a further Delimited Stroke if the reference and target polylines are nearly located (e.g. d_1, d_2 are small enough) while their terminating spots too far away from each other. The iterative process on the tricky example in Figure 3.14 will match *A*→*C* with A_1→C_1 together, where *A*→*C* consists of two Delimited Strokes *A*→*B* and *B*→*C*, and A_1→C_1 is constituted by Delimited Strokes A_1→B_1 and B_1→C_1, although neither "*A*→*B* and A_1→B_1"nor "*B*→*C* and B_1→C_1" are corresponding to each other, i.e. the example in Figure 3.14 leads to a potential matching pair with *M:N (M=2, N=2)* relationship.

In case that the set *U(A)* turns out to be empty, or no potential matching pair is calculated starting from this set, the matching process can be started around the ending node *B* of the initial *'seed polyline'* in Figure 3.14. If even the set *U(B)* is empty or failed to generate qualified potential matching pairs, then a new *seed polyline* will be picked up.

3.4.2 Exclusion of incorrect potential matching pairs

The set of potential matching pairs of Delimited Strokes contains not only correct match, but some wrong suggestions. In order to exclude the wrong matching pairs, a number of geometric constraints have been taken into account with respect to the differences of length, angle, shape, location, average area between the two polylines of each potential matching pair (denoted as PL_1 and PL_2). In the following paragraphs of this section, PL_1 and PL_2 are interpreted as two oriented sets of vertices $<p_{1,1}\ p_{1,2}\ldots\ p_{1,n}>$ and $<p_{2,1}\ p_{2,2}\ldots\ p_{2,m}>$ where $p_{N,j} = (x_{N,j}, y_{N,j})$ *(N=1 or 2)*.

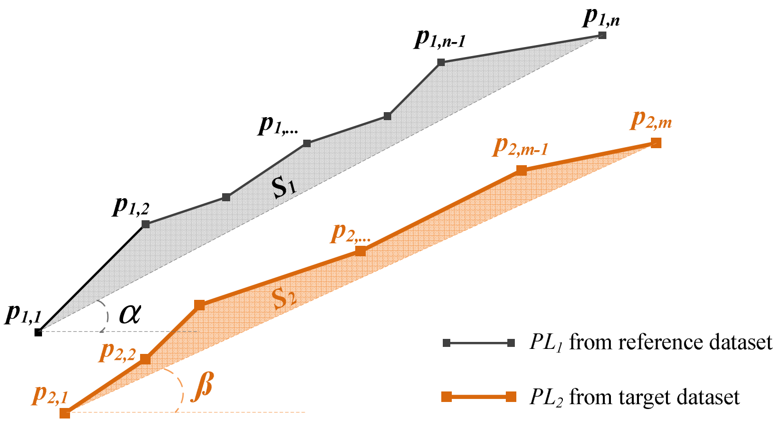


Figure 3.15 Geometric differences between PL_1 = $<p_{1,1}\ p_{1,2}\ldots\ p_{1,n}>$ *and* PL_2 = $<p_{2,1}\ p_{2,2}\ldots\ p_{2,m}>$

- **Orientation difference**

For a polyline, the orientation can be approximately described by the angle between the straight line connecting its two endpoints and the x-axis, see α of PL_1 and β of PL_2 in Figure 3.15. If the angle difference $(\alpha - \beta)$ is about 0 radians, the polylines PL_1 and PL_2 are nearly parallel and have the same orientation; if the value of the angle difference is close to π, polylines are parallel but have opposite directions; if the angle approximates π/2, then these two polylines are perpendicular. Equation [3-2] provides a normalization criterion to measure the orientation difference between PL_1 and PL_2.

$$N(\Delta\theta) = \frac{\Delta\theta}{\Delta\theta_{tolerance}} = \frac{\arccos(\frac{\vec{v}_{PL1} \cdot \vec{v}_{PL2}}{|\vec{v}_{PL1}| \cdot |\vec{v}_{PL2}|})}{\Delta\theta_{tolerance}} \qquad \ldots[3\text{-}2]$$

where, $\vec{v}_{PL1}$ is a vector formed by the starting and ending point of polyline *PL₁* = <*p₁,₁ p₁,₂... p₁,ₙ*>;

$\vec{v}_{PL2}$ is a vector formed by the starting and ending point of polyline *PL₂* = <*p₂,₁ p₂,₂... p₂,ₘ*>;

$\Delta\theta_{tolerance}$ is an empirical tolerance value of the angle difference between two matched polylines.

- **Length difference**

The length difference between the two polylines PL_1 and PL_2 of each potential matching pair can be calculated by the $\Delta l = |l_1 - l_2|$ in Equation [3-3]. The smaller the value Δl, the more appropriate the matching pair is. The length difference between the polylines PL_1 and PL_2 is normalized by the $N(\Delta l)$ by Equation [3-3], which will be employed to define the geometric constraints in the latter processes.

$$N(\Delta l) = \frac{\Delta l}{\Delta l_{tolerance}} = \frac{|l_1 - l_2|}{\min\begin{cases}\max\begin{cases}(l_1 + l_2)\cdot\Delta r(l)_{tolerance}\\ \Delta l_{\min}\end{cases}\\ \Delta l_{\max}\end{cases}} \quad \text{...[3-3]}$$

where, $l_1 = \sum_{i=1}^{n-1}[(p_{1,i+1} - p_{1,i})\cdot(p_{1,i+1} - p_{1,i})^{\mathrm{T}}]^{1/2}$ is the length of PL_1 = <$p_{1,1}$ $p_{1,2...}$ $p_{1,n}$>;

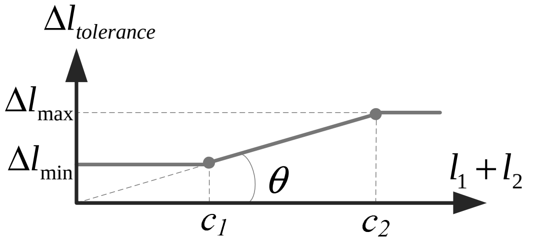
$l_2 = \sum_{j=1}^{m-1}[(p_{2,j+1} - p_{2,j})\cdot(p_{2,j+1} - p_{2,j})^{\mathrm{T}}]^{1/2}$ is the length of PL_1 = <$p_{2,1}$ $p_{2,2...}$ $p_{2,m}$>;

$\Delta l_{\max}$ is a predefined coefficient with respect to the maximal tolerance value of the length difference between two matched polylines;

$\Delta l_{\min}$ is a predefined coefficient with respect to the minimal tolerance value of the length difference between two matched polylines;

$\Delta r(l)_{tolerance}$ is a predefined coefficient with respect to the tolerance ratio divided between the length difference and summation.

Equation [3-3] shows that the tolerance value of length difference (viz. $\Delta l_{tolerance}$) is not a constant. Instead, it reveals certain of self-adaptive natures in measuring the length similarity between different pairs of polylines to be compared. The general trend is the longer the polylines to be compared, the larger the tolerance value $\Delta l_{tolerance}$ is, while the tolerance value has to be never lower than $\Delta l_{\min}$ or exceed $\Delta l_{\max}$, which can be seen in Figure 3-16.



where, $\theta = \arctan(\Delta r(l)_{toerance})$

$c_1 = \Delta l_{\min} / \Delta r(l)_{tolerance}$

$c_2 = \Delta l_{\max} / \Delta r(l)_{tolerance}$

Figure 3.16 The general trend of $\Delta l_{tolerance}$

- **Area difference**

By connecting the starting point and ending point directly, two polygons can be created for PL_1 and PL_2, viz. polygons PLg_1 = <$p_{1,1}$ $p_{1,2}$ … $p_{1,n}$ $p_{1,1}$> and PLg_2 = <$p_{2,1}$ $p_{2,2}$… $p_{2,m}$ $p_{2,1}$> (see Figure 3.15). Comparing the areas between PLg_1 and PLg_2 can also help to judge whether PL_1 and PL_2 are proper homologous polylines from different datasets or not, see the normalized area difference in Equation [3-4].

$$N(\Delta AvS) = \frac{\Delta AvS}{\Delta AvS_{tolerance}} = \frac{\left|\frac{S_1}{D_1} - \frac{S_2}{D_2}\right|}{\Delta AvS_{tolerance}} \qquad \ldots[3\text{-}4]$$

where, $S_1 = \frac{1}{2}\sum_{i=1}^{n+1} p_{1,i} \cdot \begin{vmatrix} 0 & 1 \\ -1 & 0 \end{vmatrix} \cdot p_{1,i+1}{}^{\mathrm{T}}\ (p_{1,n+1} = p_{1,1})$ is the area of polygon PLg_1 = <$p_{1,1}$ $p_{1,2}$ … $p_{1,n}$ $p_{1,1}$>;

$S_2 = \frac{1}{2}\sum_{j=1}^{m+1} p_{2,j} \cdot \begin{vmatrix} 0 & 1 \\ -1 & 0 \end{vmatrix} \cdot p_{2,j+1}{}^{\mathrm{T}}\ (p_{2,m+1} = p_{2,1})$ is the area of polygon PLg_2 = <$p_{2,1}p_{2,2}$… $p_{2,m}$ $p_{2,1}$>;

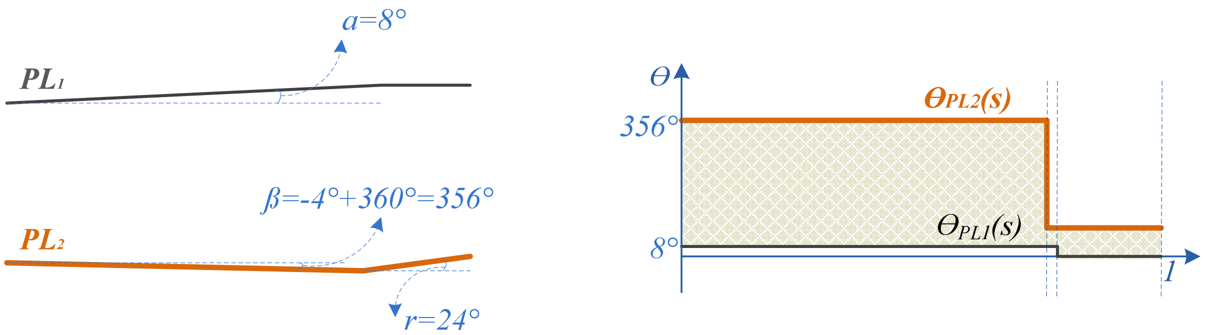
$D_1 = [(p_{1,n} - p_{1,1}) \cdot (p_{1,n} - p_{1,1})^{\mathrm{T}}]^{1/2}$ is the distance between two endpoints of the polyline PL_1;

$D_2 = [(p_{2,n} - p_{2,1}) \cdot (p_{2,n} - p_{2,1})^{\mathrm{T}}]^{1/2}$ is the distance between two endpoints of the polyline PL_2;

$\Delta AvS_{tolerance}$ is the tolerance for the average area difference between matched polylines;

- **Shape difference**

As illustrated in Section 2.2.3.1, the *cumulative angle function*, also denoted as *turning function* $\Theta_A(s)$ is a well-known shape descriptor for polygons (viz. enclosed polygonal lines) and hereby the shape difference between two polygons can be efficient measured by the Equation [2-5]. However, the Equation [2-5] suffers some limitation if it is directly applied for the polyline matching. For instance, the polyline PL_1 and PL_2 in Figure 3.17-a have similar geometric characteristics of 'shape' from the data matching point of view, whereas their *turning functions* $\Theta(s)$ are very different from each other (see Figure 3.17-b).



(a) The polylines to be compared: PL_1 and PL_2 (b) *Turning angle function* of PL_1 and PL_2

Figure 3.17 Limitation of the *turning function* for line matching

This limitation can be overcome by an improved metric function. Consider the two polylines PL_1 and PL_2 and their associated *turning functions* $\Theta_{PL1}(s)$ and $\Theta_{PL2}(s)$. The degree to which PL_1 and PL_2 are similar can be measured by Equation [3-5]:

$$L_P(PL_1, PL_2) = \left\|\Theta_{PL1} - \Theta_{PL2}\right\|_P = \left(\int f(\Theta_{PL1}, \Theta_{PL2})^P \cdot ds\right)^{\frac{1}{P}} \qquad \ldots[3\text{-}5]$$

where, $f(\Theta_{PL1},\Theta_{PL2}) = \begin{cases} \left|\Theta_{PL1}-\Theta_{PL2}\right|, & if \quad \left|\Theta_{PL1}-\Theta_{PL2}\right| \le 180° \\ 360° - \left|\Theta_{PL1}-\Theta_{PL2}\right| & if \quad \left|\Theta_{PL1}-\Theta_{PL2}\right| > 180° \end{cases}$.

In practice, the parameter *P* is often assigned by the value ‘*1*’, thus $L_P(PL_1, PL_2)$ can represent the area enclosed by the curve *f (ϴ_{PL1}(s), ϴ_{PL2}(s))* and the horizontal axis, see the shadow part of Figure 3.18.

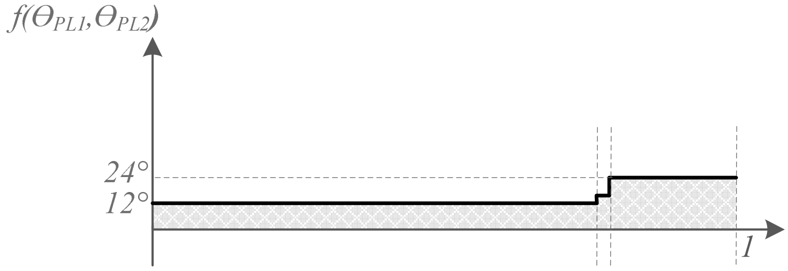


Figure 3.18 The rectangular strips formed by the curve *f (ϴ_{PL1}(s), ϴ_{PL2}(s))* and the horizontal axis

Usually, the larger the *L_P,* the less similar the polylines are in terms of their shapes. Equation [3-6] gives the normalized value of $L_P(PL_1, PL_2)$, where $L_p\|\cdot\cdot\|_{tolerance}$ is an empirically defined tolerance.

$$N(L_p) = \frac{L_p(PL_1, PL_2)}{L_p\|\cdot\cdot\|_{tolerance}} \qquad \text{…[3-6]}$$

- **Location difference**

The location difference can be measured by the *‘average distance’* between the polylines *PL₁* and *PL₂*. With a scenario similar to that of ‘*Frechét distance*’ (Eiter and Mannila 1994; Mascret et al. 2006), the so-called ‘average distance’ can be intuitively defined as follows: A man is walking a dog on a leash; the man moves on one curve, the dog on the other; each of them walks with a regular speed and arrive at the endpoints synchronously. The *‘average distance’* refers to the average length of the leash that is sufficient for traversing both curves. Given two curves that are equivalent to two continuous functions: $f[a,a'] \to V$ and $g[b,b'] \to V$, where $a,a',b,b' \in \Re, a < a', b < b'$ and (V,d) is a metric space. Then d_{AV} denotes their ‘average distance’ defined as:

$$d_{AV}(f,g) = \inf_{\substack{\alpha[0.1]\to[a,a'] \\ \beta[0.1]\to[b,b']}} \int_0^1 Dis(f(\alpha(t)), g(\beta(t))) \cdot dt \qquad \text{…[3-7]}$$

Where $Dis(f(\alpha(t)), g(\beta(t)))$ represent the Euclidean distance between positions of the man and his dog at time point *t*.

For the polylines with finite number of vertices, e.g. *PL₁* = <*p_{1,1} p_{1,2...} p_{1,n}*> and *PL₂* = <*p_{2,1} p_{2,2...} p_{2,m}*>, it is possible to find out the corresponding position for each vertices of *PL₁* along *PL₂,* denoted as *PL₁'* = <*p'_{1,1} p'_{1,2...} p'_{1,n}*> (see Figure 3.19-b) by means of interpolation. Reversely, the vertices of *PL₂* can also find their corresponding positions along *PL₁*, denoted as *PL₂'* = <*p'_{2,1} p'_{2,2...} p'_{2,m}*> (see Figure 3.19-c). Thus, the ‘average distance’ between *PL₁* and *PL₂* can be roughly calculated by these discrete pairs of corresponding points, see Equation [3-8]:

$$d_{AV}(PL_1, PL_2) = \frac{1}{2} \cdot \frac{\sum_{i=1}^{n-1}\left|l_{1,i\to1,i+1}\right| \cdot \left(\left|l_{1,i\to1,i'}\right| + \left|l_{1,i+1\to1,i+1'}\right|\right) + \sum_{j=1}^{m-1}\left|l_{2,j\to2,j+1}\right| \cdot \left(\left|l_{2,j\to2,j'}\right| + \left|l_{2,j+1\to2,j+1'}\right|\right)}{\sum_{i=1}^{n-1}\left|l_{1,i\to1,i+1}\right| + \sum_{j=1}^{m-1}\left|l_{2,j\to2,j+1}\right|} \qquad \text{…[3-8]}$$

Where $l_{N,A\to N,B}$ ($N=1$ or 2) represents the line segment from point $p_{N,A}$ to $p_{N,B}$ and its length is denoted as $|l_{N,A\to N,B}|$; $l_{N,A\to N,B'}$ represents the line segment from point $p_{N,A}$ to $p'_{N,B}$.

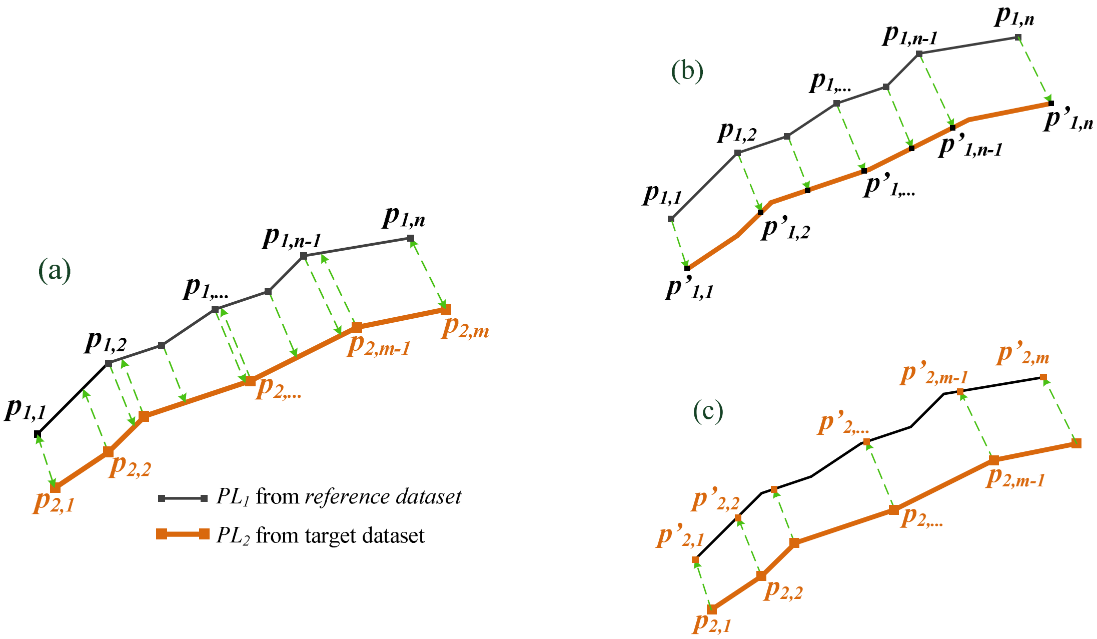


Figure 3.19 The 'average distance' between the *PL₁* and *PL₂*

The geometry constraints can thus be established in a standard way, the 'average distance' between *PL₁* and *PL₂* has been normalized by Equation [3-9], where the tolerance $d_{AV\,tolerance}$ is given empirically.

$$N(d_{AV}) = \frac{d_{AV}(PL_1, PL_2)}{d_{AV\,tolerance}} \qquad \text{...[3-9]}$$

- **Exclusion criteria**

With the availability of the variables of $N(\Delta\theta)$, $N(\Delta l)$, $N(\Delta AvS)$, $N(L_p)$ and $N(d_{AV})$, the geometric constraints can be defined as:

$$M(N) < M(I), \qquad \text{...[3-10]}$$

where $M(N) = [N(\Delta\theta), N(\Delta l), N(\Delta AvS), N(L_p), N(d_{AV})]^T$ and $M(I) = [1,...,1]^T \in R^{1\times 5}$. The potential Delimited Stroke matching pairs fitting for all of the criteria in Expression [3-10] will be confirmed as the promising matching pairs, while others will be rejected.

3.4.3 Exactness prove of promising matching pairs

In some cases, more than one potential matching pair could pass the exclusion criteria in. step 3.4.2 and each of them consists of two matched polylines with *m* to *n* relationship ($m \geq 1, n \geq 1, m \in N, n \in N$), *denoted as PL₁*= <*DS*$_{1,1}$ *DS*$_{1,2...}$ *DS*$_{1,m}$> and *PL₂*=<*DS*$_{2,1}$ *DS*$_{2,2...}$ *DS*$_{2,n}$>, *where DS*$_{1,i}$ ($1 \leq i \leq m$) and *DS*$_{2,j}$ ($1 \leq j \leq n$) are Delimited Strokes from different datasets. The *PL₁* and *PL₂*, however, often represent an inaccurate correspondence especially when either or both of *PL₁* or *PL₂* are connected to some very short Delimited Strokes.

As shown in Figure 3.20, the Delimited Strokes are circularly formed by ending nodes. Set *A₁*→*B₁* as the initial '*sPL (seed polyline)'* and the exploring matching process described by Section 3.4.1 is likely to be terminated at node C_2 considering that it falls inside the searching area of the terminating spot C_1 of the reference polyline $A_1 \rightarrow C_1$. The polyline PL_1= <$DS_{1,1}(A_1 \rightarrow B_1)$, $DS_{1,2}(B_1 \rightarrow C_1)$> and PL_2=<$DS_{2,1}(A_2 \rightarrow B_2)$, $DS_{2,2}(B_2 \rightarrow C_2)$> will be treated as one promising matching pair if they pass the exclusion criteria defined by Expression [3-10] (ref. Section 3.4.2). However from the geometric and topologic point of view, the polyline <$A_2 \rightarrow B_2 \rightarrow C_2 \rightarrow D_2$> should be the exact match of PL_1 while the polyline <$A_2 \rightarrow B_2 \rightarrow C_2$ > is a little bit too shorter. This case requires a variable to measure the matching accuracy which should reflect the *Similarity* in terms of size, shape, location, orientation as well as the similarity of their starting and end points.

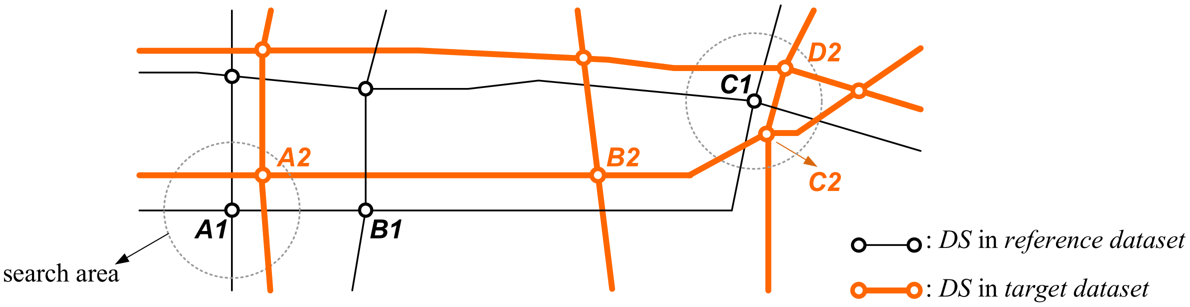


Figure 3.20 The inaccurate matching between polylines <$A_1 \rightarrow B_1 \rightarrow C_1$> and <$A_2 \rightarrow B_2 \rightarrow C_2$>

As depicted by Expression [3-11], the similarity of two points p_1 and p_2, i.e. $Similarity\_p(p_1, p_2)$, is a function of their Euclidian distance $\Delta d(p_1, p_2)$ and topologic difference $\Delta Topo\_p(p_1, p_2)$ which is essentially dependent on the node valences and the angle differences between the emanating edges from the points.

$$Similarity\_p(p_1, p_2) = f\left(\Delta d(p_1, p_2), \Delta Topo\_p(p_1, p_2)\right)^{-1} \qquad \text{...[3-11]}$$

To declare that the topologic difference $\Delta Topo\_p(p_1, p_2)$ defined in this expression is different to that of Expression [3-1] (viz. $\Delta Topo\_N(N_1, N_2)$). $\Delta Topo\_N(N_1, N_2)$ is based on the topologic typification in Section 3.1.2, whist the $\Delta Topo\_p(p_1, p_2)$ is defined as Expression [3-12] where V_1 and V_2 represent the valence of p_1 and p_2 respectively; and $\xi_{penalty}$ is a penalty coefficient which is larger than 0.

$$\Delta Topo\_p(p_1, p_2) = \qquad \text{...[3-12]}$$

$$\begin{cases} \left(\sum_{i=1}^{V} \left| Angle_{\_i}(p_1) - Angle_{\_i}(p_2) \right| \right) \Big/ V, & where\ V = V_1 = V_2 \\ \min\left\{ \sum_{i=1}^{V} \left| Angle_{\_i}(p_1) - Angle_{\_j(i)}(p_2) \right| \right\} \Big/ V + \xi_{penalty}, & where\ V = V_1,\ \forall j(i), i \le j(i) \le V_2, j(i-1) < j(i)\ \left(if V_1 < V_2\right) \\ \min\left\{ \sum_{j=1}^{V} \left| Angle_{\_i(j)}(p_1) - Angle_{\_j}(p_2) \right| \right\} \Big/ V + \xi_{penalty}, & where\ V = V_2,\ \forall i(j), j \le i(j) \le V_1, i(j-1) < i(j)\ \left(if V_1 > V_2\right) \end{cases}$$

Since there are linear objects to be referenced at present, $\Delta Topo\_p(p_1, p_2)$ can be calculated by means of Angle-Index of endpoints (viz. $Angle_{\_i}(p_i)$ *and* $Angle_{\_j}(p_2)$) which are established in Section 3.1.3 (ref. Figure 3.4 and Table 3.4). As compared to $\Delta Topo\_N(N1, N2)$, $\Delta Topo\_p(p_1, p_2)$ reflects the topologic difference of two points more accurately.

Furthermore, the variable $Geo\_Similarity(PL_1, PL_2)$ is defined in Expression [3-13] as the measurement of the geometric similarity between the polylines *PL₁* and *PL₂*. It contains six parameters: $\Delta\theta$, Δl, ΔAvS, L_P, d_{AV}, $\Delta\theta_{FS,FS}$ and $\Delta\theta_{ES,ES}$, whereby $\Delta\theta$, Δl, ΔAvS, L_P and d_{AV} represent the differences of orientation, length, average area, shape and location respectively which are defined in Step 3.4.2 from Equation [3-2] to [3-8]; while $\Delta\theta_{FS,FS}$ is the angle difference between the first line segments of the PL_1 and PL_2; and $\Delta\theta_{ES,ES}$ is the angle difference between the last line segments (Zhang and Meng 2007).

$$Geo\_Similarity(PL_1, PL_2) = f(\Delta\theta, \Delta l, \Delta AvS, L_P, d_{AV}, \Delta\theta_{FS,FS}, \Delta\theta_{ES,ES})^{-1} \qquad \ldots[3\text{-}13]$$

Based on Expression [3-12] and [3-13], the variable $Similarity(PL_1, PL_2)$ is defined in Expression [3-14] to indicate the overall similarity of two polylines, where PL_1_SP and PL_1_EP represent the starting and ending point of PL_1; Likewise, the starting and ending point of the other polyline are represented by PL_2_SP and PL_2_EP; ξ_1, ξ_2 are two empirical coefficients.

$$\begin{aligned} Similarity(PL_1, PL_2) &= \xi_1 \times Geo\_Similarity(PL_1, PL_2) \\ &+ \xi_2 \times \left(Similarity\_p(PL_1\_SP, PL_2\_SP) + Similarity\_p(PL_1\_EP, PL_2\_EP)\right) \ldots[3\text{-}14] \end{aligned}$$

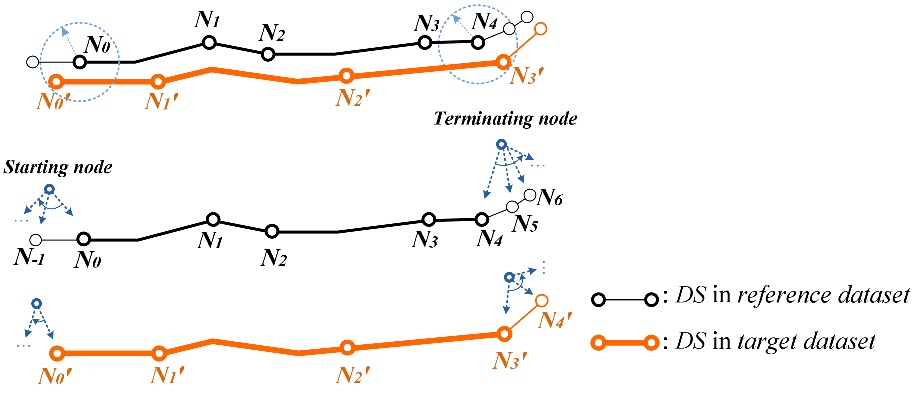


Figure 3.21 Overall process of the exactness inspection of the Delimited Stroke matching pair

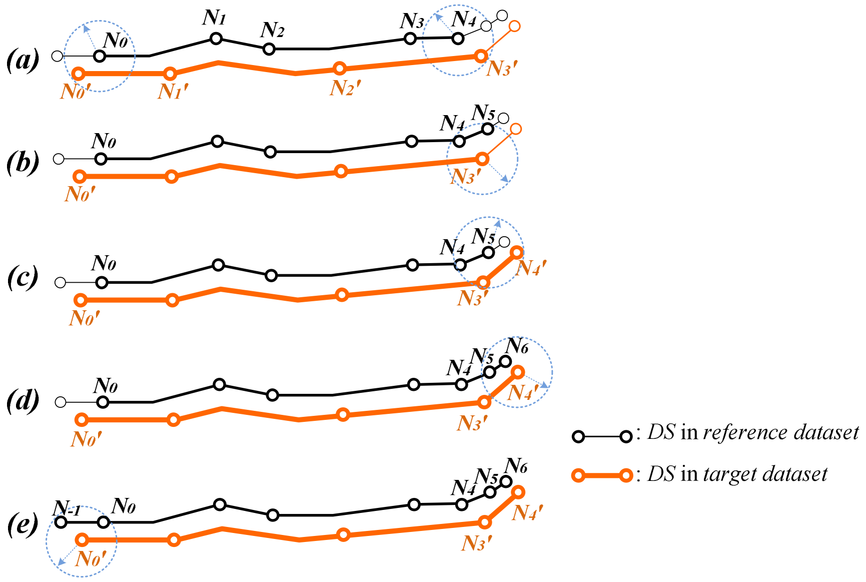


Figure 3.22 Decomposed process of the exactness inspection of the Delimited Stroke matching pair

The $Similarity\_p(p_1, p_2)$, $Geo\_Similarity(PL_1, PL_2)$ and $Similarity(PL_1, PL_2)$ can be scaled to a real number between 0 and 1, with 0 indicating ‘not similar at all’ (entirely wrong match), and 1 ‘a maximal similarity’ (perfect match). With the measurement of $Similarity(PL_1, PL_2)$, it is possible to inspect the exactness of each promising matching pair. The procedure of exactness inspection can be illustrated by the example depicted in Figure 3.21 and Figure 3.22.

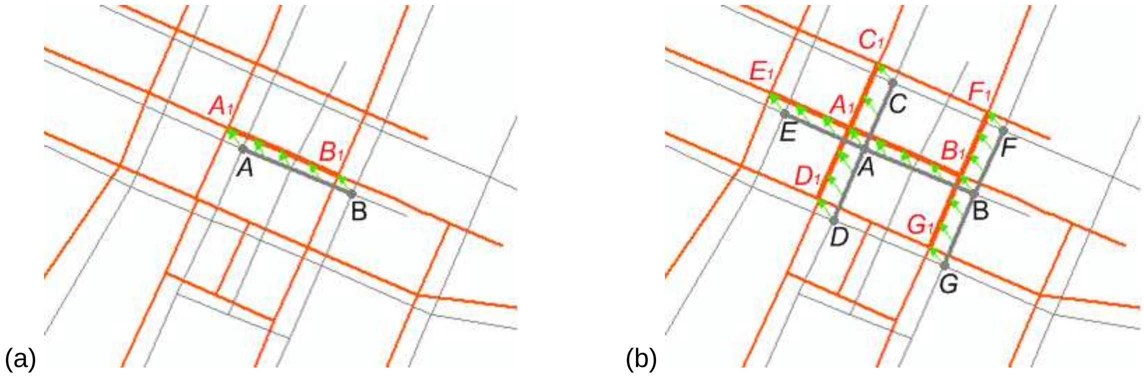
In this example, the initial promising matching pair is constituted by *PL_1= <DS_1 DS_2 DS_3 DS_4>* (viz. polyline *$N_0 \rightarrow N_4$*) and *PL_2= <DS'_1 DS'_2 DS'_3>* (viz. polyline *$N_0' \rightarrow N_3'$*), where *DS_i* represent a Delimited Stroke from reference dataset which is terminated by node *$N_{i-1} \rightarrow N_i$*; and *DS'_i* is from target dataset and terminated by node *$N_{i-1}' \rightarrow N_i'$*. The inspection process begins with the initialization of “*PL_1=$N_0 \rightarrow N_4$* and *PL_2=$N_0' \rightarrow N_3'$*” (see Figure 3.22-a) and the current exact matching result indicated by the variable *‘actual match’*

The polyline *PL_1* needs a further treatment since it is shorter than *PL_2*. Starting from the end spot of node *N_4*, the *PL_1* will be prolonged by progressively connecting short Delimited Strokes from the reference dataset which fall inside the search area of the end spot of *PL_2* (viz. node *N_3'*). The *PL_1=$N_0 \rightarrow N_4$* will be replaced by the prolonged polyline *$N_0 \rightarrow N_5$* as the latter reveals a larger $Similarity(PL_1, PL_2)$ (ref. Expression [3-14]), which will lead to a new *‘actual match’* constituted by *“PL_1 =<DS_1…DS_4 DS_5> (=$N_0 \rightarrow N_5$)* and *PL_2=<DS'_1 DS'_2 DS'_3> (=$N_0' \rightarrow N_3'$)”* (see Figure 3.22-b). Otherwise, it remains unchanged. Then, a similar progressively prolonging procedure will be conducted from the end spot of *PL_2 - node N_3'*. As the result, the *PL_2* will be prolonged to node *N_4'* considering that *PL_2= $N_0' \rightarrow N_4'$* has a larger $Similarity$ to *PL_1=$N_0 \rightarrow N_5$* than polyline *PL_2=$N_0' \rightarrow N_3'$* (see Figure 3.22-c). This prolonging procedure stops when (a) there is no further Delimited Stroke falling inside the search area of the end spot or (b) the maximum $Similarity$ has been reached (see Figure 3.22-d). Symmetrically, the prolonging procedure could be also conducted by progressively connecting the short Delimited Strokes from the other end of the matching pair (see the end spots of *N_0* and *N_0'* in Figure 3.22-e) and followed by a $Similarity$ comparison.

For the case shown in Figure 3.22, the matching pair of *PL_1= <DS_1…DS_3 DS_4>* (*$N_0 \rightarrow N_4$*) and *PL_2 = <DS'_1 DS'_2 DS'_3>* (*$N_0' \rightarrow N_3'$*) will be replaced by *PL_1= <DS_0…DS_5 DS_6>* (*$N_{-1} \rightarrow N_6$*) and *PL_2 = <$DS'_{1...}$$DS'_3$ DS'_4>* (*$N_0' \rightarrow N_4'$*) after the exactness inspection.

3.4.4 Network-based selection

The process of exclusion (section 3.4.2) and exactness inspection (section 3.4.3) may lead to nearly optimal, yet not necessarily a unique matching solution. In order to pick up one final matching among multiple solutions, a network-based selection has been suggested in the proposed matching algorithm.



black: reference dataset orange: target dataset Green arrows: links between the DS pairs

Figure 3.23 The schematic diagram of network-based matching

The concept of the network-based matching is illustrated in Figure 3.23. With the constructed *graph* in Section 3.2, the conjoint Delimited Strokes can be easily detected and treated together. For example, after identifying the corresponding pair *A→B* and $A_1 \rightarrow B_1$ in Figure 3.23-a, three further pairs can be detected from the matched nodes *A* and A_1, which are *A→C* and $A_1 \rightarrow C_1$, *A→D* and $A_1 \rightarrow D_1$, and "*A→E* and $A_1 \rightarrow E_1$. Likewise, the matched nodes *B* and B_1 can lead to two further pairs *B→F* and $B_1 \rightarrow F_1$, *B→G* and $B_1 \rightarrow G_1$. In this case, polyline *A→B* and its conjoint polylines *A→C*, *A→D*, *A→E*, *B→F* and *B→G* from the reference dataset are treated as an integral unit - a *network*, which enables the network-based matching between *C-E-D-A-B-F-G* and C_1-E_1-D_1-A_1-B_1-F_1-G_1 in Figure 3.23-b.

In this process, a *network* is regarded as a series of oriented Delimited Strokes linked by intersection points. Identical networks have the same number of Delimited Strokes and intersections. To assess the similarity between networks we tie a weight to each Delimited Stroke in the network according to its importance in the network which is generally commensurate to its length and topology. Then, the similarity of two networks is given by Expression [3-15] in the DSO algorithm, which is essentially based on the matching *Similarity* of all the involved Delimited Stroke pairs.

$$NetW\_Similarity(NW_1, NW_2) = \qquad \ldots[3\text{-}15]$$

$$Similarity(NW_1\_PL_1, NW_2\_PL_1) + f_\xi(n) \cdot \frac{\sum_{i=2}^{n} Length_i\|\ldots\| \cdot Similarity(NW_1\_PL_i, NW_2\_PL_i)}{\sum_{i=1}^{n} Length_i\|\ldots\|}$$

Knowing that $NetW\_Similarity(NW_1, NW_2)$ represents the similarity of two networks NW_1 and NW_2; n means that each network is composed of n Delimited Stroke pairs; each pair consists of two polylines $NW_1\_PL_i$ *and* $NW_2\_PL_i$), $i=1$ represent the initial Delimited Stroke matching pair and $i= 2,\ldots n$ represent further matching pairs; the variable $Similarity(PL_1, PL_2)$ is defined in Expression [3-14] to indicate the overall similarity of two polylines from different datasets; $Similarity\ (NW_1\_PL_i, NW_2\_PL_i)$ is defined by Expression [3-14], which can indicate the overall similarity between the polylines $NW_1\_PL_i$ *and* $NW_2\_PL_i$; $Length_i\|\ldots\|$ is a norm to calculate the length summation of polylines $NW_1\_PL_i$, $NW_2\_PL_i$; moreover, the $f_\xi(n)$ is a significant penalty function that affects the evaluation of the similarity between two different networks.

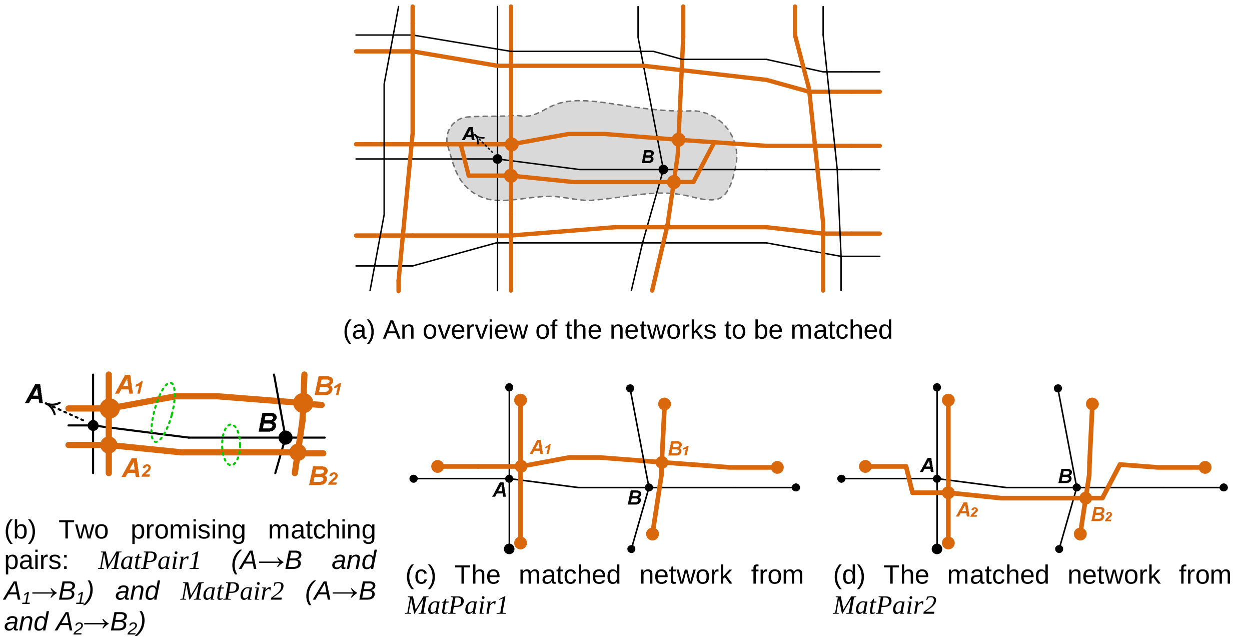
As one larger network often bears more context information than smaller ones, the $f_\xi(n)$ should be a function which has a positive association to the size of the networks. In other words, the more the further matched Delimited Stroke pairs involved, the more reliable the matching result would be. Therefore, the function $f_\xi(n)$ can be empirically defined in [3-16],

$$f_\xi(n) = \xi_{NetW} \cdot (n-1)^P \quad (0 < \xi_{NetW} < 1,\ 0 < P < 1) \qquad \ldots[3\text{-}16]$$

where $n-1$ represent the number of the further matching pairs; ξ_{New} and P are two coefficients with values between 0 and 1.

As illustrated earlier, every promising Delimited Stroke pair can lead to a pair of matched networks in this process. In case that two or more promising matching pairs are identified after step 3.4.1~3.4.3, multiple matched networks will be achieved. Among them, the matched networks with the largest $NetW\_Similarity(NW_1,NW_2)$ will be regarded as the best match, whilst the others are rejected. The network-based selection allows the consideration of more context information. Accordingly, the matching results tend to be more robust than those from context-free matching. In Figure 3.24, starting from the seed polyline *A→B*, two promising Delimited Stroke matching pairs of *MatPair1* and

$MatPair2$ are identified (see Figure 3.24-b). Hereinto the $MatPair2$ constituted by *A*→*B* and $A_2 \rightarrow B_2$ reveals larger *Similarity* than the $MatPair1$ of *A*→*B and* $A_1 \rightarrow B_1$, that is $Similarity(A \rightarrow B, A_2 \rightarrow B_2) > Similarity(A \rightarrow B, A_1 \rightarrow B_1)$ (ref. Expression [3-14]), even though the latter one is more likely to be the exact match from a contextual point of view. Taking advantage of the network-based selection, the *MatPair1* is indeed confirmed as the '*best match'* while the $MatPair2$ is rejected considering that the matched network generated from $MatPair1$ (see Figure 3.24-c) has a larger $NetW\_Similarity$ than that from $MatPair2$ (see Figure 3.24-d).



(a) An overview of the networks to be matched

(b) Two promising matching pairs: $MatPair1$ (*A*→*B* and $A_1 \rightarrow B_1$) and $MatPair2$ (*A*→*B* and $A_2 \rightarrow B_2$)

(c) The matched network from $MatPair1$

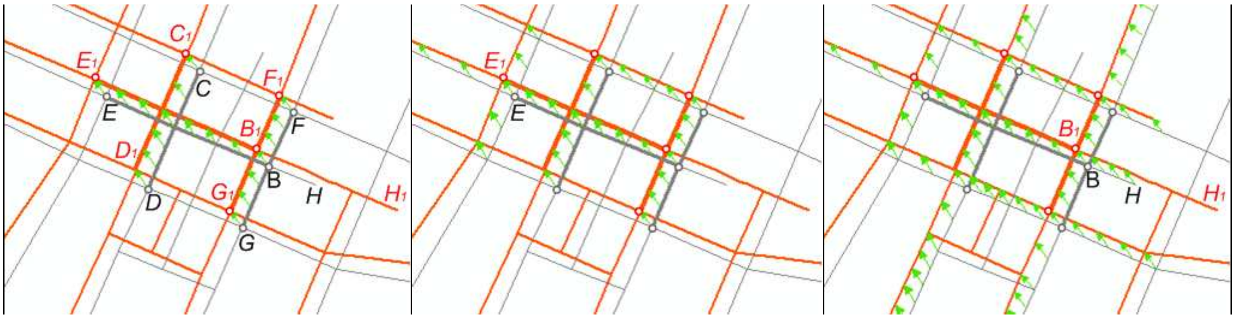
(d) The matched network from $MatPair2$

black: reference dataset orange: target dataset

Figure 3.24 An example of network-based selection

3.4.5 Matching-growing from the seeds

The node pairs on twigs of the matched networks, such as *B* and B_1, *C* and C_1, *D* and D_1, *E* and E_1, *F* and F_1, *G* and G_1 in Figure 3.25-a, can be further treated as seeds for matching-growing. Starting from each seed, the matching grows step by step. In Figure 3.25, the matching grows firstly from the seed pair *E* and E_1, to three further Delimited Stroke matching pairs (see Figure 3.25-b). If the new matched pairs reveal sufficiently similar geometric and topologic characteristics, the growing will go on, otherwise it will be terminated. Likewise, the growing process can be operated on *B* and B_1, leading to partial correspondence between *B*→*H* and $B_1 \rightarrow H_1$ in Figure 3.25-c.



(a) (b) (c)

black: reference dataset orange: target dataset green arrows: links between the Delimited Stroke pairs

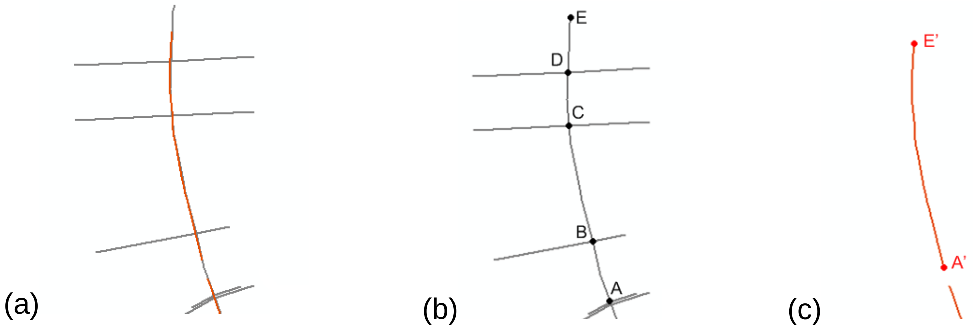
Figure 3.25 Matching growing from seeds

In Section 2.3.1, the partial correspondences have been generally categorized into three groups, viz. the matching pairs with (a) extended, (b) contained and (c) lapped relationship (ref. Figure 2.8). As illustrated in Figure 3.25-c, the conducted matching growing process in this section is able to deal with the partial corresponding road objects in group (a), i.e. with extended relationships (ref. Figure 2.8-a). The partial corresponding objects in group (b) and (c) (ref. Figure 2.8-a and -b), however, needs further treatments as described in Section 3.5.

It should be noted that the correctness of the matching-growing strongly depends on the correctness of matched networks. Wrongly matched networks could spread errors with the growing of matching. To avoid such unfavourable cases, the matching growing process will be triggered when (1) only one Delimited Stroke matching pair passes the exclusion criteria described in Section 3.4.2; and (2) the $NetW\_Similarity$ calculated by Expression [3-15] exceeds an empirical limit.

3.5 Treatment of fragmental matching areas

Section 3.3 ~ 3.4 are dedicated to an operational algorithm for the matching between different line-networks. Being supported by the extendable Delimited Strokes and network-based matching, the algorithm is able to consider the geometric and topologic information in a large context environment, hence reveal a high matching performance. However, in the area where the network is quite fragmentized, it might be very hard for this algorithm to correlate the linear objects from different datasets which are partially corresponding to each other since the fragmentized road objects provide insufficient topologic information in many cases. For example, in Figure 3.26, the road *A→E* from the reference dataset is constituted by four Delimited Strokes: *A→B, B→C, C→D & D→E* (see Figure 3.26-b). Its nearest road *A'→E'* in the target dataset is composed of only one Delimited Stroke *A'→E'* in Figure 3.26-c. However between "*A→B, B→C, C→D & D→E*" and "*A'→E'* ", it is impossible to identify any corresponding pair(s) with *1:1*, *1:N*, *M:1* or *M:N (M>1, N>1)* relationship since the polyline *A→E* is much longer (ca. 210 meters longer in this particular case of Figure 3.26) than *A'→E'*.



black: reference dataset orange: target dataset

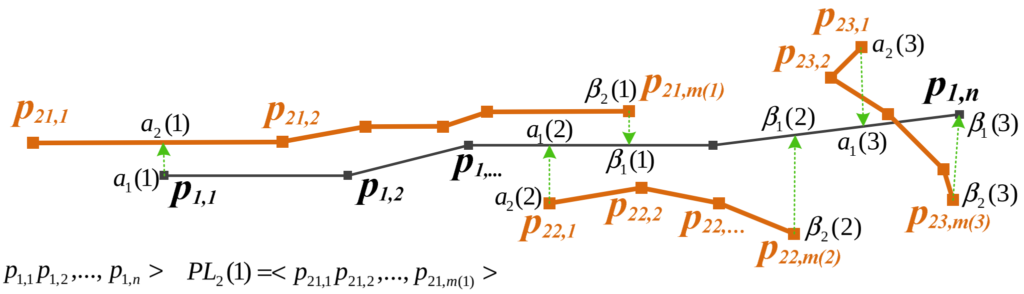
Figure 3.26 An example of fragmentized road objects

In order to solve this problem, the proposed matching algorithm suggests a four-step process to deal with the unmatched fragmentized Delimited Strokes. Worthwhile to mention is that after the matching iteration from Section 3.3 to 3.4 (ref. Figure 3.1), the Delimited Strokes are line objects corresponding to level 3 in Table 3.5.

Step 1: Instantiation of the reference polyline

Following the principle of 'good continuity' in Gestalt psychology (Zhang and Meng 2007), a sequence of line objects from the reference dataset, which have not been matched by the former processes, is chained together and acts as the matching reference, denoted as PL_1=<$p_{1,1}$ $p_{1,2...}$ $p_{1,n}$> (c.f. Figure 3.27).

In general the angle between the connected line segments can be treated as the dominant factor in the good continuity principle; however, if there are meaningful attributes available, e.g. street name, width, amount of the lane, etc., the thematic information can also play a significant role. It should be noted that neither too long nor too short matching reference is expected in this step, because too short polylines carry little geometrically characteristic information and too long ones may result in multiple incorrect matching suggestions. Moreover, the anticipated reference polyline should be delimited by the intersection (*Valence* ≥ *3*) or dead-end (*Valence* = *1*) if possible.



$PL_1 =< p_{1,1}p_{1,2},\dots, p_{1,n} >$ $PL_2(1) =< p_{21,1}p_{21,2},\dots, p_{21,m(1)} >$

$PL_2(2) =< p_{22,1}p_{22,2},\dots, p_{22,m(2)} >$ $PL_2(3) =< p_{23,1}p_{23,2},\dots,p_{23,m(3)} >$

black: reference dataset orange: target dataset

Figure 3.27 Matching of the fragmentized linear objects

Step 2: Identification of candidate polylines

All line objects from the target dataset that are close to PL_1 are selected. These line objects can also be chained to generate one or more polylines which are regarded as potential matching candidates (c.f. polylines $PL_2(1)$, $PL_2(2)$ and $PL_2(3)$ in Figure 3.27), denoted as:

$$U(PL_2) = \{PL_2(k) \mid 1 \le k \le K, k \in N\} \qquad \dots[3\text{-}17]$$

where *K* represents the number of candidate polylines and $PL_2(k)$ can be denoted as <$p_{2k,1}$ $p_{2k,2\dots}$ $p_{2k,m(k)}$> which fulfils the criterion of:

$$\min\{\Delta d(p_{1,i}, l_{2k,j}), \forall i,\ 1 \le i \le n\} < \Delta d_T,\quad j = 1,2,3,\dots,m(k)-1 \qquad \dots[3\text{-}18]$$

where $p_{1,i}$ is one of the vertices of PL_1; $l_{2k,j}$ is the line segment $p_{2k,j}p_{2k,j+1}$ of $PL_2(k)$; $\Delta d(p_{1,i}, l_{2k,j})$ represents the distance between $p_{1,i}$ and $l_{2k,j}$, and Δd_T is a predefined tolerance value.

Step 3: Decomposition to constitute potential matching pairs

As the $m:n\ (m \ge 1, n \ge 1, m, n \in N)$ matchings have been already identified in the former processes, the matching reference PL_1 and the candidate $PL_2(k) \in U(PL_2)$ tend to be partially corresponded. After the calculation of the overlapping part between PL_1 and each $PL_2(k) \in U(PL_2)$, the reference and candidate polylines will be decomposed to constitute a set of potential matching pairs represented by:

$$U(\tilde{PL_1}, \tilde{PL_2}) = \{(PL_{1\alpha_1(k)}^{\beta_1(k)}, PL_2(k)_{\alpha_2(k)}^{\beta_2(k)}) \mid k \in N, 1 \le k \le K,\ 0 \le \alpha_1(k) < \beta_1(k) \le 1,\ 0 \le \alpha_2(k) < \beta_2(k) \le 1\}$$

…[3-19]

Knowing that *K* is the number of candidate polylines; $PL_{1\alpha_1(k)}^{\beta_1(k)}$ represents a portion of candidate polyline PL_1 from the position $\alpha_1(k)$ to $\beta_1(k)$, while $PL_2(k)_{\alpha_2(k)}^{\beta_2(k)}$ is a portion of candidate polyline

$PL_2(k)$ from $\alpha_2(k)$ to $\beta_2(k)$. In the example illustrated in Figure 3.27, three potential matching pairs are identified, viz. $\left(PL_{1\alpha_1(1)}^{\beta_1(1)}, PL_2(1)_{\alpha_2(1)}^{\beta_2(1)}\right)$, $\left(PL_{1\alpha_1(2)}^{\beta_1(2)}, PL_2(2)_{\alpha_2(2)}^{\beta_2(2)}\right)$ and $\left(PL_{1\alpha_1(3)}^{\beta_1(3)}, PL_2(3)_{\alpha_2(3)}^{\beta_2(3)}\right)$.

Step 4: Exclusion of incorrect matches

The set of $U(\widetilde{PL}_1, \widetilde{PL}_2)$ contains both correct match and wrong suggestions. By employing the exclusion criteria defined in Section 3.4.2 (ref. Expression [3-10]), some incorrect matches can be removed from the set of $U(\widetilde{PL}_1, \widetilde{PL}_2)$, e.g. $\left(PL_{1\alpha_1(1)}^{\beta_1(1)}, PL_2(1)_{\alpha_2(1)}^{\beta_2(1)}\right)$ in Figure 3.27. If all remaining matching pairs in $U(\widetilde{PL}_1, \widetilde{PL}_2)$ correspond to different portions of the reference polyline PL_1, then they will be confirmed as correct match. Otherwise, if the matching pairs $(PL_{1\alpha_1(k')}^{\beta_1(k')}, PL_2(k')_{\alpha_2(k')}^{\beta_2(k')})$ and $(PL_{1\alpha_1(k'')}^{\beta_1(k'')}, PL_2(k'')_{\alpha_2(k'')}^{\beta_2(k'')})$ are overlapped, that is $(\alpha_1(k'), \beta_1(k')) \cap (\alpha_1(k''), \beta_1(k'')) \neq \phi$ where ϕ means the empty set, the matching pair with lower *Similarity* (ref. Expression [3-14]) should be eliminated as false suggestions. In Figure 3.27, for instance, the matching pair constituted by $\left(PL_{1\alpha_1(1)}^{\beta_1(1)}, PL_2(1)_{\alpha_2(1)}^{\beta_2(1)}\right)$ is chosen as the final result since it reveals larger *Similarity* than that of $\left(PL_{1\alpha_1(2)}^{\beta_1(2)}, PL_2(2)_{\alpha_2(2)}^{\beta_2(2)}\right)$.

Following the four-step process depicted above, it is possible to match the linear objects which partially correspond between different datasets. In this way, the overall matching completeness can be dramatically enhanced in the areas where the networks are fragmentized. However, the matching certainty and accuracy is hard to be ensured as the fragmentized objects provide little contextual information. For this reason, this final process is used as an optional component in the whole matching strategy, i.e. this process is not mandatory in practice especially when the matching accuracy and certainty are emphasized.

Chapter 4

Assisting Methodologies for Higher Matching Performance

Being supported by the extendable Delimited Strokes and network-based selection, the DSO algorithm introduced in Chapter 3 is able to consider the geometric and topologic information in a large context environment, hence provide a considerably improved matching performance in terms of computing speed, matching rate and matching certainty. This chapter is dedicated to three assisting methodologies to further refine the matching performance of the proposed DSO algorithm, concerning on (a) how to efficiently handle special cases; (b) how to make use of semantic attributes; and (c) how to enhance the computing speed.

Although the DSO algorithm can be applied to match road networks with high matching rate and certainty, uncertain matching would remain in areas where geometric or topologic conditions are too complex or inconsistent to allow a reliable identification of matching pairs. For some special cases of looping crosses, parallel lines, etc., this algorithm still shows a limited performance, especially when datasets to be matched were captured at different LoDs. For instance, a single polyline may correspond to a pair of dual carriageways in another dataset or a node may correspond to a loop. Adjusting some criteria or parameters in the matching approach does not help much because various types of special matching cases have very different characteristics which may require over-tuning of the available criteria or parameters and cause performance decay for normal cases. Section 4.1 proposes a structure-guided matching strategy to circumvent this problem. The whole road network is classified at first into various groups of structures according to some rules. Each structure type then triggers a specialized matching approach which can create desirable matching results.

Since the DSO algorithm does not rely on any semantic information, it is applicable to various road networks and other types of line networks like waterways or electronic pipelines. However, the matching calculation can be facilitated by semantic attributes if they are available in both datasets to be matched. Section 4.2 is concerned with a semantic-guided matching strategy for road networks. The semantic attributes are grouped into objective and subjective attributes with distinctive functionalities in different matching scenarios.

From the pragmatic point of view, a good matching algorithm should produce high-quality results at a high speed. One efficient way to increase the computing speed is to implement spatial indexes (Xiong 2000; Franklin et al. 1994; Schimandl et al. 2009). For this reason, two grid-based spatial indexes are introduced in Section 4.3. One is for point data and the other for linear data. These two spatial indexes can be embedded in the DSO algorithm and accelerate the process of computing the potential matching pairs between different datasets, especially when huge datasets are concerned.

4.1 Matching approach guided by *'structure'*

A street network can be regarded as a unit constituted by various road structures, such as roundabouts, dual-carriageways (parallel lines), narrow passages, navigation stubbles, slip roads around cloverleaf junctions and normal single carriageways (see examples in Figure 4.1).

The matching strategy guided by *'structure'* is characterized by three consecutive steps: (1) structure recognition, (2) process modelling, and (3) process execution. 'Structure recognition' can be regarded as an activity to describe, classify and identify typical object clusters, termed as 'structures'. During the process modelling, different structure categories trigger different matching algorithms as well as the associated criteria. Process execution as the final step takes place to operate the matching approach. With regard to the general problem of searching scope and matching performance, the 'process execution' is conducted according to a reasonable execution sequence for various structure categories. Since the structures of dual carriageways, roundabouts, narrow passages, navigation stubbles and slip roads have several particular characteristics, their matching processes are triggered ahead of the normal single carriageways. If one structure of dual carriageways, narrow passages, navigation stubbles or slip roads from the reference dataset can not find its corresponding structure from the target dataset, it will be treated as normal single carriageways hereafter and then execute the matching once again, so that more desirable matching results can be achieved.

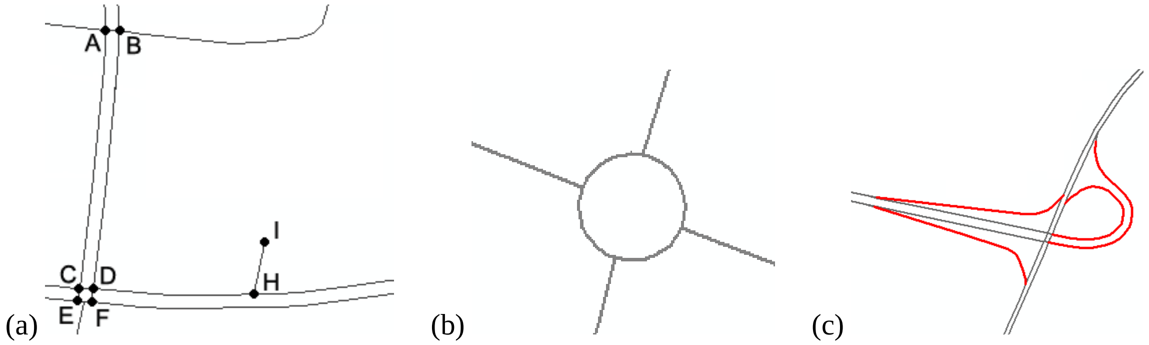


Figure 4.1 Dual carriageways: e.g. parallel lines *A→C→E* & B→*D→F* in (a); narrow passages: e.g. road *A→B* in (a); navigation stubbles: e.g. road *H→I* in (a); roundabouts in (b); slip roads around cloverleaf junctions, see red lines in (c)

The DSO algorithm along with the necessary parameters is an efficient algorithm for the general task of matching single carriageways, incl. narrow passages, navigation stubbles, slip roads around cloverleaf junctions and normal single carriageways. The roundabouts or dual-carriageways, however, take on quite different geometric or topologic characteristics from the single carriageways. Therefore, it is inappropriate to directly employ the DSO matching algorithm. The structure-guided matching offers a complementary strategy by focusing on the two challenging cases that are not considered in many reported matching approaches - dual carriageways (parallel lines) and looping crosses as illustrated in Figure 4.1-a and 4.1-b. Worthwhile to mention is that structure-guided matching strategy can handle the dual carriageways or looping crossings with both similar and distinctive LoDs.

4.1.1 Matching of the roundabouts

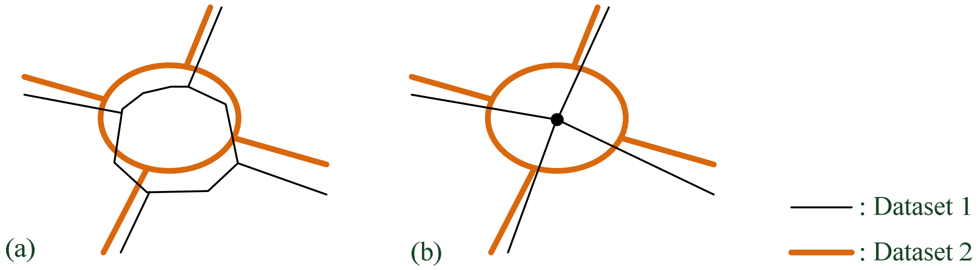


Figure 4.2 Corresponding roundabouts with (a) similar or (b) dissimilar LoDs

The following three-step matching process can be utilized to identify the corresponding roundabouts between different datasets: (a) the corresponding roundabouts with similar LoD, see the black and orange polygons in Figure 4.2-a; and (b) the corresponding roundabouts with different LoDs, e.g. in Figure 4.2-b one road roundabout is represented by a loop in one dataset whereas it is just a node in the other.

Step 1 Recognition of roundabouts

The looping crosses can be detected by comparing the coordinate pairs of a polyline: At first a sequence of line objects sharing certain characteristics (e.g. the principle of ‘good continuity’ in Gestalt psychology) is chained to form a polyline. If two points along the chain claim the same geometric position, they indicated the existence of a roundabout (Zhang et al. 2007).

Step 2 Matching the roundabouts with similar LoDs

As depicted in Figure 4.2, the corresponding roundabouts with similar LoDs are often represented by two polygons in different datasets. The correspondence between homologous polygons can be identified by comparing their geometric characteristics with respect to the area, location and shape.

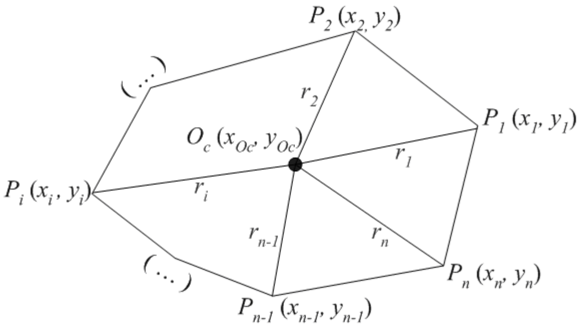


Figure 4.3 The general polygonal look of a looping cross

- ***Area difference***

The area of a polygon with the oriented vertices of $PLg = < p_1 p_2 ... p_n p_{n+1} >$ (see Figure 4.3) can be calculated according to Equation [4-1], where $p_i = (x_i, y_i)$, $p_{n+1} = p_1$ and n represents the number of polygon convexes.

$$S_{PLg} = \frac{1}{2}\sum_{i=1}^{n} p_i \cdot \begin{vmatrix} 0 & 1 \\ -1 & 0 \end{vmatrix} \cdot p_{i+1}{}^{\mathrm{T}} \qquad \text{…[4-1]}$$

Note that the area of the polygon is defined to be positive if the points are arranged in a counter-clockwise order, and negative if they are in clockwise order (Beyer 1987).

Comparing the areas between two polygons from different datasets helps to judge whether these two polygons are corresponding roundabouts or not, see the normalized area difference in Equation [4-2].

$$\hat{N}(\Delta S) = \frac{\Delta S}{\Delta S_{tolerance}} = \frac{\left|S_{PLg_1} - S_{PLg_2}\right|}{\min\begin{cases} \max\begin{cases} (S_{PLg_1} + S_{PLg_2}) \cdot \Delta\tilde{r}(s)_{tolerance} \\ \Delta\tilde{S}_{\min} \end{cases} \\ \Delta\tilde{S}_{\max} \end{cases}} \qquad \text{…[4-2]}$$

Knowing that PLg_1 and PLg_2 represent the roundabouts to be compared; $\Delta\tilde{S}_{\max}$ and $\Delta\tilde{S}_{\min}$ are pre-defined maximal and minimal tolerance values of the area difference between corresponding polygons; and $\Delta\tilde{r}(s)_{tolerance}$ is a pre-defined tolerant ratio divided by the area difference and summation.

- ***Location difference***

The centre of a polygon, denoted as $O_C = (x_{Oc}, y_{Oc})$ in Figure 4.3, is formulated by the average of all the vertices of the polygon, see Equation [4-3] where $PLg =< p_1 p_2 ... p_n p_{n+1} >$ and $p_{n+1} = p_1$.

$$O_C = \frac{1}{n} \cdot \sum_{i=1}^{n} p_i \qquad \text{...[4-3]}$$

Since the O_C represents a reference point of the polygon, the distance between the centre points of the roundabouts to be compared (denoted as PLg_1 and PLg_2) has been utilized to measure their location difference in the proposed matching model. This distance can be calculated by d_{PLg} in Equation [4-4] and then normalized by $N(d_{PLg})$, where $O_{C,PLg1}$ and $O_{C,PLg2}$ represent the centre point of PLg_1 and PLg_2 respectively; and the tolerance value $\tilde{d}_{tolerance}$ is given empirically.

$$\hat{N}(d_{PLg}) = \frac{d_{PLg}}{\tilde{d}_{tolerance}} = \frac{[(O_{C,PLg1} - O_{C,PLg2}) \cdot (O_{C,PLg1} - O_{C,PLg2})^{\mathrm{T}}]^{1/2}}{\tilde{d}_{tolerance}} \qquad \text{...[4-4]}$$

The Location difference $\hat{N}(d_{PLg})$ is a significant parameter for the detection of possible corresponding roundabouts between different datasets.

- ***Shape difference***

The shape of a polygon can be roughly represented by two variables R_{L-PLg} and σ_{r-PLg} defined in Equation [4-5] and [4-6].

$$R_{L-PLg} = \frac{|S_{PLg}|}{l_L}, \qquad \text{...[4-5]}$$

where S_{PLg} is the area of the polygon $PLg =< p_1 p_2 ... p_n p_{n+1} >$ and l_L the length of the longest chord of PLg which can be calculated by $l_L = \max\{[(p_i - p_j) \cdot (p_i - p_j)^{\mathrm{T}}]^{1/2}, \forall i \neq j\}$ $(1 \le i, j \le n, i, j \in N)$.

$$\sigma_{r-PLg} = \frac{\sqrt{(r_i - \bar{r})^2}}{n-1}, \qquad \text{...[4-6]}$$

where $r_i = [(O_C - p_i) \cdot (O_C - p_i)^{\mathrm{T}}]^{1/2}$ $(1 \le i \le n, i \in N)$ is the distance between the centre point O_C and the vertex p_i of a polygon and $\bar{r} = \sum_{i=1}^{n} r_i \Big/ n$ is the average value.

In the proposed matching model, the shape difference between two polygons PLg_1 and PLg_2 can be measured by the normalized values of R_{L-PLg} and σ_{r-PLg} defined in Equation [4-7] and [4-8]. Knowing that $\Delta\tilde{R}_{L\min}$, $\Delta\tilde{R}_{L\max}$, $\Delta\tilde{r}(R_L)_{tolerance}$, $\Delta\tilde{\sigma}_{r\min}$, $\Delta\tilde{\sigma}_{r\max}$ and $\Delta\tilde{r}(\sigma_r)_{tolerance}$ are pre-defined thresholds with respect to the tolerant differences between the corresponding polygons from different datasets; hereinto $\Delta\tilde{R}_{L\min}$, $\Delta\tilde{R}_{L\max}$, $\Delta\tilde{\sigma}_{r\min}$ and $\Delta\tilde{\sigma}_{r\max}$ are in metric space while $\Delta\tilde{r}(R_L)_{tolerance}$ and $\Delta\tilde{r}(\sigma_r)_{tolerance}$ are ratios.

$$\hat{N}(\Delta R_L) = \frac{\Delta R_L}{\Delta R_{L-tolerance}} = \frac{\left|R_{L-PLg_1} - R_{L-PLg_2}\right|}{\min\begin{cases}\max\begin{cases}(R_{L-PLg_1} + R_{L-PLg_2})\cdot\Delta\tilde{r}(R_L)_{tolerance}\\ \Delta\tilde{R}_{L\min}\end{cases}\\ \Delta\tilde{R}_{L\max}\end{cases}} \qquad \ldots[4\text{-}7]$$

$$\hat{N}(\Delta\sigma_r) = \frac{\Delta\sigma_r}{\Delta\sigma_{r-tolerance}} = \frac{\left|\sigma_{r-PLg_1} - \sigma_{r-PLg_2}\right|}{\min\begin{cases}\max\begin{cases}(\sigma_{r-PLg_1} + \sigma_{r-PLg_2})\cdot\Delta\tilde{r}(\sigma_r)_{tolerance}\\ \Delta\tilde{\sigma}_{r\min}\end{cases}\\ \Delta\tilde{\sigma}_{r\max}\end{cases}} \qquad \ldots[4\text{-}8]$$

Based on the characteristics of size (area), location and shape, the exclusion criteria for the roundabouts matching can be defined as:

$$M(\hat{N}) < M(I) \qquad \ldots[4\text{-}9]$$

where $M(\hat{N}) = [\hat{N}(\Delta S), \hat{N}(\Delta d_{PLg}), \hat{N}(\Delta R_L), \hat{N}(\sigma_r)]^T$ and $M(I) = [1,...,1]^T \in R^4$ is a standard unit vector. The roundabout PLg_1 and PLg_2 are confirmed as promising correspondences if they pass the exclusion criteria in Expression [4-9]; otherwise they will be rejected as wrong matching suggestions.

In case that the reference roundabout PLg_1 has more than one corresponding roundabout in the target dataset which passes the exclusion criteria of $M(\hat{N}) < M(I)$, interpreted as $U(PLg_2) = \{PLg_{21}, PLg_{22},..., PLg_{2M}\}$, none of the $PLg_{2i}(1 \le i \le M)$ can be simultaneously regarded as correct counterpart to PLg_1. Two options are available for the confirmation of a unique matching counterpart. One is to weigh the different criteria according to their relative contributions: the matching pair constituted by PLg_1 and $PLg_{2i}(1 \le i \le M)$ with the largest value of weighted sum is regarded as the best matching, whilst all the others are discarded. The weighted sum of measurements can be calculated by:

$$Geo\_Similarity_{PLg} = M(W)\cdot\left[M(I) - M(\hat{N})\right] \qquad \ldots[4\text{-}10]$$

where $M(W) = (W_{\Delta S}, W_{\Delta d}, W_{\Delta R_L}, W_{\Delta\sigma_r})$ is a weight vector given empirically by the user; $M(\hat{N}) = [\hat{N}(\Delta S), \hat{N}(\Delta d_{PLg}), \hat{N}(\Delta R_L), \hat{N}(\sigma_r)]^T$; and $M(I) = [1,...,1]^T \in R^4$. The other option is to stepwise decrease the tolerance value in Equation [4-2], [4-4], [4-7] and [4-8], until only one matching pair remains.

Step 3 Matching of the roundabouts at different LoDs

The same real-world roundabout could have quite different representations in different datasets, e.g. the roundabout depicted in Figure 4.4-a appears as a loop crossing in Dataset 1 (see polygon $< A_1A_2A_3A_1 >$ in Figure 4.4-b) while in the Dataset 2 at a coarser LoD it is generalized as a node (see node *A* in Figure 4.4-c). In this case, matching the corresponding roundabouts has become a task to identify the correspondence between a polygon and a node. Since polygon and node reveal distinct geometric and topologic characteristics, it is impossible to measure their similarities in a straightforward way. Bearing this artefact in mind, we propose a devious matching strategy, which can be elucidated by the example in Figure 4-4.

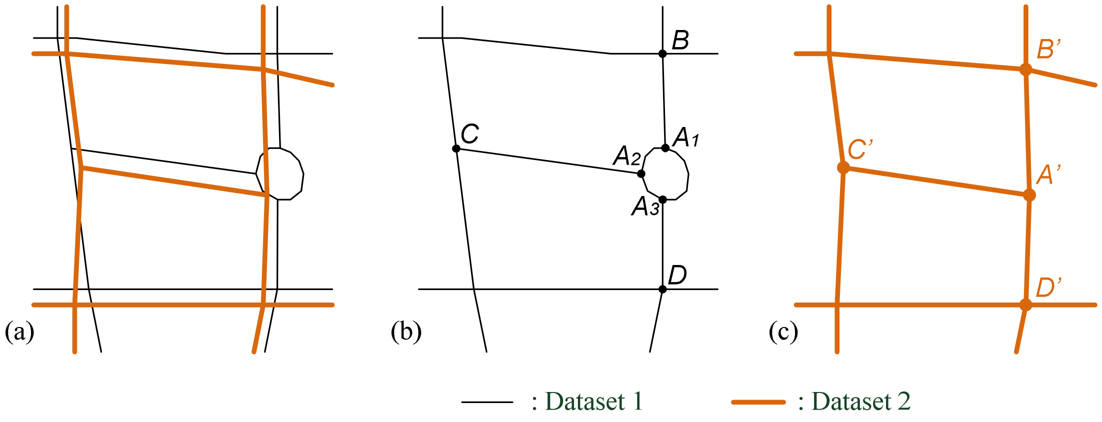


Figure 4.4 Matching roundabouts at different LoDs

In Figure 4-4, the polygon $PLg =< A_1A_2A_3A_1 >$ from Dataset 1 is selected as the 'reference'. In aforementioned matching Step 2, no corresponding polygon in Dataset 2 can be identified as the counterpart of the 'reference' as the roundabout is represented too differently in these two datasets. Therefore, the matching process between polygon and node is triggered. The branches connected to the polygon and the node are compared. As depicted in Figure 4.4-b, the 'reference' polygon has three branches, viz. $A_1 \to B$, $A_2 \to B$ and $A_3 \to B$, which can be interpreted as $U(Branch) = \{Branch_1, Branch_2 ... Branch_n\}$, where n represents the valence of the roundabout and all the $Branch_i (1 \le i \le n, i \in N)$ are polylines. Subsequently, the DSO matching process proposed in Chapter 3 is conducted for each $Branch_i \in U(Branch)$ in turn. The matching result is recorded in a set of $\widehat{U}(Branch) = \{CP_1, CP_2 ... CP_n\}$, where CP_i represents the counterpart of $Branch_i$ in dataset 2. If the DSO algorithm has successfully identified the counterpart for $Branch_i$, the CP_i in dataset 2 is denoted as $\widehat{B}ranch_i$, otherwise the CP_i is a *null set* denoted as ϕ. Two alternatives strategies - a strict and a loose one are possible to identify the correspondence between the polygon and node.

Strict strategy: if every $Branch_i$ of the 'reference' polygon PLg in Dataset 1 can find its counterpart $\widehat{B}ranch_i$ in Dataset 2, viz. $\widehat{U}(Branch) = \{\widehat{B}ranch_1, \widehat{B}ranch_2 ... \widehat{B}ranch_n\}$, and all polylines in $\widehat{U}(Branch)$ intersect at a node A', then the polygon PLg and node A' will be matched together.

Loose strategy: in case that not every $Branch_i$ of the reference polygon PLg in Dataset 1 can find its counterpart $\widehat{B}ranch_i$ in Dataset 2, the set of $\widehat{U}(Branch) = \{CP_1, CP_2 ... CP_n\}$ can be divided into two parts: viz. $\widehat{U}_1(Branch) = \{CP_i |\ CP_i = \widehat{B}ranch_i\}$ and $\widehat{U}_2(Branch) = \{CP_j |\ CP_j = \phi\}$. The polygon PLg and node A' will be confirmed as counterparts if (a) the dimension (size) of $\widehat{U}_2(Branch)$ is much smaller than that of $\widehat{U}_1(Branch)$ and (b) the node A' is shared by all polylines in set $\widehat{U}_1(Branch)$.

The matching strategy with looser criteria can lead to a higher matching rate (completeness) than the strict strategy, while the strict strategy often reveals a higher matching certainty.

4.1.2 Matching of dual carriageways

A dual carriageway is a street in which the two directions of traffic flow are separated by a central barrier or strip of land, known as a central reservation or median. This type of road (e.g. highway) is usually able to carry more traffic than normal single carriageways. As one of the challenging cases that have not yet been considered in many literatures so far, dual carriageways do not occur rarely in large databases. Similar to the classification of roundabouts (ref. Section 4.1.1), the corresponding dual carriageways can also be categorized into two groups according to their representations:

(1) The corresponding dual carriageways reveal similar LoDs, i.e. they are represented by two pairs of parallel lines (one for each direction) in both datasets, see examples in Figure 4.5-a.

(2) The dual carriageways reveal different LoDs. Figure 4.5-b shows an example, where the road *A→B* is a single polyline in Dataset 2 whereas this road is represented by dual carriageways, i.e. a pair of parallel lines in Dataset 1.

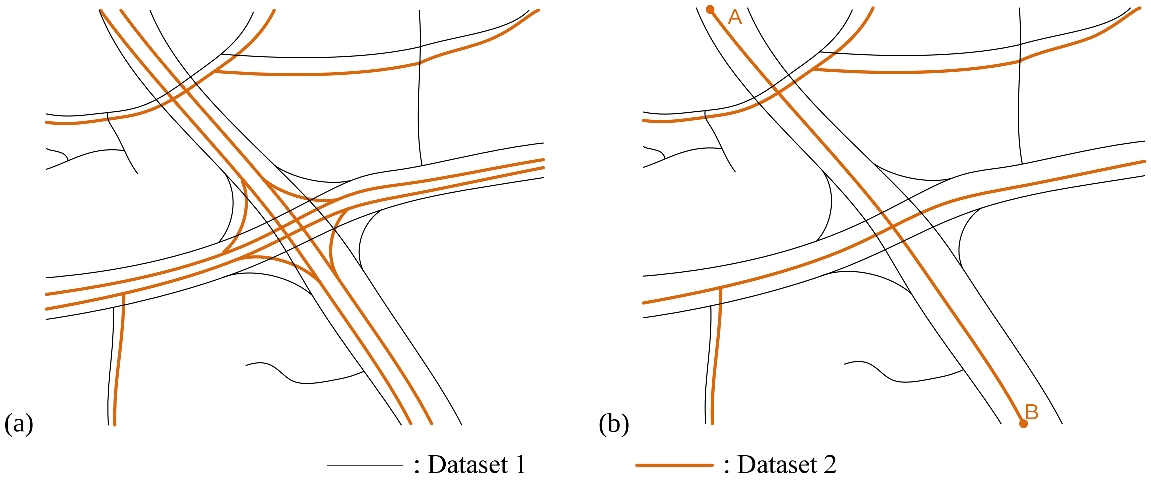


Figure 4.5 Dual-carriageways with (a) similar and (b) dissimilar LoDs in comparable datasets

The DSO matching algorithm can hardly deal with dual carriageways, especially when they are represented in different LoDs in the datasets to be compared. One efficient way to solve this problem is to embed a generalization process of the dual carriageways to the overall matching approach, which is illustrated in Figure 4.6. At first, the dual carriageways in the datasets to be matched are recognized and generalized (collapsed) to their centre lines. The subsequent matching process is then conducted on the centre lines. After the matching, however, the original dual carriageways will replace these temporary centre lines to generate the ultimate matching results.

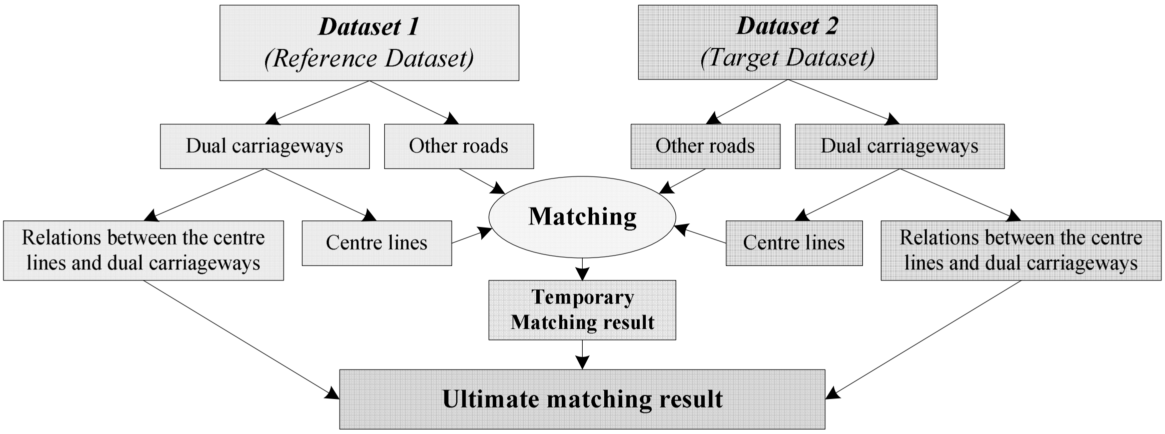


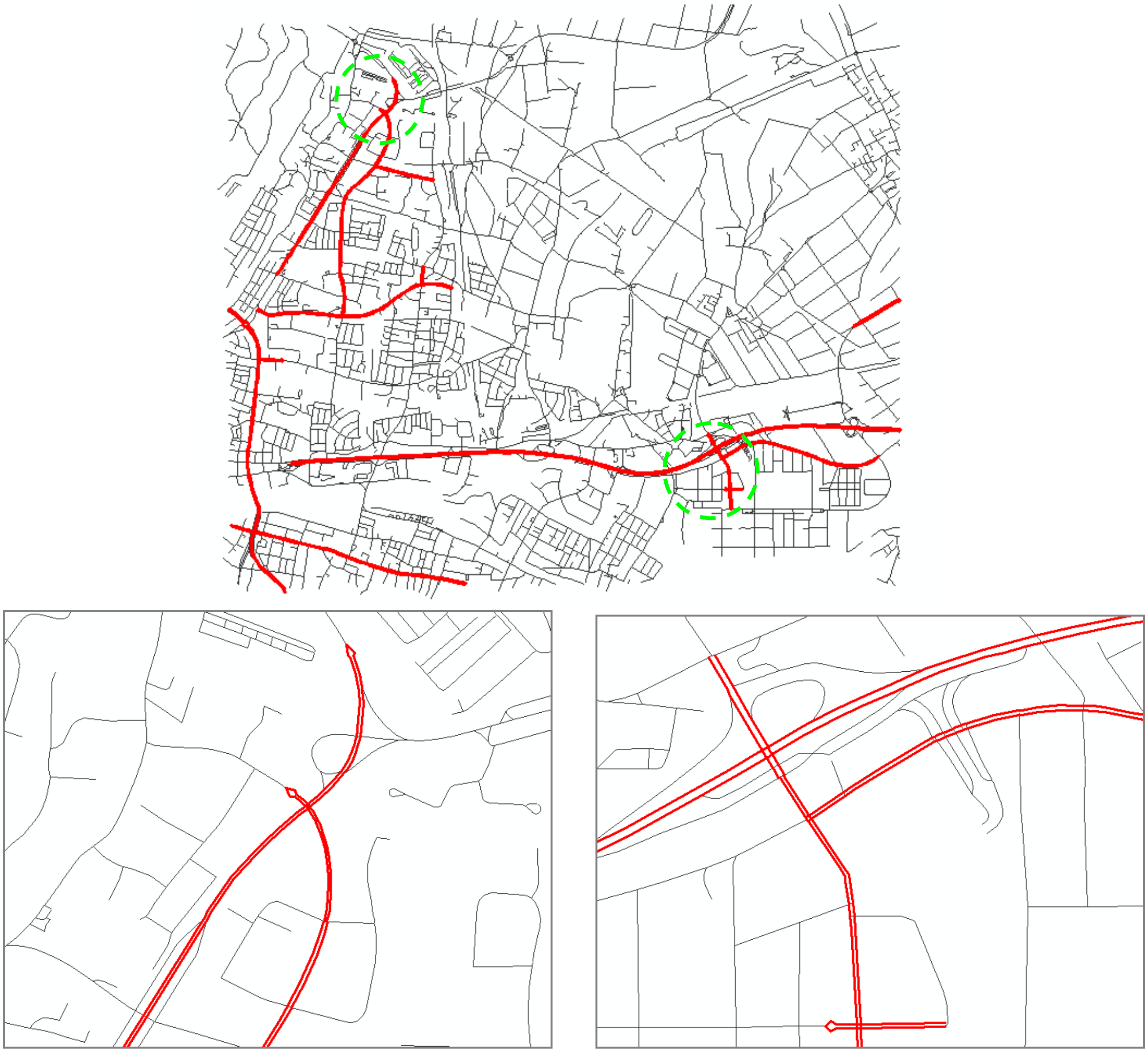
Figure 4.6 The improved matching approach for the dual carriageways

The detailed matching process has to answer the following essential questions:

- How to recognize the dual carriageways (parallel lines)?
- How to collapse the dual carriageways (parallel lines)?
- How to match the dual carriageways with similar LoDs?
- How to match the dual carriageways with dissimilar LoDs?

Step 1: Recognition of the dual-carriageways (parallel lines)

The automatic recognition of parallel roads starts with an exploring procedure in the datasets: If two closely located polylines reveal similar geometric properties, incl. angular, length, maximal chord, etc., and do not intersect, they will be associated to parallel roads and treated as one item. In order to facilitate the subsequent matching approach, each of the anticipated parallel lines should be delimited by the intersection (Valence ≥ 3) or ending node (Valence = 1) of a road. It should be noted that if the exploring procedure has to traverse all road objects, it will unnecessarily consume too much computing time because (a) the recognition of parallel roads is computing intensive and (b) most of the matching references do not belong to the special cases of parallel roads. For these reasons, the recognition process is triggered only under some conditions, e.g. when polylines are short (< 30m) and have their ending nodes with the valence of 3 or 4. The narrow passage roads *A→B*, *C→D*, *E→F, C→E*, and *D→F* in Figure 4.1-a are such polylines.



black lines: normal roads; red lines: recognized dual carriageways

Figure 4.7 Examples of recognized dual carriageways in the dataset of ATKIS*

***** The detailed introduction of the dataset ‘ATKIS’ can be found in Section 5.1.

In the example depicted in Figure 4.7, more than 90% parallel roads are detected with a computing time of 1.7 seconds (CPU: Intel Duo Core 2.0 Hz). The automatic recognition process is independent of any semantic information, therefore can be generically integrated in an on-the-fly matching approach.

Step 2: Collapsing of the dual carriageways

Once the dual carriageways (line pairs) are recognized, their centrelines can be derived. The *skeleton algorithm* is widely used to collapse double lines into a middle axis (Haunert and Sester 2004; Thom 2005; Haunert and Sester 2008). In the context of road-network matching, however, the collapse procedure has to meet the following requirements: (a) the neighbouring features, viz. the roads connecting to the initial dual carriageways, need to be adjusted so that no gaps and no overlaps occur; (b) the topologic relations between the dual carriageways and neighbours need to be preserved; and (c) the neighbours should be modified as little as possible. Since the skeleton algorithm can hardly satisfy all three requirements, the author has proposed another collapsing method.

First, the 'node pairs' are detected. Two closely located nodes which lie on different sides of the recognized dual carriageways and share certain topologic characteristics can be treated as a 'node pair'. In this step, there are five types of 'node pairs' in total, see examples in Table 4.1 where the 'node pair' is denoted as *(N_1, N_2)*. The type (a) ~ (c) can be easily detected since they are connected by narrow passages, while the detection of the type (d) needs to compare their topologic characteristics (ref. Section 3.1.2): N_1 and N_2, which are closely located from different sides of the parallel lines are considered as a pair if (1) the Valences of both N_1 and N_2 are equal to 3; (2) the $Typ_{TopoR=3}$ of N_1 and N_2 are not 0; (3) the difference between the $Angle_{TopoR=3}$ of N_1 and N_2 is approximately ±180°.

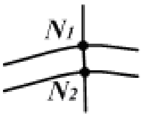
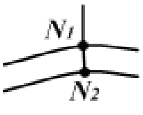
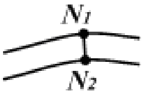
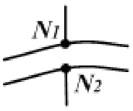
Example				
Type	(a)	(b)	(c)	(d)

Table 4.1 Different types of 'node pairs'

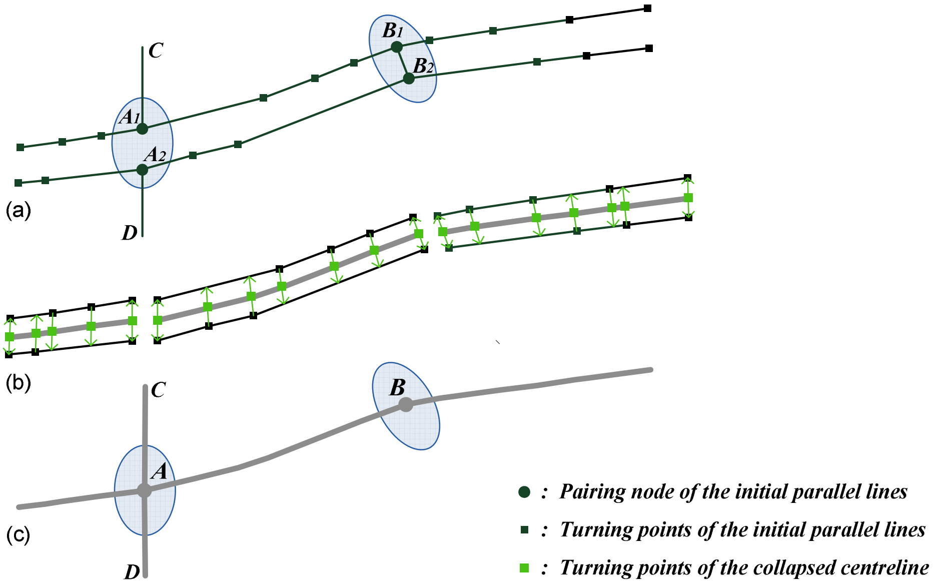

C
B1
B2
A1
A2
(a)
D
(b)
C
B
A
(c)
D
● : Pairing node of the initial parallel lines
▪ : Turning points of the initial parallel lines
▪ : Turning points of the collapsed centreline

Figure 4.8 An example of collapsing dual carriageways

Second, with the identified 'node pairs' as terminating points, the whole dual carriageway can be split into smaller sections, see the example in Figure 4.8-a,b. Each Section is constituted by two parallel polylines, denoted as $Sec_I = (Sec_I\_PL_1, Sec_I\_PL_2)$, where $Sec_I\_PL_1$ and $Sec_I\_PL_2$ are represented by two chains of oriented vertices, viz. $Sec_I\_PL_1 =< p_{1,1}p_{1,2}\cdots p_{1,n} >$ and $Sec_I\_PL_2 =$ $< p_{2,1}p_{2,2}\cdots p_{2,m} >$. By means of interpolation, each vertices $p_{1,i}$ $(1 \le i \le n)$ of $Sec_I\_PL_1$ can find its corresponding position along $Sec_I\_PL_2$, denoted as $p'_{1,i}$, where $p'_{1,1} = p_{2,1}$ and $p'_{1,n} = p_{2,m}$. Likewise, the corresponding position of $p_{2,j}$ $(1 \le j \le m)$ on $Sec_I\_PL_1$, can be also identified, denoted as $p'_{2,j}$. Subsequently, the vertices of $p_{1,i}$ $(1 \le i \le n)$, and $p_{2,j}$ $(1 \le j \le m)$ are aligned to the centre points calculated by $\overline{p}_{1,i} = (p_{1,i} + p'_{1,i})/2$ and $\overline{p}_{2,j} = (p_{2,j} + p'_{2,j})/2$. According to a proper sequence, all of the points $\overline{p}_{1,i}$ $(1 \le i \le n)$ and $\overline{p}_{2,j}$ $(2 \le j \le m-1)$ can be chained together and act as the collapsed centre line of the divided section $Sec_I = (Sec_I\_PL_1, Sec_I\_PL_2)$, denoted as $\tilde{S}ec_I = \overline{P}L_I$, see the example of thick grey lines in Figure 4.8-b.

By connecting all the $\tilde{S}ec_I = \overline{P}L_I$ together, we can get the collapsed centrelines of the dual carriageways, see example in Figure 4.8-c. Along with the collapsing operation, the neighbours of the dual carriageways are also transformed based on the displacement vectors of $p_{1,i} \to \overline{p}_{1,i}$ $(1 \le i \le n)$ and $p_{2,j} \to \overline{p}_{2,j}$ $(1 \le j \le m)$, e.g. the neighbouring roads $C \to A_1$ and $D \to A_2$ in Figure 4.8-a have been respectively extended to $C \to A$ and $D \to A$ in Figure 4.8-c, while the narrow passage $B_1 \to B_2$ is replaced by the node B.

Step 3: Matching of the dual carriageways with similar LoDs

After the generalization in Step 2, two pairs of temporal files are established to record the collapsed dual carriageways from the reference and target dataset: one is constituted by $U_{ref}(Sec_{1,I})$ = $\{Sec_{1,1}, Sec_{1,2}, ..., Sec_{1,N}\}$ and $\overline{U}_{ref}(\tilde{S}ec_{1,I})$ = $\{\tilde{S}ec_{1,1}, \tilde{S}ec_{1,2}, ..., \tilde{S}ec_{1,N}\}$ $(1 \le I \le N)$, where $Sec_{1,I} = (Sec_{1,I}\_PL_1, Sec_{1,I}\_PL_2)$ and $\tilde{S}ec_{1,I} = \overline{P}L_{1,I}$ represent the initial and collapsed dual carriageways from the reference dataset respectively, interpreted as $Sec_{1,I} \Leftrightarrow \tilde{S}ec_{1,I}$; and the other is the pair of $U_{tag}(Sec_{2,J})$= $\{Sec_{2,1}, Sec_{2,2}, ..., Sec_{2,M}\}(1 \le J \le M)$ and $\overline{U}_{tag}(\tilde{S}ec_{2,J})$=$\{\tilde{S}ec_{2,1}, \tilde{S}ec_{2,2}, ..., \tilde{S}ec_{2,M}\}$, where $\tilde{S}ec_{2,J} = \overline{P}L_{2,J}$, $Sec_{2,J} = (Sec_{2,J}\_PL_1, Sec_{2,J}\_PL_2)$ and $Sec_{2,J} \Leftrightarrow \tilde{S}ec_{2,J}$.

To identify the corresponding dual carriageways with similar LoDs between different datasets, the proposed DSO matching algorithm in Chapter 3 is conducted between the sets of $\overline{U}_{ref}(\tilde{S}ec_{1,I})$ and $\overline{U}_{tag}(\tilde{S}ec_{2,I})$, where $\tilde{S}ec_{1,I}$ and $\tilde{S}ec_{2,I}$ act as the basic items (viz. Delimited Strokes) for the matching. As a generic case of $n:m$ $(n \ge 1, m \ge 1)$ matching, the polyline chained by $< \tilde{S}ec_{1,a(1)}, \tilde{S}ec_{1,a(2)} ... \tilde{S}ec_{1,a(n)} >$ $(1 \le a(i) \le N, \forall i \in \{1,2,...,n\})$ from reference dataset could be matched to the polyline $< \tilde{S}ec_{2,\beta(1)}, \tilde{S}ec_{2,\beta(2)} ... \tilde{S}ec_{2,\beta(m)} >$ $(1 \le \beta(j) \le M, \forall j \in \{1,2,..., m\})$ from target datasets. As the ultimate matching result, the $< \tilde{S}ec_{1,a(1)}, \tilde{S}ec_{1,a(2)} ... \tilde{S}ec_{1,a(n)} >$ and $< \tilde{S}ec_{2,\beta(1)}, \tilde{S}ec_{2,\beta(2)} ... \tilde{S}ec_{2,\beta(m)} >$ is replaced by the initial dual carriageways of $< Sec_{1,a(1)}, Sec_{1,a(2)} ... Sec_{1,a(n)} >$and $< Sec_{2,\beta(1)}, Sec_{2,\beta(2)} ... Sec_{2,\beta(m)} >$, which will result in two matched pairs of poylines, viz.:

(a) $< Sec_{1,a(1)}\_PL_{k(1)}, ..., Sec_{1,a(n)}\_PL_{k(n)} >$ and $< Sec_{2,\text{ß}(1)}\_PL_{l(1)}, ..., Sec_{2,\text{ß}(n)}\_PL_{l(n)} >$;

(b) $< Sec_{1,a(1)}\_PL_{k'(1)}, ..., Sec_{1,a(n)}\_PL_{k'(n)} >$ and $< Sec_{2,\text{ß}(1)}\_PL_{l'(1)}, ..., Sec_{2,\text{ß}(n)}\_PL_{l'(n)} >$;

where $k(i), k'(i), l(j), l'(j) \in \{1,\ 2\},\ k(i) \ne k'(i), l(j) \ne l'(j)$, $\forall i \in \{1,2,...n\}$, $\forall j \in \{1,2,...m\}$, and the polygon enclosed by $< Sec_{1,a(1)}\_PL_{k(1)}, ..., Sec_{1,a(n)}\_PL_{k(n)} >$ and $< Sec_{1,a(1)}\_PL_{k'(1)}, ..., Sec_{1,a(n)}\_PL_{k'(n)} >$ has a consistent orientation to that of $< Sec_{2,\text{ß}(1)}\_PL_{l(1)}, ..., Sec_{2,\text{ß}(n)}\_PL_{l(n)} >$ and $< Sec_{2,\text{ß}(1)}\_PL_{l'(1)}, ..., Sec_{2,\text{ß}(n)}\_PL_{l'(n)} >$, for example, both of them are counter-clockwise.

Step 4: Matching of the dual carriageways with dissimilar LoDs

This step deals with the dual carriageways that can not find their counterparts with similar LoDs in the other dataset. After Step 3, the unmatched dual carriageways from reference dataset are denoted as $\hat{U}_{ref}(\hat{Sec}_{1,I}) = \{\hat{Sec}_{1,1}, \hat{Sec}_{1,2}, ..., \hat{Sec}_{1,\hat{N}}\}$ and $\hat{\hat{U}}_{ref}(\hat{\hat{Sec}}_{1,I}) = \{\hat{\hat{Sec}}_{1,1}, \hat{\hat{Sec}}_{1,2}, ..., \hat{\hat{Sec}}_{1,\hat{N}}\}$ $(1 \le I \le \hat{N})$ where $\hat{Sec}_{1,I} = (\hat{Sec}_{1,I}\_PL_1, \hat{Sec}_{1,I}\_PL_2)$ and $\hat{\hat{Sec}}_{1,I} = \hat{\hat{P}}L_{1,I}$ represent the initial and collapsed dual carriageway from the reference dataset respectively, viz. $\hat{Sec}_{1,I} \Leftrightarrow \hat{\hat{Sec}}_{1,I}$; and unmatched dual carriageways from target dataset are denoted as $\hat{U}_{tag}(\hat{Sec}_{2,J}) = \{\hat{Sec}_{2,1}, \hat{Sec}_{2,2}, ..., \hat{Sec}_{2,\hat{M}}\}$ $(1 \le J \le \hat{M})$ and $\hat{\hat{U}}_{tag}(\hat{\hat{Sec}}_{2,J}) = \{\hat{\hat{Sec}}_{2,1}, \hat{\hat{Sec}}_{2,2}, ..., \hat{\hat{Sec}}_{2,\hat{M}}\}$, where $\hat{Sec}_{2,J} = (\hat{Sec}_{2,J}\_PL_1, \hat{Sec}_{2,J}\_PL_2)$, $\hat{\hat{Sec}}_{2,J} = \hat{\hat{P}}L_{2,J}$, and $\hat{Sec}_{2,J} \Leftrightarrow \hat{\hat{Sec}}_{2,J}$.

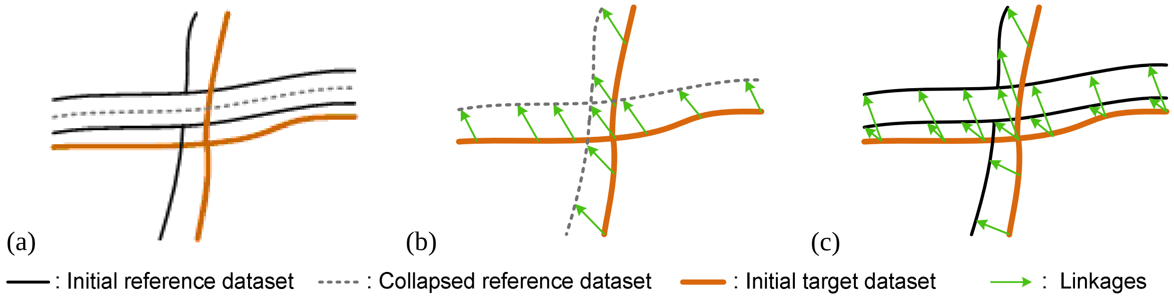


Figure 4.9 Matching of the dual carriageways with dissimilar LoDs

To deal with the unmatched dual carriageways from the reference dataset, the DSO matching process is triggered to calculate the corresponding single polylines between $\hat{\hat{U}}_{ref}(\hat{\hat{Sec}}_{1,I})$ from reference dataset and the entire target dataset. As an example of successful matching, the polylines $\hat{\hat{P}}L_{ref}$ and PL_{tag} could be identified as temporal counterparts between different datasets, where $\hat{\hat{P}}L_{ref}$ is chained by various $\hat{\hat{Sec}}_{1,I}$ from $\hat{\hat{U}}_{ref}(\hat{\hat{Sec}}_{1,I})$ and PL_{tag} is constituted by original road objects from target dataset. Replacing $\hat{\hat{Sec}}_{1,I}$ by $\hat{Sec}_{1,I} = (\hat{Sec}_{1,I}\_PL_1, \hat{Sec}_{1,I}\_PL_2)$, the $\hat{\hat{P}}L_{ref}$ becomes two parallel lines, which can result in an equivalent matching pair (see definition in Section 2.3.1) between different datasets, i.e. a pair of dual carriageways from reference dataset have been successfully matched to a single carriageway from the target dataset, see the example from Figure 4.9-a to 4.9-c. Likewise, the DSO matching algorithm can be conducted between $\hat{\hat{U}}_{tag}(\hat{\hat{Sec}}_{2,J})$ from target dataset and the entire reference dataset.

4.2 Matching guided by *'semantics'*

Besides spatial information, some semantic attributes, such as Functional Road Classification (FRC), street name or number, can be utilized to guide the matching calculation, e.g. it can bring objects together according to their *'semantic similarity'*, where the 'semantic similarity' is an attempt to characterize the degree of semantic proximity of some specific attributes between two objects from different datasets (Devogele 2002); furthermore, in many other cases, the 'semantic similarity' also acts as a 'filter' to ruling out all object types that should not be matched together or as a criterion to select the best matches.

The 'semantic similarity' can be easily determined if the datasets to be matched are based on the same data model, e.g. different datasets captured by same organization, or have similar object structures and attribute definitions. However, if the datasets to be matched have different data structures or attribute definitions, the calculation of the ‘semantic similarity’ becomes much more complicated. In order to properly and adequately utilize the semantic attributes during the matching process, three topics are discussed in this section, concerning (i) Comparison of the objective and subjective semantic attributes; (ii) Utilization of the objective semantic attributes in the matching process; and (iii) Utilization of the subjective semantic attributes in the matching process.

4.2.1 Comparison of the objective and subjective semantic attributes

In general, the semantic attributes of a geospatial object can be categorized into two groups: (a) objective semantic attributes and (b) subjective semantic attributes.

4.2.1.1 Objective semantic attributes

The objective attributes describe the inherent and objectively measurable properties/characteristics of a geospatial object. Since they are not dependent on any data structure or data specification, their attribute fields have consistent definitions in different datasets, i.e. a 1-1 mapping in any known and coherent context is possible.

In the context of road-network matching, such objective semantic attributes incl.:

- *Street name,*
- *Street width,*
- *Number of lanes,*
- *Travel time,*
- *Direction of the traffic flow,*
- *etc.*

Different objective attributes will trigger on different criteria in the process of data matching, see detailed depiction in Section 4.2.2.

4.2.1.2 Subjective semantic attributes

The subjective semantic attributes are often utilized to represent (1) fuzzy characteristics of an object; or (2) artificially introduced properties for some specific applications. In street networks, for instance, the fuzzy subjective semantic attributes can include ‘road object ID’, ‘road type’, ‘road class’ and so on, where the ‘road type’, ‘road class’ describe fuzzy characteristics of one street and the ‘road object ID’ is an artificial number to differentiate various object which does not reflect any realistic street property at all.

Since the artificially introduced properties (e.g. the object ID) seldom bear any logic relation between different datasets, it is difficult to use them for the purpose of data matching, especially when the datasets to be compared are provided by different organizations and therefore captured in accordance with distinct data specifications.

The fuzzy characteristics of a street, e.g. the road type or road class, however, always have certain correspondences among different datasets. The definitions of fuzzy properties obey several rules of abstraction, aggregation or generalization specified in data schema. Usually their attribute fields are not always consistent between different datasets. For this reason, there would be a partial one-to-many, many-to-one and many-to-many mappings between these attribute fields. As a result, their semantic correspondences become much more complex than that of the objective attributes and need to be analyzed beforehand.

4.2.2 Utilization of the objective semantic attributes in the matching process

This section describes the usage of objective semantic attributes street name, street width, number of lanes, travel time and direction of the traffic flow. Street name is a significant attribute for street-network matching. However, a street could have diverse names in different cultures, historic periods and / or language groups. The following paragraphs are focused on the datasets with the same or very similar nomenclature of alphabetic languages.

4.2.2.1 Matching criteria of 'Street Name'

Even if the datasets to be matched originate from the same or very similar nomenclature, the names of the same street are not always strictly equivalent between different datasets due to (a) tiny differences of nomenclatures, e.g. German national nomenclature could be a bit different to European nomenclature; and (b) small errors in data capturing, e.g. the street *'Arcisstrasse'* has been wrongly recorded as *'Arcistrasse'* where one letter-*'s'* is missing. To overcome these problems, the street names have to be normalized and after that the *Levenshtein Distance* (Levenshtein VI, 1965) can be employed to measure their similarities, see details in below:

Step 1: Normalization of street names

At first, the special letters in a language are replaced by letters in the common alphabet of English composed of 26 letters from *a* to *z*, for example, the German umlauts *ä*, *ö*, *ü* and *ß* are replaced by *ae*, *oe*, *ue* and *ss* (see Figure 4.10-b). Then the usual abbreviations are converted back to the complete words, e.g. '*str.*' to '*strasse*' (see Figure 4.10-c) and all of the blanks and "-" in the string are deleted (see Figure 4.10-d). In addition, all lowercases are converted to capital letters (see Figure 4.10-e).

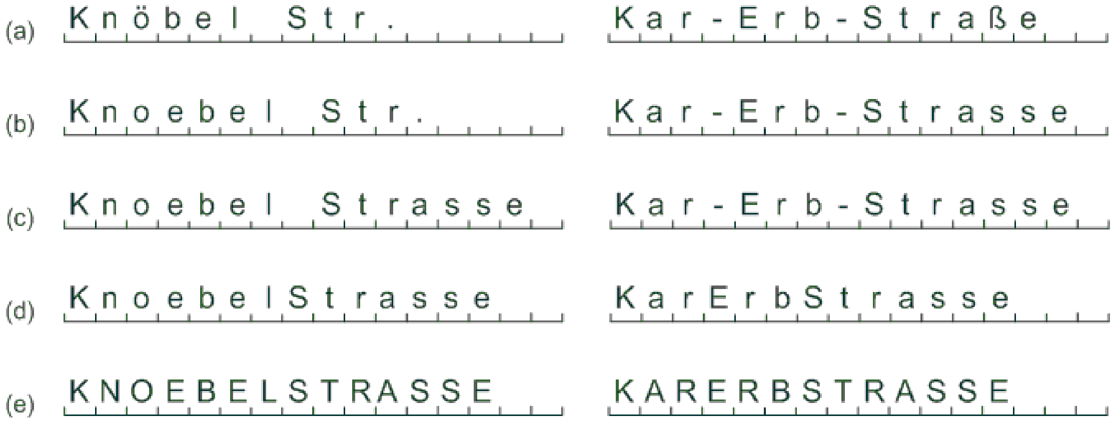

(a) Knöbel Str. Kar-Erb-Straße
(b) Knoebel Str. Kar-Erb-Strasse
(c) Knoebel Strasse Kar-Erb-Strasse
(d) KnoebelStrasse KarErbStrasse
(e) KNOEBELSTRASSE KARERBSTRASSE

Figure 4.10 Normalization of the street name '*Knöbel Str.*' and *'Kar-Erb-Straße'*

Step 2: Comparison of street names with Levenshtein Distance

The Levenshtein Distance, also called Edit-Distance, is a measurement of the similarity between two strings. It is defined as the minimum number of operations to transform one string into the other, where the operations may be deletion of a character, insertion of a character or substitution of a character. Obviously, the greater the Levenshtein Distance, the more different the strings are (Levenshtein 1965; Navarro 2001), for example,

- The Levenshtein Distance between *'KARTO'* and *'KARTO'* is 0, because no transformation is needed;
- The Levenshtein Distance between '*KATO'* and *'KARTO'* is 1, because the letter *'R'* must be inserted in the first string to make it equal to the second;
- The Levenshtein Distance between '*CARTOE'* and *'KARTO'* is 2, because the *'C'* must be substituted by *'K'* and the *'E'* must be deleted.

To compute the Levenshtein Distance, a dynamic programming algorithm can be used. We incrementally fill a bi-dimensional array *D*, where $D_{[i][j]}$ stores the distance between the initial prefix substrings of length *i* and *j* of the first and second string. To compute $D_{[i][j]}$, we need to take the minimum of $D_{[i-1][j-1]}$+*substitutionCost*, $D_{[i][j-1]}$+*insertionCost*, $D_{[i-1][j]}$ + *deletionCost,* where *insertionCost* and *deletionCost* are normally set to 1, and *substitutionCost* is either 0 if the letters in position *i* of the first string and in position *j* of the second string are equal, or 1 if they are not. Thus we can fill the array from left to right and from top to bottom, since in this way we guarantee that we always have previously calculated the values that we need to calculate $D_{[i][j]}$. Once the array is finished, we can read the final Levenshtein Distance between the two strings in its bottom-right corner (Paleo 2007).

		K	A	R	T	O
	0	1	2	3	4	5
C	1	1	2	3	4	5
A	2	2	1	2	3	4
T	3	3	2	2	2	3
O	4	4	3	3	3	2

Table 4.2 Computation of the Levenshtein Distance between *'CATO'* and *'KARTO'*

Table 4.2 shows an example of such a computation: the Levenshtein Distance between *'CATO'* and *'KARTO'* is in the lower right hand corner of the matrix, that is equal to 2. This corresponds to our intuitive realization that *'CATO'* can be transformed into *'KARTO'* by substituting *'C'* for *'K'* and adding *'R'*, one substitution and one insertion = two changes. The complexity of this algorithm is *O(m*n),* both in time and space, where *n* and *m* are the lengths of the strings. The Levenshtein Distance will be never smaller than $|m-n|$.

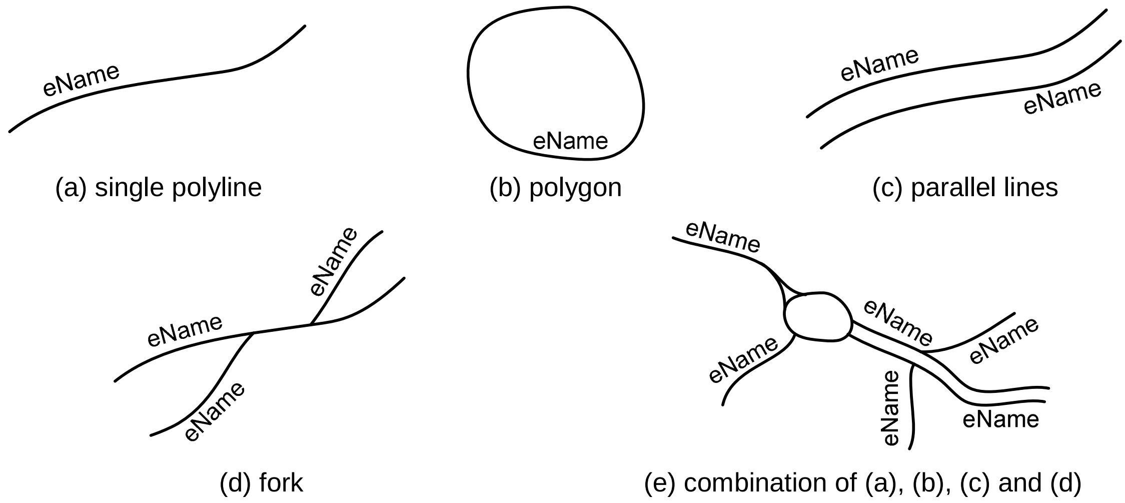
Step 3: Employment of 'Street Name' in different matching scenarios

If the Levenshtein Distance between two street names is below a fixed threshold (e.g. 2), these two strings are considered as equivalent to each other. Using such equivalence is an efficient way to enhance the performance of geometric and topologic matching: if both of the datasets to be matched are fully fulfilled with attributes, the equivalent street names can be implemented as 'identifiers' to compute the corresponding counterparts; the complexity occurs, however, when only one or none of the datasets is fully attributed. In the following, different matching scenarios are discussed in details, where the datasets to be matched are interpreted as $DSet_A$ and $DSet_B$; the $eName$ represents two equivalent street names between different datasets; the cluster chained by the objects with the street name $eName$ is denoted as $wholeStreet_{eName}$; $wholeStreet_{A,eName}$ is in $DSet_A$ and $wholeStreet_{B,eName}$ is in $DSet_B$.

- ***Scenario 1 - both datasets are fully attributed with 'Street Name'***

In this scenario, $eName$ acts as an 'identifier' to calculate the corresponding road objects between different datasets. As depicted in Figure 4.11, besides a single polyline, a street with the name $eName$ could be also formed as polygon, fork, parallel lines or their combinations.

In case that $wholeStreet_{eName}$ is a single polyline or polygon (see Figure 4.11-a & 4.11-b), the $wholeStreet_{A,eName}$ and $wholeStreet_{B,eName}$ from different datasets $DSet_A$ and $DSet_B$, can be directly matched together if they have similar size and location, ref. the geometric criteria of [3-3] and [3-9] in Section 3.4.2 (for polylines), and the Equation [4-2] and [4-4] in Section 4.1.1 (for polygons). The geometric criteria conducted here should be looser than those defined in Section 3.4.2 and 4.1.1, i.e. the tolerance of differences becomes larger.



(a) single polyline (b) polygon (c) parallel lines

(d) fork (e) combination of (a), (b), (c) and (d)

Figure 4.11 Different forms of a street with the name *eName*

If $wholeStreet_{A,eName}$ and $wholeStreet_{B,eName}$ are configured as forks, the corresponding nodes (valence>2 or valence=1) between $wholeStreet_{A,eName}$ and $wholeStreet_{B,eName}$ have to be identified first; then based on the corresponding nodes, $wholeStreet_{A,eName}$ and $wholeStreet_{B,eName}$ can be decomposed into a series of matched single polylines, e.g. the corresponding *wholestreets* in Figure 4.12 can result in three different matching pairs, viz. (1) $N \rightarrow A$ & $N' \rightarrow A'$, (2) $N \rightarrow B$ & $N' \rightarrow B'$, and (3) $N \rightarrow C$ & $N' \rightarrow C'$. The identification of the corresponding nodes can also profit from the available street names. For instance, street names of branches that are incident to the matched nodes can be checked to ensure that they are consistent with each other. If inconsistency is identified, the result of the matched nodes is invalid and additional node pairs can be evaluated (Xiong and Sperling 2004).

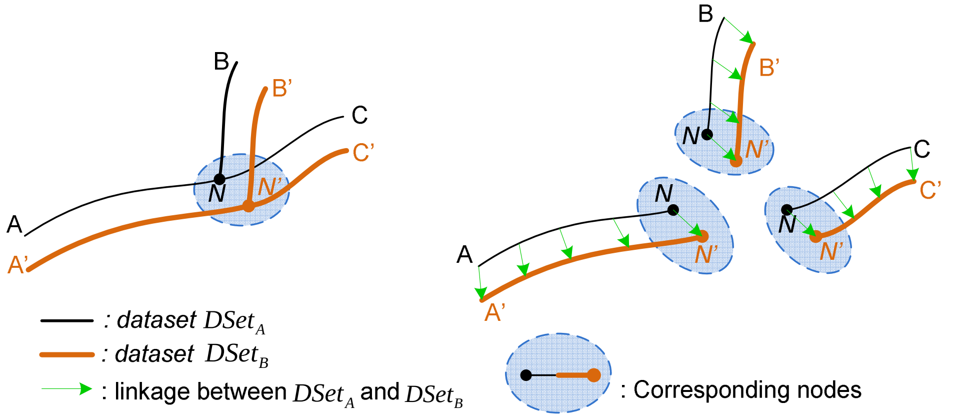


Figure 4.12 Matching of two forks based on the criteria of *Street Name*

The relative positions have to be considered when both $wholeStreet_{A,eName}$ and $wholeStreet_{B,eName}$ are parallel lines, e.g. in Figure 4.13, (a) $A_1 \rightarrow B_1$ & $A_1' \rightarrow B_1'$ and (b) $A_2 \rightarrow B_2$ & $A_2' \rightarrow B_2'$ are defined as two matching pairs, as the polygon $A_1 \rightarrow B_1 \rightarrow B_2 \rightarrow A_2 \rightarrow A_1$ has the consistent orientation to the polylgon $A_1' \rightarrow B_1' \rightarrow B_2' \rightarrow A_2' \rightarrow A_1'$ - both of them are clockwise. However if only one of them is configured as parallel lines while the other is just a single line, the matching concerning on different LoDs will be conducted, ref. Section 4.1.2 - Step 4.

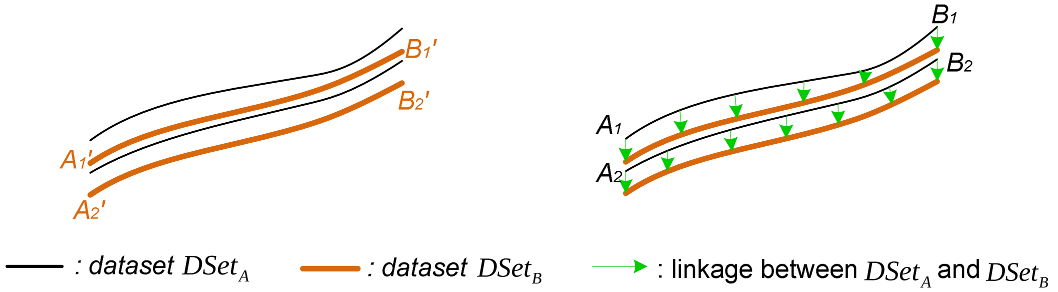


Figure 4.13 Matching of parallel lines based on the criteria of *Street Name*

If $wholeStreet_{A,eName}$ and $wholeStreet_{B,eName}$ are complex combinations as illustrated by Figure 4.11-e, the $wholeStreet_{A,eName}$ and $wholeStreet_{B,eName}$ should be decomposed to a series of single polylines and/or polygons at first, then based on the geometric or topologic matching strategy described in Chapter 3 and Section 4.1, the corresponding road counterparts can be identified.

- ***Scenario 2 - one or two of the datasets is not or only partly attributed with 'Street Name'***

In this scenario, the equivalent street name can not be utilized as 'identifier' for data matching. The reason can be explained by the example in Figure 4.14: where the real-world street *'Memeler Str.'* is represented by *A→B→C in* $DSet_A$ and by *K→M→N* in $DSet_B$. In $DSet_A$, the road segment *A→B* is attributed with the name *'Memeler Str.'* whereas the *B→C* is not; in $DSet_B$, however, the *M→N* has the name while *K→M* does not. Even though the *A→B* and *M→N* have the equitant street name, these two road segments are not homologous to each other because *A→B* and *M→N* represent different parts of the whole street *'Memeler str.'* The street name here is less reliable than the geometric or topologic characteristics like size, shape, location, valence of node etc. In such cases, street name is better to be used as a 'filter' to exclude incorrect matching pairs. If two matched polylines PL_A and PL_B have different street names to each other, i.e. the Levenshtein Distance between overruns a user-defined tolerance value, the matching pair constituted by PL_A and PL_B is unlikely correct and therefore should be excluded or at least treated as uncertain matching result in the post-processing of human interaction. However, this exclusion criterion becomes invalid when either PL_A or PL_B is not fully attributed with street name.

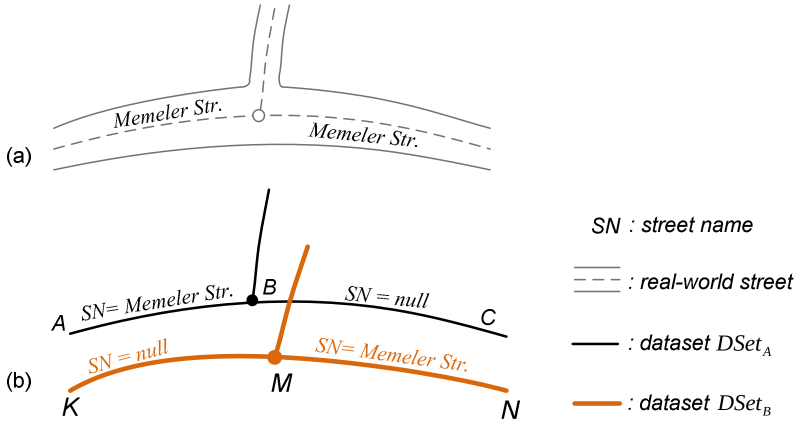


Figure 4.14 Street *'Memeler Str.'* in $DSet_A$ and $DSet_B$

4.2.2.2 Matching criteria of other objective semantic attributes

In the context of road-network matching, the street width, number of lanes, travel time and direction of the traffic flow are four representative semantic attributes besides street name. Given two polylines PL_A and PL_B from different dataset, where PL_A and PL_B are chained by two oriented sets of objects, viz. $<Obj_{A,1}, Obj_{A,2}, ..., Obj_{A,n}>$ and $<Obj_{B,1}, Obj_{B,2}, ..., Obj_{B,m}>$, then their differences with respect to street width, number of lanes, travel time and direction can be measured by Equation [4-11]~[4-14]. As depicted below, such measurements are available only if all of the road objects of $Obj_{A,i}, \forall i \in \{1,2,..n\}$ and $Obj_{B,j}, \forall j \in \{1,2,...,m\}$ are filled with the corresponding attributes.

- ***Difference of street width***

The difference of street width between PL_A and PL_B can be calculated by:

$$\Delta SW = \left| \frac{\sum_{i=1}^{n} sw_{A,i} \cdot l_{A,i}}{\sum_{i=1}^{n} l_{A,i}} - \frac{\sum_{j=1}^{m} sw_{B,j} \cdot l_{B,j}}{\sum_{j=1}^{m} l_{B,j}} \right| \qquad \text{...[4-11]}$$

where $sw_{A,i}$ and $l_{A,i}$ represent the street width and length of the road object $Obj_{A,i}$; $sw_{B,j}$ and $l_{B,j}$ represent the street width and length of $Obj_{B,j}$. The larger the difference of street width (ΔSW) is, the less likely that PL_A and PL_B are homologous.

- ***Difference of number of lanes***

The difference of number of lanes between PL_A and PL_B has a quite similar measurement to street width, see Equation [4-12].

$$\Delta NL = \left| \frac{\sum_{i=1}^{n} nl_{A,i} \cdot l_{A,i}}{\sum_{i=1}^{n} l_{A,i}} - \frac{\sum_{j=1}^{m} nl_{B,j} \cdot l_{B,j}}{\sum_{j=1}^{m} l_{B,j}} \right| \qquad \text{...[4-12]}$$

where $nl_{A,i}$ and $l_{A,i}$, $nl_{B,j}$ and $l_{B,j}$ respectively represent the number of lanes and length of $Obj_{A,i}$ and $Obj_{B,j}$.

- ***Difference of travel time***

The *average speed* can be utilized to measure the travel time difference between PL_A and PL_B, see Equation [4-13]:

$$\Delta AS = \left| \frac{\sum_{i=1}^{n} l_{A,i}}{\sum_{i=1}^{n} tt_{A,i}} - \frac{\sum_{j=1}^{m} l_{B,j}}{\sum_{j=1}^{m} tt_{B,j}} \right| \qquad \text{...[4-13]}$$

where ΔAS is the difference of the average travel speed; $tt_{A,i}$ and $l_{A,i}$, $tt_{B,j}$ and $l_{B,j}$ represent the travel time and length of the objects $Obj_{A,i}$ and $Obj_{B,j}$ respectively.

- ***Difference in direction of traffic flow***

Equation [4-14] defines the Boolean variable ΔDTF to represent the difference in the direction of traffic flow. ΔDTF is either 1 or 0, respectively representing conflict and consistent directions of traffic flow between PL_A and PL_B.

$$\Delta DTF = \begin{cases} 0 & if \quad case\ A \quad or \quad case\ B \\ 1 & if \quad case\ C \quad or \quad case\ D \end{cases} \qquad \text{...[4-14]}$$

Case A: PL_A and PL_B are one-way streets and have consistent directions of traffic flow;
Case B: neither PL_A nor PL_B is one-way street;
Case C: PL_A and PL_B are one-way streets; however, they have inverse directions of traffic flow;
Case D: one of PL_A and PL_B is one-way street while the other is not.

In the matching process, the variables ΔSW , ΔNL , ΔAS and ΔDTF defined in Equation [4-11]~[4-14] are useful as exclusion criteria: if one of these four variables exceeds a user-defined threshold, the matching pair constituted by PL_A and PL_B tend to be wrong and therefore has to be automatically excluded or treated as uncertain matching result; moreover, ΔSW , ΔNL , ΔAS and ΔDTF can also support the computing of $Similarity$ between each identified matching pair (ref. Equation [3-14] in Chapter 3), which helps to make a more proper decision on scrutinizing the matching result.

4.2.3 Utilization of the subjective semantic attributes in the matching process

Assume the datasets to be matched are $DSet_A$ and $DSet_B$. $DSet_A$ consists of m objects denoted as $Set\{Obj_A\} = \{Obj_{A,i} | i = 1,2,...,m\}$ and each object has a number of subjective semantic attributes $Set\{Attr(Obj_{A,i})\} = \{Attr(Obj_{A,i})_p = \alpha_p |\ p = 1,2,..., P\}$; likewise, $DSet_B$ has $Set\{Obj_B\} =$ $\{Obj_{B,j} | j = 1,2,...,n\}$, and each Obj_B has the subjective attributes of $Set\{Attr(Obj_{B,j})\} =$ $\{Attr(Obj_{B,j})_q = \text{ß}_q |\ q = 1,2,...,Q\}$. In general, correspondences between subjective semantic attributes can be classified into two types: equivalence and inclusion.

- **Equivalent correspondence**

Two sets of attribute values from different datasets are said to be *'equivalent to each other'* if the subset of $DSet_A$ with the attributes $\{Attr_{A,p'} = \alpha_{p'} |\ p' = 1,2,..., P'\ \ ,1 \le P' \le P\}$ is equivalent to the subset of $DSet_B$ with the attributes $\{Attr_{B,q'} = \text{ß}_{q'} |\ q' = 1,2,..., Q'\ \ ,1 \le Q' \le Q\}$, i.e.

$$Set\{Obj_A'\} \Leftrightarrow Set\{Obj_B'\} \quad \text{...[4-15]}$$

where, $Set\{Obj_A'\} = \{Obj_{A,i'} | Attr(obj_{A,i'})_{p'} = \alpha_{p'}, p' = 1,2,...P'\} \subset DSet_A$;

and $Set\{Obj_B'\} = \{Obj_{B,j'} | Attr(obj_{B,j'})_{q'} = \text{ß}_{q'}, q' = 1,2,...Q'\} \subset DSet_B$.

- **Inclusive correspondence**

Two sets of attribute values from different datasets, are defined as *'inclusive corresponded to each other'* if they satisfy the conditions defined by Expression [4-16] or [4-17].

$$Set\{Obj_A'\} \subseteq Set\{Obj_B'\} \quad \text{...[4-16]}$$

$$Set\{Obj_A'\} \supseteq Set\{Obj_B'\} \quad \text{...[4-17]}$$

where, $Set\{Obj_A'\} = \{Obj_{A,i'} | Attr(obj_{A,i'})_{p'} = \alpha_{p'}, p' = 1,2,...P'\} \subset DSet_A$;

and $Set\{Obj_B'\} = \{Obj_{B,j'} | Attr(obj_{B,j'})_{q'} = \text{ß}_{q'}, q' = 1,2,...Q'\} \subset DSet_B$.

Similar to the objective semantic attributes of street width, number of lanes, travel time etc., the identified corresponding attribute values can be also employed as exclusion criteria in the matching process: e.g. the polylines PL_A and PL_B will be rejected as a proper matching pair if (a) $PL_A \in Set\{Obj_A'\}$, $PL_B \notin Set\{Obj_B'\}$ and (b) $Set\{Obj_A'\} \subseteq Set\{Obj_B'\}$ or $Set\{Obj_A'\} \Leftrightarrow Set\{Obj_B'\}$.

Due to the fact that different datasets may have distinct values or classes of corresponding subjective attributes, it is very difficult to automatically identify the correspondences. In practice, a human-computer interactive process is often inevitable.

4.3 Matching guided by *'spatial Index'*

Besides the completeness and accuracy, the matching performance is also indicated by the computing speed. Two computing processes are necessary during the matching approach: one is to identify all nodes within a given tolerance distance from a query point, e.g. in the operation of 'Identification of the potential Delimited Stroke matching pairs' depicted by Section 3.4.1; the other is to identify all the line objects within a tolerance distance from each vertex of a given line object, e.g. in the matching process of 'Dealing with the fragmental matching areas' in Section 3.5. The distance between a point and a line segment is different from that between a point and a whole line object. The former can be expressed by Equation [2-2] (see Section 2.2.3.1), while the latter is a process of finding the shortest distance between the point and all line segments constituting the line.

Usually, the road network is represented by vectors of coordinate pairs which are ordered by their connectivity. Given a random query point in space, it will consume much computing time to find the nearly located points or line segments by comparing the objects one by one. In common cases, spatial indexes are utilized to optimize spatial queries within large spatial databases. The spatial indexes needed here have the purpose of accelerating the point-to-point and line-to-point matching. A number of special constraints should be satisfied (Zhang et al. 2009):

(a) It should be easily established and implemented;
(b) It should lead to a maximum speed improvement of the matching process.

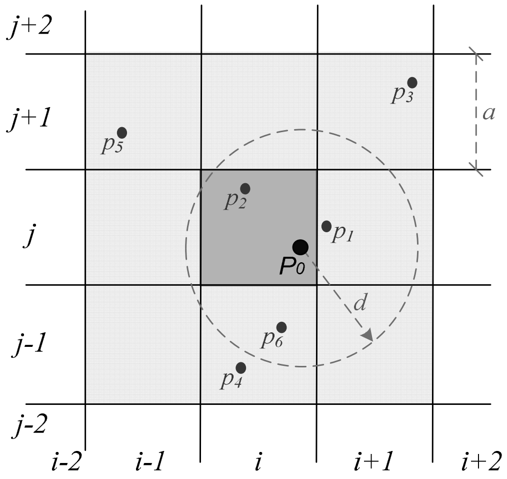
Two grid-based spatial indexes have been established - one is for point and the other for linear data. In the domain of spatial indexes, a grid is a regular tessellation of a manifold or 2-D surface that divides it into a series of contiguous cells where unique identifiers will be assigned and used for spatial indexing purposes. Various grids have been proposed or are currently in use, including grids based on square or rectangular cells, triangular grids or meshes, hexagonal grids, grids based on diamond-shaped cells, and possibly many more. In practice, construction of grid-based spatial indexes entails allocation of relevant objects to their position or positions in the grid, then creating an index of object identifiers versus grid cell identifiers for rapid access (Wikipedia 2009^A). In the proposed matching approach, the grids based on 'Square' are employed to partition the large spatial space so that the point and linear data can be efficiently indexed.

4.3.1 The index model for point data

The grid-based spatial indexes for point data are relatively simple to use. It has been widely discussed and used to organize vector-based objects in large spatial databases. As mentioned earlier, the proposed spatial index for points data aims at accelerating the process of identifying the point set $P(P_0) = \{p_i | d_e(P_0, p_i) \le d\}$, where $d_e(P_0, p_i)$ represents the Euclidean distance between the point p_i and the query point P_0, d is a tolerance distance.

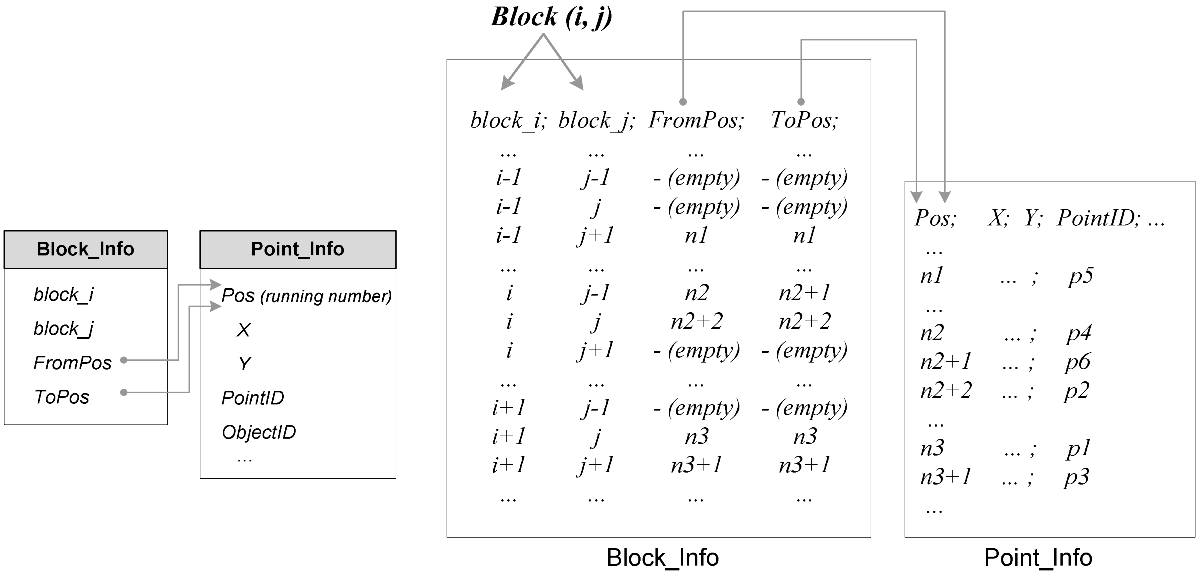
As depicted by Figure 4.15, the whole spatial space is divided into a series of smaller square blocks with varying numbers of points and each point belongs to one block that covers its entity. In the first step of finding all points within a given distance d to a query point P_0, the *block* $_{(i,\ j)}$ is confirmed as *base region* since it contains the query point, but points with the distance d to the query point P_0 are not necessarily located in the same base region, e.g. in Figure 4.15, the point p_1 is very close to P_0

but falls outside *block* $_{(i,\ j)}$. This means it is not sufficient to merely search the base region. To overcome this limitation, the base region $block_{(i,j)}$ and its neighbors $set(N\_block_{(i,j)}) = \{\ block_{I,J} \mid\ i-1 \le I \le i+1, j-1 \le J \le j+1, I \ne i, J \ne j, I, J \in N\ \}$ are often calculated together and termed as *affected blocks*, viz.: $set(A\_block_{(i,j)}) = \{\ block_{I,J} \mid\ i-1 \le I \le i+1, j-1 \le J \le j+1\ \}$. To ensure that the affected blocks can cover all of the points within the tolerance distance d to the query point P_0, the side of the square block has to be larger than the tolerance distance, i.e. $a > d$ in Figure 4.15.



dark grey square: base region;   light grey squares: neighbour-blocks

Figure 4.15 Grid-based spatial index for point data



block_i;	*block_j;*	*FromPos;*	*ToPos;*
...	...	...	...
i-1	*j-1*	*- (empty)*	*- (empty)*
i-1	*j*	*- (empty)*	*- (empty)*
i-1	*j+1*	*n1*	*n1*
...	...	...	...
i	*j-1*	*n2*	*n2+1*
i	*j*	*n2+2*	*n2+2*
i	*j+1*	*- (empty)*	*- (empty)*
...	...	...	...
i+1	*j-1*	*- (empty)*	*- (empty)*
i+1	*j*	*n3*	*n3*
i+1	*j+1*	*n3+1*	*n3+1*
...	...	...	...

Pos;	*X; Y;*	*PointID; ...*
...		
n1	... ;	*p5*
...		
n2	... ;	*p4*
n2+1	... ;	*p6*
n2+2	... ;	*p2*
...		
n3	... ;	*p1*
n3+1	... ;	*p3*
...		

(a) (b)

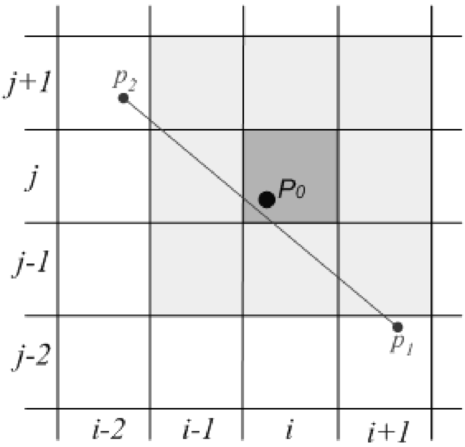
Figure 4.16 Spatial index to organize the point data with an example of p_1 *to* p_6 illustrated in Figure 4.15

According to the data structure illustrated in Figure 4.16-a, the established spatial index for point data can be compactly stored in the physical memory of computer, which is treated as a global variable in the whole matching approach. As shown in Figure 4.16-b, instead of scanning the whole space, the matching algorithm will only scan the points inside the affected blocks of $set(A\_block_{(i,j)}) = \ \{\ block_{I,J} \mid\ i-1 \le I \le i+1, j-1 \le J \le j+1\ \}$, viz. the points p_1 *to* p_6 illustrated in Figure 4.15 to check whether the distance to the query point P_0 is smaller than the pre-defined tolerance distance d or not. In this way, the computation complexity has been cut from *O(n)* to *O(9*n/g)* where *n* is the number of points in the whole space and *g* is the total number of blocks, i.e.

*O(n)/O(9*n/g) = g/9* times faster. Taking a 30km*30km matching area as an example and assuming the tolerance distance is 60 meters, with the established spatial index, the computing speed will become around twenty eight thousand times faster ($g=(30\times10^3\times30\times10^3)/60^2 \;=2.5\times10^5$, $g/9=2.5\times10^5/9\approx2.78\times10^4$).

4.3.2 The index model for line segments

The index for line segments aims to accelerate the process of finding all the line segment l_{ij} (viz. $l(p_i,p_j)$) within a tolerance distance d from a query point P_0, i.e. to identify the set of line segments $L(P_0)$ = $\{\, l_{ij} | d_{PtoL}(P_0,l_{ij})\le d\}$ as fast as possible, where $d_{PtoL}(P_0,l_{ij})$ represents the distance between point P_0 and line segment l_{ij} calculated by Equation [2-2]. As depicted by Section 4.3.1, the grid-based spatial indexes can efficiently organize randomly distributed point objects into a hash directory. However, complications will occur in dealing with line segments. Applying the index techniques only to the start and end point of a line segment proves inadequate because the terminating points do not carry the actual shape information between them (Hoelf and Samet 1991). Often a line segment can stretch into a wide scope of space. If such an extensive line segment is indexed only to the blocks where its start and end- points are located, its parts across the blocks can not be handled, e.g. the start point p_1 of the line segment l_{12} in Figure 4.17 falls inside the lower-right *block*$_{(i+1,j-2)}$ and the end point p_2 belongs to the *block*$_{(i-2,j+1)}$. Since neither *block*$_{(i+1,j-2)}$ nor *block*$_{(i-2,j+1)}$ is an affected block in $set(A\_block_{(i,j)})=\{\, block_{I,J} |\; i-1\le I\le i+1, j-1\le J\le j+1 \;\}$, the line segment l_{12} could be ignored during the searching process even though it is very close to the query point P_0. For this reason, new index methods are required for line segments.



dark grey square: base region; light grey squares: neighbour-blocks

Figure 4.17 Limitation of the grid-based spatial index for point data

Figure 4.18 illustrates another example of a plane corresponding to a set of 25 square grids in a 2D spatial space. In this example, the query point is denoted as P_0 and the line segment as $l(p_i,p_j)$ (abbreviated as l_{ij}), the side of the square grid is represented by a and the predetermined tolerance between the line segment and the query point is d. In particular, the worst case may occur due to the following reasons:

(a) The query point P_0 is located on a corner of the base region, e.g. the upper-right corner of the *block*$_{(i,j)}$ illustrated in Figure 4.18;

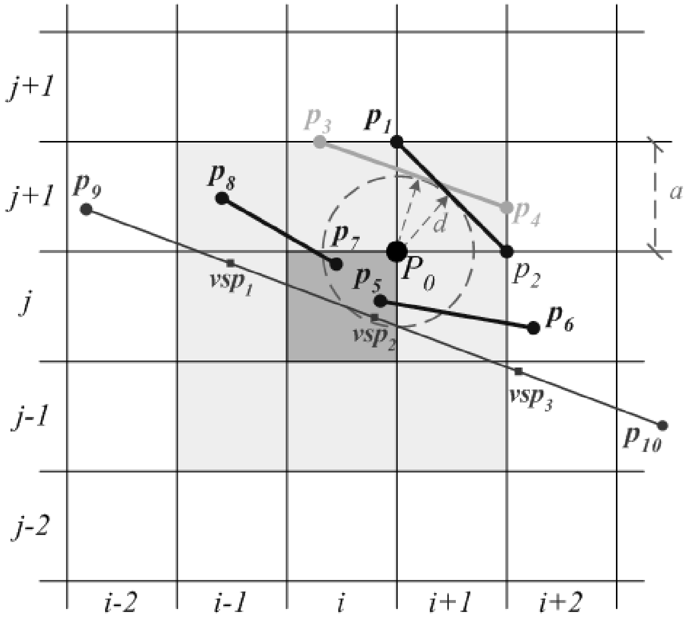
(b) The line segment l_{12} is tangent to the circle with its centre located on query point P_0 and the radius equal to tolerance d;

(c) The side of the square-grid a is equal to $\sqrt{2}\cdot d$;

(d) l_{12} is restricted by two corners of a neighbouring block, e.g. the upper-right neighbour of the base region *block* $_{(i,j)}$, see *block* $_{(i+1,j+1)}$ in Figure 4.18; and

(e) The angle $\angle p_1 P_0 p_2$ is a right angle ($=\pi/2$).

The worst case for line segment l_{12} and query point P_0 indicates that if a randomly given line segment l is not longer than l_{12}, i.e. $l \le l_{12} = 2d$ and has a distance to the query point P_0 smaller than the tolerance *d, i.e.* $d_{PtoL}(P_0, l) \le d$ (ref. Equation [2-2]), and if the side of the square grid is longer than $\sqrt{2}\cdot d$, i.e. $a > \sqrt{2}\cdot d$, then there has to be at least one endpoint of l falling inside one of the affected blocks of the base region covering the query point P_0, see examples of l_{56} and l_{78} in Figure 4.18.



dark grey square: base region; light grey squares: neighbour-blocks

Figure 4.18 Grid-based spatial index for line segments

In fact, a network may contain line segments with a longer length than $l_{12} = 2d$ in the real-world geospatial databases. To avoid such line segments, two decomposition rules are defined:

(i) During the partition process of the whole spatial space, the employed square grid should be larger than $\sqrt{2}d \times \sqrt{2}d$;

(ii) If the line segment l is longer than $2d$, then a set of equidistant *virtual specific points* (VSP) will be inserted to l, e.g. the small square-shaped points *vsp*$_1$, *vsp*$_2$ and *vsp*$_3$ along the line segment $l_{9,10}$ as shown in Figure 4.18. As the result, l is split into several smaller portions and each portion has a length of not longer than $2d$. These virtual points, together with the start and end- point of the line segment l, will be treated as *characteristic points* for the further index calculations. Line segment l is assigned to all blocks when any of the characteristics points fall inside them.

Following these two decomposition rules, a spatial index model for line segments is established. In this model, a block is allowed to have a varying number of line segments; and vice versa a line segment can also be spanned over different blocks. The data structure of the spatial index is illustrated in Figure 4.19.

To identify the set of line segments $L(P_0)$ defined by $\{l_{ij} | d_{PtoL}(P_0, l_{ij}) \leq d\}$, the line segments are first organized by the above-mentioned spatial index and the base region containing the query point P_0 is identified. Then, all line segments occupied by the affected blocks are selected as the potential candidates to the set $L(P_0)$. In the example illustrated by Figure 4.20, besides the line segments l_{12}, l_{34}, $l_{56}$and l_{78}, the $l_{9,10}$ will also be treated as a potential item in $L(P_0)$= $\{l_{ij} | d_{PtoL}(P_0, l_{ij}) \leq d\}$ due to the fact that its virtual specific points *vsp*$_1$ and *vsp*$_2$ fall inside the affected blocks. Finally, each of the potential candidates is verified to see whether the distance to the query point P_0 is smaller than the pre-defined tolerance distance d or not. In this way, the search speed can be dramatically enhanced. Instead of computing every line segment distributed in the whole spatial space, only the potential candidates are calculated.

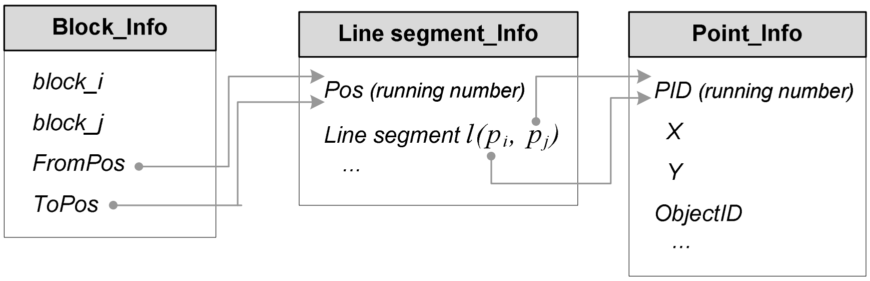
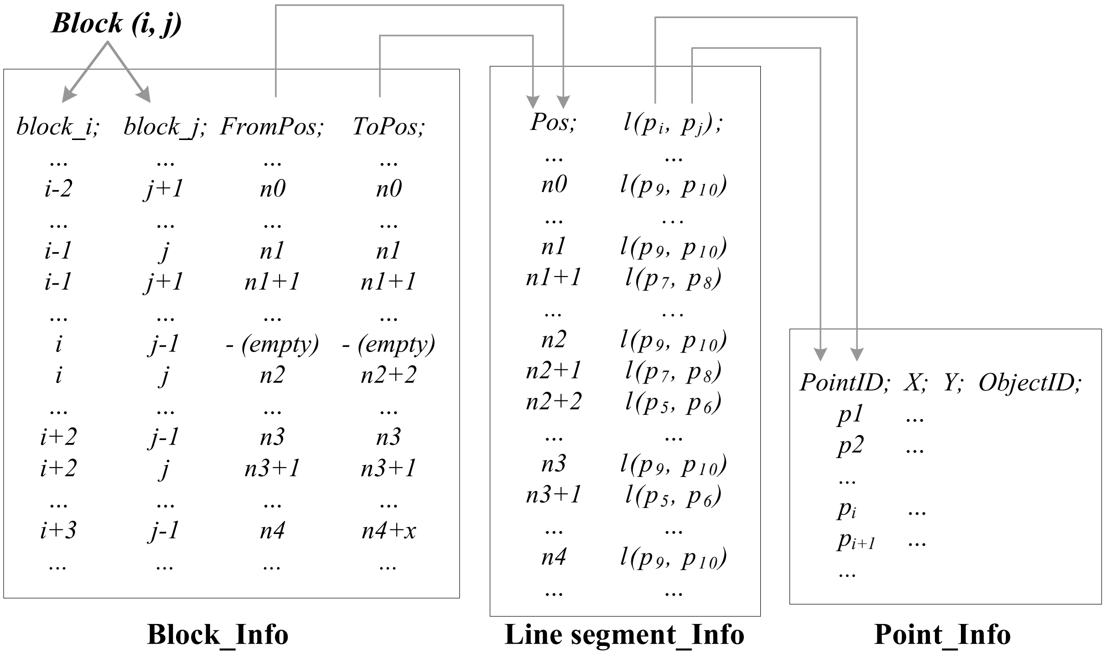


Figure 4.19 Data structure of the spatial index for line segments



block_i;	block_j;	FromPos;	ToPos;
...	...	...	...
i-2	j+1	n0	n0
...	...	...	...
i-1	j	n1	n1
i-1	j+1	n1+1	n1+1
...	...	...	...
i	j-1	- (empty)	- (empty)
i	j	n2	n2+2
...	...	...	...
i+2	j-1	n3	n3
i+2	j	n3+1	n3+1
...	...	...	...
i+3	j-1	n4	n4+x
...	...	...	...

Block_Info

Pos;	$l(p_i, p_j)$;
...	...
n0	$l(p_9, p_{10})$
...	...
n1	$l(p_9, p_{10})$
n1+1	$l(p_7, p_8)$
...	...
n2	$l(p_9, p_{10})$
n2+1	$l(p_7, p_8)$
n2+2	$l(p_5, p_6)$
...	...
n3	$l(p_9, p_{10})$
n3+1	$l(p_5, p_6)$
...	...
n4	$l(p_9, p_{10})$
...	...

Line segment_Info

PointID;	X;	Y;	ObjectID;
p1	...		
p2	...		
...			
p_i	...		
p_{i+1}	...		
...			

Point_Info

Figure 4.20 The established spatial index for the line segments l_{56}, l_{78} *and* $l_{9,10}$ illustrated in Figure 4.18

The spatial indexes to organize the points and line segments have been embedded in the proposed DSO matching algorithm and substantially accelerated the matching process, especially when large-scope matching areas are concerned.

Chapter 5

Evaluation of the Matching Performance

To examine how well the proposed contextual matching approach works, a computer program using above described programming logic has been developed in the Windows Environment, which uses ESRI's commercial software ArcGIS 9.x as a platform for data management, user-interface, and map display. The matching program can directly read and write networks in ESRI's shape file format and has accurately and efficiently search for the matching pairs between different datasets. This Chapter will evaluate the matching approach by conducting a number of experiments on various real-world data.

The test datasets are described in details in Section 5.1. Section 5.2 discusses the experiment results of matching various versions of street networks. Section 5.3 assesses the matching quality that is reflected at various certainty levels with the aim to detect and refine mismatches created by the automatic routines. Section 5.4 addresses the statistical investigations on geometric differences between the matching pairs from different datasets. These differences provide clues for user to define proper threshold values in an interactive matching process.

5.1 Test datasets

The conducted matching experiments involve four different datasets:

5.1.1 ATKIS

ATKIS was captured through map digitization in combination with semiautomatic object extraction from imagery data. The data structure is defined in accordance with the Official Topographic Cartographic Information System. ATKIS is a general topographic dataset that stores data of different topographic object categories. It is not targeted to a certain application domain but rather serves as an information basis on top of which application-dependent data can be added (Volz 2006). The objects of this dataset are defined by their position, shape, name and other properties based on attribute-oriented catalogue. In this way, a real-world object should be firstly assigned to an object class and then precisely described by some specific attributes, i.e. a landscape feature that has been assigned to an object class can be represented in higher detail by descriptive attributes. An object class has a specific scope of applicable attributes that describe quantitative properties. Each of these attributes has its own key values.

As the most important part for the transportation, the road layer of ATKIS is composed of geometries and general-purposed attributes of road lines. Each captured road is displayed in the cartographic database as a line representing the centreline of the road which reveals an accuracy of ±3m at important positions (AdV 2003). However the attributes are not completely covered with values, especially the street names which are essential clues for the matching are only sporadically available.

5.1.2 Tele Atlas

As a high-end geospatial database product, Tele Atlas is a fully attributed dataset containing detailed road network, water features, parks and landmarks, county, city and civil divisions, ZIP codes, urbanized area codes and census tracts. The Tele Atlas road network contains geometries and navigation-oriented attributes of road lines (middle axes) which were captured through map digitization, GPS-supported field measurement and dynamic supervision of traffic information. In Europe, it has an absolute accuracy of within 10 meters inside built-up areas and 25 meters outside built-up areas. In dense urban areas where up-to-date source material is present, an accuracy of a few meters can be expected.

Tele Atlas is one of the most important data suppliers for the routing of motorcars. Its dataset contains a number of routing-relevant attributes:

- Street name
- Functional Road Class (FRC): Classification of a transportation element based on its functional importance within the transportation network;
- Form of Way (FOW): Indication of the physical appearance and/or traffic characteristics of a transportation element that can be important for routing applications.
- Network Class: Creation of a closed and efficient routing network with a hierarchical structure.
- Restrictions: A restriction limits access to part or a whole transportation element, e.g. blocked passage, restricted time validity, vehicle type, direction of traffic flow, etc.
- Maneuvers: A Maneuver is a mandatory, preferred, or prohibited access of a transportation element in relation to another transportation element.
- Signpost Information: Signposts are useful for directional and destination information. Transportation elements that lead to a destination indicated by a signpost can be linked.
- Points of Interest (POI): A Point of Interest, also called ‘Service’, is a point representation of an activity at a specific location, such as hotel, gas station, restaurant, showplace, beauty spot etc.

One of major assets of the Tele Atlas database is its detailed classification of road elements that constitute the transportation network, which can be important for selection and extraction of certain types of roads, e.g. routing calculations can use the classifications to estimate speed limits for assigning weights (Tele Atlas 2003).

5.1.3 NAVTEQ

NAVTEQ is a world leader in the creation, maintenance and distribution of digital navigable maps, which provides highly accurate representations of the detailed and fully attributed road networks in 7 different formats for 17 European countries, North America and expanding areas of the Middle East and Far East (2004). Similar to Tele Atlas, the NAVTEQ road network contains geometries and navigational attributes of road lines (middle axes), which were captured through map digitization, GPS-supported field measurement and dynamic supervision of traffic information. Every single road object can be described with more than 160 attributes such as street names, address ranges, access limitations, time and turn restrictions, physical barriers and gates, one-way streets, restricted access and relative road heights, speed limits, etc., which allows, for example, routing guidance applications and location services.

NAVTEQ's digital map database originates from precisely captured real-world roads. Every day, hundreds of trained field technicians drive and re-drive highways, streets, alleys and rural roads to acquire or update road data. As a result, they construct a database from a driver's view and stick to a single global standard. Being continuously updated, NAVTEQ digital map data not only enables door-to-door routing throughout Europe and North America, but it contains millions of POIs in 44 categories, covering hotels to petrol stations, sports facilities to tourist attractions, helping users identify their required destinations.

NAVTEQ digital map data offers accuracy, detail, reliability, and flexibility and hence has a vast variety of different applications on the leading express mail services, web mapping services (WMS), emergency and government routing plans, efficient field service management, as well as numerous other fleet operations and Location Based Services (LBS).

5.1.4 OpenStreetMap (OSM)

OpenStreetMap (OSM, http://www.openstreetmap.org/) is a collaborative project to create and provide free geographic data such as street maps of the world based on Web 2.0. The maps are created using data from portable GPS devices, aerial photography and other free sources.

OpenStreetMap was founded in July 2004 and in April 2006 a foundation was established, with the aim to encourage the growth, development and distribution of free geospatial data and provide geospatial data for anybody to use and share. The initial OSM map data was built from scratch by volunteers performing systematic ground surveys using a handheld GPS unit and a notebook or a voice recorder, which was then entered into the OSM database from a computer. More recently the availability of aerial photography and other data sources from commercial and government sources has greatly increased the speed of this work and has allowed land-use data to be collected more accurately (Wikipedia 2009 ^{B}), e.g.:

- In December 2006 Yahoo confirmed that OSM could use their aerial photography as a backdrop for map production.
- In April 2007 Automotive Navigation Data donated a complete road dataset for the Netherlands and trunk road data for India and China to the project.
- In October 2007 OSM completed the import of a US Census TIGER road dataset.

The ground surveys in OSM project are performed by volunteers on foot, bicycle or in a car. Some devoted contributors help digitize whole towns over a period of time. Mapping parties are organized to bring a number of contributors together who can digitize a particular area for an evening or a weekend. In addition to structured surveys, a large amount of smaller editing work such as correct errors or add features, is made by contributors. By August 2008, there were over 50 000 registered users with over 5 000 active contributors. In March 2009, 100 000 users were registered.

Up to date, the OSM has revealed highly accurate and detailed geographic data in many urban areas of Europe and North America; however, in some rural areas the data quality of OSM is still very poor.

5.2 Experimental results and interpretations

The experiments involve (a) Matching between NAVTEQ and ATKIS; (b) Matching between Tele Atlas and ATKIS; (c) Matching between OpenStreetMap and NAVTEQ; and (d) Matching between Tele Atlas and NAVTEQ. The selected test areas cover a number of federal states in Germany, such as Berlin, Hessen, Lower Saxony and Bavaria, containing coastal and inland areas, mountainous and flat areas, rural and built-up areas.

5.2.1 Evaluation metrics

Before presenting the test results, the author prefers to illustrate the metrics to evaluate the automatic matching performance.

The automatic matching results can be generally categorized into matched objects and unmatched objects. As illustrated by Figure 5.1, the matched objects can be further classified into accurate matches, mismatches and false positive matches; while the unmatched objects involves the cases of proper non-matches and false negative matches (Cobb et al., 1998):

Accurate match *- a geospatial object in reference dataset is matched to its counterparts in target dataset with a sufficient accuracy;*

Mismatch *- an object in reference dataset is correctly judged to have a match in target dataset, but its correspondent is wrongly identified;*

False positive match *- an object in reference dataset is incorrectly labelled as having a match in target dataset when in fact it does not;*

Proper non-match *- an object in reference dataset is correctly judged as having no match in target dataset;*

False negative match *- an object in reference dataset is identified as not having a match in target dataset when in fact it does.*

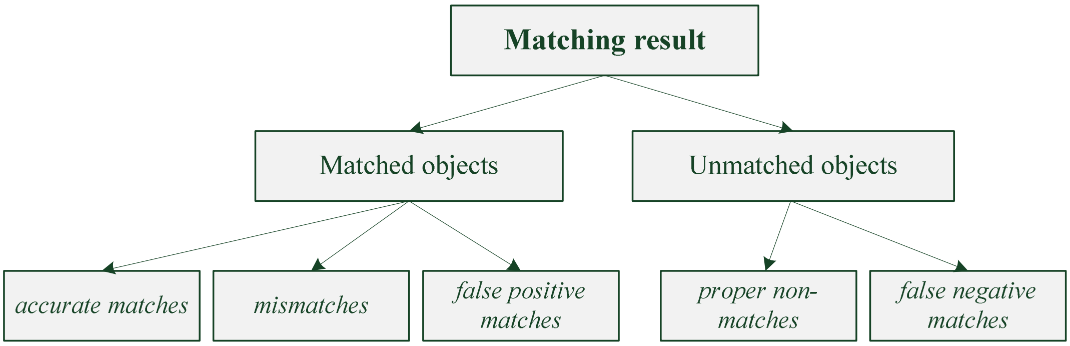


Figure 5.1 Hierarchical classification of the automatic matching results

The iterative matching approach identifies new matches at each stage and does not label non-matches until the final stage; false negatives are a residual and do not present a problem at an intermediate iteration; false positive errors and mismatches are less desirable and less manageable than false negatives because they may introduce additional errors in subsequent iterations, and an operation can find nowhere in the automatic matching approach to correct the false positives and mismatches (Saalfeld 1988). In line with the occurrences listed above, the metrics of matching rate, matching correctness and matching speed can be respectively defined.

- **Matching rate**

The matching rate, also termed as matching completeness, is calculated in Equation [5-1], where N_{aM} , N_{mM} , N_{fnM} represent the number of objects which belong to accurate matches, mismatches and false negative matches respectively.

$$matchRate = \frac{N_{aM} + N_{mM}}{N_{aM} + N_{mM} + N_{fnM}} \times 100\% \qquad \text{...[5-1]}$$

- **Matching correctness**

The matching correctness, also termed as matching accuracy, is the percentage of accurate matches with respect to the total matched objects including accurate matches, mismatches and false positive matches, see Equation [5-2].

$$matchCorrectness = \frac{N_{aM}}{N_{aM} + N_{mM} + N_{fpM}} \times 100\% \qquad \text{...[5-2]}$$

- **Matching speed**

As indicated in Chapter 3 and 4, the contextual matching approach takes turns to conduct an iterative searching operation on each reference road no matter whether this road has a counterpart or not. Hence, the matching speed defined by Equation [5-3] is equal to the total number of reference objects ($\sum N$) divided by the computing time ($matchTime$), where 'reference objects' cover all the occurrences of accurate matches, mismatches, proper non-match, false positive and false negative matches.

$$matchSpeed = \frac{\sum N}{matchTime} \qquad \ldots[5\text{-}3]$$

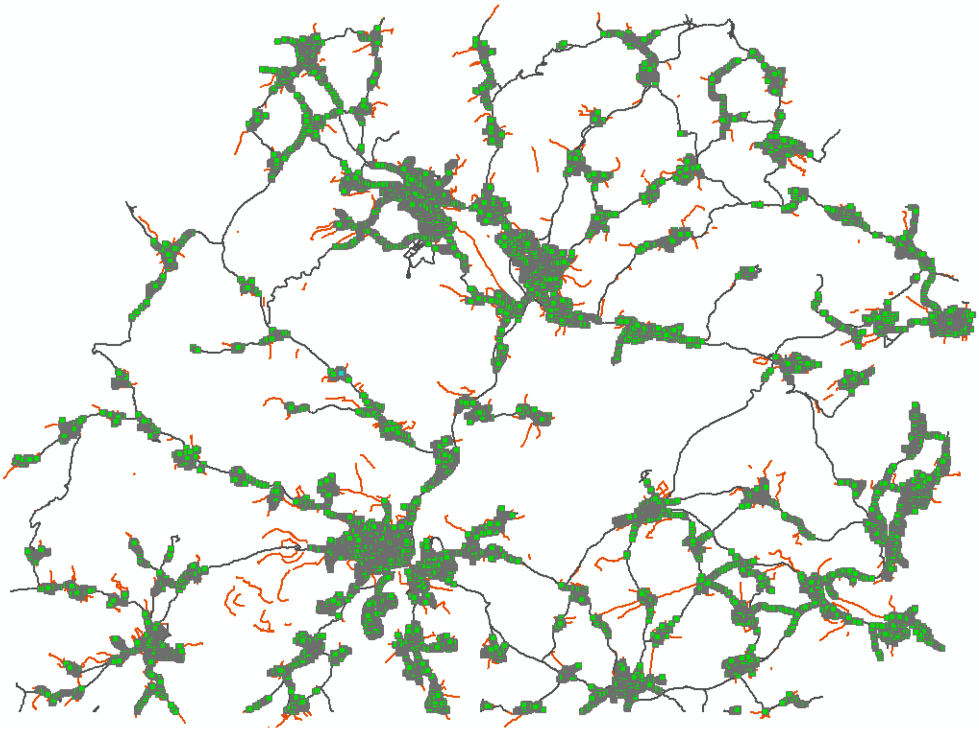
Obviously, the defined metrics of matching rate, correctness and speed can reflect the performance of the contextual matching approach: the larger these values are, the better the matching performance. However, in the practice, computing these metrics requires some reference matching results which rarely exist and are very expensive to obtain. Thus, for our experiments all these indicators are evaluated through an interactive counting of errors. Since the task is laborious, it can be performed on an excerpt of the data, and then extrapolated to the entire dataset (Mustière and Devogele 2008).

5.2.2 Quantitative results

After comparing the automatic matching results with the manually produced ones, the performance of the proposed contextual matching algorithm is evaluated with respect to the three measurements of automatic matching rate, correctness and speed. As demonstrated in Section 5.2.1.1~5.2.1.4, this approach revealed satisfactory matching performances on the conducted experiments.

5.2.2.1 Matching between ATKIS and Tele Atlas

Figure 5.2 illustrates a matching experiment between ATKIS and Tele Atlas in a large test area of Hessen, Germany (ca. 1200 km^2), which involves two versions of the real-world road networks: one is ATKIS represented in dark grey lines and the other is Tele Atlas represented in red lines. In this test area, there are 10959 ATKIS objects and 10681 Tele Atlas objects in total. To accomplish the automatic matching, the automatic matching program takes 22 seconds by a CPU of Intel Duo 2.0 Hz, i.e. about 500 objects per second (incl. data preprocessing and data matching). More than 97% of the ATKIS objects were matched to their Tele Atlas counterparts during the automatic procedure; among the matched objects more than 99.1% are correct. The detailed statistic matching result is summarized in Table 5.1.



dark grey lines: ATKIS red lines: Tele Atlas green arrows: links

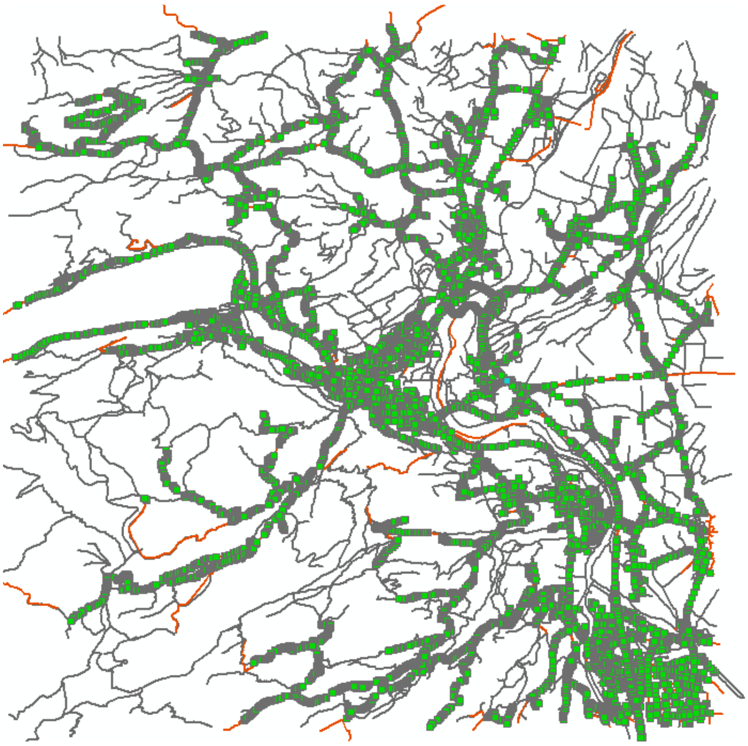
Figure 5.2 Matching between ATKIS and Tele Atlas in Hessen, Germany (ca.1200 km^2)

Datasets to be matched (ca. 1200 km2)		Reference dataset ATKIS	Target dataset Tele Atlas
Total length of the road network		1237.5 km	1282.2 km
Total amount of the turning points		42411	54747
Total amount of the road objects		10947	10360
Matching result (measured by the amount of the objects of reference dataset)	*accurate matches*	8001	
	mismatches	52	
	false positive matches	21	
	false negative matches	228	
	proper non-matches	2645	
Automatic matching rate		(8001+52)/(8002+51+228) = 97.2%	
Automatic matching correctness		8001/(8001+52+21) = 99.1%	
Automatic matching time (incl. data preprocessing)		22 seconds	
Automatic matching speed		10947/22 = 498 objects/second	
Computer (CPU)		Intel Centrino Core Duo 2.0 Hz	

Table 5.1 Statistic result of the matching experiment illustrated in Figure 5.2

As illustrated in Figure 5.2, three major differences can be identified between the road network of ATKIS and Tele Atlas. Firstly, the Tele Atlas is fully attributed while the ATKIS is not, i.e. the conducted automatic matching approach can merely compare the geometry and topology. Second, ATKIS and Tele Atlas follow different rules of data acquisition. The Tele Atlas network contains some road elements that do not have correspondences in ATKIS and vice versa. Third, unsymmetrical location shift often occurs between these two road networks, and in some cases such location shift has even exceeded 60 meters.

5.2.2.2 Matching between NAVTEQ and ATKIS



dark grey lines: ATKIS red lines: NAVTEQ green arrows: links

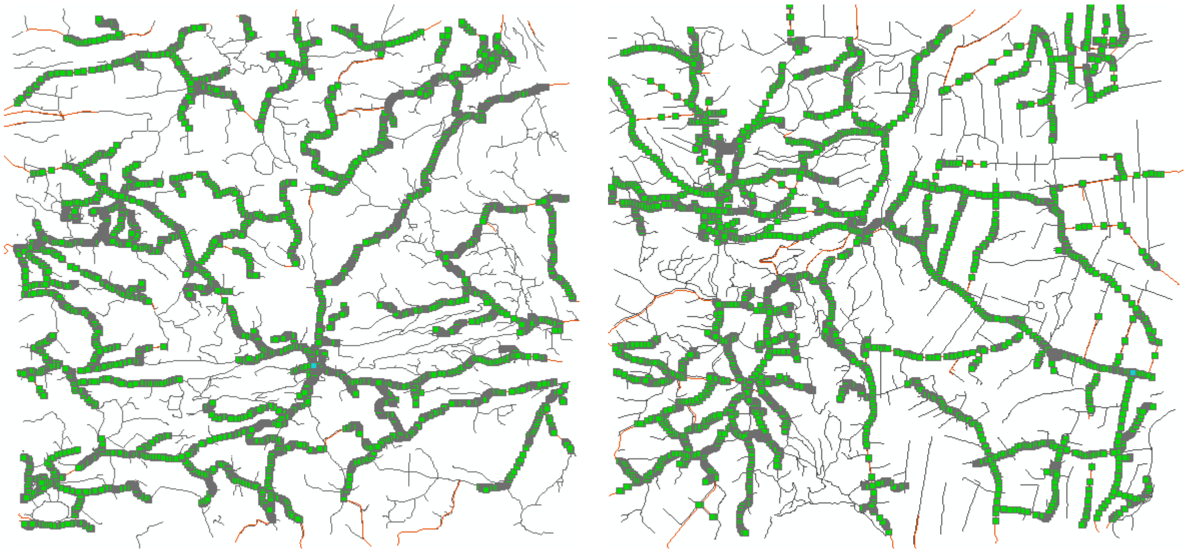
Figure 5.3 Matching between NAVTEQ and ATKIS in Immenstadt, Germany (ca.120 km^2)

Datasets to be matched (ca. 120 km2)		Reference dataset NAVTEQ	Target dataset ATKIS
Total length of the road network		307.5 km	688.7 km
Total amount of the turning points		6551	38901
Total amount of the road objects		2214	5011
Matching result (measured by the amount of the objects of reference dataset)	*accurate matches*	1951	
	mismatches	14	
	false positive matches	3	
	false negative matches	87	
	proper non-matches	159	
Automatic matching rate		(1951+14)/(1951+14+87) = 95.8%	
Automatic matching correctness		1951/(1951+14+3) = 99.1%	
Automatic matching time (incl. data preprocessing)		6 seconds	
Automatic matching speed		2214 / 6= 369 objects/second	
Computer (CPU)		Intel Centrino Core Duo 2.0G Hz	

Table 5.2 Statistic result of the matching experiment illustrated in Figure 5.3

Figure 5.3 illustrates the matching experiment between NAVTEQ (red lines) and ATKIS (dark grey lines) in the city of Immenstadt, Germany (ca. 120 km^2). As the dataset of ATKIS does not have any valuable semantic attributes, only geometric and topologic matching are used in this experiment (similar to Section 5.2.2.1).

As illustrated, the two datasets have different representations (scales) - the road network of ATKIS consists of 5011 objects with the total length of 688.7 kilometres while the road network of NAVTEQ only covers 2214 objects with the total length of 307.5 kilometres. In spite of this discrepancy, the contextual matching approach reveals a satisfying matching performance. As summarized in Table 5.2, the automatic matching rate reaches 95.8%; meanwhile the overall matching correctness exceeds 99.1%. Particularly worthwhile to mention is that a number of corresponding roundabouts (loops) and dual carriageways (parallel lines) have been accurately matched together.



Area (a) (ca. 100 km^2) Area (b) (ca. 100 km^2)

dark grey lines: ATKIS red lines: NAVTEQ green arrows: links

Figure 5.4 Matching between ATKIS and NAVTEQ in the mountain areas of Garmisch, Germany

Matching areas		Area (a)		Area (b)	
Datasets to be matched		Reference (ATKIS)	Target (NAVTEQ)	Reference (ATKIS)	Target (NAVTEQ)
Total length of the road network		382.7 km	167.4 km	381.7 km	169.7
Total amount of the turning points		19000	4865	14998	4402
Total amount of the road objects		1762	571	1857	682
Matching result (measured by the amount of the objects of reference dataset)	*accurate matches*	762		892	
	mismatches	2		2	
	false positive matches	0		1	
	false negative matches	43		57	
	proper non-matches	955		905	
Automatic matching rate		764 / 805 = 94.7%		894 / 951=94.0%	
Automatic matching correctness		762 / 764 = 99.7%		892 / 895=99.7%	
Automatic matching time (incl. data pre-processing)		2 seconds		2 seconds	
Automatic matching speed		881 objects/second		928 objects/second	
Computer (CPU)		Intel Centrino Core Duo 2.0G Hz			

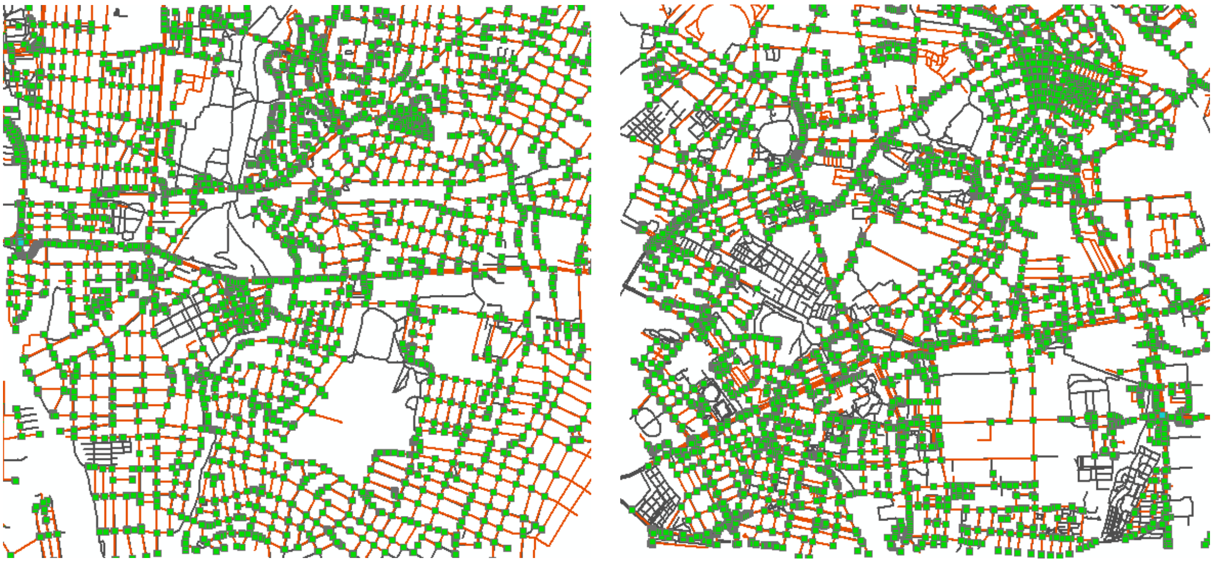
Table 5.3 Statistic result of the matching experiment illustrated in Figure 5.4

Figure 5.4 shows two matching areas which are randomly selected from the mountain areas of Garmisch, Germany. There are 3619 (1762+1857) objects with the total length of 764.4 (382.7+381.7) kilometres in the ATKIS dataset while only 1253 (571+682) objects with the total length of 337.1(167.4+169.7) kilometres in NAVTEQ dataset. Nevertheless, some road objects are represented in the less detailed dataset of NAVTEQ but not in ATKIS dataset. Being constrained by the different representations between these two datasets, the automatic matching rate only reaches 94%. On the other hand, as the topologic conditions in mountain areas are not so complex as the built-up areas, the conducted matching experiments in Garmisch, Germany reveals a nearly perfect matching accuracy: e.g. in the two examples illustrated in Figure 5.4, ca. 99.7% of the automatic matched objects are correct, see details in Table 5.3.

5.2.2.3 Matching between OSM and NAVTEQ

Figure 5.5 depicts two matching experiments which are conducted in the urban areas of Berlin, Germany: one is the north-west part (see Figure 5.5-a) and the other is in south-middle (see Figure 5.5-b).

Comparing to the matching areas of Hessen (ref. Figure5.2) and Garmisch (ref. Figure 5.4), Germany, the roads become much denser in the urban areas of Berlin and accordingly the topologic relationships becomes much more complex. For instance, the matching areas illustrated in Figure 5.5 involve a number of complex crossings and dual carriageways which reveal either similar of distinct LoDs between NAVTEQ and OSM. Moreover as the OSM streets can be contributed by any internet user, its data quality can hardly be assured. In some areas the corresponding roads may have entirely different geometry and/or topology with respect to the location, shape, topologic connection, etc. Taking advantage of the network-based matching strategy and the capability to calculate more context information, however, the contextual matching approach shows satisfying matching results in these two urban areas: the overall matching rate reaches 97.9% while the matching correctness is a little bit lower - ca. 98.9%, see detailed statistics in Table 5.4. In this matching experiment, the semantic attributes have not been considered either.



Area (a) (ca. 40 km^2) Area (b) (ca. 40 km^2)

dark grey lines: OSM red lines: NAVTEQ green arrows: links

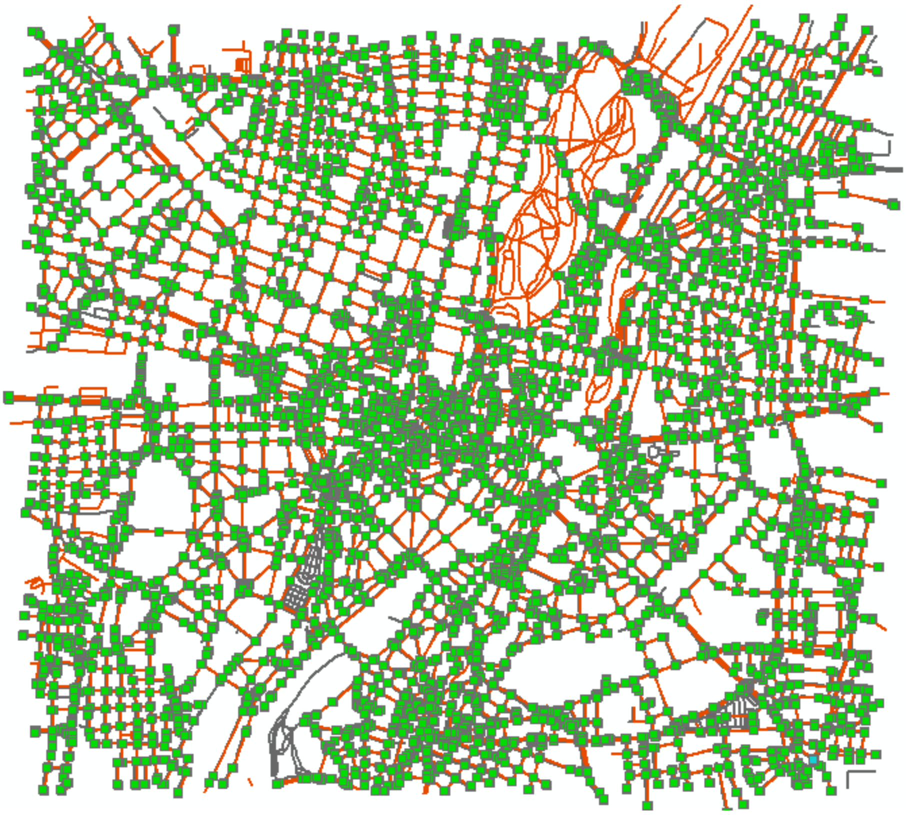
Figure 5.5 Matching between NAVTEQ and OSM in the urban areas of Berlin, Germany

Matching areas		(a)		(b)	
Datasets to be matched		Reference (OSM)	Target (NAVTEQ)	Reference (OSM)	Target (NAVTEQ)
Total length of the road network		353.4 km	313.2 km	438.5 km	341.8 km
Total amount of the turning points		7953	6510	11209	8355
Total amount of the road objects		2918	2555	4226	3122
Matching result (measured by the amount of the objects of reference dataset)	*accurate matches*	2244		2460	
	mismatches	20		17	
	false positive matches	4		13	
	false negative matches	44		57	
	proper non-matches	606		1679	
Automatic matching rate		2264/2308=98.1%		2477/2534=97.8%	
Automatic matching correctness		2244/2268=98.9%		2460/2490=98.8%	
Automatic matching time (incl. data pre-processing)		3 seconds		4 seconds	
Automatic matching speed		973 objects/second		1057 objects/second	
Computer (CPU)		Intel Centrino Core Duo 2.0G Hz			

Table 5.4 Statistic result of the matching experiment illustrated by Figure 5.5

5.2.2.4 Matching between Tele Atlas and NAVTEQ

In Figure5.6, the road networks form Tele Atlas and NAVTEQ have been successfully matched together in downtown area of Munich, Germany. As both the datasets of Tele Atlas and NAVTEQ are almost fully attributed, the semantic information, such as ‘Street Name’, becomes possible being utilized for the identification of the homologous counterparts between different datasets. Based the semantic matching strategies elucidated in Section 4.2, the proposed approach has revealed a nearly perfect matching performance in this case in spite of the complex geometric and topologic conditions: the matching rate goes near to 98%; and the matching correctness is over 99.6%.



dark grey lines: Tele Atlas red lines: NAVTEQ green arrows: links

Figure 5.6 Matching between Tele Atlas and NAVTEQ in the centre part of Munich, Germany (ca.50 km^2)

Datasets to be matched (ca. 120 km2)		Reference dataset Tele Atlas	Target dataset NAVTEQ
Total length of the road network		520.0 km	528.5 km
Total amount of the turning points		21479	12211
Total amount of the road objects		6302	5528
Matching result (measured by the amount of the objects of reference dataset)	*accurate matches*	5638	
	mismatches	21	
	false positive matches	0	
	false negative matches	122	
	proper non-matches	521	
Automatic matching rate		(5638+21)/(5638+21+122)= 97.9%	
Automatic matching correctness		5638/(5638+21+0)=99.6%	
Automatic matching time (incl. data preprocessing)		7 seconds	
Automatic matching speed		6302/7= 900 objects/second	
Computer (CPU)		Intel Centrino Core Duo 2.0G Hz	

Table 5.5 Statistic result of the matching experiment illustrated by Figure 5.6

In despite of the availability of the semantic attributes in both datasets to be compared, a few unfavourable matches - 21 mismatchings and 122 false negative matches still exist as shown in Figure5.6 due to the fact that:

(i) Some pedestrian ways are not named in either dataset;

(ii) The corresponding road objects from Tele Atlas and NAVTEQ have different street names in some cases;

(iii) The geometric and (or) topologic characteristics of the corresponding roads are too inconsistent between the two datasets; and

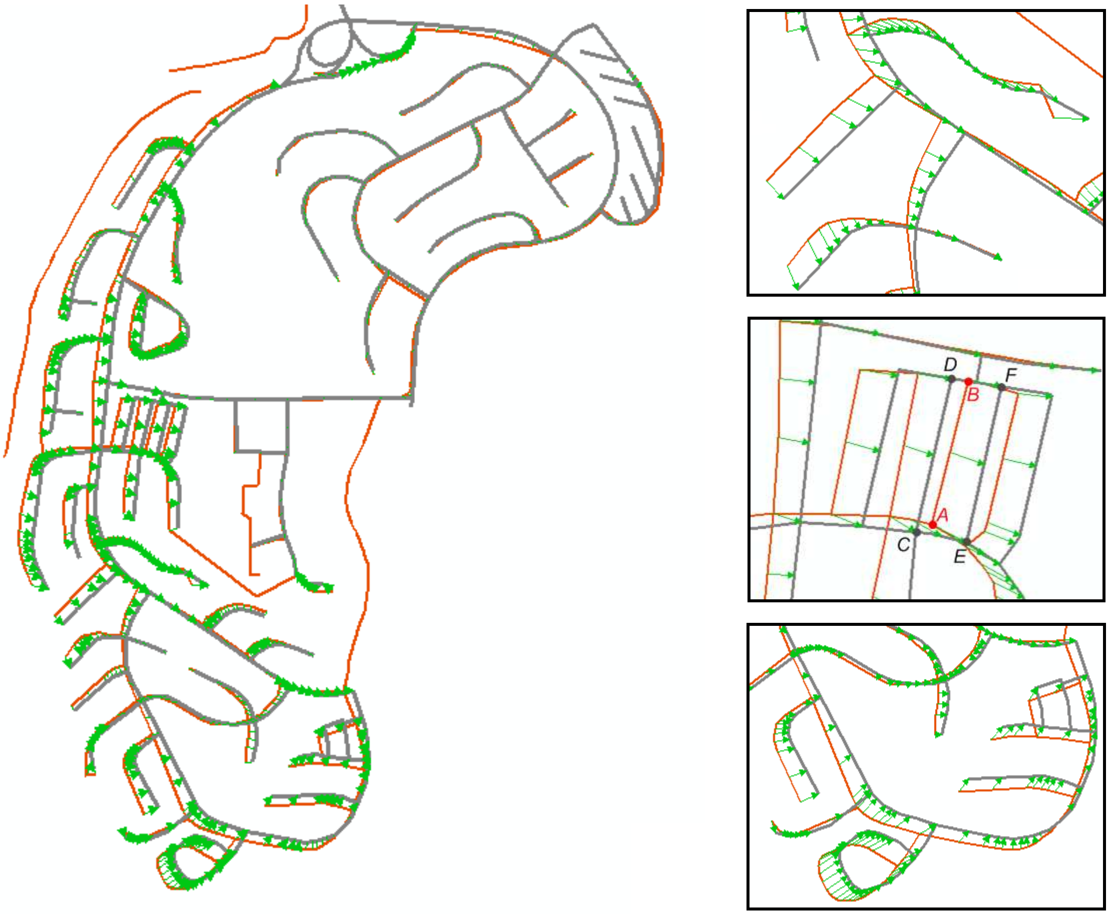
(iv) The topologic connections are too complex to be implemented for the data matching.

The matching experiment between Tele Atlas and NAVTEQ leads to a general insight. On one hand, the semantic information which exists in both datasets can bring useful knowledge into the matching process and thus enhance the matching efficiency and accuracy; and (b) on the other hand, the semantic attributes are not 100% reliable. For this reason, the semantic matching approach is often combined with the geometric and topologic methods in practice.

5.2.3 In-depth discussion on the matching results

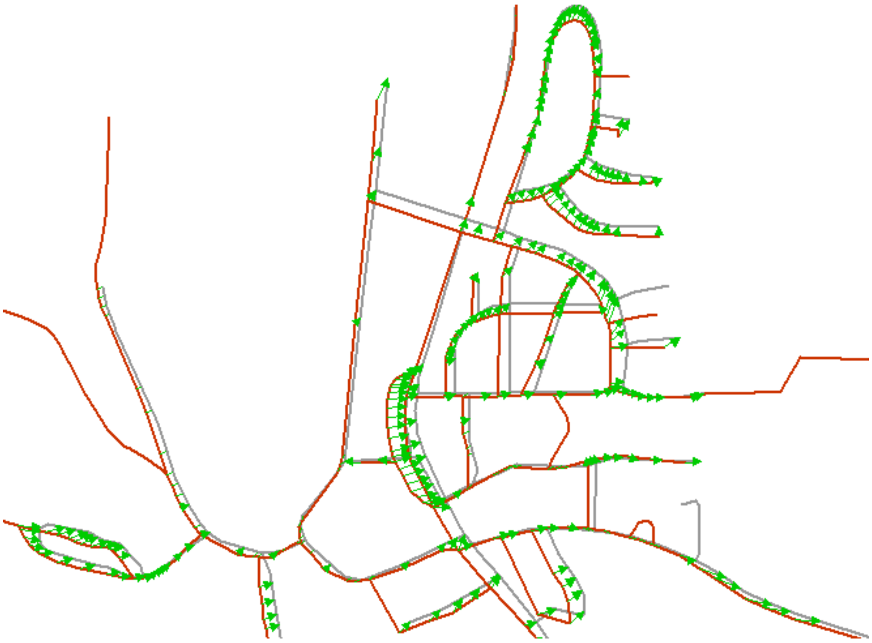
In order to further assess the performance of the contextual matching approach, a few detailed matching cases are discussed below.

Figure 5.7, 5.8 and Figure 5.9 show some efficient cases for the street matching between Tele Atlas and ATKIS. In these cases, the contextual matching approach reveals a perfect matching completeness and accuracy: not only the lines but also the nodes are accurately matched through the comparison of the topologic connection between nodes and lines. Owing to the network-based matching, the road *A→B* in Figure 5.7 is correctly matched to *E→F* although it lies much closer to road *C→D*. The similar occasion also occurs in many other cases, e.g. the central part of Figure 5.8 and Figure 5.9-b.



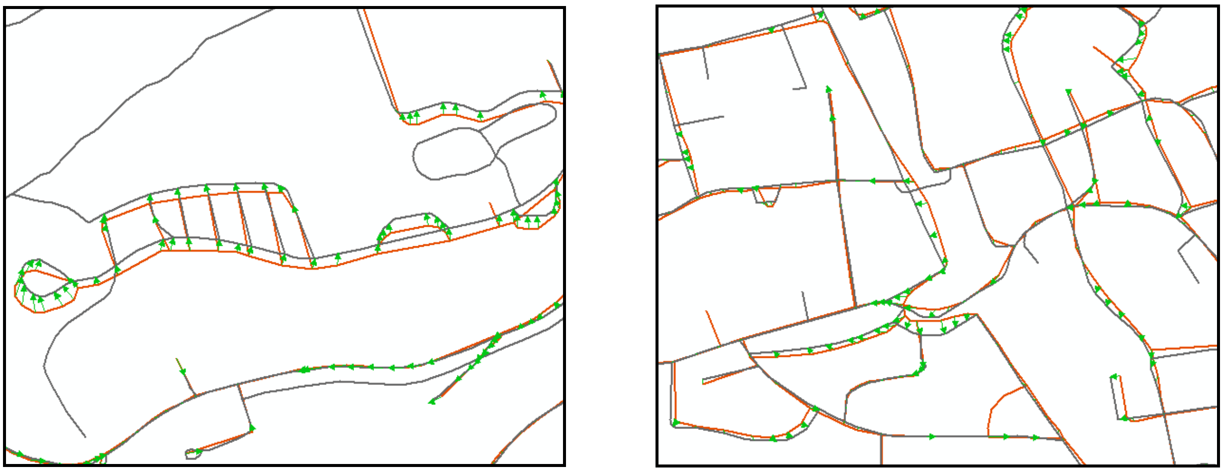
dark grey lines: ATKIS red lines: Tele Atlas green arrows: links

Figure 5.7 An efficient matching case with three detailed parts



dark grey lines: ATKIS red lines: Tele Atlas green arrows: links

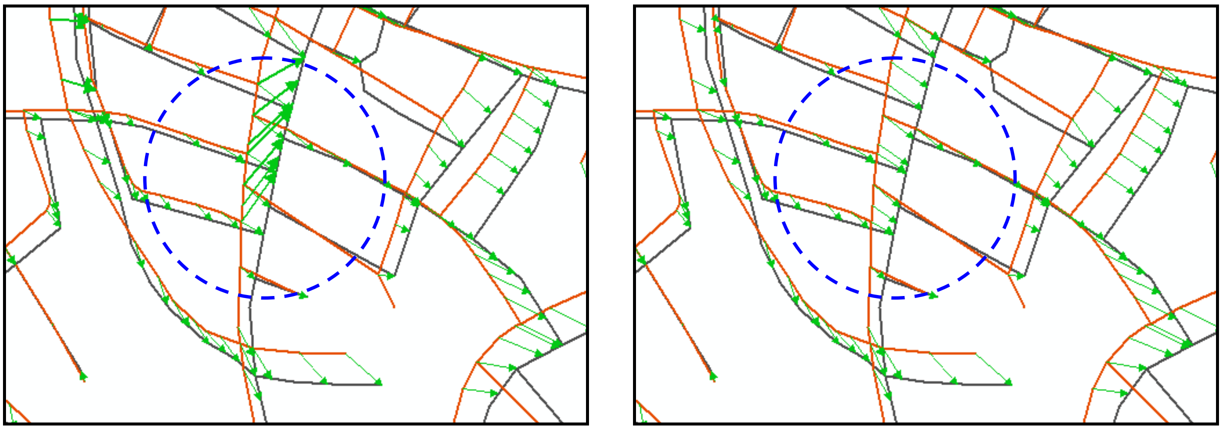
Figure 5.8 An efficient matching case



(a) (b)

dark grey lines: ATKIS red lines: NAVTEQ green arrows: links

Figure 5.9 Two efficient matching cases

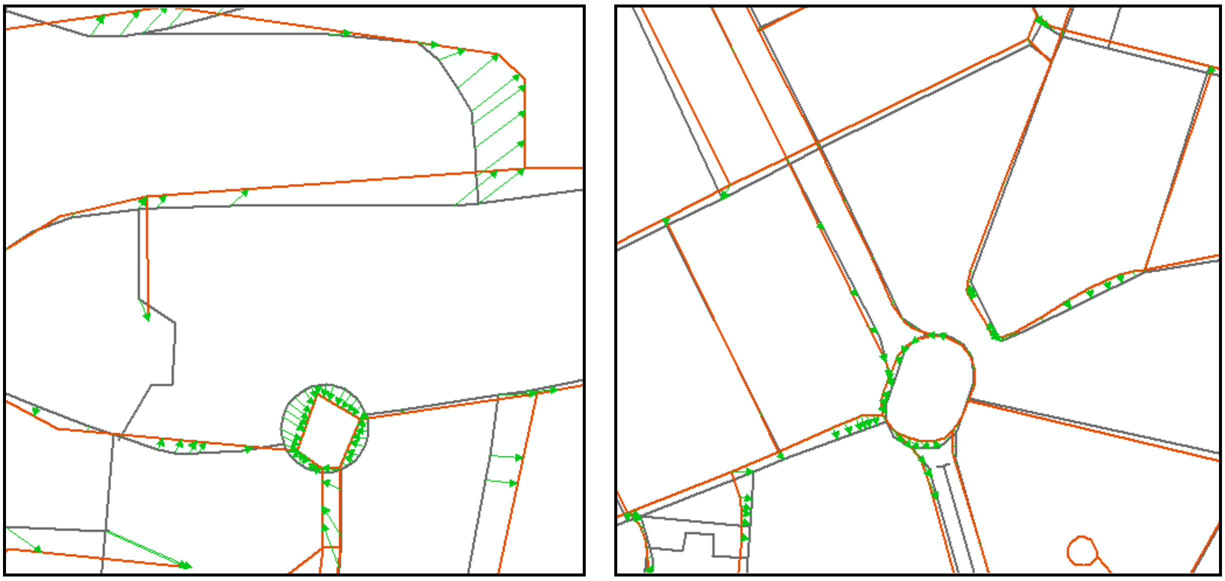


(a) (b)

dark grey lines: ATKIS red lines: Tele Atlas green arrows: links

Figure 5.10 The matching performances (a) with data preprocessing and (b) without data preprocessing

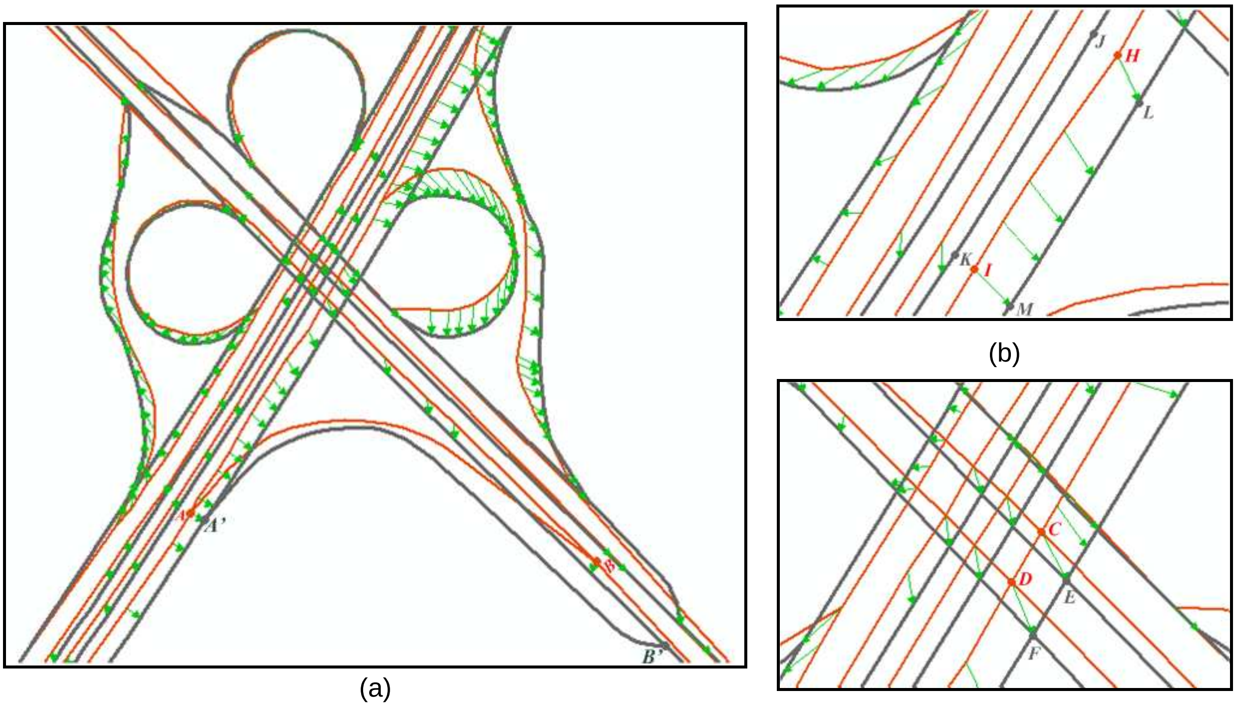
Being aided by a data preprocessing that can eliminate topologic ambiguity (ref. Section 3.1.4), the contextual matching approach can yield better matching performances. In the example shown in Figure 5.10, the matching results with and without the data preprocessing are compared to each other.



(a) (b)

dark grey lines: OSM red lines: Tele Atlas green arrows: links

Figure 5.11 Efficient matching cases for looping crosses



(b)

(a)

(c)

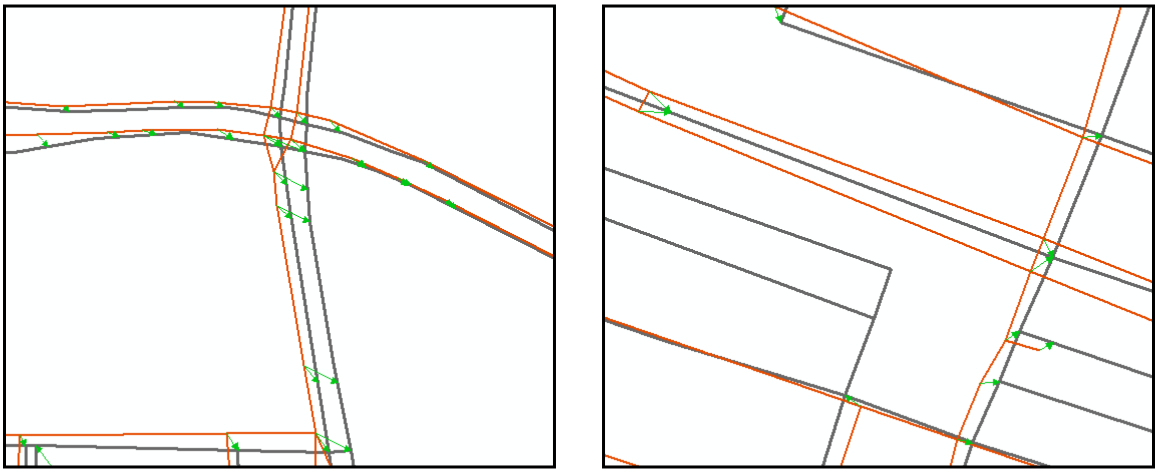
dark grey lines: ATKIS red lines: Tele Atlas green arrows: links

Figure 5.12 A complex matching case around highways

Taking advantage of the assisting technology - matching guided by ‘structure’, the contextual matching approach also revealed high matching rate and accuracy for loops, parallel roads and even complex crossings, e.g., Figure 5.11 shows two efficient matching cases for looping crosses; and

Figure 5.12 illustrate a complex matching cases around high ways. In Figure 5.12, the dual carriageways and some other slip roads are accurately matched between different datasets. As the parallel roads are treated as one item, the contextual matching approach does not necessarily match the closest objects together, e.g., the Tele Atlas road $H \rightarrow I$ shown in Figure 5.12-(b) has two promising candidates in ATKIS - road $M \rightarrow L$ and $K \rightarrow J$, and is matched to the farther one - road $M \rightarrow L$. Since the network-based matching takes the holistic sequence of the nodes into consideration, the node matching proves efficient around the cloverleaf junction, such as node *C* to *E* and *D* to *F* in Figure 5.12-(c), where the topologic connections are very complex.

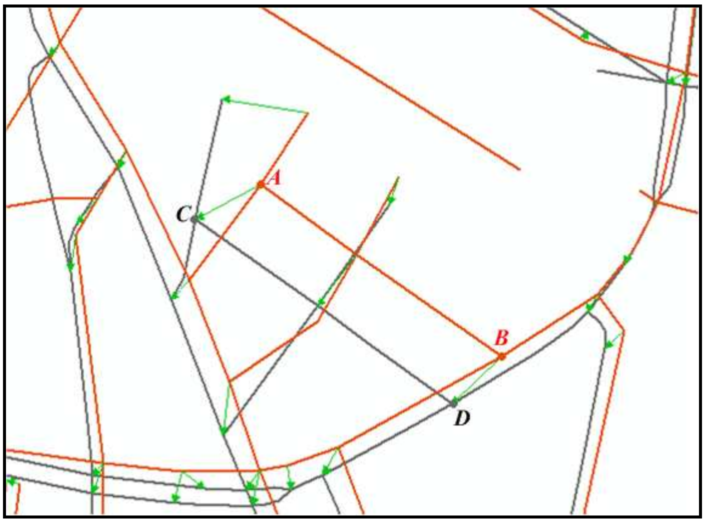
The contextual matching approach can be employed to match the streets at very different LoDs since the generalization process has been integrated to the matching calculations. Figure 5.13, for example, shows two matching cases which take place on the dual carriageways between the datasets of OpenStreetMap (OSM) and NAVTEQ, where the dual caraways are represented by parallel lines in one dataset and in the other they are just single polylines. Still they are properly matched together by the proposed matching approach.



(a) (b)

dark grey lines: OSM red lines: NAVTEQ green arrows: links

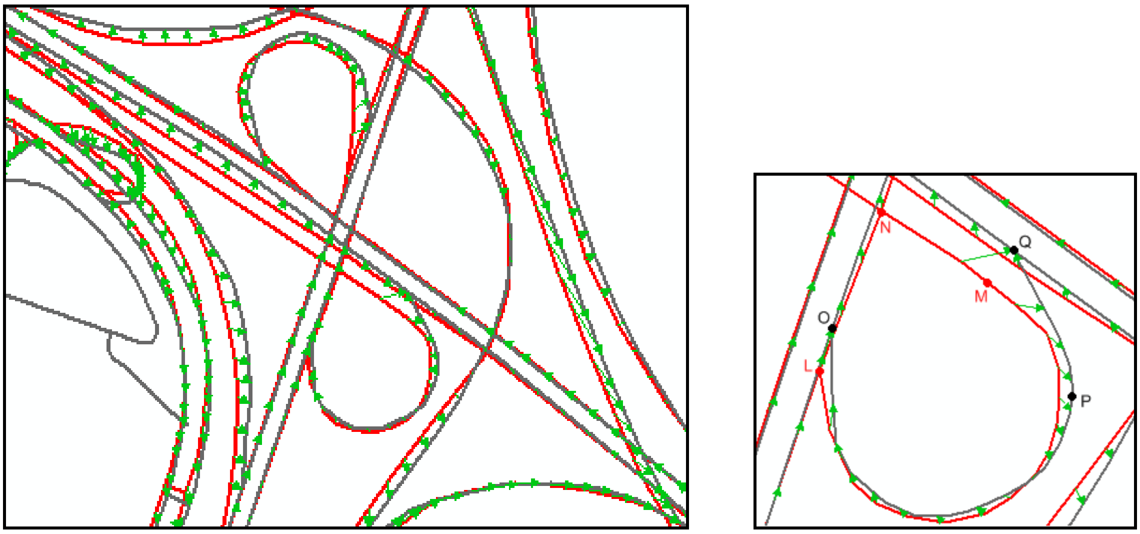
Figure 5.13 Matching of the dual carriageways at different LoDs between the datasets of OSM and NAVTEQ



dark grey lines: Tele Atlas red lines: NAVTEQ green arrows: links

Figure 5.14 A successful matching case between Tele Atlas and NAVTEQ

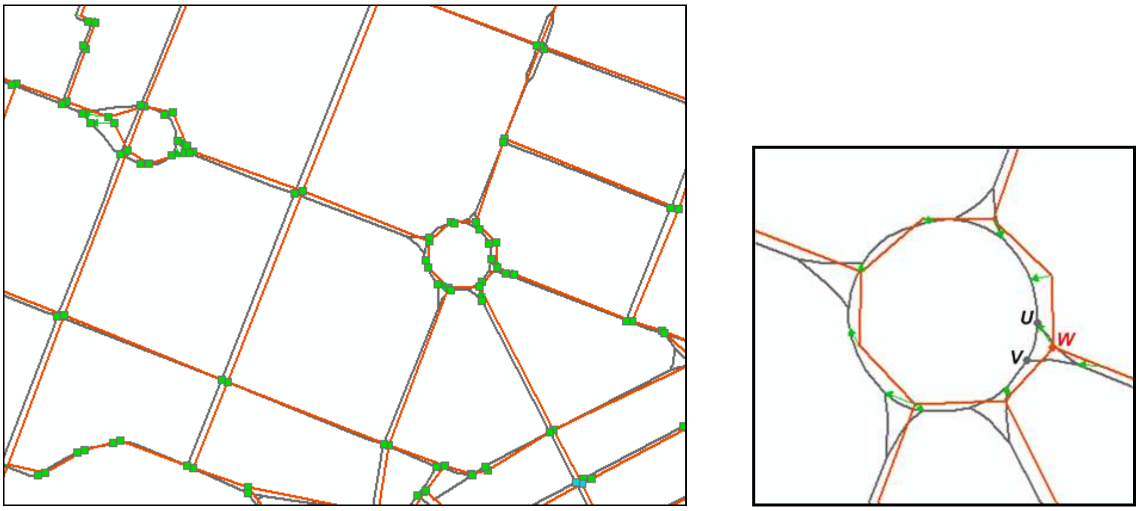
Nevertheless, the matching results are not always perfect. The contextual matching approach may be mistaken in some cases if the datasets to be matched are not sufficiently clean or can not be thoroughly preprocessed. The ATKIS slip road $A' \rightarrow B'$ in Figure5.12-a, for example, has its topologic counterpart $A \rightarrow B$ in the dataset of Tele Atlas. Unfortunately, these two roads were not matched together due to the too large distance (about 80 meters) between the node *B* and *B'*. Such occasions do not rarely occur, especially when the datasets to be matched reveal a substantial discrepancy in data accuracy. In the process of semantic matching, this kind of problem can be solved by employing looser geometric constraints, i.e. the geometric tolerance values will become larger if the comparing road objects from different dataset have quite similar semantic properties, e.g. in Figure 5.14, with the availability of the street name in both of datasets to be matched, the roads $A \rightarrow B$ from NAVTEQ has been successfully matched to the road $C \rightarrow D$ from Tele Atlas regardless the fact that they were placed very far apart. However, in some cases the corresponding road objects between different datasets could have different street names. For this reason, the matching process based on street name could also lead to mismatches if the geometric and topologic characteristics are not adequately considered.



(a) (b)

dark grey lines: ATKIS red lines: Tele Atlas green arrows: links

Figure 5.15 Topologic inconsistency between ATKIS and Tele Atlas



(a) (b)

dark grey lines: Tele Atlas red lines: NAVTEQ green arrows: links

Figure 5.16 Ambiguous corresponding nodes between NAVTEQ and Tele Atlas

Furthermore, the topologic inconsistency can influence the matching performance as well, e.g. the proposed contextual matching approach can not correctly match the roads *L→M→N* and *O→P→Q* in Figure 5.15-(b), since these two corresponding roads do not connect the same way. Figure 5.16 illustrates another typical error. The NAVTEQ node *W* has two corresponding nodes *U* and *V* in Tele Atlas. Ideally the node *W* should not be matched neither to *U* nor to *V* but to the centre point of them. However this occasion has not been considered so far in the current matching approach. In addition, the proposed matching algorithm also suffers some limitations if the topologic contexts are too complex; or if the objects in the dataset are highly fragmental.

5.2.4 Summarization of the matching performance

As depicted in Section 5.2.3, the proposed approach has been successfully implemented to match different versions of road network between (1) ATKIS and Tele Atlas, (2) ATKIS and NAVTEQ, (3) OSM and NAVTEQ and (4) NAVTEQ and Tele Atlas. Based on the statistics of the overall matching results and the discussions of some detailed matching cases, the matching performance of the proposed approach has been summarized from different points of view.

- **Reliability and accuracy**

Being supported by the extendable Delimited Strokes and network-based matching, the proposed approach is able to consider the geometric and topologic information in an extensive context, hence provide a high completeness and matching accuracy. Taking advantage of the recognition categorization and generalization of the road structures, the proposed approach also shows us a satisfying matching performance for challenging cases of roundabouts, dual carriageways (parallel lines) and complex crossings, even in areas where topologic conditions are quite complex or the road networks to be matched are represented in different LoDs. The overall matching rate of the conducted experiments in Section 5.2.2 (ref. Figure 5.2~5.6 and Table 5.1~5.5) exceeds 97%; among the matched objects more than 99.2% are correct, i.e. on average 96.6% (21948/22714) of the 22714 (21948+128+638) objects were accurately matched by the automatic procedure in the same way as manually matched reference datasets (see Table 5.6).

Matching areas	Accurate matches	Mis-matches	False positive matches	False negative matches	Matching rate	Matching correctness
Area 1	8001	52	21	228	97.2%	99.1%
Area 2	1951	14	3	87	95.8%	99.1%
Area 3	762	2	0	43	94.7%	99.7%
Area 4	892	2	1	57	94.0%	99.7%
Area 5	2244	20	4	44	97.9%	98.9%
Area 6	2460	17	13	57	97.2%	98.8%
Area 7	5638	21	0	122	97.9%	99.6%
∑	**21948**	**128**	**42**	**638**	**97.2%**	**99.24%**

Table 5.6 Statistic results of different matching experiments

As shown in Table 5.6, there are still c.a. 3% unmatched objects and 0.7% matching errors, i.e. the automatic matching results are not perfect. As mentioned in Chapter 2 (ref. Section 2.2.2), there are two causes that can lead to imperfections of the data matching performances: *algorithm limitations* and *data ambiguity.* The *algorithm limitations* can be overcome by human interactions at certain stages in the matching process. In the case of *data ambiguity*, however, it may not even be possible for a human operator to determine a correct course of action (Blasby et. 2003). In the conducted matching experiments listed in Table 5.6, *data ambiguity* is the primary inducement to generate unfavourable matching results, e.g. the matching problems often occur in areas with completely different acquisition rules or when the topologic connections are too inconsistent.

- **Effectiveness**

The time measurements showed that the contextual matching approach is very effective: e.g. in the test area depicted by Figure 5.2 (ca. 1200 km^2), there are 10947 Basis DLM objects and 10360 Tele Atlas objects in total. To match all of them, the contextual matching approach takes 22 seconds by a CPU of Intel Duo 2.0 Hz, i.e. about 500 objects per second. If we assume that an experienced human operator is able to match 5 to 6 pairs in a minute (Walter and Fritch 1999), he is able to match the data thousands times slower than the automatic approach.

This computational effectiveness has much to do with the employment of the grid-based spatial indexes for points and lines introduced in Section 4.3. With help of such a grid-based system, the exponential computational complexity of the search problems can be dramatically reduced. For instance, in the process of selecting the initial spots depicted by Section 3.3.1, in order to find the nodes within a tolerance distance *d* to a given node *A* from the reference dataset, all nodes in the target datasets have to be calculated if the spatial index for points is not used, that is N_{Node} computing cycles in total where N_{Node} represent the node number of objects in target dataset. When the grid-based index is utilized, however, the computing cycles can be cut to $9^* N_{Node} / N_{Grid}$, where N_{Grid} is the total number of grids (viz. square blocks in Section 4.3). The factor 9 is based on the assumption that the maximum search range is equal to or less than the side length of the cell, so a maximum of nine cells will be traversed in each of the searches (Xiong 2000). Take a matching area with 10*10 km^2 as an example, if the size of the each grid is 50*50 m^2, then the whole matching region will be divided into (10000/50)^2=40,000 square blocks and thus the searching time becomes $40000/9 \approx 4.44\times10^3$ times shorter.

With the use of the spatial indexes described in Section 4.3, an increase in network size will not significantly slow down the matching speed. As can be seen in Figure 5.17, the results demonstrate that the computing time has a linear relationship to the amount of the road objects falling inside the matching areas: the sample data (crossing points) distributed along the straight line of *y=0.002x+0.3325* where the *y* and *x* represent the computing time and the number of the reference objects respectively; and hereby the overall matching speed can be approximately measured by 1/0.002 = 500 objects/second.

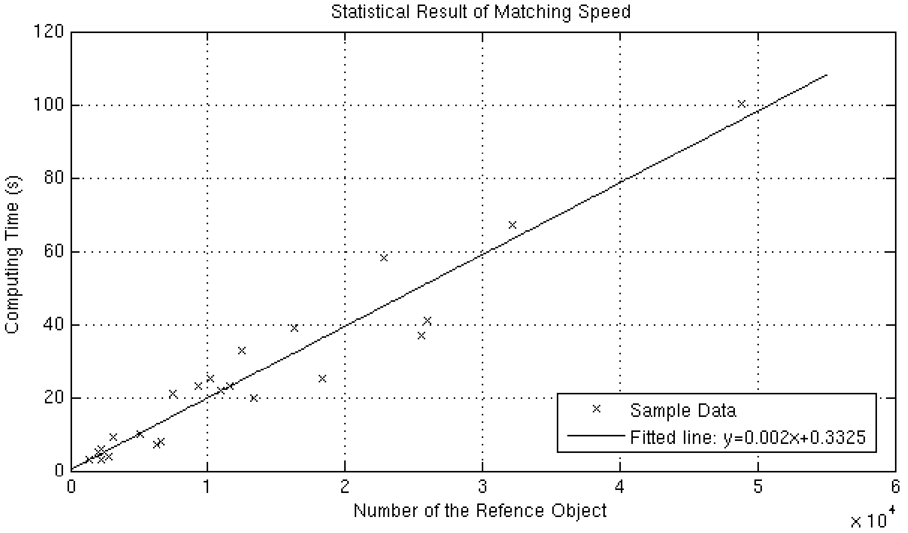


Figure 5.17 The computing speed of various matching experiments

As illustrated in Figure 5.17, the computing speed did not keep stable among the conducted matching experiments: in some cases it is more than 800 objects/seconds whereas in some other cases it is below 400 objects/seconds. To our knowledge, there are at least four causes which may result in such a fluctuating of the matching speed: (1) the status of the computer, e.g. the temperature of the CPU; (2) one object could have different number of shape points from the other - the more the shape points, the lower the matching speed; (3) one object could be much longer than another - the longer the object, the more computing time it needs; (4) as depicted in Chapter 3, the proposed DSO algorithm involves an iterative matching process. Hence, if the corresponding road objects reveal similar geometry and topology between the datasets to be matched, most of the corresponding objects will be calculated after the first iteration, thus only a few objects will be taken into account in the second or third iteration. Otherwise, if the corresponding objects have dissimilar geometric or topologic characteristics, a number of road objects would be remained as ‘unmatched’ after the first matching iteration, which have to be further processed in the second and third matching iterations and therefore claim longer computing time.

Nevertheless, the extremely large networks, such as the whole cities like Berlin and Hamburg in Germany, may overwhelm computer’s physical memory if they are loaded at once. In these cases, special care must be taken (Xiong 2000). One way to solve this problem is to divide a large database, if it exceeds a predefined tolerance size, into several sub areas, thus limit the physical memory intensity. Keeping in mind that the region division may cause unwanted border effects, we set up a buffer surrounding each sub area. The actual searching scope is thus a little bit larger than each individual sub area. Such a division also facilitates the “Unsymmetrical data matching” introduced by Zhang and Meng (2007).

- **Generic nature**

Worthwhile to mention is also the generic nature of the proposed matching approach: it can work with the worst case - one or both of the datasets to be matched have no or little semantic information (e.g. ATKIS and OSM data). The conducted matching experiments proved that this approach can be used to match corresponding road networks among a variety of datasets, incl. ATKIS, Tele Atlas, OSM, NAVTEQ etc. As the proposed DSO matching algorithm does not depend on any semantic attributes (ref. Chapter 3), the proposed approach can be also utilized in the same way for other road-network data models, such as U.S. Tiger etc. Moreover, the gained insight from road matching can therefore be easily adapted to the task of matching other linear features like hydrological networks and electronic pipe lines.

Although the DSO algorithm is principally insensitive to the amount of semantic information, it is open-ended and can be very well combined with the semantic-guided matching. Thus, the matching accuracy and reliability can be enhanced. For instance, the contextual approach revealed a nearly perfect matching result between Tele Atlas and NAVTEQ in Section 5.2.2.4. Nevertheless, the contextual matching approach is essentially based on comparisons of the geometries and topologies between different datasets while the matching guided by semantic attributes such as road names or functional road classes is just an optional operator.

- **Robustness**

As illustrated earlier, the contextual matching approach has been successfully tested on a large number of matching areas in Germany and worked smoothly for the different matching tasks between (i) NAVTEQ and ATKIS, (ii) Tele Atlas and ATKIS, (iii) OpenStreetMap and NAVTEQ, and (iv) Tele Atlas and NAVTEQ.

Taking advantage of its robust matching performance, the proposed approach acquires capabilities for the real-world geospatial data enrichments, e.g. the developed matching program has been applied by Corp. United Maps to conflate the motorways and pedestrian roads from different datasets for the purpose of multi-modal navigation (see detailed illustration in Section 6.3). Up to date, the Corp. United Maps has completed the data conflation between NAVTEQ and ATKIS in whole Germany: the total matching area is more than 300,000 km^2 with ca. 6.7 million NAVTEQ features and 15.4 million ATKIS objects inside.

In general, the contextual matching approach shows a high automatic degree, accurate, speedy, robust and generic performance on different matching applications, which indicates that more than a significant matching prototype the developed matching program has the potentialities to be extended to a commercial software production.

5.3 Assessment of the matching quality

As depicted in Section 5.2, the contextual matching approach has yielded a quite positive performance in the conducted matching experiments: the overall matching rate exceeds 97%; among the matched objects more than 99.2% are correct. However, a few unfavourable matches are inevitable in most matching approaches. Although less than 3% unmatched objects (viz. false negative match) and circa 0.8% matching errors (incl. mismatch and false positive match) can be thought of as a good result compared to the complexity of the networks, it still represents a perplexing manual work for human interactive editing. In practice the matching errors are much more intractable than the unmatched objects. This can be explained by the fact that, for the time being, the unmatched objects can be very easily marked as doubtful, whereas the process to detect the errors is time consuming and labour intensive, because erroneous matches need to be analysed one by one. Moreover, even if a match is visually correct, it is still coupled with a degree of uncertainty.

Hence, in order to minimize the costs of human interaction, the matching quality is accessed by classifying the matching results into different certainty levels, where the measurement of matching certainty is defined by the '*MCertainty*' in Equation [5-4].

$$MCerta\mathrm{int}\,y(PL_1, PL_2) = Similarity\ (PL_1, PL_2) - \Delta Semantics\ (PL_1, PL_2) - (N-1) \times \xi_{Penalty}$$

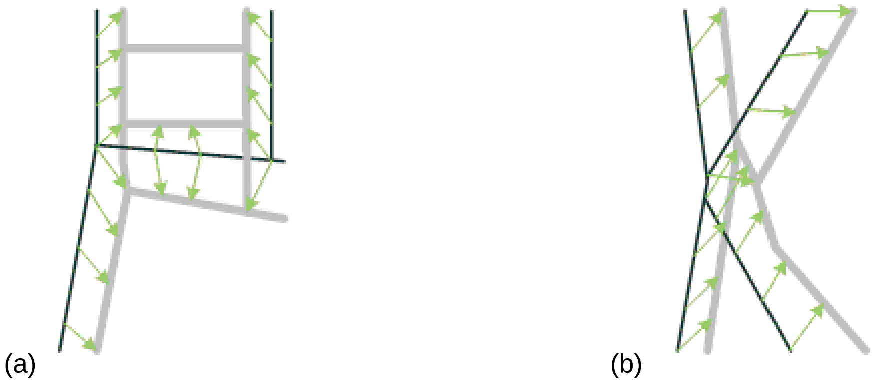
… [5-4]

The certainty of a matching pair is based upon the variable $Similarity(PL_1, PL_2)$ defined by expression [3-14]; meanwhile the impact of semantic difference $\Delta Semantics(PL_1, PL_2)$, which concerns on street name, street width, number of lanes, travel time, direction of traffic flow etc. (ref. Section 4.2) has to be also calculated when the streets to be compared are both fulfilled with the corresponding attribute values.

Moreover, the matching has a reduced reliability if the reference polyline has more than one promising candidate after the matching process of Section 3.4.2. Hence the value of certainty of the ultimate matching result should be subtracted by a term according to $(N-1) \times \xi_{Penalty}$ where N means the number of promising matching candidates and $\xi_{Penalty}$ is a user-defined penalty coefficient. The variable *MCertainty* can be scaled to a number between 0 and 1 - the smaller the value the more likely it is that the matching pair constituted by PL_1 and PL_2 is wrong.

In addition, the matching conflicts can also infract the certainty of the identified matching pair. If two or more line objects from the reference dataset correspond to a single object from the target dataset as shown in Figure 5.18-a, they are said to be in conflict because the matching usually means a one-to-one relation. To note that the matched streets which reveal dissimilar LoDs in different datasets (ref. Section 4.1.1.3 and 4.1.2.4) are not necessary to be considered as matching conflicts. Besides, the overcrossing linkages have also to be considered as matching conflicts (see an example in the central part of Figure 5.18-b).

For conflicting matches, the problem can be firstly solved by comparing the values of variable *MCertainty*: the matching pair with the largest *MCertainty* is regarded as the best match, whilst all the others will be rejected; after that the variable *MCertainty* will be tuned to *Min{MCertainty,0.15}*, where 0.15 is an empirically defined constant which indicates a very low matching reliability.



grey: dataset 1; black: dataset 2; arrows: linkages

Figure 5.18 Examples of the matching conflicts

Using the contextual matching approach in three different experiments with the total matching area of ca. 1400 km^2, more than 19000 objects from the reference dataset are matched to their corresponding objects in the target dataset, among which 101 matching errors are detected manually, incl. mismatch and false positive match. Then, we conducted a matching certainty analysis on the overall matched pairs and the falsely matched pairs respectively. Based on the statistical work, the distribution of *MCertainty* (Matching Certainty) as well as its classification is illustrated in Figure 5.19 and Table 5.7.

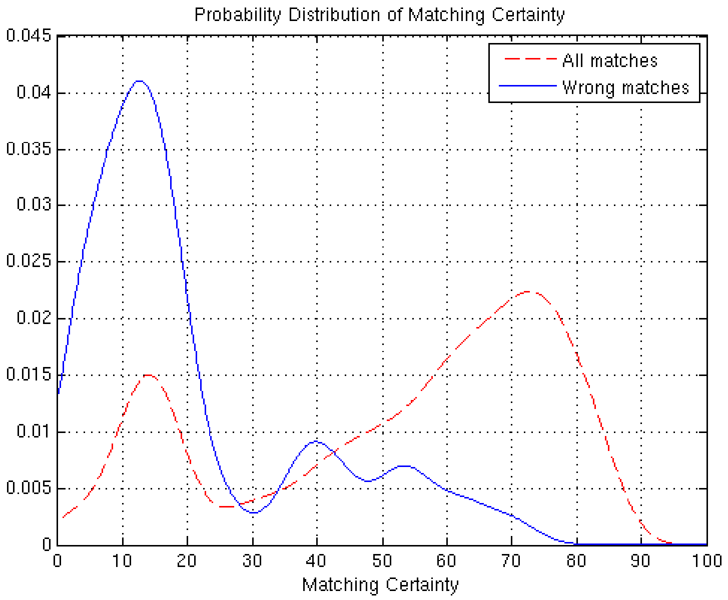


Figure 5.19 Probability distribution of Matching Certainty

	Matching Certainty	[0.00, 0.20]	(0.20, 0.70)	[0.70, 1.00]	∑
Density distribution	Falsely matched objects	80 (79.21%)	20 (19.80%)	1 (0.99%)	**101**
	Overall matched objects	3454 (18.06%)	10174 (53.19%)	5498 (28.75%)	**19126**

Table 5.7 Distribution of Matching Certainty

As shown in Figure 5.19 and Table 5.7, among the falsely matched objects, around 80% have a Matching Certainty of lower than 0.2, whilst nearly none of them reveals a Matching Certainty larger than 0.70, which indicates: (a) A wrong match can be associated with a very small Matching Certainty ($\leq$0.2); (b) A match with a Matching Certainty of over 0.70 can be confirmed as a correct match.

Classification	possible	good	perfect
Matching Certainty	[0.00, 0.20]	(0.20, 0.70)	[0.70, 1.00]

Table 5.8 Classification of the Matching Certainty

According to this rule, the author suggests to classify the matching results into three certainty levels (see Table 5.8):

Level 1 contains all the matching pairs with the Matching Certainty lower than 0.20. Approximately 18% of all matching pairs of the test areas and 80% of the wrong matching pairs are within this class, which indicates most of the matching errors are falling inside this class with the Matching Certainty ≤ 0.20. Therefore the matching pairs within this class are classified as ‘possible’.

Level 2 contains all the matching pairs which have the Matching Certainty between 0.20 and 0.70. This class contains approximately 53% of all matching pairs which are classified as ‘good’. Wrong matches are very few but still appear in this class - the matching accuracy is about 99.8% (1-20/10174=99.8%).

Level 3 contains the remaining 29% of all matching pairs which have the Matching Certainty higher than 0.70. These matching pairs are treated as ’perfect’ matches due to the fact that among the 19126 matched pairs as mentioned in Table 5.7, there is only one error detected within this class. The accuracy of the matching pairs within this class exceeds 99.98% (1-1/5498=99.98%).

As illustrated above, the definition of the Matching Certainty and the concomitant classification of the matching results can simplify the process of detecting the matching errors as most of the matching errors are concealed at the certainty level of ‘possible’. Thus, the tedious manual post-processing work to interactively refine the automatic matching results will be highly reduced.

5.4 Statistical analysis on geometric deviations

In the proposed DSO matching algorithm, several geometric constraints have been employed to exclude unlike matching pairs, which requires several pre-defined tolerance values with respect to:

- the ‘distance’ between the corresponding nodes, ref. T_d in Section 3.4.1;
- the ‘orientation’ difference between the linear counterparts, ref. $\Delta\theta_{tolerance}$ in Equation [3-2];
- the ‘length’ difference between the linear counterparts, ref. $\Delta l_{\max}$, $\Delta l_{\min}$ and $\Delta r(l)_{tolerance}$ in Equation [3-3];
- the ‘average area’ difference between the linear counterparts, ref. $\Delta AvS_{tolerance}$ in Equation [3-4];
- the ‘shape’ difference between the linear counterparts, ref. $L_p\|\cdot\cdot\|_{tolerance}$ in Equation [3-6];
- the ‘location’ difference between the linear counterparts, ref. $d_{AV\,tolerance}$ in Section [3-9];

These values may vary with the data models and can not be formulated in a universal way. In order to estimate the overall deviations between different datasets, a series of statistic investigations on various matching tasks are conducted, which can help to assign proper user-defined tolerance values for the different geometric constraints listed above. In the following, three matching areas are randomly selected for the investigations:

(1) Matching between Tele Atlas and ATKIS in an area (ca. 100 km^2) of Hessen, Germany, with 2421 road objects in ATKIS and 2523 objects in Tele Atlas;

(2) Matching between ATKIS and NAVTEQ in a rural area (ca. 100 km^2) of Garmisch, Germany, with 1762 objects in ATKIS and 571 objects in Tele Atlas;

(3) Matching between ATKIS and NAVTEQ in a built-up area (ca. 100 km^2) of Immenstadt, Germany, with 5011 objects in ATKIS and 2214 objects in NAVTEQ.

After running the contextual matching approach in these three areas, an interactive post-processing is triggered to refine the automatic matching result, add new matches as well as correct the falsely identified matching pairs. Based on the refined matching result, a series of statistic analysis are conducted, so that the frequency distribution of the geometric deviations between corresponding counterparts in different datasets can be calculated.

5.4.1 ‘Distance’ between corresponding nodes

As depicted in Figure 5.20, the ‘distances’ between corresponding nodes have dissimilar frequency distributions in different matching experiments. For instance, in experiment (1), the average distance between the corresponding nodes is around 7 meters, whereas in experiment (2) and (3), such average distances are equal to 12 and 14 meters respectively.

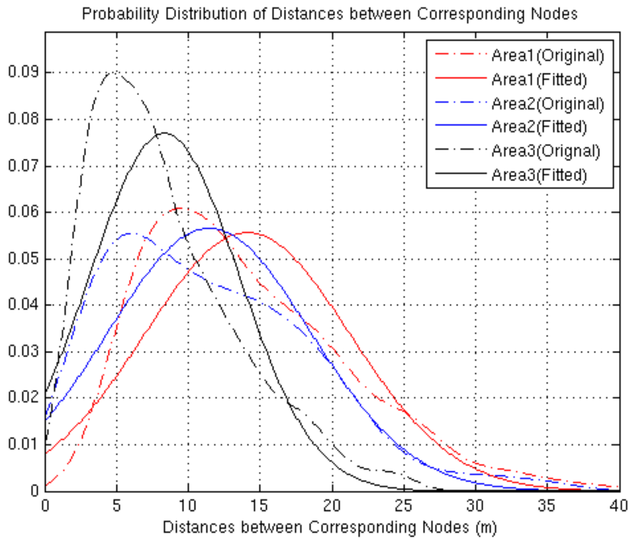


Figure 5.20 Probability distribution of the ‘distance’ between corresponding nodes

5.4.2 ‘Location’ difference

The ‘location’ difference of the corresponding roads can be measured by the ‘average distance’ between the matched linear objects, which reveals dissimilar frequency distributions in different experiments either (see Figure 5.21).

As illustrated by Figure 5.20 and 5.21, the ‘distance’ between corresponding nodes and the ‘location’ difference between the matched roads do not show similar frequency distributions in different matching experiments, which indicates that the tolerance values of these two parameters are quite case-dependent. Bearing in mind this characteristics, the tolerances of T_d and $d_{AV\,tolerance}$ are interactively given by the human operator.

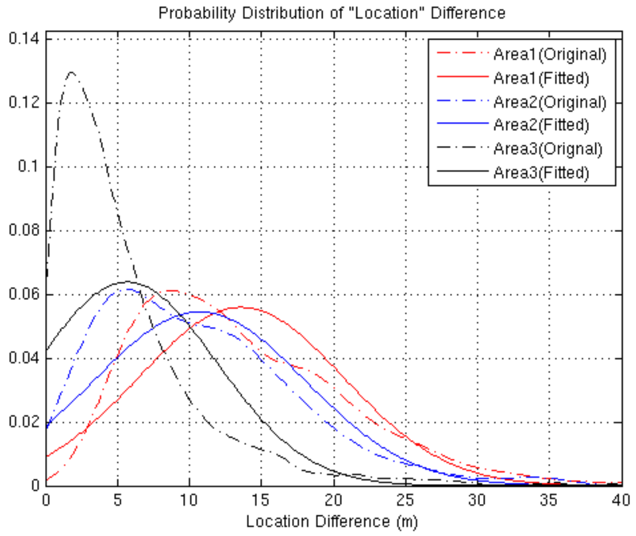


Figure 5.21 Probability distribution of the ‘location’ difference between corresponding road objects

5.4.3 Differences of ‘orientation’, ‘shape’ and ‘average area’

The orientation can be calculated by the starting point and ending point of a line: the larger the orientation difference between two roads, the more unlikely they match with each other. In our matching program, the tolerance of ‘orientation’ difference, viz. $\Delta\theta_{tolerance}$ in Equation [3-2], has been empirically settled as 25°, due to the fact that (i) different matching experiments have revealed similar frequency distributions on the ‘orientation’ differences – the distributions curves wrapped to each other; and (ii) in all of the conducted experiments, almost all of the matching pairs have an orientation differences less than 25°, see Figure 5 .22.

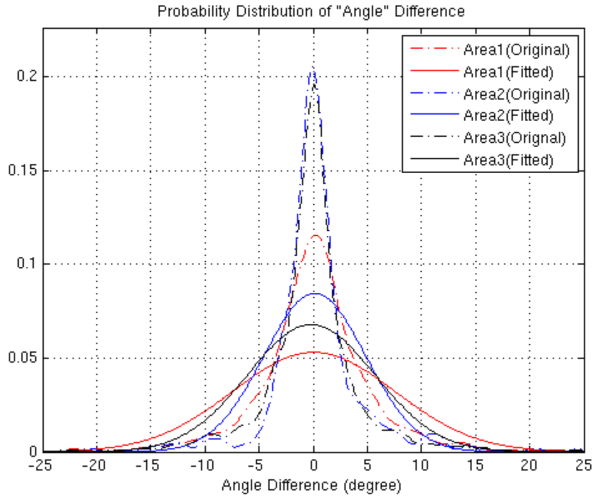


Figure 5.22 Similar frequency distribution of the ‘orientation’ difference on different matching experiments

Likewise, the tolerance of ‘shape’ difference (ref. $L_p\|\cdot\cdot\|_{tolerance}$ in Equation [3-6]) and the tolerance of the ‘average area’ difference (ref. $\Delta AvS_{tolerance}$ in Equation [3-4]) can also be empirically determined as constant values since the ‘shape’ and ‘average area’ difference between the corresponding road objects revealed similar frequency distribution in different experiments as well, see Figure 5.23 and Figure 5.24.

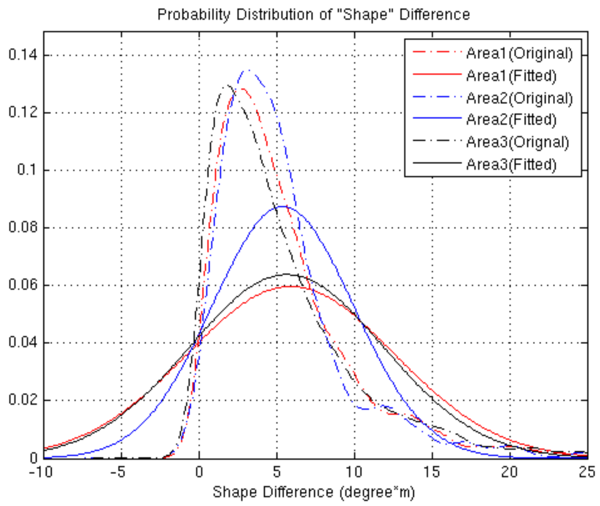


Figure 5.23 Similar frequency distributions of the ‘shape’ difference on different matching experiments

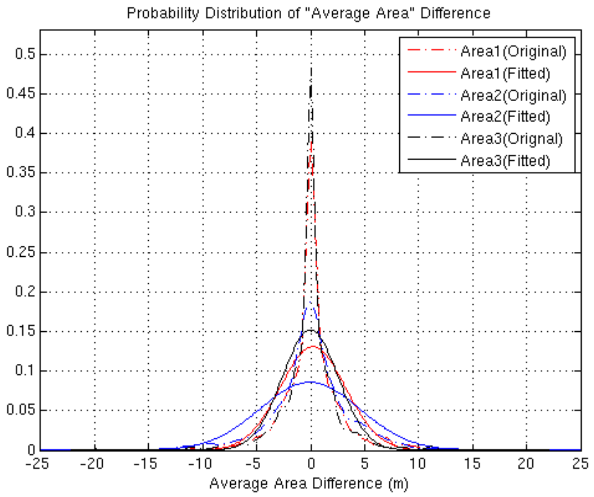
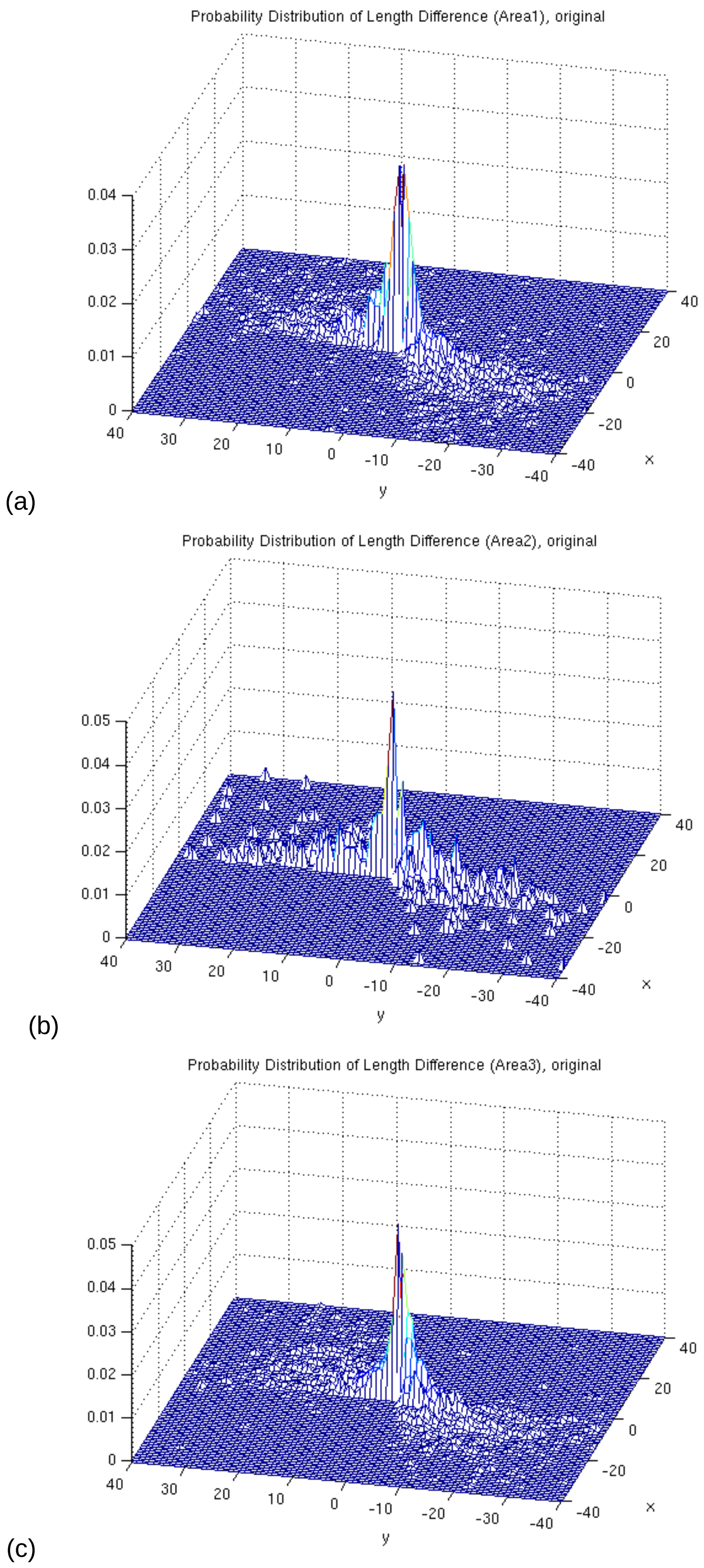


Figure 5.24 Similar frequency distributions of the ‘average area’ difference on different matching experiments

5.4.4 Differences of ‘length’

Figure 5.25 (a)~(c) illustrate the frequency distribution surfaces (3-dimension) of the ‘length’ difference of the corresponding road counterparts on different matching experiments. In order to properly measure the length difference between the matched roads, not only the absolute value of the difference but also the ratio divided by the length difference and the length summation have to be considered, see the X and Y -axis in Figure 5.25.

As illustrated in Figure 5.25, the ‘length’ differences also have quite similar frequency distributions on different matching experiments. Taking into account this characteristic, the tolerant parameters of $\Delta l_{\max}$, $\Delta l_{\min}$ and $\Delta r(l)_{tolerance}$ in Equation [3-3] can be settled as constant empirical values in the matching program.



X-axis: absolute value of the ‘length’ difference; Y-axis: ratio divided by the absolute value of ‘length’ difference and the summation; Z-axis: probability distribution

Figure 5.25 Similar distribution surfaces of the ‘length’ difference on different matching experiments of (a) Area 1, (b) Area 2 and (c) Area 3.

Chapter 6

Implementations of the Matching Approach

As compared with the traditional matching algorithms such as Buffer Growing and Iterative Closest Point, the contextual matching approach has yielded a considerably improved performance in terms of (a) Automatic matching reliability and accuracy: in a number of large test areas in Germany, the overall matching rate exceeds 95%; among the matched objects more than 99% are correct; (b) Robustness and generic nature: the DSO matching algorithm does not rely on any semantic information, therefore, it can be employed to match all kinds of road networks from different data resources; the investigated datasets consist of Tele Atlas, NAVTEQ, ATKIS, OpenStreetMap etc., and the test matching areas exceed 300 000 km^2 which involves more than 20 million objects in total; (c) High computing speed: e.g. in a test region of ca. 1 200 km^2, there are 10 959 ATKIS objects and 10 681 Tele Atlas objects. The matching has taken only 22 seconds in a normal PC (personal computer) with the CPU of Intel Duo 2.0Hz, i.e. about 500 objects per second. This computing speed remains stable for much larger matching areas.

Based on the performance the contextual matching approach has the substantial capability to handle mega data. So far it has been successfully implemented in three practical cases:

- Postal data integration;
- Integration of the routing-relevant information from different datasets; and
- Conflation of the pedestrian ways between different datasets.

6.1 Case 1 - Postal data integration

The project “Postal data integration” was sponsored by German Federal Agency for Cartography (BKG) and aims at enriching the Basic Digital Landscape Model (Basis DLM*)* with geo-referenced house numbers of post addresses, where the Basis DLM is one of the sub-datasets of ATKIS (see detailed illustration of ATKIS in Section 5.1.1). The available post addresses are from German Federal Post Office, which were manually collected by postmen on the basis of road geometries from Tele Atlas Corp. (see detailed illustration of Tele Atlas in Section 5.1.1). The individual house numbers are stored as discrete points distributed along the two road sides. The location of each individual house number was estimated or interpolated by the postman based on the beginning and terminating house number delimiting each street segment and the known house-numbering rules. For this reason, it may deviate more or less from the true location of the corresponding house entrance, but the topologic relationship is preserved. Since the road database of Tele Atlas reveals a different geometric / semantic accuracy from that of Basis DLM, the attempt of a direct integration of postal data into Basis DLM is doomed to fail. Therefore, we divide the enrichment process into two main stages: the first stage is dedicated to the matching of road objects between Basis DLM dataset as reference and Tele Atlas dataset as target. In the second stage, a projection based on the Rubber-Sheet principle is established for each pair of matched road lines. Thus, the discrete locations of house numbers along a Tele Atlas road line can find their corresponding positions along the homologous line in Basis DLM.

This section is focused on the second stage as the methodology of data matching has been elicited in detail (ref. Chapter 3~5). As depicted in Figure 6.1, after the data matching between Tele Atlas and Basis DLM, two processes are employed to accomplish the task of postal data integration:

(a) establishment of linkages from Tele Atlas to Basis DLM; (b) alignment of the house numbers based on the Rubber-Sheet Transformation.

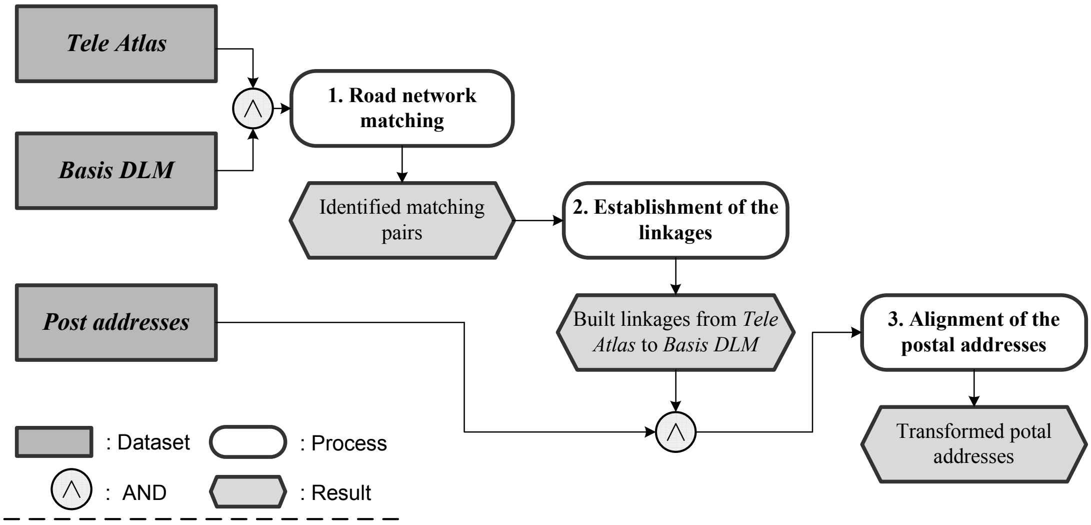
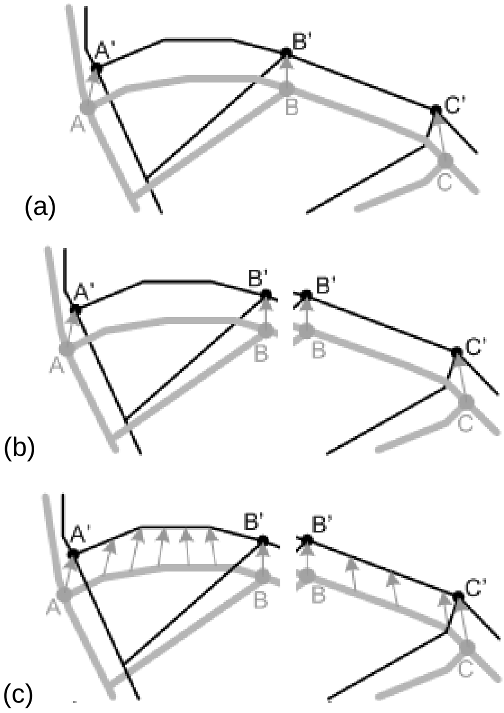

Tele Atlas
Basis DLM
Post addresses
1. Road network matching
Identified matching pairs
2. Establishment of the linkages
Built linkages from Tele Atlas to Basis DLM
3. Alignment of the postal addresses
Transformed potal addresses
: Dataset
: Process
: AND
: Result

Figure 6.1 The strategy of postal data integration

6.1.1 Establishment of linkages from Tele Atlas to Basis DLM

In order to interactively visualize the matching results and align the postal data from Tele Atlas to Basis DLM, it is necessary to establish the linkages between the matched road lines. The linkages can be generated with the following steps:



black lines: Basis DLM grey lines: Tele Atlas arrows: linkages

Figure 6.2 Establishment of the linkages between Tele Atlas and Basis DLM

a) Find out the corresponding nodes with *Valence >=3* between two datasets, see examples of point *B & B'* in Figure 6.2-a;

b) Split the polyline into smaller sections with such nodes as terminating points, e.g. in Figure 6.2-b, the matched roads pair of *A→C & A'→C'* has been divided into two parts, one is *A→B & A'→B'* and the other is *B→C & B'→C';*

c) Match the turning points in every divided section by means of interpolation; see Figure 6.2-c.

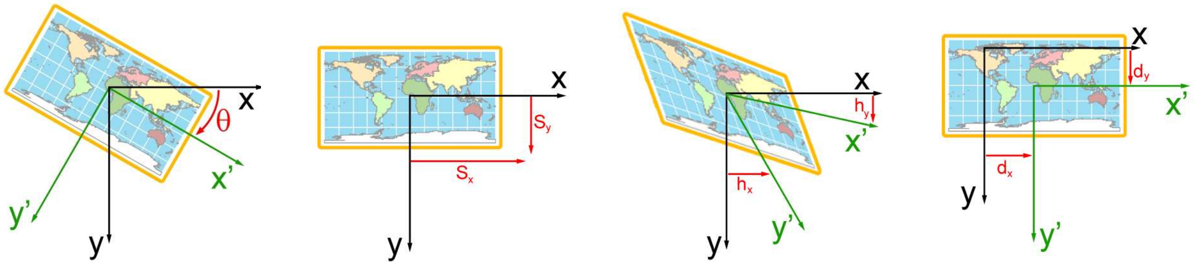
Since the contextual matching approach works in ArcGIS (ref. Chapter 5), it allows the operator to interactively refine the established linkages, add new linkages between corresponding points as well as correct the false ones. In addition, the definition of the matching certainty and the concomitant classification of the matching results can simplify the process of detecting the wrongly built linkages, as most of the errors are concealed at the certainty level of 'possible' (ref. Section 5.3). After the interaction, a desirable matching result can be reached.

6.1.2 Alignment of the postal addresses based on the linkages

For the tasks of feature alignment, *Affine Transformation* and *Rubber-Sheet Transformation* are two of the most popular and well-known algorithms reported in literature hitherto, see details below:

- **Affine Transformation**

An Affine Transformation is an important class of linear geometric transformations which maps variables into new variables by applying a linear combination of translation, rotation, scaling and/or shearing operations (Fisher et al. 2003; Unser et al. 1994).



(a) Rotation (b) Scaling (c) Shear (d) Translation

Figure 6.3 Affine Transformation

In a 2D environment, one Affine Transformation can be expressed as a transformation that fixes some special point (the 'origin') followed by a simple translation of the entire plane. Given a point $P=(p_x, p_y)$, for example, its new position $P'=(p_x', p_y')$ after the Affine Transformation can be calculated by a matrix multiplication as:

$$\begin{pmatrix} p_x' \\ p_y' \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix}$$

…[6-1]

Where, (1) $\begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} + \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} + \begin{pmatrix} 1 & h_x \\ h_y & 1 \end{pmatrix}$ represents a summation of the transformations of Rotation $R(\theta)$ (see Figure 6.3-a), Scaling $S(s_x, s_y)$ (see Figure 6.3-b) and Shear $H(h_x, h_y)$ (see Figure 6.3-b); and (2) $\begin{pmatrix} m_{13} \\ m_{23} \end{pmatrix}$ carries out a pure translation, see an example in Figure 6.3-d.

Since the general Affine Transformation is defined by 6 constants, it is possible to define this transformation by specifying the new output image locations of any three input control coordinate pairs (Fisher et al. 2003). In practice, many more points are measured and *a least-squares method* can be used to find the best fitting transformation (Markovsky and Mahmoodi 2009).

- **Rubber-Sheet Transformation**

A Rubber-Sheet Transformation, also called Rubber-Sheeting, is a mathematically defined function between regions which are divided into sub-regions. Each sub-region in the first has a unique counterpart in the second. Also, every point and line of the boundaries and the sub-regions themselves maintain their relative positions from the first region to the second, i.e., topology is preserved. The transformation is specified in pieces - a specific sub-transformation for each sub-region. Each sub-region is transformed differently, much like taking a piece of rubber and stretching it in sections to make it fit over some object (Gillman 1985; White and Griffin 1985). The Rubber-Sheet Transformation is often used to make small geometric adjustments in the data, usually to align features with more accurate information.

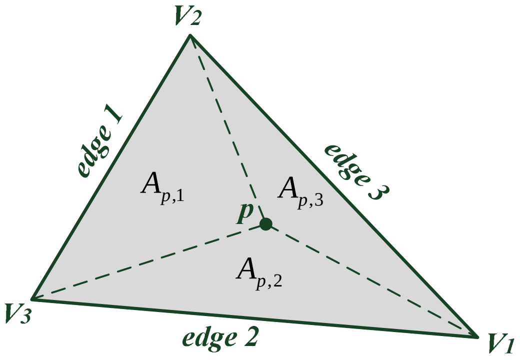


Figure 6.4 Rubber-Sheet Transformation

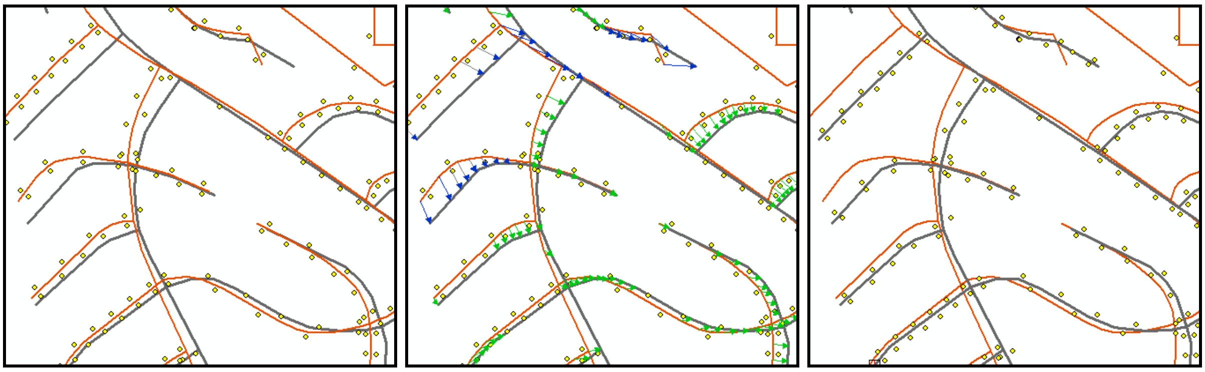
Triangulation is a term that can refer to either the actual triangular regions or the method that is used to generate the regions. One such triangulation method is the Delaunay triangulation, considered to be the *'best'* triangulation for Rubber-Sheeting (Cobb et al. 1998). The Rubber-Sheet Transformation, based on triangulation, is easy to compute. The method and underlying theory can be summarized as the following facts (Gillman 1985). For details, see (Saalfeld 1985).

1) Any point $p = (p_x, p_y)$ inside a triangle T can be expressed as $p = a_1 \cdot V_1 + a_2 \cdot V_2 + a_3 \cdot V_3$, where $a_1 + a_2 + a_3 = 1$, and $V_i = (V_{i,x}, V_{i,y})$ are the vertices of the triangle. a_i with $i = 1,3$ and $a_i \geq 0$ are convex coefficients of p .
2) For any point p in a triangle T , the *i-th* convex coefficient of p is $a_i = A_{p,i}/A_T$, where $A_{p,i}$ is the area of the triangle with the *i-th* edge and p, see Figure 6.4; A_T is the area of the triangle T . $A_{p,i}$ is non-negative if the vertices of the triangle T are in counter clockwise order.
3) Let T' with the vertices of V_1', V_2', V_3' be the triangle corresponding to T in the other map. Then the image of p is $p' = a_1 \cdot V_1' + a_2 \cdot V_2' + a_3 \cdot V_3'$.

After comparing the performance of different transformation algorithms listed above, the Rubber-Sheeting has been employed to do the alignment of the post addresses: the starting and ending point of the established linkages are treated as *control point pairs (CPPs)*; using the refined CPPs after human interaction, a geometric transformation based on the Rubber-Sheeting principle can be done; in this way, any arbitrary points along Tele Atlas road lines, e.g. the post addresses, can find their corresponding positions in Basis DLM.

6.1.3 Results of the postal data integration

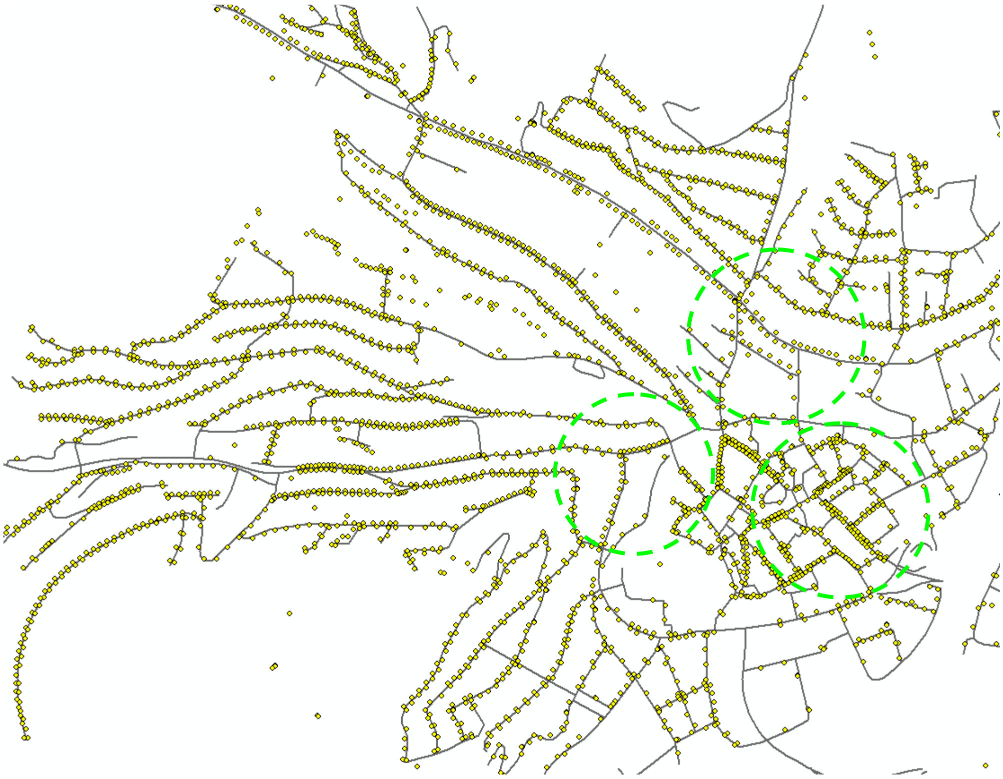
As depicted in Figure 6.5, with the availability of the established linkages (arrows), the house numbers of post addresses (yellow points) have been properly transferred from Tele Atlas (red lines) to Basis DLM (grey lines) after the Rubber-Sheet adjustment.



black lines: Basis DLM red lines: Tele Atlas points: postal data arrows: linkages

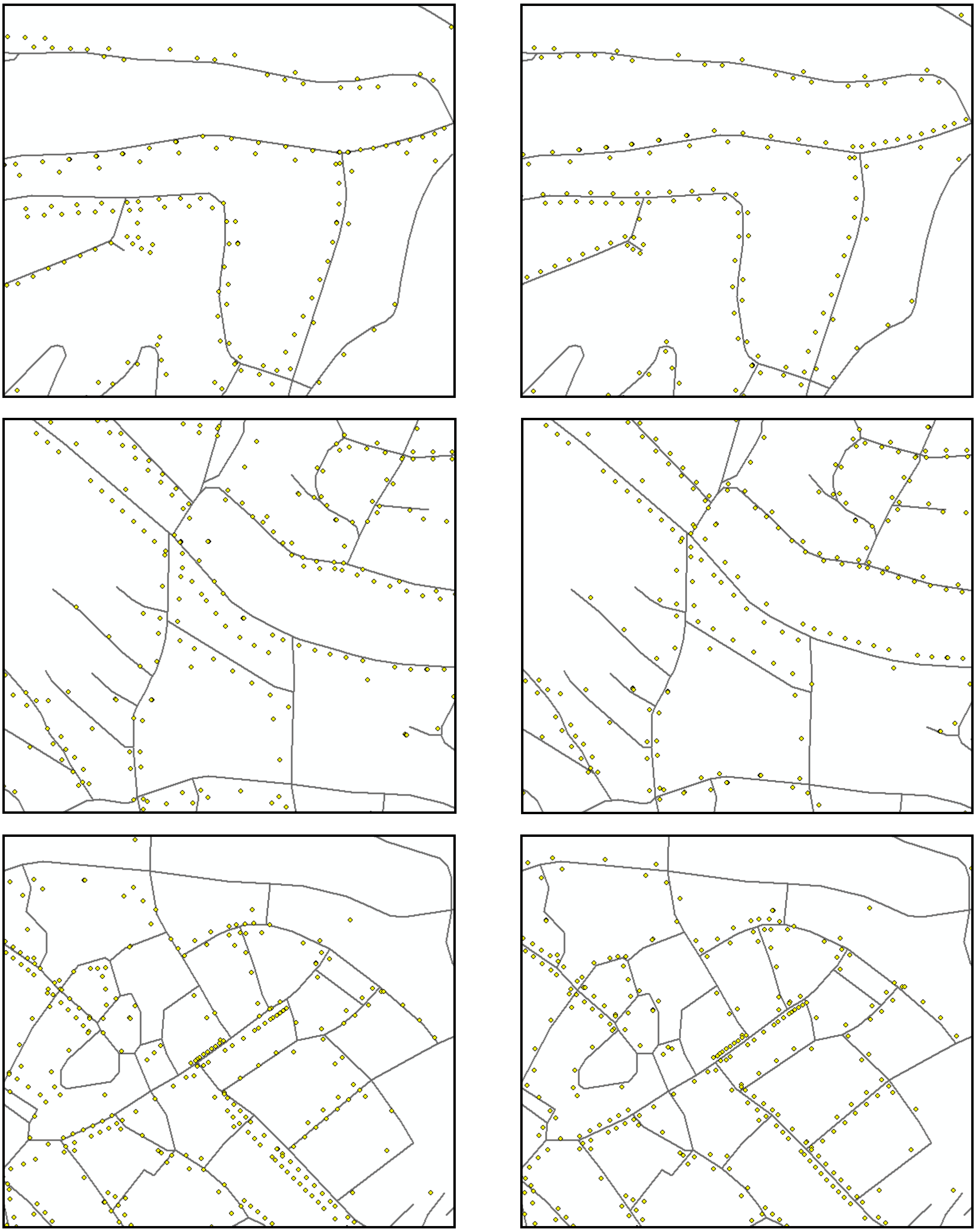
Figure 6.5 Integration process of postal data: superimposition of three datasets (left), road linkages between Tele Atlas and Basis DLM (middle), transfer based on the Rubber-Sheet principle (right)

The contextual matching approach has been successfully applied for the real-world data enrichment in the federal state of Hessen, Germany. Figure 6.6 shows the enriched dataset of Basis DLM with the transformed post address in an area of Hessen, Germany, where three enlarged sections are illustrated in Figure 6.7.



black lines: Basis DLM yellow points: postal data

Figure 6.6 The result of the postal integration: Basis DLM with the transformed postal data



(a) Basis DLM with initial post data (b) Basis DLM with transformed post data

black lines: Basis DLM yellow points: postal addresses

Figure 6.7 Three enlarged sections of Figure 6.6

6.2 Case 2 - Integration of the routing-relevant information from different datasets

With the growing demand on multi-purpose navigations, the route calculation becomes more and more complex. It is no longer limited to the search for the shortest route from one point to another. Rather it aims at identifying *'optimal'* ways with regard to multiple objectives or comprehensive criteria such as time, energy consumption, cost, convenience and comfort, scenic spot etc. Unfortunately, the currently operational route planning algorithms reveal rather limited performances due to unavailable or insufficient interoperation among the underlying data that are separately maintained in different spatial databases. Accordingly the study of integrating the routing-relevant information from different datasets becomes increasingly necessary because in many cases a dataset from one single source is insufficient for the calculation of the *'best'* route. With the frame of the project "Development of a method for the construction of a routing-capable street database in medium-scale range" funded by German Federal Agency for Cartography and Geodesy (BKG), the contextual matching approach was implemented to transfer the routing-relevant information from Tele Atlas to DLM De*.* Similar to Basis DLM, DLM De is a sub-dataset of the evolving ATKIS. The detailed illustrations of ATKIS and Tele Atlas can be found in Section 5.1.1.

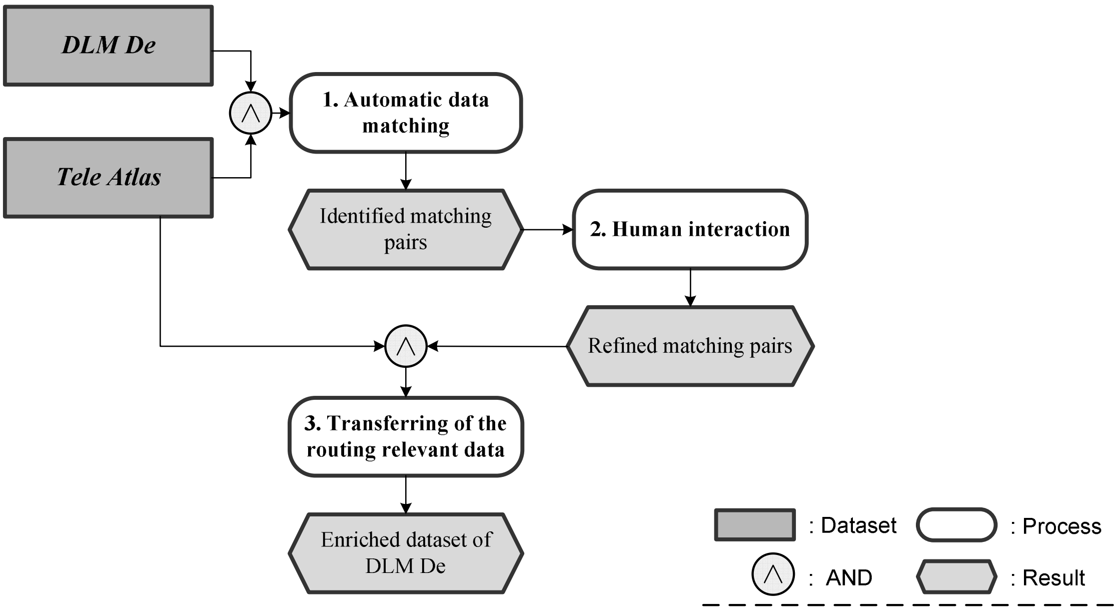
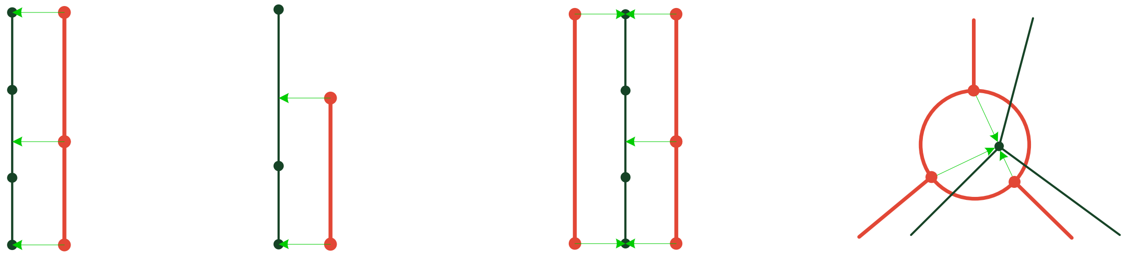

DLM De
Tele Atlas
∧
1. Automatic data matching
Identified matching pairs
2. Human interaction
Refined matching pairs
∧
3. Transferring of the routing relevant data
Enriched dataset of DLM De
: Dataset
: Process
∧ : AND
: Result

Figure 6.8 The strategy to achieve the routing data integration

As depicted by Figure 6.8, the flow diagram of enriching the DLM De with the routing-relevant information from Tele Atlas can be characterized by three processes: (a) Automatic matching to identify the corresponding road objects between different datasets; (b) interaction to refine the automatic matching result; and (c) transferring the routing-relevant information from Tele Atlas to DLM De. In Sections 6.2.1 ~ 6.2.3, the three processes are introduced in detail, while Section 6.2.4 is dedicated to a brief discussion on the enriched dataset of DLM De.

6.2.1 Identification of the corresponding road objects

A true data integration is much more than just overlaying data in a geographic information system (GIS) as it must set up the explicit relations between individual objects in different datasets (Butenuth et al. 2007), which indicates that the integration of routing-relevant information from various data sources requires the identification of the corresponding road objects, namely data matching, between different road networks. As mentioned earlier, with the contextual matching approach, not only the matching pairs with $m:n\ (m \geq 1, n \geq 1,\ m,n \in N)$ relationship, i.e. m DLM De road objects are corresponding to n objects in Tele Atlas (c.f. Figure 6.9-a), but also the matching pairs with partial or equivalent corresponding relationships can be identified (c.f. Figure 6.9-b ~ Figure 6.9-d).



(a) *m:n* matching (b) *patial* matching (c) *equivalent* matching (d) *equivalent* matching

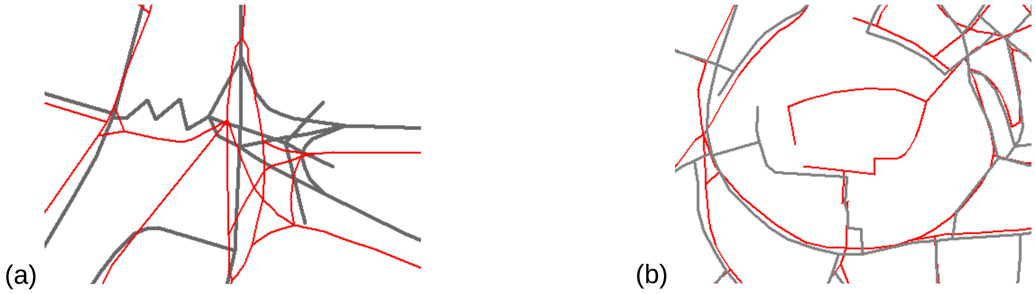
black lines: DLM De red lines: Tele Atlas green arrows: correspondences

Figure 6.9 Identified matching pairs with different matching relationships

It should be noted that for the matching pairs with equivalent corresponding relationships, e.g. the corresponding dual carriageways which reveal distinctive LoDs in different datasets (ref. Figure 6.9-c), it is complicated and error-prone to automatically exchange the routing-relevant information between different datasets, especially in case that the attributes need to be transferred from the dataset with lower LoD to the dataset with higher one. As the equivalent correspondences do not occur frequently in this project, we suggest processing them together with human interactions, i.e. this project is focused on the automatic data transferring between the matching objects with $m:n$ $(m \geq 1, n \geq 1,\ m,n \in N)$ or partial corresponding relationships, while the matching pairs of equivalent correspondences have to be dealt with by a human-assisted process.

6.2.2 Interactive refinement of the automatic matching result

In spite of the apparent progresses of the contextual matching approach (ref. Chapter 5), a completely automatic road matching between different datasets with 100% matching rate or accuracy is still difficult to reach. The matching algorithm suffers some limitations if the datasets to be matched reveal a substantial topologic inconsistency (see Figure 6.10-a); or if the objects in the dataset are highly fragmental (see Figure 6.10-b).



black lines: DLM De red lines: Tele Atlas

Figure 6.10 Examples of topologic inconsistency and geometric fragmentation

However the integration of the routing-relevant information requires a comprehensive and accurate matching between different datasets. Hence, the human operators should be provided with some interactive tools to evaluate and refine the results of automatic data matching processes. In our approach, a series of Plug-Ins (extensions) in ArcGIS 9.x has been developed which allows the users to visualize the identified corresponding objects between different datasets and then interactively add new matching pairs or correct the false matches, so that a more desirable matching result can be reached. On the basis of the refined matching results, the process 6.2.3 is eligible to automatic integration of the routing-relevant information from different data sources.

6.2.3 Transferring routing-relevant information from Tele Atlas to DLM De

Depending on the relationship to the road objects, the routing-relevant information in Tele Atlas can be categorized into three groups: (a) attributes of the road itself; (b) attributes at the intersections; and (c) Points of Interest (POIs) bound to the road objects. Different data groups will call upon different methodologies for the transferring of the routing-relevant information.

6.2.3.1 Attributes of the road itself

In Tele Atlas, the routing-relevant information of the road itself consists of street name, street width, direction restriction, speed limitation, FRC (Function Road Class), FOW (Form of Way), etc.

While attribute transfer is a relatively trivial task if a *1:1* correspondence exists between the datasets of DLM De and Tele Atlas (ref. Figure 6.11-a), such ideal cases are rare (Song et al. 2006; Zhang et al. 2008). In the real world, many other scenarios arise, e.g. the matching pairs with *M:1, 1:N or M:N (M>1, N>1)* or partial correspondence. In order to deal with the general cases of $m:n\ (m \geq 1, n \geq 1)$ or partial matchings, the matching pairs with *1:n/m* relationship are identified, i.e. one object from DLM De is matched to a cluster of connected objects or object-parts in Tele Atlas.

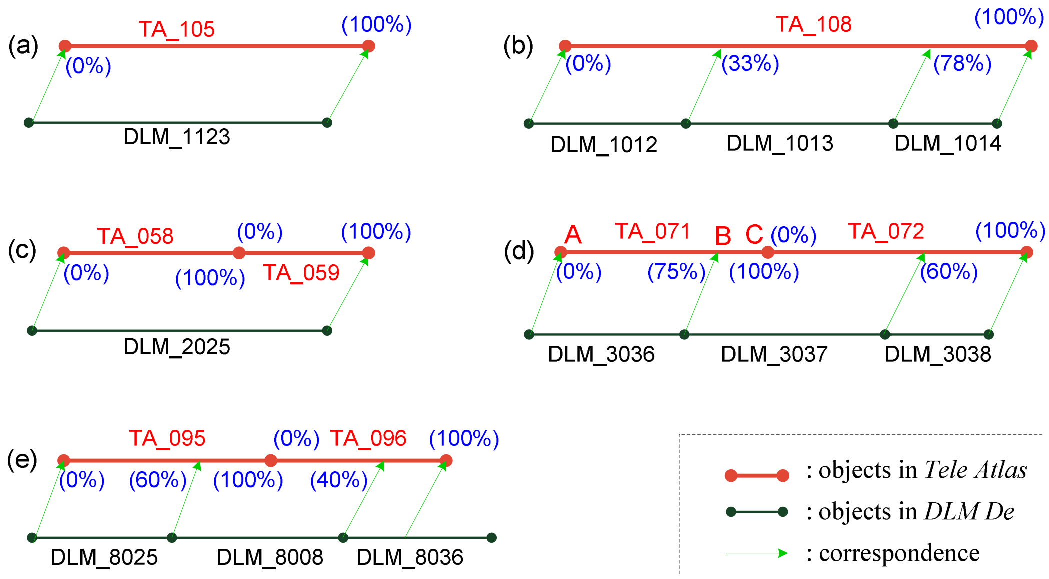


Figure 6.11 Examples of matching pairs with different corresponding relationships: (a) Matching pair with *1:1* relationship; (b) Matching pair with *M:1(3:1)* relationship; (c) Matching pair with *1:N (1:2)* relationship; (d) Matching pair with *M:N (3:2)* relationship; (e) Matching pair with partial relationship

***Note** in Figure 6.11-d, 75% means $\frac{Length\ \ of\ \ AB}{Length\ \ of\ \ AC} = 75\%$. Other percentages are indicated in a similar way.

For the matching cases with $m:n\ (m \geq 1, n \geq 1,\ m,n \in N)$ relationships between the datasets of DLM De and Tele Atlas, each DLM De object can lead to a matching pair with *1:n/m* relationship. E.g. three objects from DLM De and two objects from Tele Atlas are matched together in Figure 6.11-d, which will result in three matching pairs with *1:n/m* relationship as shown in the rows marked with "*" in Table 6.1. However, for the matching pairs with partial correspondences, it is not always possible to identify the matching pairs with *1:n/m* relationship for all of the DLM De objects. Figure 6.11-e illustrates that three DLM De objects partially corresponding to two Tele Atlas objects; hereinto only two DLM De objects have their *1:n/m* counterparts in the dataset of Tele Atlas, see the rows marked with "**" in Table 6.1.

DLM De	**Tele Atlas**		
Object ID	**Object ID**	**From Position**	**To Position**
DLM_1123	*TA_105*	*0%*	*100%*
DLM_1012	*TA_108*	*0%*	*33%*
DLM_1013	*TA_108*	*33%*	*78%*
DLM_1014	*TA_108*	*78%*	*100%*
DLM_2025	*TA_058*	*0%*	*100%*
	TA_059	*0%*	*100%*
***DLM_3036*	*TA_071*	*0%*	*75%*
***DLM_3037*	*TA_071*	*75%*	*100%*
	TA_072	*0%*	60%
***DLM_3038*	*TA_072*	*60%*	*100%*
***DLM_8025*	*TA_095*	*0%*	*60%*
***DLM_8088*	*TA_095*	*60%*	*100%*
	TA_096	*0%*	*40%*
***DLM_1013*	*No matching pair with 1:n/m relationship*		

Table 6.1 Matching pairs with *1:n/m* relationship derived from Figure 6.11

As soon as the matching pairs with *1:n/m* relationship are identified, it becomes possible to transfer the routing-relevant attributes of the road itself from Tele Atlas to DLM De. If one object from DLM De is related to more than one object in Tele Atlas, c.f. the rows containing the object ‘DLM_3037’ in Table 6.1, the attributes of the different objects in Tele Atlas must be firstly generalized and then transferred to the DLM De object. In this process, the generalization of attribute values is guided by a knowledge base, i.e. the generalization of attribute values should be triggered by a set of rules such as: (a) Keep the same attribute, e.g. street name can be directly transferred to another dataset; (b) Transfer the maximum or minimum value, e.g. maximally allowed speed of the vehicles on the streets, or minimum bridge height over a road; (c) Transfer the sum of object values, e.g. the travel time of the streets; and (d) Transfer the average value, e.g. CO_2 emission value of the vehicles on the streets; etc.

6.2.3.2 Attributes at road intersections

As the contextual matching approach is able to match both the roads and the nodes along them (ref. Chapter 5), it is possible to transfer the information at the road intersections from Tele Atlas to DLM De. In the dataset of Tele Atlas, the significant routing-relevant information at the road intersections, such as turning restrictions and manoeuvres, signpost information, traffic light, etc., are represented by the combination of the node (‘junction’) and its emanating edges (‘road objects’) .

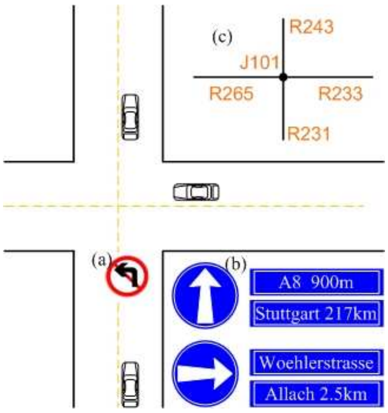


Figure 6.12 Attributes at the road intersection - (a) left forbidden; (b) signpost information; and (c) digital map

Type	*From object*	*Junction*	*To Object*
Prohibited Maneuver	*R231*	*J101*	*R265*

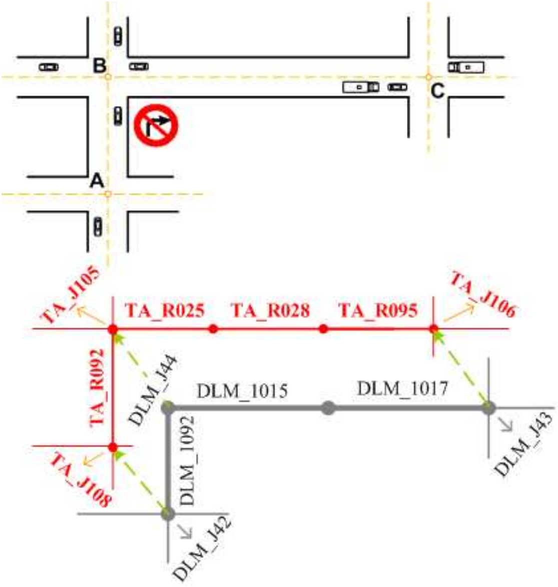
Table 6.2 ‘Left forbidden’ in Tele Atlas

Type	*From Junction*	*To Object*	*Name*	*Distance*
Highway Entry	*J101*	*R243*	*A8*	*900m*
Spot	*J101*	*R243*	*Stuttgart*	*217km*
Street	*J101*	*R233*	*Woehlerstrasse*	*-*
Spot	*J101*	*R233*	*Allach*	*2.5km*

Table 6.3 ‘Signposts’ in Tele Atlas

For instance, the ‘turning restrictions’ are stored in the way of “road object→ junction→ road object” as shown in Figure 6.12-a and Table 6.2; and the ‘signposts’ are described by relationships between junction and the relevant road objects, see Figure 6.12-b and Table 6.3.

In order to transfer such kind of routing-relevant information from Tele Atlas to DLM De, it is necessary to identify the matching pairs with *pseudo 1:1* relationship around the road intersections in spite of the fact that some of the matching pairs do not have *1:1*, but hold *M:1, 1:N, M:N (M>1, N>1)* or partial corresponding relationships between different datasets. The matching pairs with *pseudo 1:1* relationship should be recorded with the related corresponding junction information as illustrated in Figure 6.13 and Table 6.4.



black: DLM De red: Tele Atlas

Figure 6.13 Matching around a road intersection

No.	***Corresponding Objects***		***Corresponding Junctions***	
	DLM De	***Tele Atlas***	**DLM De**	***Tele Atlas***
***** *1.*	*DLM_1015*	*TA_R025*	*DLM_J44*	*TA_J105*
2.	*DLM_1017*	*TA_R095*	*DLM_J43*	*TA_J106*
** 3.*	*DLM_1092*	*TA_R092*	*DLM_J44*	*TA_J105*
4.	*DLM_1092*	*TA_R092*	*DLM_J42*	*TA_J108*

Table 6.4 Matching pairs with *pseudo 1:1* relationship

The matching pairs with *pseudo 1:1* relationship make it possible to transfer the routing-relevant information at road intersections between different datasets, e.g. based on the pseudo 1:1 matching pairs marked with '*****' in Table 6.4, the 'right- forbidden' -*'TA_R092→TA_J105→TA_R025'* in Tele Atlas can be replaced by *'DLM_1092→DLM_J44→DLM_1015'* at first, then integrated with the dataset of DLM De.

6.2.3.3 Points of Interest bound to the road objects

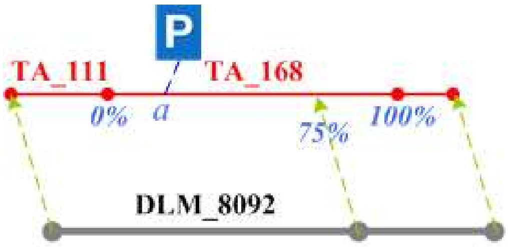
POIs (Points of Interest), are a series of point representations bound to the road lines, such as hotel, gas station, restaurant, showplace, beauty spot etc. The POI information is also relevant for routing although it is not always used for the routing purposes. The transferring of the POI information from Tele Atlas to DLM De involves two steps.

(a) Geometric displacement

Initially the individual POIs are stored as discrete points along the road features in Tele Atlas. In order to adapt the POIs position to the DLM De roads, it is necessary to establish the linkages between the shape points of matched road pairs by means of interpolation; with the availability of these links, the POIs can be automatically displaced from Tele Atlas to DLM De following the Rubber -Sheet principle explained in Section 6.1.3.

(b) Semantic transferring

The incipient POIs are semantically related to the road objects of Tele Atlas. Based on the identified matching pairs with *1:n/m* relationship in Table 6.1, the semantic relationships between the POIs and road objects in Tele Atlas can be also transferred to the dataset of DLM De.



black: DLM De red: Tele Atlas

Figure 6.14 Semantic transferring of the POI

DLM De	Tele Atlas		
Object ID	***Object ID***	***From Position***	***To Position***
DLM_8092	*TA_111*	*0*	*100%*
	TA_168	*0*	*75%*

Table 6.5 One identified matching pair *(1:n/m)* from Figure 6.14

In the example depicted in Figure 6.14, the POI **'P'** is related to the road object *TA_168* in *Tele Atlas.* Considering that the entry point *'a'* drops in the interval [0%, 75%] of the object *TA_168,* which belongs to the matching pair recorded in Table 6.5, a semantic connection between **'P'** and object *DLM_8092* should be appended to the dataset of DLM De.

6.2.4 Enrichment of DLM De with the routing-relevant information from Tele Atlas

As known, in the dataset of DLM De, the attributes are not completely covered with values, especially the routing applications are not yet considered. Following the methodologies defined in Section 6.2.3, all of the significant routing-relevant information can be automatically transferred from Tele Atlas to DLM De.

Figure 6.15 and 6.16 show the enriched DLM De with a few routing-relevant information from Tele Atlas, incl. Street Name, turning restrictions and manoeuvres, and some POIs like park place, gas station, hospital etc.

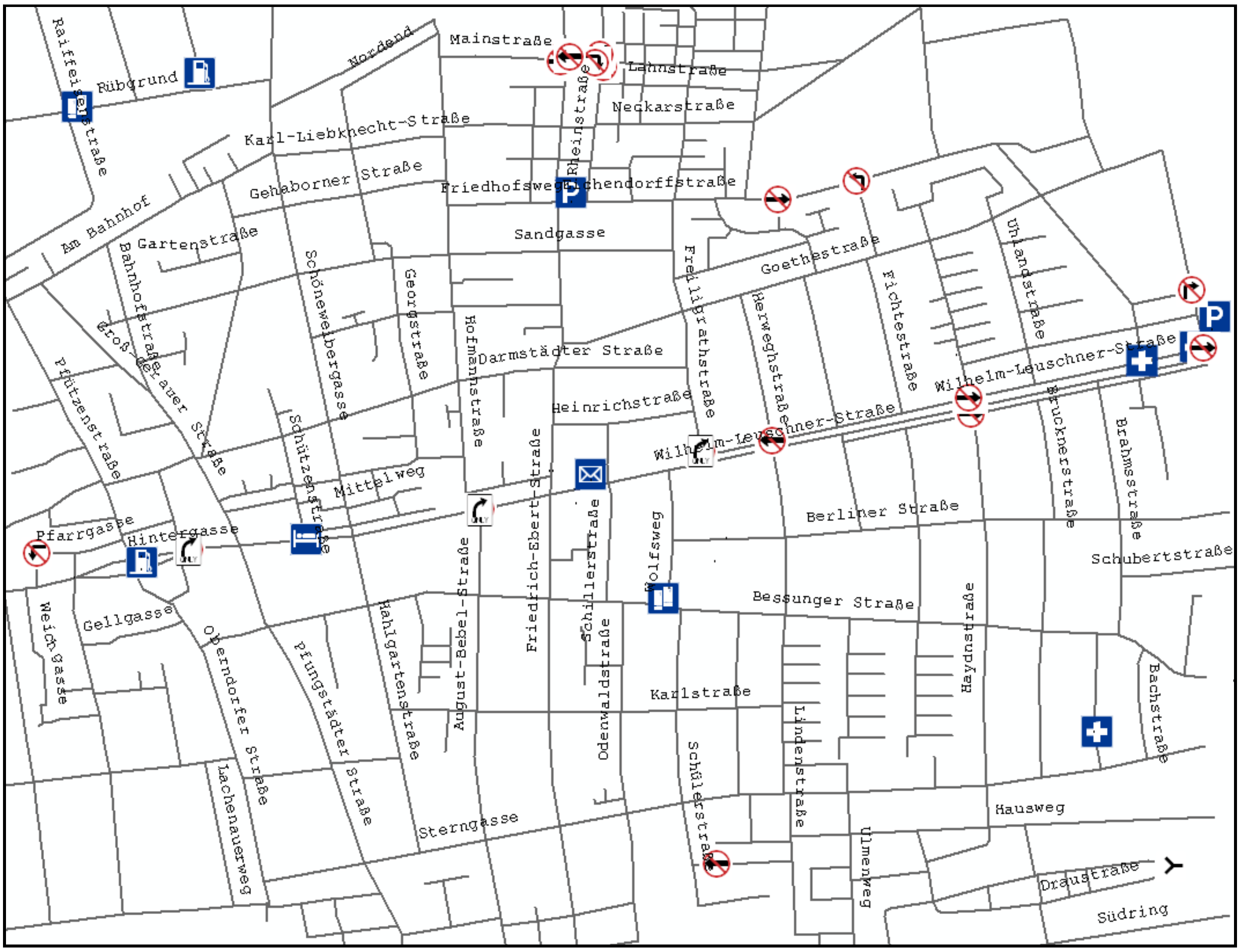


Figure 6.15 The enriched dataset of DLM De with some routing-relevant information from Tele Atlas

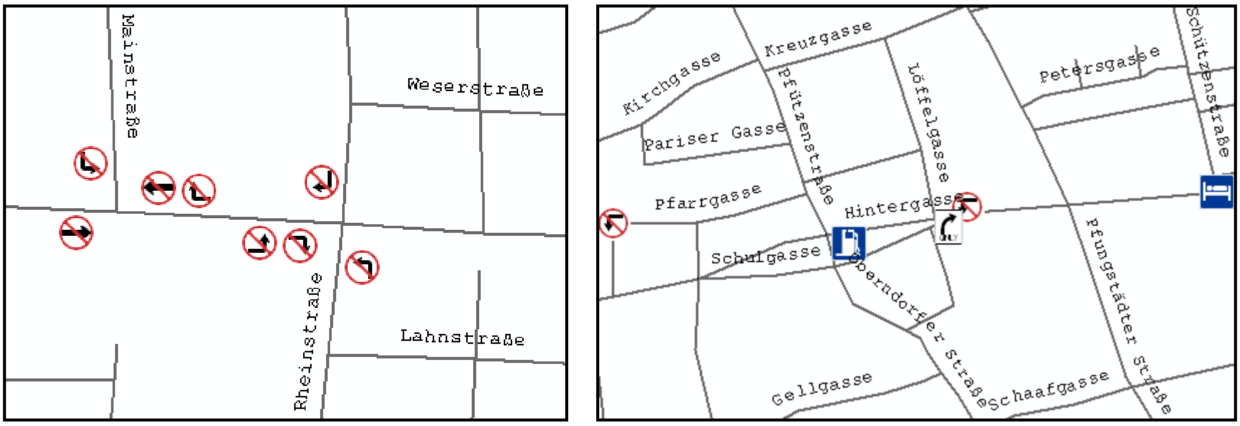


Figure 6.16 An enlarged section of Figure 6.15: integrated turning restrictions and manoeuvres (left) and POIs (right)

It should be noted that not all road objects from DLM De can be enriched with the attributes from Tele Atlas since some kinds of roads (e.g. pedestrian ways in parks) only exist in DLM De, but not be captured in Tele Atlas, see examples in the lower-left and upper-right of Figure 6.15.

After the data enrichment, the DLM De has gained several capabilities for routing calculation and car navigation. In order to prove the routing capability of the enriched DLM De, experiments are being conducted which can:

- **Transform the enriched DLM De from ATKIS to GDF**

The data structure of DLM De is defined in accordance with the Official Topographic Cartographic Information System (ATKIS), which is not suitable for routing applications. Thereby, the enriched DLM De with the routing-relevant information from Tele Atlas should be transformed from ATKIS standard to GDF (Geographic Data Files). As a European standard, GDF has been widely applied for the purpose of car navigation and many other transport and traffic applications such as fleet management, dispatch management, traffic analysis, traffic management, and automatic vehicle location.

- **Verify the routing capability of the enriched DLM De**

As soon as the enriched DLM De data are transformed to GDF format, various existing routing algorithms can be tested and compared with the navigational road database Tele Atlas. If the algorithms can always attain the same route solution in both Tele Atlas and DLM De, we can believe that the significant routing-relevant information in Tele Atlas has been successfully transferred to the dataset of DLM De. However in some cases, the algorithms may achieve different route solutions between Tele Atlas and DLM De. On such occasions, it is necessary to compare the two different route solutions with each other.

The better solution in the enriched DLM De should demonstrate that (a) the integration of the significant routing-relevant information between DLM De and Tele Atlas is comprehensive; and (b) the initial DLM De data has been convincingly associated with the integrated routing-relevant information from Tele Atlas and therefore the enriched DLM De becomes more valuable for the purpose of vehicles routing. Meanwhile, the better solution in Tele Atlas can also indicate that some significant routing-relevant-information was lost or false transferred during the process of data matching and integration.

6.3 Case 3 - Conflation of pedestrian ways between different datasets

In recent years, several navigator producers, such as Nokia and TomTom have begun to address pedestrians for its navigation services rather than drivers, which indicate that the GPS navigators will be no longer restricted on motor ways, but also serve the pacers on the pedestrian ways. However, as most well-known routing-capable database in the world, neither Tele Atlas nor NAVTEQ is qualified for such services due to the data acquisition ways. In either Tele Atlas or NAVTEQ, the road networks were captured primarily by GPS-supported equipments on cars, i.e. the roads which are prohibited to motor vehicles, e.g. around commercial centres and in public parks, are not possible to be captured in either Tele Atlas or NAVTEQ. Whereas, the ATKIS data, from German Mapping Agency, were captured through map digitization in combination of semiautomatic object extraction from imagery data and thereby roundly covers the pedestrian areas. This problem can be illustrated using the example of road-network data in Munich, Germany (see Figure 6.17), where the grey lines represent the ATKIS data and the orange represent NAVTEQ. In the centre area of this example, which is a part of the park named *'English Garten'*, a lot of pedestrian ways captured by ATKIS do not appear in the dataset of NAVTEQ. However the ATKIS dataset itself is not capable for navigation purposes either although it involves both motor ways and pedestrian ways because the attributes are not completely covered with values, especially the routing-relevant information is rarely considered in this dataset.

In order to achieve the objectives of multi-modal navigation, i.e. “pacer + car”, the project of “conflation of pedestrian ways between NAVTEQ and ATKIS” has been launched by Ltd. Corp.

United Maps. It aims at appending the pedestrian ways of ATKIS to the road network of NAVTEQ, which indicates that in the final conflated dataset the road network of NAVTEQ acts as the backbone and the ATKIS contributes the additional pedestrian ways.

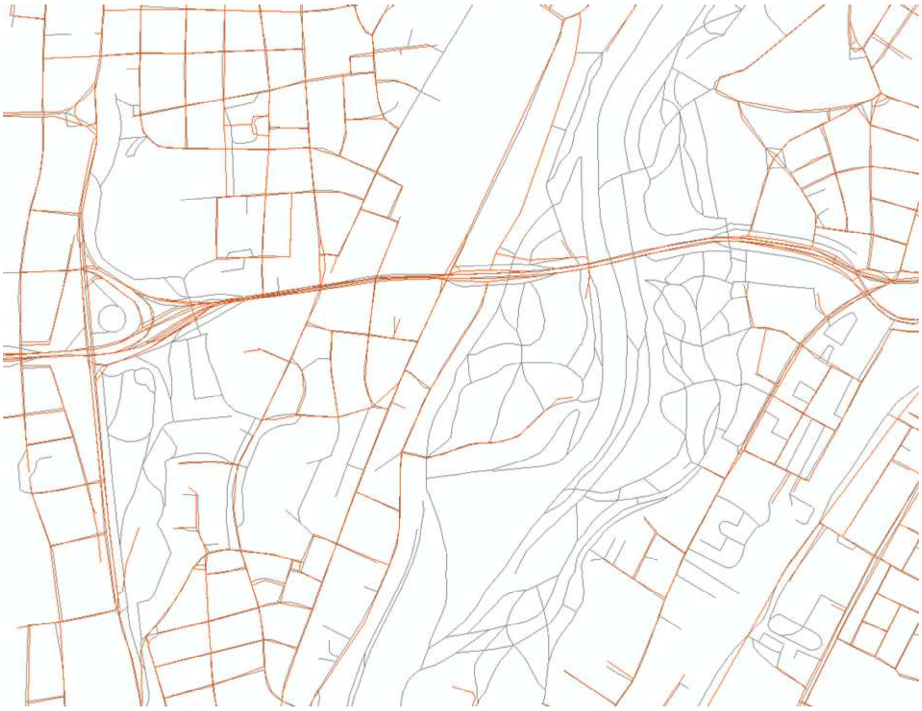


Figure 6.17 Differences of the road networks between NAVTEQ (orange) and ATKIS (grey)

Clearly, one can not rely on manual approach to conflate diverse geospatial datasets, as the area of interest may be anywhere in the world and manually conflating a large region (e.g. the whole Germany) is very time consuming and error-prone. For this reason, an automatic routine for the conflation of different road networks between NAVTEQ and ATKIS has been developed, which involves five processes as depicted in Figure 6.18:

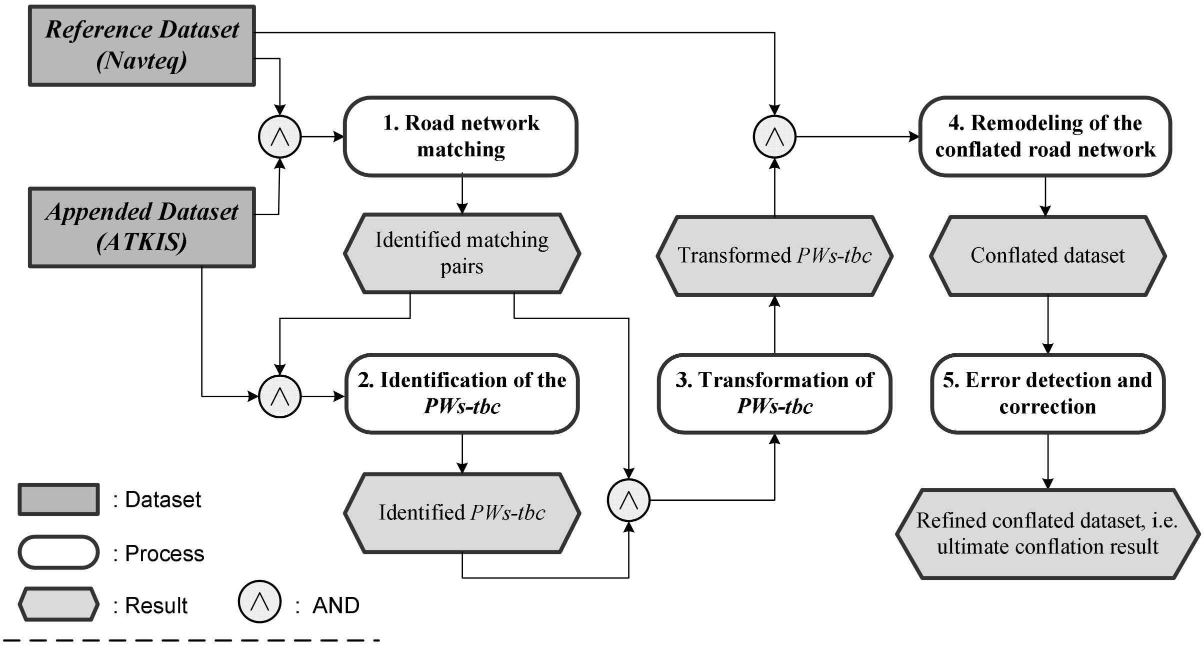

Reference Dataset (Navteq)
Appended Dataset (ATKIS)
1. Road network matching
Identified matching pairs
2. Identification of the PWs-tbc
Identified PWs-tbc
3. Transformation of PWs-tbc
Transformed PWs-tbc
4. Remodeling of the conflated road network
Conflated dataset
5. Error detection and correction
Refined conflated dataset, i.e. ultimate conflation result
: Dataset
: Process
: Result
: AND

Figure 6.18 Strategy to achieve the conflation of different road networks

(i) Road-network matching between participating datasets (viz. ATKIS and NAVTEQ)

(ii) Identification of the pedestrian ways to be conflated (PWs-tbc) in ATKIS*;*

(iii) Transformation of PWs-tbc to eliminate geometric inconsistency;

(iv) Remodelling of the conflated road network; and

(v) Error checking and correction.

From Section 6.3.1 to 6.3.5, the five processes are introduced in detail. The conflation result is analyzed in Section 6.3.6. The terminologies of conflation, combination and merging are utilized interchangeably in this section.

6.3.1 Network matching between participating datasets

A critical first task in the process of data conflation is to identify the correspondences between networks so that data can be merged or transferred from one network to another. Although correspondences between different networks can be visually recognized, their automatic matching was regarded as a non-trivial procedure (Xiong 2004). With the development of our contextual matching approach, it is now possible to bring datasets from NAVTEQ and ATKIS together with a high matching rate and matching accuracy (see Section 5.2), which then allows the automatic conflation of the pedestrian ways from different datasets.

6.3.2 Identification of the PWs-tbc in ATKIS

We denote the road network of ATKIS as NW_{AT}, and the road network of NAVTEQ as NW_{Na}. The goal of this process is to identify all the pedestrian ways which have not yet been captured in NW_{Na} but do exist in NW_{AT}, i.e. to calculate the set of *PWs-tbc* (viz. pedestrian ways to be conflated) defined by the expression $Set(PWs-tcb)=\{PW_i | PW_i \in NW_{AT}, PW_i \notin NW_{Na}\}$, where PW represents 'pedestrian way'.

After the automatic data matching, the road objects in NW_{AT} (ATKIS) can be classified into three groups:

a. Matched;
b. Unmatched;
c. Partially matched;

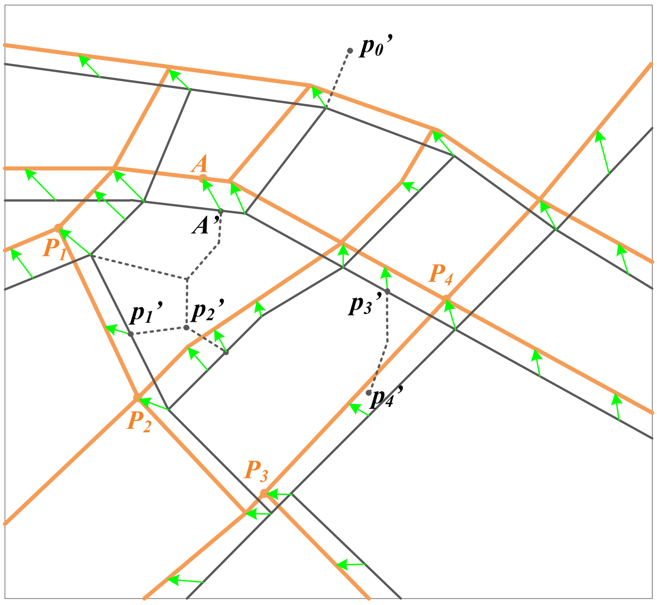
The 'matched' can indicate that the road objects in NW_{AT} (ATKIS) have successfully found their counterparts in NW_{Na} (NAVTEQ), which breaks the condition of $PW_i \notin NW_{Na}$. Therefore, only the unmatched or partially matched objects in NW_{AT}, i.e. the roads which can not find their counterparts in NW_{Na}, can be treated as the potential pedestrian ways to be conflated to the dataset of NAVTEQ (viz. *potential PWs-tbc*), see examples of grey dashed line in Figure 6.19.

However, not all of the potential PWs-tbc should be conflated to the dataset of NAVTEQ due to the following reasons:

(a) In spite of the apparent progresses, the employed matching approach still can not guarantee a completely automatic data matching between different datasets with 100% matching rate or accuracy, which indicates that some of road objects in ATKIS and their corresponding partners in NAVTEQ can not be matched together in the automatic matching process (ref. Section 6.3.1). Due to the mismatching, a certain number of objects could be identified as „unmatched" or "partially matched" regardless the fact that their counterparts do exist.

(b) In the dataset of ATKIS, several road objects have no counterparts in NAVTEQ although they are not pedestrian ways. These roads obviously do not belong to the set of PWs-tbc either.

To achieve more accurate identification of PWs-tbc, the semantic information can be considered. In this project, no semantic information that indicates whether a road object belongs to pedestrian way or not is available. However, some attributes may help to partially eliminate several road objects which do not belong to pedestrian ways.



orange lines: road network of NAVTEQ; grey lines: road network of ATKIS;
dashed lines: pedestrian ways to be conflated (PWs-tbc); green arrows: linkages
Figure 6.19 Matching between NAVTEQ and ATKIS

6.3.3 Transformation of PWs-tbc to eliminate geometric inconsistency

The identified PWs-tbc can not be directly implemented for the data conflation since their geometries might be in conflict with the road network of reference dataset in some cases. For example, in Figure 6.19 the pedestrian way p_2'→p_1' from ATKIS should be connected to the street P_1→P_2 in NAVTEQ, but in fact they are detached here; and (b) instead of intersecting to each other, the road p_3'→p_4' lies apart from road P_3→P_4. Such a case requires an adaptive transformation to harmonize the shape and location of the PWs-tbc to the road network of NAVTEQ. This transformation process can be characterized by two steps:

Step 1: Establishment of the control point pairs

A *control point pair* (abbreviated as *CPP*) consists of a point in one dataset and a corresponding point in the other dataset. Finding proper control point pairs is an important step in the transformation process as all the other points are aligned based on them (Chen 2005). Essentially, based on the identified matching pairs, the control point pairs can be generated by means of interpolation (ref Section 6.1.1) The identified corresponding coordinates (see the example of green arrows in Figure 6.19) are stored in the physical memory and act as control point pairs in the next step of 'Alignment based on control point pairs'. Here, the control point pair is constructed by the *fromPoint* in ATKIS (appended dataset) and *toPoint* in NAVTEQ (reference dataset) which tend to represent the same position in the real world.

Step 2: Alignment based on control point pairs

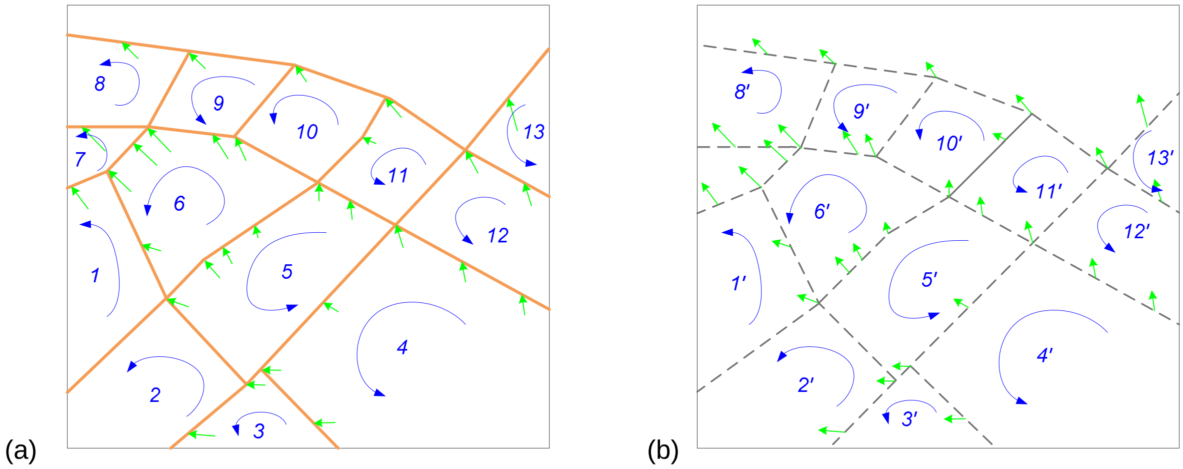
The overall transformation of the PWs-tbc from ATKIS needs to satisfy several cartographic constraints, such as preservation of the orientation, the relative spatial position, and the continuity between adjacent objects. This means, the turning points of the PWs-tbc should be properly aligned on the basis of the control point pairs (CPPs). According to the topologic characteristics and their relationship to the CPPs, these turning points can be categorized into three groups, (a) Turning points which are duplicated to the *fromPoints* of CPPs; (b) Road crossings (valence ≥3) or dead-ends (valence =1) which are not duplicated to the *fromPoint* of any CPP; and (c) Other shape-points along the PWs-tbc. Different categories will call upon different methodologies for the point alignment.

(a) Turning points which are duplicated to the fromPoints of CPPs

For this kind of turning points, the alignment is conducted by displacing the turning point between the control point pair, e.g. from the point *A'* to *A* in Figure 6.19. Such an alignment preserves the topologic continuity and assures that the transformation can sew together joined road objects between diverse datasets.

(b) Road crossings or dead ends which are not duplicated to the fromPoint of any CPP

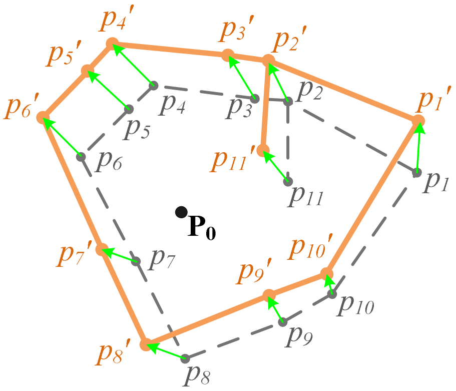
The established control point pairs in *step 1* form a distortion map for the whole conflation area. In order to properly adjust the position of the crossings (valence ≥3) and dead ends (valence =1) of the PWs-tbc which are not duplicated to the *fromPoint* of any CPP, e.g. the nodes *p_2'* and *p_4'* in Figure 6.19, the Euclidean transformations are calculated based on the distortion map. In principle, all cells in the distortion map may influence the transformation of the crossings and dead ends, either evidently or subtly. However, it has been generally recognized that the changes in one region usually do not affect the geometries at remote distances or the influences are ignorable in the practice. Furthermore, it is common to find different systematic distortion trends in different parts and none of these trends can represent the global analytical shape distortion between the datasets to be conflated. To solve these disagreements, the local transformation is applied, which employs space partition of the whole conflation area into much smaller regions and therefore can better handle the local distortions in each region. In a road network, the linear topologic structure provides a natural way to spatially subdivide the datasets, i.e. mesh-based partition. A mesh, also called face, can be regarded as a closed region that does not contain any other region. The meshes, e.g. $\{\, mesh_i \mid i = 1,2,3,...13\}$ based on the road network of NAVTEQ depicted in Figure 6.20-a, define boundaries in a natural way and also form the zones that separate the objects insides the zones from those outside (Doytsher et al. 2001). Considering that this process aims at transforming the PWs-tbc from the ATKIS, the meshes of $\{\, mesh_i \mid i = 1,2,3,...13\}$ based on NAVTEQ have to be distorted around the CCPs. As the result, a set of new meshes $\{\, Mesh_I ' \mid I = 1,2,3,...13\}$ (see Figure 6.20-b) will be established, which fit the geometries of the dataset of ATKIS*.*



orange solid lines: initial NAVTEQ; grey dash lines: distorted NAVTEQ; green arrows: linkages

Figure 6.20 Space partition based on meshes: (a) initial meshes based on NAVTEQ; and (b) distorted meshes that fit the geometries of ATKIS

With each distorted mesh (see examples in Figure 6.20-b), the CCPs can build up a local distortion map, which influences the alignment of the points within or on the boundary of this mesh. Let's define the CPP *as* vector $\vec{V}(p_i, p_i')$, where $p_i = (x_i, y_i)^{\mathrm{T}}$ is the *fromPoint* and $p_i' = (x_i', y_i')^{\mathrm{T}}$ is the endpoint; the neighbour of p_i is denoted as $p_{i,j}$. The concept of 'neighbour' can be illustrated by the example in Figure 6.21, where the point p_2 has three neighbours of $\{ p_{2,1}, p_{2,2}, p_{2,3} | p_{2,1} = p_3, p_{2,2} = p_1, p_{2,3} = p_{11} \}$; point p_1 has two neighbours of $p_{1,1} = p_2$ and $p_{1,2} = p_{10}$; and point p_{11} has only one neighbour p_2.



orange solid lines: initial NAVTEQ; grey dash lines: distorted NAVTEQ; green arrows: linkages

Figure 6.21 Local distortion map based on mesh partition

Thus, given a road crossings or dead ends $P_0 = (X_0, Y_0)^{\mathrm{T}}$ falling inside the distorted mesh (see the example of polygon $p_1 p_2 ... p_{10} p_1$ in Figure 6.21, its new position $P_0' = (X_0', Y_0')^{\mathrm{T}}$ in the conflated dataset can be calculated by equation [6-2].

$$P_0' = P_0 + \frac{\sum_{i=1}^{n}[\sum_{j=1}^{m}(p_{i,j} - p_i)^{\mathrm{T}} \cdot (p_{i,j} - p_i)]^{\alpha} \cdot [(P_0 - p_i)^{\mathrm{T}} \cdot (P_0 - p_i)]^{-\beta} \cdot (p_i' - p_i)}{\sum_{i=1}^{n}[\sum_{j=1}^{m}(p_{i,j} - p_i)^{\mathrm{T}} \cdot (p_{i,j} - p_i)]^{\alpha} \cdot [(P_0 - p_i)^{\mathrm{T}} \cdot (P_0 - p_i)]^{-\beta}} \qquad ...[6\text{-}2]$$

Where, m - number of the neighbous of p_i;

n - number of the CPPs

α, β - two experimental coefficients larger than 0;

Equation [6-2] demonstrates that when a given point is duplicated to one of the reference vertexes p_i, the weight of this vertex $[(P_0 - p_i)^{\mathrm{T}} \cdot (P_0 - p_i)]^{-\beta}$ approaches infinity. It indicates that the transformation of the given point will be calculated only according to the vertex's own displacement, which is in accordance with the alignment of the turning points in *Group (a)* and therefore can provide us a consecutive transformation model.

In practice, the set of CPPs includes more points than those forming closed meshes. It is common to encounter open-end edges or edges that link meshes, which indicates several crossings or dead ends of the PWs-tbc could be outside all of the closed meshes (see point *p₀'* in Figure 6.19). For such cases, the proposed local transformation model will build up a well-defined buffer around the given point; then all the CPPs that fall inside this buffer will be taken into account for the transformation; however, if there is no CPP falling inside, the point will keep its initial position after the data conflation.

In order to enhance the computing efficiency, the alignment of the crossings and dead ends in this group could be ignored if the overall geometric deviation between different datasets is small enough (e.g. <5 meters).

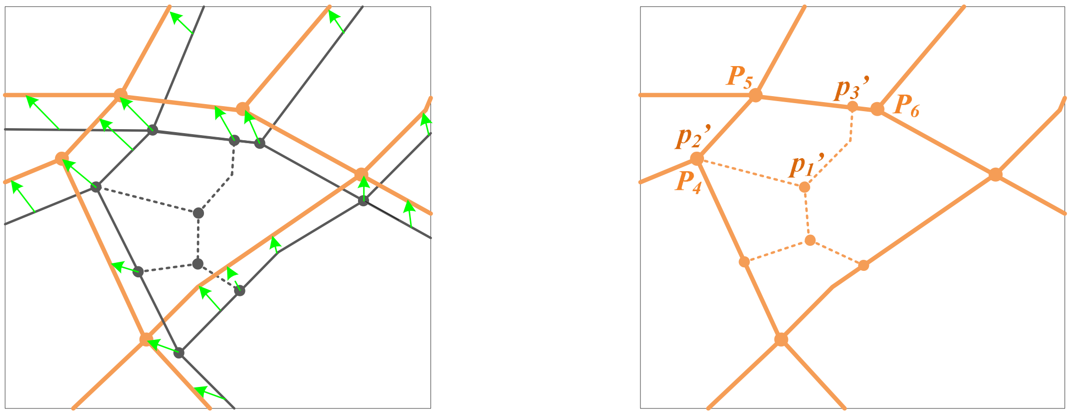
(c) Other Shape points of the PWs-tbc

The shape points in this group are (i) neither road crossings nor dead end and (ii) not duplicated to the *fromPoint* of any CPP. In order to preserve the initial orientation and form of the PWs-tbc, these shape points are aligned based on the point transformations in *Group (a)* and *Group (b)*. For example, the PWs-tbc $p_1 p_2 ... p_{n-1} p_n$ is restricted by p_1 and p_n, where $p_1 = (x_1, y_1)^{\mathrm{T}}$ is a turning point in *Group (a)* with the transformation $\Delta T_{p1} = (\Delta x_1, \Delta y_1)^{\mathrm{T}}$ and $p_n = (x_n, y_n)^{\mathrm{T}}$ is a road crossing in *Group (b)* with the transformation $\Delta T_{pn} = (\Delta x_n, \Delta y_n)^{\mathrm{T}}$. Then, the transformation of the shape point p_i $(2 \le i \le n-1)$ can be calculated by Equation [6-3], where ΔT_{pi} represents the transformation of the shape point p_i and γ is an experimental coefficient between (0, 1].

$$\Delta T_{pi} = (\Delta x_i, \Delta y_i)^{\mathrm{T}} = \frac{\Delta T_{pn} \cdot [(p_i - p_1)^{\mathrm{T}} \cdot (p_i - p_1)]^{\gamma} + \Delta T_{p1} \cdot [(p_i - p_n)^{\mathrm{T}} \cdot (p_i - p_n)]^{\gamma}}{[(p_i - p_1)^{\mathrm{T}} \cdot (p_i - p_1)]^{\gamma} + [(p_i - p_n)^{\mathrm{T}} \cdot (p_i - p_n)]^{\gamma}} \quad (2 \le i \le n-1)$$

…[6-3]

After the alignment of all the turning points in *Group (a), (b) and (c)*, the PWs-tbc will have their new forms and positions in the conflated road network, see an example in Figure 6.22.



orange: the reference dataset of NAVTEQ; grey: the appended dataset of ATKIS;
dashed lines: conflated pedestrian ways; green arrows: linkages (control point pairs)

Figure 6.22 Transformation of PWs-tbc from one road network to the other

6.3.4 Remodelling of the conflated dataset

In the conflated dataset the newly appended PWs-tbc and the initial road network of NAVTEQ should be well organized from both topologic and semantic perspective. To demonstrate this issue, several changed representations in the conflated dataset are discussed in the following subsections.

6.3.4.1 Creating new intersections (nodes)

After the adaptive geometric transformation, one *PW-tbc* is able to be aligned to the new coincident position in the conflated road network. Topologically, the *PW-tbc* will have nothing to do with the conflated road network if it is totally apart from the initial road network of NAVTEQ or its touching point to the road network of NAVTEQ is an existing node, *see* example of the conflated road $p_1' \rightarrow p_2'$ in Figure 6.22 where the p_2' and P_4 are overlapped.

The condition, however, becomes complicated when the *PW-tbc* touches one road object from road network of NAVTEQ and the touching point is neither from-node nor to-node of this object. In such cases, the conflated road network requires new intersections (nodes) to rearrange the topologies of the conflated road network. For example in Figure 6.22, the conflation of the *PW-tbc* $p_1' \rightarrow p_3'$ necessitates a new intersection p_3' to split the object $P_5 \rightarrow P_6$ into two parts, which reserves the connectivity between the PWs-tbc and the road network from NAVTEQ.

6.3.4.2 Decomposition and transferring of semantic information

The decomposition and transferring of the attributes from the reference dataset into the new one is an important function for the map conflation. This is a straightforward task for topologically unchanged road objects because these objects will lead to 1:1 attribute transferring. However, difficulties may occur for those that have been divided by new created intersections, e.g. in Figure 6.22, the road object $P_5 \to P_6$ from the initial road network of NAVTEQ has been split into two objects $P_5 \to p_3'$ and $P_5 \to p_3'$ in the conflated dataset. In this case, the initial attribute of the object should be first decomposed and then transferred to the split parts. The non-spatial attributes of the original object, such as street name, Functional Road Class, Form of Way, etc., can be directly assigned to the newly generated objects, whereas the spatial attributes, such as the street length and travel time, should be fairly assigned to the new ones by means of interpolation (Zhang and Couloigner 2005).

6.3.4.3 Entity ID issues

In the routing capable geospatial database, each geographic entity should have a unique identifier (ID) to distinguish it from all other geographic entities. In general, either object ID or node ID (for both from-node and to-node) can be concerned for routing purposes. Though the assignment of an ID has a great impact on the design and implementation of a routing-capable database, there is no unique way to make a decision on when to assign a new ID or when it has to remain the same. In this project, the decision depends on the various conditions listed below:

For the conflated objects, e.g. $p_1' \to p_2'$ and $p_1' \to p_3'$ in Figure 6.22, we should assign new object IDs for them. Meanwhile, the node ID of P_4 in NAVTEQ is transferred to the to-node of the object $p_1' \to p_2'$ (viz. p_2')*;* while new node IDs are required by the nodes which are either initial from the ATKIS (e.g. p_1' in Figure 6.22) or newly created intersections (see Section 6.4.3.1 and p_3' in Figure 6.22).

For the unchanged objects from reference dataset of NAVTEQ, e.g. $P_4 \to P_5$ in Figure 6.22, we will keep all the IDs for road object, from-node and to-node. However, if a road object from the reference dataset of NAVTEQ is divided into different parts after the data conflation (e.g. $P_5 \to P_6$ in Figure 6.22), then each part (see $P_5 \to p_3'$ and $p_3' \to P_6$ in Figure 6.22) has to be assigned a new object ID since it acts as an individual road object in the conflated dataset.

Moreover, all of the original object IDs should be reserved to keep the communications between the final conflated dataset and the sources of (a) reference dataset (NAVTEQ) and (b) appended dataset *(*ATKIS*)*.

6.3.5 Error detection and correction

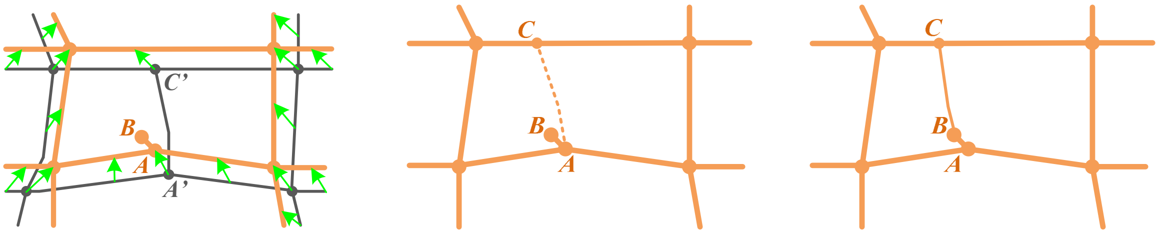
Instead of providing comprehensively accurate data conflation between different datasets, the automated routine defined in Section 6.3.1 to 6.3.4 often leads to an accurate result up to a certain percentage, which indicates that after the automatic data conflation a post-processing is necessary to improve the data quality. Error checking and correction is thereby needed to help the operators to detect and then remove or refine the wrongly conflated pedestrian ways. In comparison to the initial road network of NAVTEQ, the conflated pedestrian ways from ATKIS can be classified into four Categories:

Category 1*: Duplicated conflated pedestrian ways*

In the proposed approach, the conflated pedestrian ways, which are overlapped or located very closely to the roads from the reference dataset, are regarded as duplications and can be automatically removed from the conflated dataset.

Category 2*: Partial duplications*

Figure 6.23 depicts a very typical instance of partial duplication that could be corrected by the automatic routine. In this example, $A \rightarrow C$ (see Figure 6.23-b) is a conflated pedestrian way which comes from the road network of ATKIS initially (see $A' \rightarrow C'$ in Figure 6.23-a) and $A \rightarrow B$ is a road stubble from the dataset of NAVTEQ. As the $A \rightarrow B$ reveals quite different geometries to $A' \rightarrow C'$, e.g. $A \rightarrow B$ is much shorter than $A' \rightarrow C'$, these two roads have not been matched together by the automatic routine even though they are partially corresponding in the reality. Considering that the pedestrian way $A \rightarrow C$ and the road $A \rightarrow B$ intersect at point A and angle $\angle BAC$ is small enough, the pedestrian way $A \rightarrow C$ should be automatically transformed to $B \rightarrow C$ in the ultimate conflated dataset to avoid the partial duplications (see Figure 6.23-c).



(a) data matching *(b) data conflation* *(c) error correction*

orange lines: NAVTEQ; grey lines: ATKIS;

dashed lines: conflated pedestrian ways; green arrows: linkages (control point pairs)

Figure 6.23 The process to solve the problems of partial duplications

Category **3**: *Conflated pedestrian ways that are possibly wrong*

The possible wrong conflation refers to the conflated pedestrian ways which are **(i)** located nearly to the roads from the dataset of NAVTEQ; or **(ii)** crossing over a road from the dataset of NAVTEQ without any intersection; or **(iii)** open-ended on both from-node and to-node of the conflated pedestrian ways, etc.

Interaction tools are developed to deal with these possibly wrong pedestrian ways. Traditionally, human operators take the responsibility to identify the errors, find out corresponding solutions and then manually correct the geometry or attribution with provided interaction tools. Our interaction tools are rather intelligent as they are computer-assisted and therefore allow semi-automatic interaction processes. At first, these tools will focus on the possibly wrong conflated pedestrian ways one by one; then the list of all the possible solutions for them will be calculated. Thus, what the human operators have to do is just choosing the best solution for error corrections. In this way, the human interaction processes are substantially simplified which leads to an enhancement of working efficiency, which is highly desirable for the treatment of mass data.

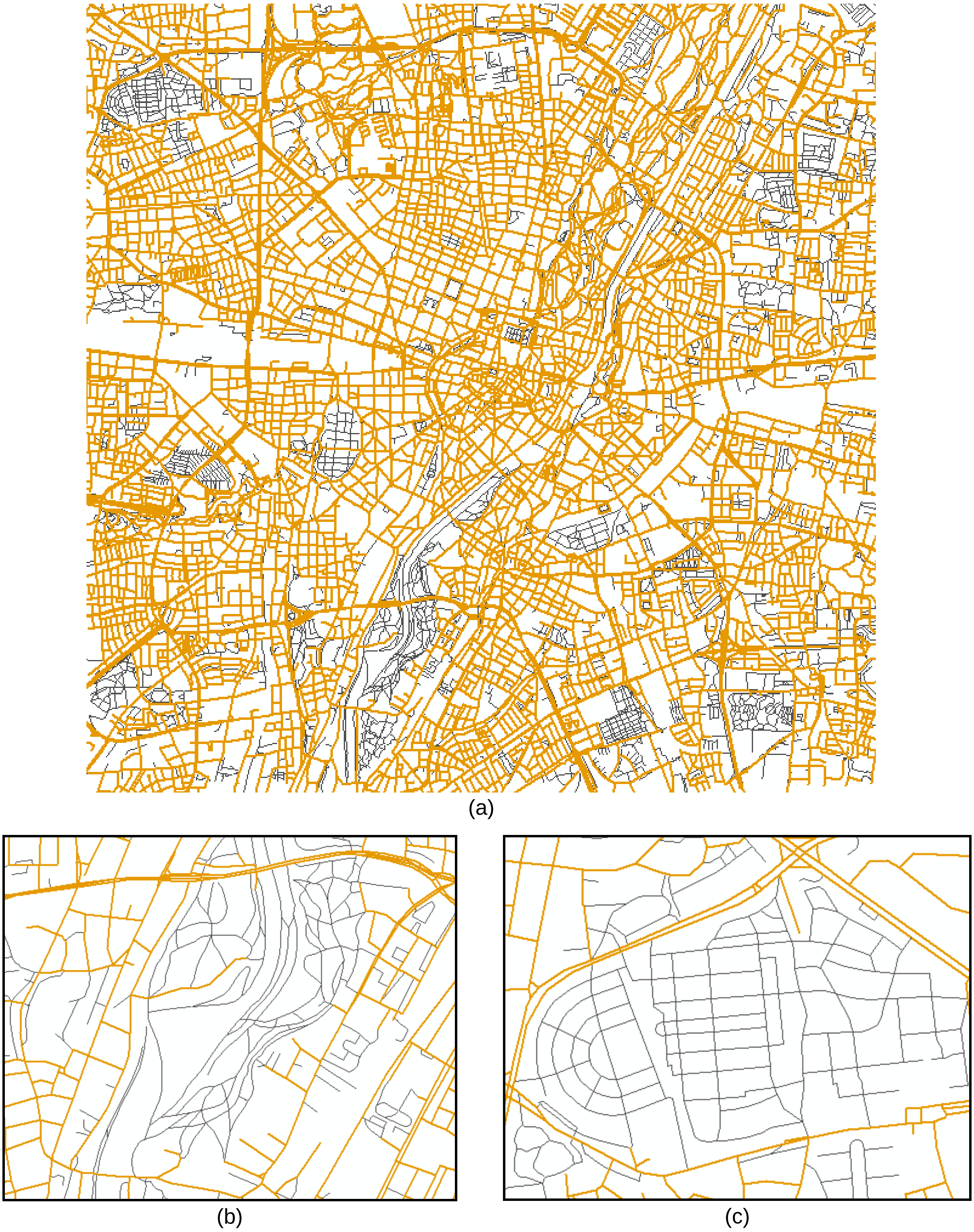
Category **4***: Reliable conflated pedestrian ways*

The cases not belonging to Category 1, 2, 3 can be treated as reliable conflations, which provide very accurate results - the overall statistical correctness exceeds 99.5% in the conducted experiments.

6.3.6 Discussion of the conflation results

As mentioned earlier, the initial dataset of NAVTEQ does not cover many pedestrian ways which are prohibited to motor vehicles. Following the conflation processes defined from Section 6.3.1 to 6.3.5, the ATKIS pedestrian ways which do not exist in NAVTEQ are selected, transformed, remodelled and then appended to the road network of NAVTEQ.

Figure 6.24~6.25 show two examples of the enriched NAVTEQ with additional pedestrian roads from ATKIS: one is in built-up area (10*10 km^2, Munich, Germany) and the other is in rural area (7*7 km^2, Garmisch, Germany); where the orange lines represent the initial road network of NAVTEQ while the grey lines represent the conflated pedestrian ways from ATKIS. The enlarged sections in Figure 6.24b-c~6.25b-c illustrate that the appended pedestrian ways and initial NAVTEQ roads have rather consistent position and topologic connection in the conflated road networks, which is very important for routing calculations.

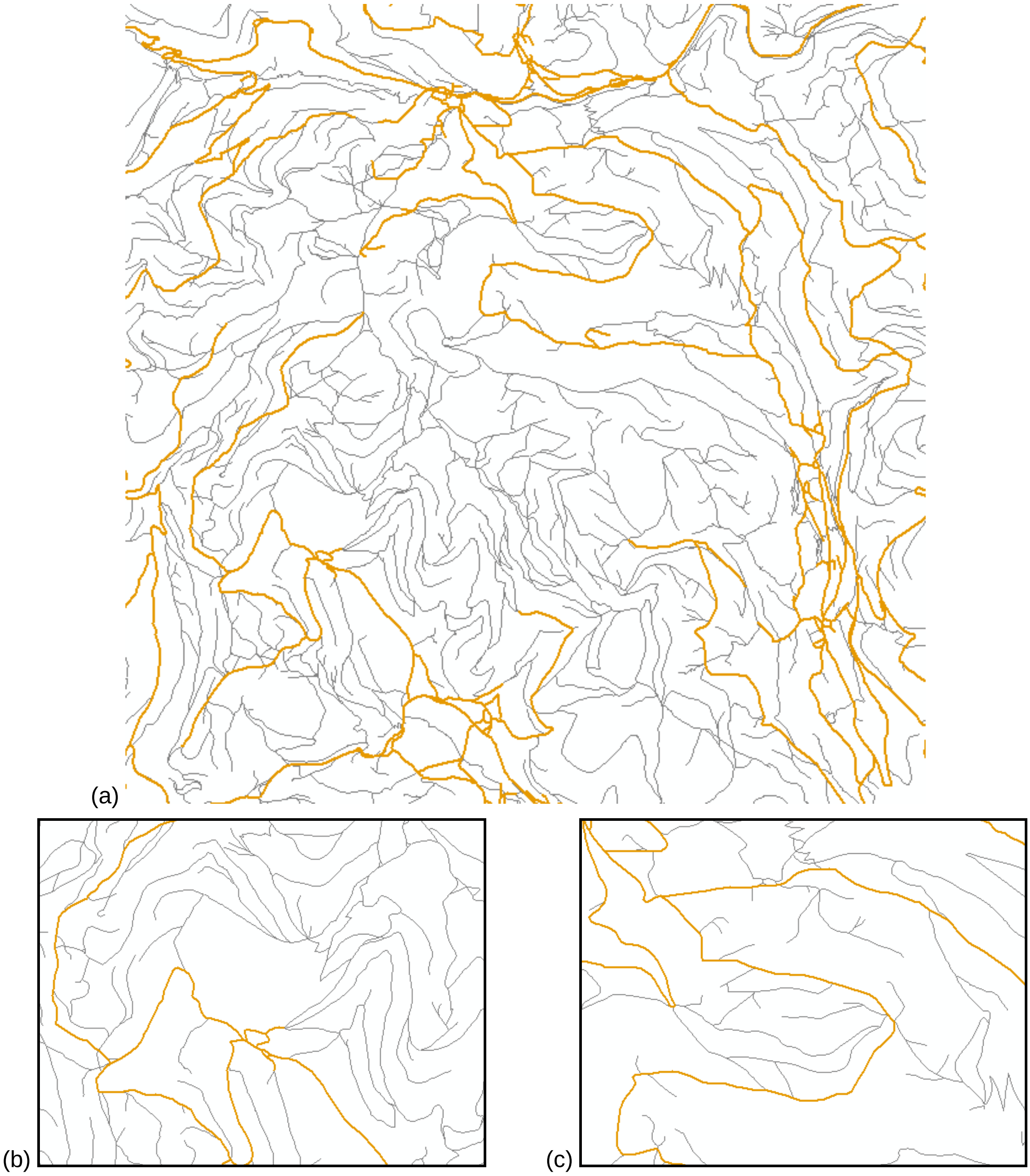

(a)

(b)

(c)

orange: the initial road network of NAVTEQ; grey: the conflated roads from ATKIS;

Figure 6.24 An example of the conflated road network in a built-up area (10*10 km^2, Munich, Germany)



orange: the initial road network of NAVTEQ; grey: the conflated roads from ATKIS;

Figure 6.25 An example of the conflated road network in a rural area (7*7 km^2, Garmisch, Germany)

Obviously, the conflated road network allows now the multi-modal navigations of ‘driving + walking’ due to the fact of that (i) it involves both motor roads and pedestrian ways; (ii) the motor roads are fully attributed with the necessary routing-relevant information from NAVTEQ; and (iii) the appended roads do not require so many routing-relevant attributes for the navigational purposes since they are anyway prohibited to motor vehicles. Usually the average travel speed on the pedestrian ways can be approximately set as 4 km/hour.

Worth mentioning is that the network conflation approach has been successfully implemented for the purpose of creating multi-modal navigational database in the region of whole Germany and is being tested for other European Countries and Metroregions, incl.: Austria, Switzerland, France, Belgium (BE), Netherlands, Luxembourg, Denmark, Poland, Czech Republic, etc. (United Maps 2009).

Chapter 7

Conclusions and Outlook

7.1 Conclusions

The thesis is dedicated to methods of road-network matching as well as their implementations in various case studies. The author has developed an efficient DSO algorithm on the basis of Delimited Strokes, extended the algorithm step by step to a contextual matching approach which is then further strengthened by three assisting methodologies (a) matching guided by ‘structure’, (b) matching guided by ‘semantics’ and (c) matching guided by ‘spatial index’.

The DSO algorithm consists of five processes: (1) data pre-processing to reduce the noise of irrelevant details and eliminate topologic ambiguities in the datasets to be matched; (2) construction of the graph to record the relationships between conjoint objects; (3) construction of the Delimited Strokes; (4) matching of the Delimited Strokes; and (5) treatment of fragmental matching areas. With the help of graph, the conjoint objects to a Delimited Stroke can be easily brought together. The resulted network is then treated as an integral unit in the matching process, i.e. the DSO algorithm is essentially a network-based matching method. As compared with point- or line-based matching methods such as Iterative Closest Point (ICP) and Buffer Growing which are not adequate to handle topologic information, the network-based approach allows a context-related topologic analysis, thus helps to improve the results of geometric matching. To overcome the dilemma that longer Delimited Strokes lead to higher matching accuracy and efficiency but lower matching rate, process (3) and (4) run iteratively at three different levels where the Delimited Strokes are progressively constructed from more aggregated to more fragmental level. Furthermore, the contextual DSO algorithm is enhanced by extendable Delimited Strokes and the matching procedure dealing with fragmental areas. In the end, it is not only able to identify the matching pairs with $m:n\ (m \geq 1, n \geq 1, m, n \in N)$ relationship, but can calculate the partial correspondences which have been seldom considered by previous algorithms reported in literature. Noteworthy is also the generic nature of the DSO algorithm because it can work with the worst case, for example, one or both of the datasets to be matched have no or little semantic information. The gained insight from road-network matching can therefore be easily adapted to the task of matching other linear features such as hydrological networks. More generally, the link cardinalities are also applicable for the matching of polygon data types.

To certain extent, a road network can be regarded as a unit constituted by various road structures, such as dual-carriageways (parallel lines), roundabouts, narrow passages, navigation stubbles, slip roads around cloverleaf junctions and single carriageways. Since the various road structures take on different geometric or topologic characteristics, it is hardly possible to efficiently match all of them using the same criteria or methods. In order to circumvent this problem, the contextual matching approach employs a strategy of matching guided by ‘structure’, which focuses on two challenging cases that have not yet been considered in many matching programs reported so far - parallel lines and looping crosses. Following such a matching strategy, the special processes dealing with structures of carriageways and roundabouts are triggered in advance of the single carriageways. If a ‘structure’ from the reference dataset can not find its corresponding structure from the target dataset, it will be treated as a single carriageway hereafter. Being supported by certain generalization operators, the ‘structure’-guided matching strategy is able to correlate the corresponding roundabouts and dual carriageways at either similar or dissimilar LoDs, and thus leads to an enhanced matching rate as well as the matching accuracy.

Another significant contribution of this thesis is the matching strategy guided by ‘semantics’. The semantic attributes are divided into two categories: objective semantic attributes which describe the inherent and explicit properties/characteristics of geo-objects; and subjective semantic attributes which represent fuzzy characteristics and artificial properties of geo-objects. Different semantic categories trigger different criteria in the matching process. The functionalities of ‘street name’ as an important attribute have been extensively investigated in different scenarios of data matching. It is important to reiterate that the semantic matching is an optional operator in the contextual matching approach. If the semantic attributes are available in both of the datasets to be matched, then they will be fully utilized to improve the matching performance; otherwise, the matching approach will only depend on the geometry and topology.

In addition to strive for high matching performance in terms of completeness and accuracy, further efforts have been made to accelerate the computing speed. The contextual matching approach uses two grid-based spatial indexes which are developed to organize point and linear data respectively. With help of the spatial indexes, the process of road-network matching has been dramatically accelerated, especially when the datasets are huge. More important is that an increase of network size will not lead to a perceivable slowing down of the matching speed.

To summarize, the experiments conducted in this thesis work prove that the proposed contextual matching approach has yielded a considerably improved matching performance in many aspects:

(a) Automatic matching rate and accuracy: in a number of large test areas in Germany, the overall matching rate exceeds 97%; among the matched objects more than 99.2% are correct; i.e. on average 96.5% (99.2%*97%) of the objects in reference datasets were accurately matched to their counterparts in target datasets together (ref. Table 5.6). As mentioned earlier, there are two reasons that can lead to imperfections of the automatic data matching - algorithm limitations and data ambiguity. In our experiments, the ‘data ambiguity’ has been confirmed as the primary inducement for the unfavourable matching results.

With less than 3% unmatched objects (viz. false negative match) and 0.8% matching errors (incl. mismatch and false positive match) in spite of data complexity and ambiguity, the contextual matching approach is highly successful. Human interactive editing is still required in most of computer-aided systems. The definition of the matching certainty and the concomitant classification of the matching results can significantly simplify the process of detecting the matching errors as most of the matching errors are concealed at the certainty level of ‘possible’. We believe that holistically perfect matching approaches are possible only if the context information can be adequately considered.

(b) High computing speed: the contextual matching approach is very effective. The matching of a test area of ca. 1200 km2 with 10947 ATKIS objects and 10360 Tele Atlas objects in total takes only 22 seconds by a personal computer with a CPU of Intel Duo 2.0 Hz, i.e. about 500 objects per second. Comparing to an experienced human operator who is able to match 5 to 6 pairs in a minute, the contextual matching approach is able to match the data thousands times faster and this computing speed tends to be stable in much larger matching areas.

(c) Robustness and generic nature: With the contextual matching approach, not only $m{:}n\ (m \geq 1, n \geq 1,\ m,\ n \in N\,)$ matching pairs, i.e. m objects in reference dataset is corresponding to n objects in target datasets, but also the matching pairs with partial or equivalent correspondences can be identified. As a result, two homologous road networks can be directly matched together regardless whether they have similar representations or not. Moreover, the contextual matching approach essentially relies on comparisons of the geometries and topologies; therefore, it is insensitive to the availability of semantic information. This generic nature makes it applicable for all kinds of road networks from different data resources. The experiments have addressed different representative matching tasks between (i) NAVTEQ and ATKIS, (ii) Tele Atlas and ATKIS, (iii) OpenStreetMap and

NAVTEQ, and (iv) Tele Atlas and NAVTEQ, on a number of urban, rural and mountain areas. Up to date, the test matching areas has exceeded 300,000 km^2 which covered more than 20,000,000 road objects in total.

The advantages of the contextual matching approach have been extensively confirmed in three real-world projects:

- **Postal data integration**

This project was sponsored by German Federal Agency for Cartography (BKG) and aims at enriching Basis DLM with geo-referenced house numbers of post addresses. The available post addresses from German Federal Post Office were manually collected by postmen on the basis of road geometries from Tele Atlas Corp. Since the road database of Tele Atlas reveals a different geometric / semantic accuracy from that of Basis DLM, the attempt of a direct integration of postal data in Basis DLM is doomed to fail. Therefore, we divide the enrichment process into two main stages. The first stage is dedicated to matching road objects between Basis DLM dataset as reference and Tele Atlas dataset as target. In the second stage, a projection based on the Rubber-Sheet principle is established for each pair of matched road lines. Thus, the discrete locations of house numbers as well as any other arbitrary points along a Tele Atlas road line can find their corresponding positions along the homologous line in Basis DLM.

- **Integration of the routing-relevant information from different datasets**

As an extension of “Postal data integration”, this project was sponsored by BKG as well. In this project, the contextual matching approach was refined and implemented to integrate routing-relevant information from different data sources of Tele Atlas and DLM De. Other methods of road data integration reported in literature are mostly focused on the identification of matching pairs and transferring the individual attributes that directly describe the underlying road objects. In this project, however, more complicated relationships between the road geometries and meaningful network attributes for special purposes such as navigation have been considered. The approach involves three processes: (a) automatic matching to identify the corresponding road objects between different datasets; (b) interaction to refine the automatic matching result; and (c) transferring the routing-relevant information from one dataset to another.

- **Conflation of the pedestrian ways between different datasets**

This project is funded by the Corp. United Maps for the purpose of multi-modal navigation. In the recent years, several navigator producers, such as Nokia and TomTom, began to target pedestrians for its navigation services rather than drivers, which indicate that the GPS navigators will be no longer restricted on motor ways, but also serve the pacers on the pedestrian ways. However, as one of the most well-known existing routing-capable database, NAVTEQ does not contain sufficient pedestrian road features since their data were primarily captured by GPS-supported equipments on cars; whereas the ATKIS data were captured through map digitization in combination of semiautomatic object extraction from imagery data and therefore roundly covers the pedestrian areas. In order to achieve the objectives of multi-modal navigation, the contextual matching approach has been implemented to conflate the road networks from different sources. The data conflation in whole Germany with a total matching area of ca. 360,000 km^2, more than 15,388,000 ATKIS objects and 6,690,000 NAVTEQ objects has been successfully accomplished. The same method is now being experimented with data from South Africa and many other European counties, such as Austria, Switzerland, France, Belgium (BE), Netherlands, Luxembourg, Denmark, Poland, Czech Republic, etc.

The matching performance and the successful applications in different real-world projects demonstrate that the contextual matching approach has the substantial potential for processing and enrichment of mega data.

7.2 Outlook

In spite of the convincing performance of the contextual matching approach, a number of further improvements and extensions are anticipated in order to keep pace with rapid technological evolutions:

- **Fine-tuning of parameter setting**

The matching performance of the proposed approach often depends on the appropriate parameters settings. For example, in the DSO matching processes, a number of geometric constraints have been employed to exclude unlike matching pairs. These values essentially depend on the data models and can not be formulated in a universal way. Therefore, a series of statistic investigations on various matching tasks have been conducted in this thesis (ref. section 5.4), which can help to assign proper user-defined tolerance values for the different geometric constraints: some of them should be settled as static numbers, whereas the others have to combine a default setting with an interactive user interface that allows the operator to visually estimate the threshold parameters based on the apparent matching pairs from the actual datasets. As the result, a more reliable matching result can be achieved. To confirm the best ‘match’, however, it is necessary to assess the similarity of two correlated object chains from different datasets, where the geometric, topologic and (or) semantic criteria have to be considered together. One of the most popular ways for such an assessment is to weigh the different criteria according to their relative contributions. In the contextual matching approach, the weights of different criteria are empirically determined, therefore, they may not precisely represent the relative importance of each threshold. One possible way to solve this problem is to develop another learning module, for instance, based on the Artificial Neural Network so that the weights can be automatically determined after sufficient training cycles with examples.

- **Development of an automatic process to update the matching results**

After two datasets are matched together, the matching results will be stored somewhere for the further applications. Obviously, once the reference dataset or target dataset is updated, the matching results have to be changed as well. Even if the contextual matching approach has revealed high matching rate and certainty, some uncertain matching problems remain in areas where topologic conditions are too inconsistent or ambiguous to allow a reliable identification of matching pairs. Although the definition of the matching certainty as well as the classification of the matching results facilitates a more comfortable interaction, it is still a tedious manual post-processing work to refine the automatic matching results. Therefore, it will consume a lot of resources if the whole matching process is conducted again. In order to reduce the computer calculation as well as the manual interaction work, an automatic updating process has to be developed for the mega-data enrichment. According to our experiences, the updating process could include two steps: (1) detecting the changes; and (2) updating the matching results. In the first step, the changes involve the removed, appended and modified geospatial objects. For the removed objects, the updating process is just the deletion of the relevant matching pairs. For the appended objects, however, the updating process becomes a bit more perplex, which requires a matching calculation to identify new matching pairs for the appended objects. In practice, the objects modification is often treated as the combination of ‘removing old objects and then appending new ones’. Consequently, the expected updating process of modified objects should also consist of two operations, viz.: ‘deleting the relevant matching pairs’ and ‘identifying new matching pairs’.

- **Construction of a Web Processing Service for on-line data matching**

Based on the service specifications of the Open Geospatial Consortium (OGC), a Web Processing Service (WPS) offers web access to a variety of geoprocessing functionalities by a standardized interface. A WPS may offer calculations as simple as subtracting one set of spatially referenced numbers from another (e.g., determining the difference in influenza cases between two different seasons), or as complicated as a global climate change model (Lanig and Zipf 2009). The contextual matching approach proposed in this thesis is based on the principle of *“What You Can See Is What*

You Can Match (WYCSIWYCM)" and thereby can lead to an on-the-fly matching process that can deal with various kinds of road networks. With the availability of WPS, the on-the-fly matching process can be directly published on-line, which allows a new way to provide virtual IT productions. Without having to download and install the matching program to the local computer, the authorized users can also perform the matching program to process their source data. Worthy to mention is the source data required by the WPS can be available locally or at the server, or delivered across a network using data exchange standards such as Geography Markup Language (GML) or Geolinked Data Access Service (GDAS).

All further tasks are intended to consolidate the advantages of the contextual matching approach on the one hand and make it accessible as a sharable web service for a wide spectrum of internet users on the other hand. At present, the approach is undergoing the commercialization process which will be finally embedded in an extensive platform of 'spatial data matching and integration'. In the future, the contextual matching technologies will play more and more significant roles for various industrial or commercial purposes, such as (1) enhancing the applicability of the existing data by transferring attributes or object classes from one dataset to another; (2) evaluating and improving the data quality by comparison of different datasets; (3) automatic maintaining or updating datasets in MRDB; (4) appending methodologies for data mining and knowledge discovery; etc..

Bibliography

AdV, Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (2003). Amtliches Topographisch- Kartographisches Informationssystem ATKIS, 2003

Anders, K. -H. & Bobrich J. (2004). MRDB approach for automatic incremental update, ICA Workshop on Generalisation and Multiple Representation, Leicester, England, August 2004

Arkin, E., Chew, P., Huttenlocher, D., Kedem, K. & Mitchel, J. (1991). An efficiently computable metric for comparing polygonal shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13(3), pp. 209-216, 1991

Badard, T. (1999). On the automatic retrieval of updates in geographic databases based on geographic data matching tools, ICC-Proceedings, '99, Ottawa, pp. 1291-1300

Baltsavias, E.P., (2004). Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems, ISPRS Journal of Photogrammetry and Remote Sensing 58 (3–4), pp. 129-51

Besl, P. & McKay, N. (1992). A method for reginstration of 3-d shapes, Trans. PAMI, Vol. 14(2), pp. 239-256

Beyer, W. g. (1987). CRC Standard Mathematical Tables, 28th ed. Boca Raton, FL: CRC Press, pp. 123-124, 1987

Blasby, D., Davis, M., Kim, D. & Ramsey, P. (2003). A white paper - GIS Conflation using Open Source Tools, http://www.jump-project.org/assets/JUMP_Conflation_Whitepaper.pdf (accessed on 2009-07-27)

Brown, J., Rao, A. & Baran, J. (1995). A Full-Featured ArcInfo Conflation System, Proc. of the Fifteenth Annual ESRI User Conference, 1995, http://training.esri.com/campus/library/Bibliography/RecordDetail.cfm?ID=9621 (accessed on 2009-07-27)

Butenuth, M., Gösseln, G.v., Tiedge, M., Heipke C., Lipeck, U. & Sester M. (2007). Integration of heterogeneous geospatial data in a federated database, ISPRS Journal of Photogrammetry & Remote Sensing 62 (2007), 328-346

Cecconi, A. (2003). Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping, dissertation, University Zurich, 2003

Chen, C.-C. (2005). Automatically and Accurately Conflating Road Vector Data, Street Maps and Orthoimagery, Ph. D. Dissertation, University of Southern California, May 2005

Chen, C.-C., Thakkar, S., Knoblok, C.A. & Shahabi, C. (2003). Automatically Annotating and Integrating Spatial Datasets, Proc. of the 8th International Symposium on Spatial and Temporal Databases (SSTD'03), Santorini Island, Greece, July 24-27, 2003, pp. 469-488 http://www.isi.edu/info-agents/papers/chen03-sstd.pdf (accessed on 2009-10-08)

Cichociski, P. (2008): Application of Advanced Topological Rules in the Process of Building Geographical Databases Supporting the Valuation of Real Estates, Integrating Generations, FIG Working Week 2008 Stockholm, Sweden, 2008

Cobb, M.A., Chung, M.J., Foley, H., Petry, F.E., Shaw, K.B. & Miller, H.V. (1998). A rule-based approach for the conflation of attributed vector data, Geoinformatica, Vol. 2, No.1, pp. 7-35

Cohen, W.W. (2000). Data integration using similarity joins and a word-based information representation language, ACM Transactions on Information Systems (TOIS), Vol. 18(3), pp. 288 ~ 321, 2000

Cormen, T.H., Leiserson, C.E., Rivest, R.L. & Clifford Stein (2001). Introduction to Algorithms, Second Edition, MIT Press and McGraw-Hill, pp. 527-529 of section 22.1: Representations of graphs. ISBN 0-262-03293-7

Dempster, A. (1968). Upper and lower probabilities induced by multivalued mapping, Annals of Mathematical Statistics, AMS-38, pp. 325-339

Deng, M., Chen, X. & Li, Z. (2005): A Generalized Hausdorff Distance For Spatial Objects In GIS, in Proc. of the 4th ISPRS Workshop on Dynamic and Multi-dimensional GIS, pp. 10 -15, Pontypridd, UK, 2005

Deretsky, Z. & U. Rodny (1993). Automatic Conflation of Digital Maps, Proc. of IEEE-IEE Vehicle Navigation & Information Systems Conference, A27-A29, 1993

Devogele, T. (2002). A new Merging Process for Data Integration Based on the Discrete Frechét Distance, Proc. of the Joint International Symposium on Geospatial Theory, Processing and Applications. Ottawa

Devogele, T, Parent, C. & Spaccapietra S. (1998). On spatial database integration, International Journal of Geographical Information Science, Vol.12(4), pp. 335-352.

Devogele, T., Trevisan, J. & Raynal, L. (1996). Building a multi-scale database with scale-transition relationships, Advances GIS Research II, Taylor & Francis, London, pp. 337-351

Dowman, I., (1998). Automated procedures for integration of satellite images and map data for change detection: the archangel project, International Archives of Photogrammetry and Remote Sensing 32 (Part 4), pp. 162-169.

Doytsher, Y., Filin, S. & Ezra, E. (2001). Transformation of Datasets in a Linear-based Map Conflation Framework, Surveying and Land Information Systems, Vol. 61, No. 3, 2001, pp.159-169

Dunkars, M. (2003). Matching of datasets, ScanGIS'2003 - The 9th Scandinavian Research Conference on Geographical Information Science, Espoo, Finland, June 4-6, 2003, http://www.scangis.org/scangis2003/papers/19.pdf (accessed on 2009-07-27)

Dunkars, M. (2004): Multiple representation databases for topological information, Dissertation, ISBN: 91-7323-100-2, KTH, Infrastructure, Stockholm, Sweden

Eiter, T. & Mannila H. (1994). Computing discrete Fréchet distance, Technical report of Christian Doppler Labor für Expertensensyteme, Technical University of Vienna, Austria, http://www.kr.tuwien.ac.at/staff/eiter/et-archive/cdtr9464.ps.gz (accessed on 2009-07-27)

Fisher, R., Perkins, S., Walker, A. & Wolfart, E. (2003). Affine Transformation, 2003, http://homepages.inf.ed.ac.uk/rbf/HIPR2/affine.htm (accessed on 2009-07-27)

Franklin, W.R., Sivaswami, V., Sun, D., Kankanhalli, M. & Narayanaswami, C. (1994). Calculating the area of overlaid polygons without constructing the overlay, Cartography and Geographic Information Systems 21, pp. 81-89

Gabay, Y. & Doytsher, Y. (1994). Automatic adjustment of line maps, Proc. of GIS/LIS'94 Annual Convention. American Congress on Surveying and Mapping, American Society for Photogrammetry and Remote Sensing, Association of American Geographers, Urban and Regional Information Systems Association, and AM/FM International, Phoenix, Arizona, pp. 333-341

Gabay, Y. & Doytsher, Y. (1995). Automatic feature correction in merging of line maps, Proc. of the 1995 ACSM-ASPRS Annual Convention 2, pp. 404-410

Gillman, D.W. (1985). Triangulations for Rubber-Sheeting, Proc. of Auto-Carto 7, Falls Church, VA: ACSM/ASP

Gösseln, G. v. & Sester, M. (2003). Semantic and geometric Integration of geoscientific Data Sets with ATKIS - Applied to Geo-Objects from Geology and Soil Science, Proc. of ISPRS Commission IV Joint Workshop "Challenges in Geospatial Analysis, Integration and Visualization II", Stuttgart, Germany, pp. 111-116, September 8 - 9, 2003

Gösseln, G.v. & Sester, M. (2004). Intergration of geoscientific data sets and the German digital map using a matching approach, XXth ISPRS Congress, Istanbul, Turkey, pp. 1249-1254, July 12-23, 2004

Gösseln, G.v. (2005). A matching approach for the integration, change detection and adaptation of heterogeneous vector data sets, XXII International Cartography Conference, 2005

Hampe, M. & Sester, M. (2004). Generating and Using A Multi-Representation Data-Base (MRDB) for Mobile Applications, Papers of the ICA Workshop on Generalisation and Multiple Representation, Leicester, 9 pages in CD-Proc., August 20-21, 2004

Hangouët, J.-F. (1995). Computation of the Hausdorff Distance between Plane Vector Polylines. Auto-Carto12, ACSM/ASPRS Annual Convention & Exposition Technical Papers, Charlotte, NC, USA, February 27-March 1, 1995

Haunert, J.-H. & Sester, M. (2004). Using the Straight Skeleton for Generalisation in a Multiple Representation Environment, 7th ICA Workshop on Generalisation and Multiple Representation, 2004

Haunert, J-H and Sester, M. (2008). Area Collapse and Road Centerlines based on Straight Skeletons, Geoinformatica (2008) 12, pp.169-191, Springer, 2008

Hoelf, E.G. & Samet, H. (1991). Efficient processing of spatial queries in line segment databases, Proc. of 2nd Symposium on Large Spatial Database (SSD’ 91), Zurich, Switzerland, August 1991

Hu, Y., Chen, J., Li, Z., Zhao R. (2008): Road Data Updating Using Tools of Matching and Map Generalization, ISPRS Vol. XXXVII, Part B4. Beijing, 2008

Jones, C B, Kidner, D B, Luo, L Q, Bundy Gl & Ware J M. (1996). Database design for a multi-scale spatial information system, International Journal of Geographical Information Science, Vol. 10(8), pp. 901-920

Jong-Sun, P., Dong-Ho, S. & Tae-Kyung, S. (2001). Development of a map matching method using the multiple hypothesis technique, IEEE Intelligent Transportation Systems Conference. Oakland, CA, USA

JUMP (2006). The JUMP Project, http://www.jump-project.org/, (accessed on 2009-10-08)

Kampshoff, S. (2005). Integration heterogener raumbezogener Objekte aus fragmentierten Geodatenbeständen, Dissertation, Rheinisch-Westfälischen Technischen Hochschule Aachen, Germany, 2005

Kang, H. (2001). Spatial Data Integration: A Case Study of Map Conflation with Census Bureau and Local Government Data, University Consortium for Geographic Information Science, Summer Assembly, June 2001, http://www.cobblestoneconcepts.com/ucgis2summer/kang/kang_main.htm (accessed on 2009-10-10)

Kashyap, V. & Sheth, A. (1996). Semantic and Schematic Similarities between Database Objects: A Context-based approach, In. International Journal on Very Large Data Bases, Vol. 5/4, pp.276-304, Spinger, 1996

Kolahdouzan, M., Chen, C., Shahabi C. & Knoblock C. A. (2005). GeoMatchMaker Automatic and Efficient Matching of Vector Data with Spatial Attributes in Unknown Geometry Systems, UCGIS Summer Assembly 2005, USA, http://www.ucgis.org/summer2005/studentpapers.htm (accessed on 2009-07-28)

Lanig, S. & Zipf, A. (2009): Interoperable processing of digital elevation models in grid infrastructures, Earth Science Informatics, volume 2, pp.107-116, Springer, 2009

Levenshtein VI (1965). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals, Soviet Physics - Doklady, 10(8), pp. 707-710, Translated from Doklady Akademii Nauk SSSR, 163(4), pp. 845-848

Löcherbach, T. (1994). Fusion of multi-sensor images and digital map data for the reconstruction and interpretation of agricultural landuse units, International Archives of Photogrammetry and Remote Sensing 30 (Part 3/2), pp. 505-511

Lupien, A. E. & Moreland, W.H. (1987). A general approach to map compilation, AUTOCARTO 8, ACSM-ASPRS, Falls Church, Va., pp 630-639

Lüscher, P., Burghardt, D. & Weibel, R. (2007). Matching road data of scales with an order of magnitude difference, CD-Proc. of the XXIII International Cartographic Conference (ICC), Incremental Updating and Versioning of Spatial Data, Moscow, Russia, August 4-10, 2007

Lv, W., Liao, W., Wu, D. & Xie, J. (2008). A New Road Network Model and Its Application in a Traffic Information System, Proc. of the Fourth International Conference on Autonomic and Autonomous Systems (ICAS 2008)

Mantel, D. & Lipeck, U. (2004). Matching Cartographic Objects in Spatial Databases, ISPRS Vol. XXXV, ISPRS Congress, Commission 4, Istanbul, Turkey, July 12-23, 2004

Marchal, F., Hackney, J. & Axhausen, K. W. (2004). Efficient Map-Matching of Large GPS Data Sets - Tests on a Speed Monitoring Experiment in Zurich, Vol. 244 of Arbeitsbericht Verkehrs und Raumplanung

Markovsky, I. & Mahmoodi, S. (2009). Least squares contour alignment, IEEE Signal Processing Letters, 16 (1). pp. 41-44. ISSN 1070-9908

Mascret, A., Devogele, T., Le Berre, I. & Hénaff, A. (2006). Coastline matching process based on the discrete Fréchet distance, Proc. of the 12th International Symposium on Spatial Data Handling (SDH), pp. 383-400, Springer: Vienna, Austria, 2006.

Meng, L. & Töllner, D. (2004). Ein Reverse-Engineering-Ansatz zur Generalisierung topographischer Daten, KN 4/2004, pp.159-163

Min, D., Zhilin, L. and Xiaoyong, C. (2007). Extended Hausdorff distance for spatial objects in GIS, International, Journal of Geographical Information Science, Vol. 21(4), pp. 459 - 475

Moosavi A. & Alesheikh A. A. (2008). Developing of Vector Matching Algorithm Considering Topologic Relations, Proc. of Map Middle East 2008, Dubai, UAE, 2008

Mustière, S. & Devogele, T. (2008). Matching networks with different levels of detail, GeoInformatica (2008), Vol.12, pp. 435-453, Springer

Mustière, S. (2006). Results of experiments of automated matching of networks at different scales, ISPRS Vol. XXXVI. ISPRS Workshop - Multiple representation and interoperability of spatial data, Hannover, Germany, February 22-24, 2006

Navarro, G. (2001). A guided tour to approximate string matching, ACM Computing Surveys, Vol. 33(1), pp. 31-88, 2001

Neuland, M. & Kürner, T. (2007). Analysis of the impact of map-matching on the accuracy of propagation models, Advances in Radio Science, Vol. 5, pp.367-372

Novak, K., (1992). Rectification of digital imagery, Photogrammetric Engineering and Remote Sensing 58, pp. 339-344

Nystuen, J.D., Frank, A.I., Frank, Jr., L., (1997). Assessing topological similarity of spatial networks, Proc. of International Conference on Interoperating Geographic Information Systems, Santa Barbara, California, USA

Ochieng, W.Y., Quddus, M.A. & Noland, R.B. (2003). Map-Matching in Complex Urban Road Networks, Brazilian Journal of Cartography (Revista Brasileira de Cartografia), Vol. 55 (2), pp. 1-18

Olteanu A.-M. (2007). Matching geographical data using the Theory of Evidence, CD-Proc. of the XXIII International Cartographic Conference (ICC), Incremental Updating and Versioning of Spatial Data, Moscow, Russia, August 4-10, 2007

Olteanu, A.-M., Mustière, S., & Ruas, A. (2006). Matching imperfect spatial data, 7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, Lisbon, Portugal, July 5-7, 2006

Paleo, B.W. (2007). An approximate gazetteer for GATE based on levenshtein distance, Student Section of the European Summer School in Logic, Language and Information (ESSLLI), http://www.logic.at/people/bruno/Papers/2007-GATE-ESSLLI.pdf, (accessed on 2009-07-27).

Parent, C. & Spaccapietra, S. (2000). Database Integration: the Key to Data Interoperability, Advances in Object-Oriented Data Modeling, Papazoglou, M. P., Spaccapietra, S. & Z. Tari (Eds.), The MIT Press, 2000, http://lbdwww.epfl.ch/e/publications_new/articles.pdf/OObook.pdf (accessed on 2009-07-27)

Pendyala, R.M. (2002). Technical Report: Development of GIS-based Conflation Tools for Data Integration and Matching, Final Report: Executive Summary, http://www.dot.state.fl.us/research-Center/Completed_Proj/Summary_PL/FDOT_BC353_21_rpt.pdf (accessed on 2009-10-01)

Pyo, J.-S., Shin, D.-H. & Sung T.-K. (2001). Development of a Map Matching Method Using The Multiple Hypothesis Technique, Proc. of IEEE Intelligent Transportation Systems Conference, pp. 23 - 27

Quddus, M. A., Noland, R. B., & Ochieng, W. Y. (2006). A High Accuracy Fuzzy Logic Based Map Matching Algorithm for Road Transport, Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, Vol. 10 (3), pp.103-115

Quddus, M.A., Ochieng, W.Y., Zhao, L. & Noland, R.B. (2003). A General Map Matching Algorithm for Transport Telematics Applications. GPS Solutions Journal, Vol. 7 (3), pp. 157-167.

Raimond, A.-M. O. & Mustière, S. (2008). Data Matching - a Matter of Belief, Headway in Spatial Data Handling, 13th International Symposium on Spatial Data Handling (SDH), pp. 501-519, Spinger Montpellier, France, 2008

Rosen, B. & Saalfeld, A. (1985). Match Criteria for Automatic Alignment, Proc. of Auto-Carto VII, 1985

Rossum, G. v. (1998). Python Patterns - Implementing Graphs, Python Software Foundation, http://www.python.org/doc/essays/graphs.html, (accessed on 2009-07-27)

Rucklidge, W.J. (1996). Efficient visual recognition using the Hausdorff distance, a book of Lecture Notes in Computer Science, 1173, Berlin: Springer.

Saalfeld, A. (1985). A Fast Rubber-Sheeting Transformation Using Simplicial Coordinates," October issue of The American Cartographer

Saalfeld A. (1988). Conflation: Automated map compilation, International Journal of Geographic Information Systems, Vol. 2(3): pp. 217-228

Safra, E & Doytsher, Y. (2006). Using matching algorithms for improving locations in cadastral maps, XXIII FIG Congress, Munich, Germany, October 2006

Safra, E., Kanza Y., Sagiv Y. & Doytsher Y. (2006). Efficient integration of road maps, Proc. of the 14th annual ACM international symposium on Advances in geographic information systems SESSION:Data integration, pp. 59-66, Arlington, Virginia, USA, 2006

Schimandl, F., Zhang, M., Mustafa, M., Meng, L. (2009). Real time application for traffic state estimation based on large sets of floating car data, International Scientific Conference on Mobility and Transport - ITS for larger Cities, Munich, Germany, May 12-13th, 2009

Sehgal, V., Getoor, L. & Viechnicki, P.D. (2006). Entity resolution in geospatial data integration, Proc. of the 14th annual ACM international symposium on Advances in geographic information system, SESSION: Data integration, pp. 83-90, Arlington, Virginia, USA, 2006

Sester, M., Anders, K. & Walter, V. (1998). Linking Objects of Different Spatial Data Sets by Integration and Aggregation, GeoInfomatica, Vol.2, No.4, pp 335-358

Shafer. G (1976). A Mathematical Theory of Evidence, Princeton University Press

Sheeren, D., Mustière, S. & Zucker, J.-D. (2004). How to Integrate Heterogeneous Spatial Database in a Consistent Way? ADBIS 2004: advances in databases and information systems, pp. 364-378, LNCS 3255, Springer, 2004

Sheth, A. (1991). Semantic issues in Multidatabase Systesms, SIGMOD Record, special issue on Semantic Issues in Multidatabases, ed. 20(4), 1991

Song, W., Haithcoat, T.L. & Keller, J.M. (2006). A Snake-based Approach for TIGER Road Data Conflation, Cartography and Geographic Information Science, Vol. 33, No. 4, pp. 287-298, 2006

Stigmar, H. (2005). Matching Route Data and Topographic Data in a Real-Time Environment, 10th Scandinavian Research Conference on Geographical Information Science, June 13-15, 2005, Stockholm, http://www.scangis.org/scangis2005/papers/stigmar.pdf (accessed on 2009-07-28)

Stigmar, H. (2006). Some Aspects of Mobile Map Services, Licentiate Dissertation, ISSN 1652-4810, Lund University, Sweden

Stilla, U., (1995). Map-aided structural analysis of aerial images. ISPRS Journal of Photogrammetry and Remote Sensing 50, pp. 3-10

Tele Atlas, MultiNet™ User Guide Shapefile Format, 2003

Theobald, D. M. (2001). Understanding Topology and Shapefiles, Arc User, http://www.esri.com/news/arcuser/0401/topo.html (accessed on 2009-07-27)

Tiedge M., Lipeck, U. & Mantel, D., (2004). Design of a Database System for Linking Geoscientific Data, Geotechnologien Science Report "Information Systems in Earth Management", No. 4, Koordinierungsbüro Geotechnologien, Potsdam, 2004, pp. 83-87, http://www.dbs.uni-hannover.de/schriften/publications/TLM2004a.pdf (accessed on 2009-10-01)

Thom, S. (2005). A Strategy for Collapsing OS Integrated Transport Network™ dual carriageways, 8th ICA WORKSHOP on Generalisation and Multiple Representation, A Coruña, July 7-8, 2005

Tomaselli, L. (1994). Topological transfer: evolving linear GIS accuracy. URISA Proc., pp. 245-259

United Maps (2009). http://www.unitedmaps.net/ (accessed on 2009-08-27)

Unser, M., Neimark, M.A. & Lee, C. (1994). Affine Transformations of Images: A Least Squares Formulation, Proc. of the 1994 IEEE International Conference on Image Processing (ICIP'94), Austin TX, USA, vol. III, pp. 558-561, November 13-16, 1994

Veltkamp R. C. (2001). Shape Matching: Similarity Measures and Algorithms, Technical report of Department of Computing Science, Utrecht University, Netherlands, http://www.cs.uu.nl/research/techreps/repo/CS-2001/2001-03.pdf (accessed on 2009-07-27)

Vivid Solutions (2005). http://www.vividsolutions.com/JCS/ (accessed on 2009-07-27)

Volz, S. & Walter, V. (2004). Linking Different Geospatial Databases by Explicit Relations, Proc. of the XXth ISPRS Congress, Comm. IV, Istanbul, Turkey, pp. 152-157

Volz, S. & Bofinger, J.M. (2002). Integration of spatial data within a generic platform for location-based applications, Proc. of the Joint International Symposium on Geospatial Theory, Processing and Applications, Ottawa, Canada, 2002

Volz, S. (2006). An iterative approach for matching multiple representations of street data, Hampe, M., Sester, M. and Harrie, L. (eds.): ISPRS Vol. XXXVI., ISPRS Workshop - Multiple representation and interoperability of spatial data, Hannover, Germany, February 22-24, 2006

Walter, V. & Fritsch, D. (1999). Matching Spatial Data Sets: a Statistical Approach, International Journal of Geographical Information Science, Vol.13, No.5, pp. 445-473

Walter, V. (1997). Zuordnung von raumbezogenen Daten - am Beispiel der Datenmodelle ATKIS und GDF, Dissertation, Deutsche Geodätische Kommission (DGK) Reihe C, Nummer 480.

Wang, Y. (1998). Principles and applications of structural image matching, ISPRS Journal of Photogrammetry and Remote Sensing 53, 154±165

White, M. & Griffin, P. (1985). Piecewise Linear Rubber-Sheet Map Transformations, October issue of The American Cartographer

Wikipedia (2009) ^{A}: http://en.wikipedia.org/wiki/Grid_(spatial_index), accessed on 2009-07-02

Wikipedia (2009) ^{B}: http://en.wikipedia.org/wiki/OpenStreetMap, accessed on 2009-10-02

Wu, D., Zhu, T., Lv, W. & Gao, X. (2007). A Heuristic Map-Matching Algorithm by Using Vector-Based Recognition, International Multi-Conference on Computing in the Global Information Technology, Guadeloupe, French Caribbean

Xiong, D. & Sperling, J. (2004). Semiautomated matching for network database integration, Vol. 59, Issues 1-2, ISPRS Journal of Photogrammetry and Remote Sensing, Special Issue on Advanced Techniques for Analysis of Geo-spatial Data, pp. 35 - 46, August 2004

Xiong, D. (2000). A three-stage computational approach to network matching, Transportation Research Part C: Emerging Technologies Vol. 8, pp. 71-89

Yang, J.-S., Kang, S.-P. & Chon, K.-S. (2005). The Map Matching Algorithm Of GPS Data With Relatively Long Polling Time Intervals, Journal Of The Eastern Asia Society For Transportation Studies, Vol. 6, pp. 2561-2573

Yuan, S. & Tao, C. (1999). Development of Conflation Components, Proc. of Geoinformatics'99, Ann Arbor, pp. 1-13, June 19-21, 1999

Zhang, M. & Meng, L. (2007). An iterative road-matching approach for the integration of postal data, Computers, Environment and Urban Systems, Vol. 31/5, pp. 598-616, Elsevier, 2007

Zhang, M. & Meng, L. (2008). Delimited Stroke Oriented Algorithm- Working Principle and Implementation for the Matching of Road Networks, Journal of Geographic Information Sciences, Vol. 14/1, pp. 44-53, June 2008

Zhang, M., Meng, L. & Qian, H. (2007). A structure-oriented matching approach for the integration of different road networks, CD-Proc. of the XXIII International Cartographic Conference (ICC), Incremental Updating and Versioning of Spatial Data, Moscow, Russia, August 4-10, 2007

Zhang, M., Mustafa, M., Schimandl, F. & Meng, L. (2009). A Grid-Based Spatial Index for Matching between Moving Vehicles and Road-Network in a Real-Time Environment, Proc. of the XXIV

International Cartographic Conference (ICC), Santiago, Chile, November 15-21, 2009, pages pending

Zhang, M., Shi, W. & Meng, L. (2005). A generic matching algorithm for line networks of different resolutions, Workshop of ICA Commission on Generalization and Multiple Representation Computering Faculty of A Coruña University - Campus de Elviña, Spain

Zhang, M., Liu, L., Gong, H. & Meng, L. (2008). An automatic approach to integrate routing-relevant information from different resources. 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), Irvine, CA, USA, November 5-7, 2008

Zhang, Y., Heipke, C., Butenuth, M., & Hu, X. (2006). Automatic extraction of wind erosion obstacles by integration of GIS data, DSM and stereo images. International Journal of Remote Sensing 27 (8), pp. 1677-1690.

Zhang, Q. & Couloigner, I. (2005). Spatio-Temporal Modeling in Road Network Change Detection and Updating, Proc. of the International Symposium on Spatio-temporal Modeling, Spatial Reasoning, Analysis, Data Mining and Data Fusion, Peking University, China, August 27-29, 2005

Zhou, J. (2005). A Three-step General Map Matching Method in the GIS Environment: Travel /Transportation Study Perspective., UCGIS Summer Assembly 2005, USA, http://www.ucgis.org/ summer2005/studentpapers.htm (accessed on 2009-06-07)

List of Figures

List of Tables

Abbreviations

ATKIS	Amtliches Topographisch-Kartographisches Informationssystem, viz. Official Topographic Cartographic Information System
Basis DLM	Basic Digital Landscape Model
BG	Buffer Growing
BKG	Bundesamt für Kartographie and Geodäsie, viz. German Federal Agency for Cartography
CPP(s)	Control Point Pair(s)
DSO Algorithm	Delimited-Stroke-Oriented Algorithm
DS(s)	Delimited Stroke(s)
FOW	Form of Way
FRC	Function Road Class
GIS	Geographic Information Science
GPS	Global Positioning System
ICP	Iterative Closest Point
LoD	Levels of Details
MRDB	Multiple representation Database
OSM	OpenStreetMap
POI(s)	Point(s) of Interest
PW(s)	Pedestrian Way(s)
PWs-tbc	Pedestrian Ways to be conflated
WYCSIWYCM	What You Can See Is What You Can Match

Acknowledgements

This research was conducted from October 2004 until October 2009 at the Department of Cartography, Technische Universität München, Germany.

I would like to extend my special gratitude and appreciation to my supervisor, Prof. Dr. Liqiu Meng, for giving me the opportunity to work on very interesting and challenging projects in an optimal environment. She always knew how to encourage me and how to fulfill my potential. Through her guidance, I have learned how to do the research, how to read and write scientific papers, how to give presentations and demonstrations, etc. In a word, I do not think I could accomplish my research and dissertation without her support and patient supervisions. Besides, I want to thank her for giving me a lot of freedom to conduct this research, which allows me to explore my own ideas to solve various problems.

Moreover, I am very grateful to my co-supervisors, Prof. Dr. Udo Lipeck, for reviewing my work and giving me valuable comments.

My gratitude also goes to all of my colleagues, viz. Luise Fleißner, Fritz Meier, Robert Kauper, Christian Murphy, Theo Geiß, Masria Mustafa, Stefan Peters, Olivier Swienty, Tumasch Reichenbach, Mathias Jahnke, Holger Kumke, Jukka Krisp, Hongchao Fan, Lu Liu, Yueqin Zhu, Jiantong Zhang and Stephan Angsüsser, for their warm-hearted help in the past five years.

Furthermore, I want to show my gratitude to my close friends Zheng Wang, Shuyan Wang, Bin Yang, etc. who gave my life richness in so many ways.

The research described in this dissertation is funded in part by German Federal Agency for Cartography and Geodesy (BKG), and in part by Corp. United Maps. In the end, I would like to thank Carsten Recknagel, Andreas Wiedmann, Dr. Joachim Bobrich for their valuable advice and significant support on providing the amount of test data.

Curriculum Vitae

Personal Data

Name	Meng Zhang
Date of Birth	April 14, 1979
Nationality	Chinese

Education

09/2001 ~ 07/2004	M. Sc. School of Civil Engineering, Tsinghua University, China
09/1997 ~ 07/2001	B. Sc. School of Civil Engineering, Tsinghua University, China
09/1994 ~ 07/1997	High School Senior section of Xi'an No. 83 middle school, Shann'xi, China
09/1991 ~ 07/1994	Middle School Junior section of Xi'an No. 3 middle school, Shann'xi, China
09/1985 ~ 07/1991	Primary School NO.2 primary school of Construction Engineering Corporation of Shann'xi Province, China

Working Experience

10/2004 ~ 12/2009	Research and Teaching Assistant Technische Universität München, Germany
06/2002 ~ 07/2002	Assistant of the President Waterpower Conservancy of Nanping City, Fujian Province, China

Honour

04/2009	2008 Chinese Government Award for Outstanding Self-Financed Students Abroad

I am a slow walker, but I never walk backwards.
(Abraham Lincoln)