TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Kartographie

Integration of time-dependent features within 3D city model

Herrn Dipl.-Ing. Hongchao Fan

Vollständiger Abdruck der von der Fakultät für Bauingenieur- und Vermessungswesen der Technischen Universität München zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften (Dr.-Ing.) genehmigten Dissertation.

Vorsitzender: Univ. -Prof. Dr.-Ing. Uwe Stilla
Prüfer der Dissertation:
1. Univ. -Prof. Dr.-Ing. Liqiu Meng
2. Univ. -Prof. Dr.-Ing. Rüdiger Westermann

Die Dissertation wurde am 19.07.2010 bei der Technischen Universität München eingereicht und durch die Fakultät für Bauingenieur- und Vermessungswesen am 04.10.2010 angenommen.

Abstract

During the recent decade, a growing number of municipalities have decided to build up 3D city models for various applications like urban planning and facility management, environmental and training simulations, disaster management and homeland security, and personal navigation etc. In these applications, the 3D city models serve the purpose for management of spatial data and as a decision-supporting tool. However, the current existing 3D city models are reconstructed on the basis of information frozen at a certain time point. They are not able to represent the changes in cities. Therefore, their applications remain rather limited. On the other hand, lots of available spatiotemporal data models have been reported in the literature, but they can hardly manage complex 3D geometries along with their semantic changes at different levels of details. More critically, they do not support continuous spatiotemporal changes.

This thesis presents an object-oriented event-state spatiotemporal data model for storage and management of both semantic and geometric changes of 3D building objects in a city. The data model is mainly composed of two parts: an event model that describes events happened to building objects; and a hierarchical spatial data model that describes 3D geometries and semantics of building objects including their valid time span. In this way, histories of building objects are modeled. The data model can be "double indexed" by events happened to objects and by objects involved in events. Correspondingly, queries can be triggered by both events and objects. On this base, a set of spatiotemporal queries are proposed.

In the object-oriented event model, events are modeled with five attributes: what, where, when, how, and who – with "what" for the type or class of events, "where" for location of events, "when" for time point or duration of events, "how" for the changing modes of events, and "who" for the objects involved in the events. In further, event-induced changes are analyzed for 3D buildings.

The notion of event is introduced in the proposed data model to describe how an object is changed from one state to another. Therefore, all the changes of an object are stored as events in the data model. However, if some of these changes should be retrieved by queries, they might be either significant to be noticed or not significant for one's attention because they can only be observed in a certain range of spatial and temporal scales. In this sense, the events stored in the data model are actually event-induced changes. Whether they can be retrieved as events depends on if they are significant for a referred spatiotemporal environment. In order to find out the significant changes to the spatiotemporal objects at different spatiotemporal scales, spatiotemporal generalization has to be conducted. The author proposes a framework for spatiotemporal generalization, which comprises event generalization, and 3D spatial generalization. The algorithms developed within this framework have been implemented and evaluated. The experiments have verified that 3D buildings can be efficiently generalized, while their characteristics can be well preserved after the generalization.

For the implementation of the spatiotemporal data model, CityGML as an XML-based OGC (Open Geospatial Consortium) standard modeling language for the storage and exchange of 3D data of city objects is adopted and extended to deal with events. In addition, software modules are developed as a platform and interface for the interactive handling of spatiotemporal city model.

The spatiotemporal data model proposed in this work combines the advantages of event-based model and object-based spatiotemporal data model. On one hand, dynamic processes are modeled as events with their types/classes, locations, time points/durations, modes of the processes, and the involved city objects. On the other hand, the life of an object is represented by a time-ordered sequence of its states and the dynamic processes indicating how the object changes from one state to another. The approach of storing events and city objects separately reveals a number of benefits: (i) the multiple storage due to n-to-m relations among events and objects are avoided, (ii) the spatiotemporal data model is double-indexed. Events and 3D objects can be queried independently and efficiently. In addition, the proposed spatiotemporal data model takes the hierarchy and inherent relations between events and objects into account, so that both events and 3D objects can be represented at different levels of detail.

Zusammenfassung

Im letzten Jahrzehnt haben sich immer mehr Städte und Gemeinden dazu entschlossen, 3D Stadtmodelle für unterschiedliche Anwendungen, beispielsweise Stadtplanung, Facility Management, Umweltanalyse, Lehrsimulationen, Simulation von Hochwasser und anderen Katastrophenmanagement, Zivilschutz, und mobile Navigationssysteme, zu erfassen. Dabei dienen 3D Stadtmodelle dem Zweck zur Verwaltung räumlicher Daten und als Entscheidungshilfe.

Allerdings sind die vorhandenen 3D Stadtmodelle noch nicht in der Lage, zeitabhängige Veränderungen zu präsentieren, denn die Daten, aus denen sie rekonstruiert wurden, beziehen sich nur auf einen bestimmten Zeitpunkt. Daher bleiben ihre Anwendungen eher beschränkt. Obwohl heutzutage zahlreiche Datenmodelle zur Verwaltung von zeitabhängigen Geodaten zur Verfügung stehen, ist es nach wie vor ein ungelöstes datenbanktechnisches Problem zur Archivierung und Unterhaltung der Veränderungen komplexer 3D Geometrien sowie ihrer semantischen Attribute in unterschiedlichen Auflösungen (levels of detail). Außerdem wurde bei der Konzipierung dieser Datenmodelle die kontinuierliche Natur mancher dynamischer Prozesse nicht berücksichtigt.

In dieser Arbeit wird ein objektorientiertes und Ereignis-Zustand-basiertes raumzeitliches Datenmodell zur Speicherung und Verwaltung semantischer und geometrischer Veränderungen der digitalen 3D Gebäude eines Stadtmodells vorgestellt. Dieses Datenmodell besteht weitgehend aus zwei Teilen: (i) einem Ereignismodell, um Veränderungen an städtischen 3D Objekten zu modellieren, und (ii) einem hierarchischen räumzeitlichen Datenmodel für die Speicherung der 3D Gebäude mit ihrer gültigen Lebensdauer. Auf diese Weise werden die Geschichten der individuellen 3D Gebäude modelliert. Das Datenmodell lässt sich doppelt indexieren, nämlich, nach Ereignissen, die an den Objekten stattfinden, oder nach Objekten, die beim Ereignis beteiligt sind. Diese Indexierung erlaubt sowohl die Ereignis-basierten als auch Objekt-basierten Abfragen. Auf dieser Grundlage, werden eine Auswahl von raumzeitlichen Anfragen vorgestellt.

Ähnlichen wie der Beschreibung von Vorkommnissen werden Ereignisse im objektorientierten Ereignis-Modell mit fünf Attribute beschrieben, nämlich, mit „was“ wird die Ereignisklasse gefragt, „wo“ der Standort, „wann“ der Zeitpunkt bzw. die Zeitspanne, „wie“ die Erscheinungsform, und „wer“ die beteiligten Objekte. Darüber hinaus werden Ereignis-induzierte Veränderungen von 3D Gebäuden analysiert und klassifiziert.

Der Begriff „Ereignis“ beschreibt den Prozess, wie sich ein Objekt von einem Zustand in einen anderen verändert. Alle raumzeitlichen Veränderungen eines Objektes werden im Datenmodell als Ereignisse gespeichert. Jedoch können Probleme auftreten, wenn die abgefragten Ereignisse für die Aufmerksamkeit der Beobachtung nicht ausreichend signifikant sind. Der Hauptgrund dafür ist, dass jedes Ereignis einschließlich der beteiligten Objekte raumzeitlich maßstababhängig ist.

In diesem Sinn sind alle im Datenmodell gespeicherten Ereignisse Ereignis-induzierte Veränderungen. Sie werden aber nur dann als Ereignisse abgefragt, wenn sie sich signifikant auf eine raumzeitliche Umgebung beziehen. Um signifikante Veränderungen für ein raumzeitliches

Objekt in unterschiedlichen raumzeitlichen Maßstäben zu erkennen, bedarf es einer raumzeitlichen Generalisierung. Daher wird in dieser Arbeit ein Konzept für Generalisierung entwickelt, das sowohl Ereignisgeneralisierung als auch die 3D Gebäudegeneralisierung umfasst. Die im Rahmen der vorliegenden Arbeit entwickelten Algorithmen wurden implementiert und durch Testdaten evaluiert. Die Untersuchungen haben gezeigt, dass 3D Gebäude mit hoher Effizienz und ohne Verlust ihrer charakteristischen Eigenschaften generalisiert werden können.

Für die technische Implementierung des neuen Datenmodells wurde CityGML herangezogen, da CityGML als OGC-Standard die Beschreibung der semantischen und geometrischen Informationen in seinem Datenformat unterstützt. Für das Ereignis-Modell wurde CityGML erweitert und mit neuen Objekt/Attributen in den Instanzdokumenten versehen. Darüber hinaus wurden Softwaremodule zur Abfrage und Visualisierung raumzeitlicher Veränderungen der 3D Gebäude entwickelt.

Das neue Datenmodell kombiniert die Vorteile der Ereignis-Orientierung mit der Objektorientierung. Einerseits werden dynamische Prozesse als Ereignisse mit entsprechenden Eigenschaften modelliert. Andererseits lässt sich das dynamische Leben eines Objekts durch zeitlich geordnete Zustände und Ereignisse repräsentieren. Die getrennte Speicherung von Ereignissen und 3D Objekten hat eine Reihe von Vorteilen: (i) die Mehrfachspeicherung aufgrund der n:m-Beziehungen zwischen Ereignissen und Objekten wird vermieden, (ii) das Datenmodell kann doppelt indiziert werden. Ereignisse und 3D Objekte lassen sich unabhängig voneinander effizient abfragen. Darüber hinaus werden die charakteristischen Hierarchien und die inhärenten Relationen zwischen Ereignissen und 3D Objekten bei der Modellierung berücksichtigt. Infolgedessen können 3D Objekte und ihre Veränderungen in unterschiedlichen Detaillierungsgraden repräsentiert werden.

Contents

1 Introduction 9

1.1 Motivation 9

1.2 Goal of the work 10

1.3 The structure of the work 12

2 Theoretical background and the state of the art of spatiotemporal modeling 15

2.1 Representation of city objects 15

2.2 Time as a dimension in city evolution 19

2.3 The state of the art of spatiotemporal modeling 23

3 Event analysis for city objects 29

3.1 The concept of event 29

3.2 Spatial aspects in events 34

3.3 Event-induced changes for buildings 36

4 An object-oriented event-state spatiotemporal data model for a 4D city environment 39

4.1 Object-oriented event modeling 39

4.1.1 Description of events with five Ws 39

4.1.2 Structure and variables for event model 40

4.2 The object-based hierarchical 3D city model 42

4.2.1 The hierarchical characteristics of city structures 42

4.2.2 The spatial object 43

4.2.3 The implementation of the data model 45

4.3 The integration of temporal information within 3D city model 45

4.3.1 Time-stamped city objects 46

4.3.2 Event as transition from one state to another 47

4.4 Spatiotemporal operations in the data model 49

5 Spatiotemporal queries and visualization in the 4D City environment 51

5.1 Spatiotemporal queries 52

5.1.1 Object-based queries 52

5.1.2 Event-based queries 53
5.2 Spatiotemporal visualization 54
5.2.1 Static visualization 54
5.2.2 Dynamic visualization 57

6 Spatiotemporal generalization 59
6.1 Introduction 59
6.2 The issue of spatiotemporal scale 60
6.3 Event generalization 61

7 Spatial generalization 63
7.1 Generalization for 3D buildings – the state of the art 63
7.2 The concept of LoD for 3D buildings 64
7.3 Deriving building models at different LoDs using generalization 67
7.3.1 Extraction of SLoD3 from LoD3 67
7.3.2 Transition from SLoD3 to SLoD2 70
7.3.3 Transition from SLoD2 to LoD2 71
7.3.4 Transition from LoD2 to LoD1 79
7.4 Summary 79

8 Implementation and experimental results 81
8.1 Extension of CityGML for spatiotemporal city model 81
8.2 Preparation of spatial data at different LoDs using 3D generalization 83
8.2.1 The implementation of deriving LoD2 models from SLoD2 models 83
8.2.2 The experimental results of deriving LoD2 models from SLoD2 models 84
8.2.3 Quality assessment for ground plan simplification 88
8.3 Visualization environment 90

9 Conclusion and outlook 91
9.1 Conclusion 91
9.2 On-going research work and the works in the future 93

List of figures 95
Bibliography 97
Acknowledgments 105
Curriculum Vitae 106

Chapter 1

Introduction

1.1 Motivation

Recent technological advances such as aerial photogrammetry, laser scanning, terrestrial measurement, 3D computer graphics etc. have greatly eased data acquisition, construction and visualization of detailed 3D city models. As a result, a growing number of municipalities have decided to build up 3D city models during the recent decade. In general, 3D city models are required for various applications such as:

- Urban planning
  Many of the existing models serve the main purpose of supporting urban planning processes by means of visualization of virtual scenes. Administrative departments find themselves often confronted with complex decision processes of large-scale reconstruction projects of old town areas and investment projects (e.g. a new shopping mall, commercial area, industrial site). Using a 3D city model the current situation of the city or the involved city part can be visualized as an overview. Besides 3D building models and a terrain model, various 3D city facilities like transportation objects, vegetation objects etc. can also be described in a 3D city model whose interactive visualization helps to present and evaluate the visual impact of planned constructions.

- Civil engineering calculations
  Civil engineering calculations require 3D building models with the highest possible details, preferably the architectural models with detailed wall and roof structures, balconies, bays, outlook as well as the internal organization. These detailed models allow calculations of volume and area of part of the building for cost estimation on one hand and render a vivid and intuitive impression of the building on the other hand.

- Safety and security
  Detailed 3D city models containing buildings and other city objects can facilitate accurate positioning and inspection of incidents, which may relieve the lifesaving rescue work in emergency cases and result in a reasonable situation awareness of safety among the inhabitants.

- Disaster management
  3D city models with detailed digital terrain model (DTM) in combination with computational fluid dynamics can serve as a basis for numerical flood simulations to predict whether parts of a city will be affected by the flood or not and how severe the impact on the buildings will be. These simulations help to plan efficient flood protection measures. Moreover, flooding can be simulated more precisely if there is sufficient thematic information embedded in the 3D city model.

- Noise modeling and investigations
  It is well known that vertical topographic structures, both artificial and natural, may affect noise propagation in a city area. A 3D city model is useful for the simulation of noise propagation over a wide area and investigation of the population under the harmful influence of noise. Contaminants in the air, line of sight for planning of radio network etc. can also be calculated.

- Route planning and navigation system
  Cognitive psychology studies have shown that persons who give directions do not only use basic elements such as street names and curves but also landmarks as orientation points. On this note, suitably textured 3D city models and suitably enhanced 3D landmarks are necessary to ensure the visual recognition of the urban environments for users of route planning and navigation systems. Moreover, interior structures of public buildings, park houses, and subway stations etc. can also be modeled for indoor navigation.

- Business development and tourism
  3D visualization of terrain and city models including buildings, streets and other city furniture combined with tourism-relevant data (e.g. hotel and restaurant information, shops etc.) can support the potential visitors to make optimal decisions and increases the occupancy within the tourist region.

In comparison to 2D city models or 2D urban information systems, adding the third dimension brings rich information in geometric and semantic sense. As a result, it enables many applications as the above listed cases. However, these applications remain rather limited when simulation about dynamic processes and queries about time-related information are needed. For example, 3D city models can support the decision making in urban planning. But if the historic city model is available, consequences of cultural and architectural development may be considered in the decision. On the other hand, a city model with the integrated historic information can make people more knowledgeable and attract more attention from business and tourism branches. Another example is that a disaster management system may conduct more precise calculation and prediction if the dynamic changes of the city objects are taken into account.

Despite of the abovementioned limitations of 3D city model, there is a growing interest in the concepts and methodologies about how the current existing city models will be updated or integrated in new versions since the urban processes in the 21st Century are accelerated at different levels thanks to the rapid development of science and technologies.

The thesis attempts to overcome the existing limitations by creating a 4D virtual city environment in which the time-dependent information is integrated within a 3D city model.

1.2 Goal of the work

The thesis addresses the following research questions about

- how to define a suitable data structure that supports not only spatial queries, but also queries of events, processes as well as temporal topological relationships,

- how to contemplate a data model that is able to handle temporal information associated with 3D city objects at different granularity levels,
- how to develop mechanisms of compressing spatiotemporal information for fast rendering,
- what can be operated and queried spatiotemporally in the 4D city environment, and
- how to visualize the results of spatiotemporal queries

These questions involve the following tasks:

- Analysis of event-induced changes
  Like other objects in an urban environment, buildings are changing more or less over time. The notion of event is introduced in order to represent the changes. The variations of events in terms of structures, tenses, characteristics, geometries and types are analyzed. The temporal relationships between different events are specified on the basis of the existing models.

- Representation of changes
  An object-oriented event-state spatiotemporal data model is proposed which allows to manage the changes of 3D buildings at different levels of detail (LoD), not only geometrically but also temporally, and visualize the 3D buildings statically or dynamically. Moreover, the data model takes the inherent dependence of geo-spatial change at spatial and temporal scales into account.

- Generalization of 3D buildings and their changes
  A series of generalization algorithms are developed to prepare 3D building models in a hierarchical order composed of different LoDs. Starting from the most detailed model, 3D buildings can be derived at four different LoDs: (i) the exterior shell model, (ii) model with generalized facades including typified windows, (iii) model with simple roof structures, and (iv) block model with flat roof structure. Different than other 3D generalization approaches, the approach presented in this thesis considers semantics associated with geometric features. In addition, the temporal domain is investigated so as to conduct the generalization on spatiotemporal features.

- Design of spatiotemporal queries
  A set of spatiotemporal queries are specified for a 4D city environment on the basis of four types of spatiotemporal queries proposed by Yuan (1999): (i) queries about attributes of entities, (ii) queries about location, spatial properties, and spatial relationships, (iii) queries about time, temporal properties, and temporal relationships, and (iv) queries about spatiotemporal behaviors and relationships. In our spatiotemporal data model, queries are classified into two groups at first: object-based queries and event-based queries, whereby in object-based queries city objects including their geometries, attributes as well as spatial and temporal relationships (comparisons) between objects and others can be inquired, while in event-based queries events and their attributes like time, location, objects involved in them can be asked.

This thesis makes contributions to the general context of spatiotemporal analysis and 3D generalization in the following aspects:

- The inherent dependence of spatial and temporal scales is studied, which has made it possible to model geometrically but temporally 3D buildings at different LoDs.

- Representing a 3D building by its exterior shell can substantially reduce the storage space. The algorithm for derivation of exterior shell from a detailed building model is developed and implemented.

- A three-step approach of generalizing 3D buildings is proposed - simplification of walls, simplification of roof, reconstruction of the 3D building by intersecting the wall and roof polygons. The algorithm can well preserve the characteristics of roof and wall structures.

1.3 The structure of the work

The remaining parts of the thesis are structured as follows. Chapter 2 introduces the theoretical background and the state of the art, beginning with an overview about the representation of city objects hitherto and the currently predominant models for representing 3D buildings. A description about how the city objects change with the development of cities follows. Finally, the state of the art of spatiotemporal data model is outlined.

Chapter 3 is dedicated to the analysis of events with respect to changes happened to buildings in a city. Firstly, events are explained and investigated in terms of their structures, tenses, characteristics, geometries in general. Then the temporal relationships between events are introduced in line with 13 basic temporal relationships reported in (Allen, 1983). Secondly, the spatial parts of events are analyzed following a discussion on the similarities between events and objects. Thirdly, events specified for buildings are abstracted to ten types in terms of geometric change and more than seven types in terms of attributive change.

Chapter 4 presents an object-oriented event-state spatiotemporal data model for storage and management of both semantic and geometric changes of 3D objects in a city. At first an object-oriented structure for modeling events in 3D objects is introduced. Then a hierarchical data model is presented for modeling both geometries and semantics in 3D objects. The connection of the event model with the spatial object model is explained leads to an event-state model for storing and managing dynamical changes of city objects. Finally, five spatiotemporal algebras within the frame of the proposed data model are described.

Chapter 5 explains and lists a set of spatiotemporal queries in line with specifications of our data model: object-based queries and event-based queries. Then the concepts for visualizing retrievals of queries are introduced. After that, spatiotemporal generalization is represented. The spatiotemporal generalization is composed of two parts: event-generalization and spatial generalization. The former is addressed in Chapter 6. The latter is presented in Chapter 7, starting with a literature study of 3D generalization. Then the concept of LoD is introduced. Several LoDs are defined by extending the LoD framework in CityGML. Accordingly, algorithms are proposed to derive coarse building models progressively from the most detailed 3D building models at LoD4.

Chapter 8 presents the implementation works and experimental results, which consists of three parts: (i) extension of CityGML for the proposed spatiotemporal city model, (ii) experimental results using the generalization algorithms, and (iii) a prototype system for the interactive visualization of spatiotemporal datasets.

Finally, Chapter 9 concludes the thesis by answering the five questions raised in the introduction. Further suggestions are given for the works in the future.

Chapter 2

Theoretical background and the state of the art of spatiotemporal modeling

The geographic features including city objects change over the time. Aiming to understand, collect, store, manage, simulate, analyze and visualize these changes, numerous spatiotemporal data models have been proposed, since Berry (1964) described geographic data using a matrix comprised of times, places and attributes. In this chapter, beginning with a historic overview of how a city was represented the modeling of 3D city objects will be listed, including the most popularly used models such as parametric model, boundary representation model, constructive solid geometry, and the newest OGC standard of CityGML. Then time features will be emphasized with respect to city evolution. Following that, the state of the art will be described by a literature review in the field of spatiotemporal data modeling.

2.1 Representation of city objects

Historical context

The first time we human beings tried to document geography can be traced back to about 15,000 years ago when our forefathers scratched the geography surrounding on a piece of mammoth bone (Harley & Woodward, 1987). From then on, for a very long time, maps have been used as fairly convenient, efficient and effective representation of our environment or portions of it (Peuquet, 2001). Among others, representing city environment and city objects has been a never-ending endeavor during the history of the cartography. In the ancient Babylon, Greece, Rom as well as in the old China, people drafted city with city walls, city moat, main streets and landmarks etc. for e.g. military defense and irrigation purposes.

On the other hand, maps with more detailed outlines of building footprints served in public administration, primarily for ownership and taxation purposes. These maps (i.e. Tongguan map, Liu 1998) can be treated as embryo of cadastre. Later, for several centuries, cadastre diverted from forms of representation. Furthermore, more and more information about buildings on the footprint were contained in the cadastre, for instance, the properties of the buildings.

Besides maps and cadastres, city or part or it was also represented on painting, handscroll or post cards. In these cases, city objects like buildings, bridges etc. are represented in 3D perspectives. Usually, people liked to depict important buildings in order to record a certain architecture event. In most of cases, people draw landmarks and their surroundings in order to make advertisements for a city. Apart from the abovementioned representation of single or a few buildings, paintings with a large number of buildings such as a castle including a group of buildings, even a city etc. could be found in museums and archives. For example, in the middle Ages castles and their surroundings were usually represented in 3D as voyages or for a good overview of military purpose. One of the famous works is the historical-geographic work of Constantinople from the Liber insularum Archipelagi by Cristoforo Buondelmonti in 1420.

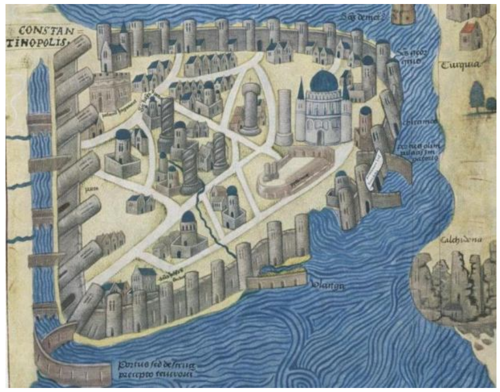


Fig. 2.1 The historical-geographic work of Constantinople (Bagrow & Skelton, 1963)

With the development and improvement of the painting skills, the representation of city objects evolved from 2D sketches to 3D illustrations. But most works were depicted as central perspective representation that rendered a panoramic view of a city or part of city. In these paintings, topological relationships reflected the way they were visually perceived by the artists. The precise geometries of objects especially geometrical relationships among objects, however, were hardly considered due to the missing orthographic projection techniques.

An exception was made by the panoramic painting "Along the River During the Qingming Festival" (Figure 2.2), which is generally attributed to the artist Zhang Zeduan (1085-1145) in the Song Dynasty. It is also regarded as one of the most remarkable representations for 3D city in the old time. Supposedly, the artist painted it for capturing the grand occasion of the celebration at the Qingming festival in Capital city of Kaifeng during the Song Dynasty (Tsao, 2003). The whole scroll of the original version is 5.28 meters long and 0.248 meters wide. The objects on the scroll include more than hundred buildings, numerous trees, a lot of vehicles and river meandering with many bridges, as well as innumerable humans and animals. In terms of building, there are hotels, temples, private residences, and official buildings varying in grandeur and style, from huts to mansions with grand front- and backyards, in addition to the shops and diners. And all the objects were drawn with very high geometric accuracy by virtue of the special technique of the parallel projection which allows measurability of distance and size – which is also an important property of modern 2D maps.

Fig.2.2. Panoramic painting "Along the River During the Qingming Festival" (By Zhang Zeduan, 11th century (Song Dynasty)) can be viewed as a 3D representation for city Kaifeng during the Song Dynasty.

This painting indicated an initial stage of map projection techniques which were rapidly developed especially in European countries since the Medieval Times. Using these techniques, it is then possible to represent objects in large area with high geometrical and topological accuracies. However, another problem arose. The city objects could not be represented in detail due to the limited display space on i.e. silk, painting cloth, paper, wood etc.

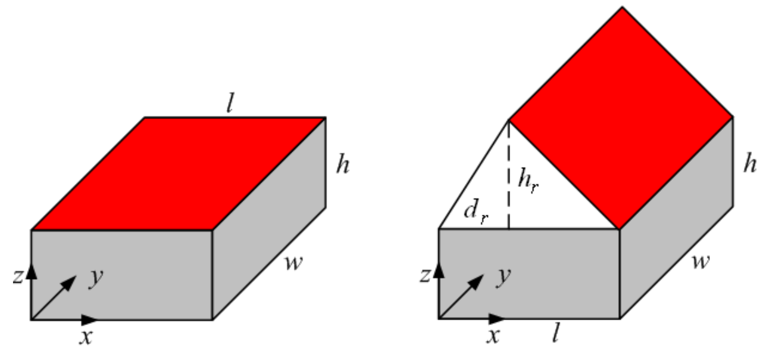
The revolution began when Waldo Tobler created the first "computer map" using raster format (Tobler, 1959) in the late 1950s. Around 40 years later, after many breakthroughs in the computer technique, interest in modeling, managing and visualizing a city in 3D environment has risen significantly, due to: (i) the development of computing technology; (ii) the availability of large, shared databases in digital form; (iii) the technique of high performance graphic card; (iv) the advanced technologies for 3D data acquisition; (v) the efficient methodologies for generating 3D objects; and (vi) the development of geospatial solutions.

3D city modeling

In general, a 3D city model can be regarded as an abstract taxonomy of city objects like buildings, vegetation, water bodies, transportation facilities etc. and the underlying landscape form. Since buildings are accounted for the vast majority of city objects, the main issue of 3D city modeling is modeling buildings in 3D.

Although buildings reveal a hug variety in structure, most of them show some regularity, which makes the reconstruction of 3D buildings less complicated because they can be represented by a set of rules or by providing a database of explicit building models (Suveg & Vosselman, 2004). There are several representations which are most commonly used for buildings such as parametric model, boundary representation (B-rep) and Constructive Solid Geometry (CSG) (Brunn et al. 1998; Förstner, 1999; Haala et al. 2007; and Meng & Forberg, 2007).

- Parametric model
  Parametric model utilizes shape parameters to represent the geometry of the building primitives and pose parameters to determine the relation of the building primitives. It relies on some a priori knowledge about the structural properties of a class of buildings. For example, the geometrical properties of a flat roof building as shown in Figure 2.3a. can be encoded by a rectangular volume. Seven parameters are needed to describe this kind of building: (i) width w, length l and height h as the shape parameters and (ii) the coordinates (x, y, z) for the spatial location and the orientation in the *xy*-plan as the pose parameters. Another slightly complicated example can be a gable roof building which can be composed of a rectangular volume and a triangular volume (Figure 2.3b). In comparison with the flat roof building, two extra parameters have to be added to describe the gable roof: one parameter for the height of the ridge h_r and the other parameter for the distance from the roof reference point to the ridge base point d_r. Likewise, L-shaped buildings could be described. However, the extension is limited due to the a priori fixation of the number of the parameters (Förstner, 1999).



(a) flat roof building (b) gable roof building

Fig.2.3. Parametric building models

- Boundary representation (B-rep) model
  Boundary representation model is a well-known polyhedral model consists of four basic geometric-topological primitives: solid, face, edge and node. A solid is bounded by faces, a face by edges, and an edge by nodes. Whereby a face is restricted to be planar and an edge must be a straight line. A curved surface can be approximated with small planar faces. As a result, an arbitrary complex 3D building can be described as an aggregation or combination of edges, faces and solids.

  The most popular form of 3D boundary representation for buildings is the Virtual Reality Modeling Language (VRML) which has been superseded by X3D. VRML defines a text file format where, e.g. vertices and edges for 3D geometry of building can be specified along with the surface color, UV mapped textures, shininess, transparency, and so on. Besides, 3D geometrical components can be networked and hyperlinked in the Internet. Moreover, in the 3D scenes animations, sounds, lighting and other aspects of the virtual world can interact with the user.

- Constructive Solid Geometry (CSG)
  Constructive Solid Geometry is well suitable to describe complex shapes which can be composited from a set of primitives (Hoffmann 1989). The standard CSG primitives consist of the block (i.e., cube), triangular prism, sphere, cylinder, cone and torus. A 3D building can be then constructed from a few of these primitives which can be arbitrarily combined with each other using Boolean operations such as intersection, union, and difference. Finally, the building is described as a CSG tree, where the leaves of the tree contain primitives, and the internal nodes contain Boolean operations. Usually, the depth and width of the CSG tree reflects the relative complexity or irregularity of the corresponding 3D building.

All the abovementioned models allow describing the form of 3D buildings. This may be sufficient for visualization. For a multi-functional and semantic 3D city model thematic attributes are important. This information should be integrated as attribute within the geometrical description of city objects. While it is possible to specify the semantics of each primitive in CGS-models, complex buildings represented as polyhedral does not allow a simple attribution (Förstner, 1999). Due to the lack of a unified format for adding semantic attributes, the functionalities like data exchange or web-based thematic queries are limited.

CityGML attempts to solve this problem by defining a standard format for coupling semantic information with 3D geometries of city objects. CityGML stands for City Geography Markup Language. It is an information model and GML application schema for the representation and exchange of 3D city model. Since 20th, August, 2008 CityGML version 1.0.0 has been adopted as a standard by the Open Geospatial Consortium, Inc. (OGC) for the modeling of 3D urban objects, especially 3D buildings.

CityGML represents a 3D city model in four different aspects, i.e. semantics, geometry, topology and appearance (Kolbe et al. 2005).

- In terms of semantics, CityGML provides a rich, general purpose information model in addition to geometry. It defines classes and relations for the most relevant topographic objects in cities comprising buildings, elevation, water bodies, transportation, vegetation, city furniture, and more. City objects (entities) are represented by features, such as building groups, buildings, walls, windows, doors or rooms etc. The description also includes attributes, relations and aggregation hierarchies between features.

  For specific domain areas CityGML also provides Application Domain Extensions (ADE) as an extension mechanism to allow the model to be enriched with additional properties and feature types.

- In terms of geometry and topology, city objects are represented according to the well-known Boundary Representation (B-Rep). The geometry model consists of primitives, which may be combined to form complexes, composite geometries or aggregates. When bounding a solid the interior and exterior side of a surface can be distinguished by using an explicit orientation which results from the order of the defining points.

  Moreover, CityGML defines special surfaces as specification of Triangulated Irregular Networks (TIN) to represent the terrain. The triangulation may be reconstructed using standard triangulation methods (Delaunay triangulation).

- In terms of appearance, an appearance model is defined in CityGML. In order to fulfill different requirements in different application, appearances are not limited to visual data but represent arbitrary categories called themes such as infrared radiation, noise pollution etc.

Furthermore, CityGML defines five consecutive LoDs (LoD0 to LoD4 with increasing accuracy and structural complexity), where objects become more detailed with increasing LoD regarding both their spatial and thematic differentiation. An object can be represented simultaneously in different LoDs by providing distinct geometries for the corresponding LoDs (Kolbe & Gröger, 2003; Gröger et al., 2008; Kolbe et al., 2009).

2.2 Time as a dimension in city evolution

The entire world changes over the time, so do city objects. Since the birth of the first city after the Neolithic Revolution in Middle East (Bairoch, 1988), changes in cities have never ceased.

Ancient times

In the ancient times, a city or township functioned mainly as a place for trading surpluses from agriculture, hunting or fishing. Therefore, new constructions of buildings for markets, taxation or private residences as well as roads, ports or other transportation facilities were essential, Later on, trade played more and more important role in the everyday life and handicraft arose. As a result, new buildings for houses, sanitation, business and public administration were required due to the increasing population migration and the growth of the urban residents.

After a period of time, some buildings might be damaged and repaired again. Some might be destroyed for different reasons. As indicated in history, cities became sprawled with new markets, new buildings for different utilities. New roads were paved, while old ones were extended or changed according to city plans. As the time went by, cities might be developed and expended thank to the increasing prosperity, population etc. and the development of cultures and techniques. Or they (parts of them) might be damaged, even destroyed by wars or nature disasters.

Middle Ages

During the European Middle Ages, a city was as much a political entity[1] as a collection of houses. Therefore, structures for military defense such as city walls, city moat, watch towers etc. became standard features in a city, as well as structures for public administration such as city hall, congress etc.

In the Early Middle Ages, most of cities, regarding the administration, were ruled through the network of bishops, because the Catholic Church was the major unifying cultural influence at that time. Hence, a wave of building of cathedrals and smaller parish churches occurred across Western Europe. In terms of the functionality, the cathedral or parish church was used by the community in other ways, in addition to a being place of worship, i.e. a meeting place for guilds, a hall for banquets, a place for mystery plays or a place for fairs. At the same time, the church was used as a place to thresh and store grain. Besides, many monasteries were built in this duration.

Although the Early Middle Ages was regarded as Dark Age in the history, the achievements in art and architecture have their dazzling sides. As their representation in city construction, few buildings were attempted with a form of stone architecture loosely based upon Roman forms and hence later named Romanesque. Where available, Roman brick and stone buildings were recycled for their materials.

The High and Late Middle Ages were characterized by the urbanization, military expansion, and intellectual revival in Europe. This time period saw an explosion in population. Correspondingly, urban area and size grew by adding more buildings and city facilities. On the other hand, cities showed their complete functionalities with a mass of infrastructures i.e. canals, bridges, facilities for systematic delivering, schools, various academies, theaters, and so on. Because of the great religious movements, the controller or rather holders of cities were taken by kingdoms. Then castles, even palaces were built in cities. Meanwhile, a city might be occupied by others. Castles

[1] According to Oxford English Dictionary (OED), an entity is a thing with distinct and independent existence, and an object is a material thing that can be seen and touched. From this point of view, entities cover more things than objects, for instance, a building can be both an entity and an object, but the function or ownership of the building can only be an entity. But in the domain of object-oriented programming, an object is regarded as a compilation of attributes (object elements) and behaviors (methods) encapsulating an entity. In this way, an entity is equivalent to an object. For convenience, in this thesis the concept of object and entity will follow the principle in the domain of object-oriented programming. Exceptions are the cases when the term “Entity-Relationship” is mentioned in which the notion “entity” has the same meaning as in Entity-Relationship Modeling.

might be damaged and either repaired and extended or they be destroyed and new built. At the same time, monumental buildings would normally be built as symbols of conquest.

From the industrial age to the time being

The growth of modern industry from the late 18th century led to massive urbanization and the rise of new large cities, first in Europe and then in other regions, as new opportunities brought huge number of migrants from rural communities into urban areas. On one hand, railroads, railway stations, harbors and new roads were built in cities to transport passengers, materials and productions for industries. On the other hand, factories, warehouses, bulk stores and other facilities for manufacturing especially for heavy industry were established. The living conditions during the Industrial Revolution varied from the splendor of the homes of the riches to the squalor of the lives of the poor. While the businesses and the wealthy elite built villas surrounded by massive gardens, there were only small houses in cramped streets for the normal workers. And the middle class built their own houses or share with others in a row house. The situation changed from the middle of 19th century, when public health acts were introduced covering things such as sewage, hygiene and making some boundaries upon the construction of homes.

Furthermore, authorities began to making planning, in order to improve the built and social environment of communities. Therefore, cities were characterized with regular patterns from the geographical point of view. For example, most of cities deployed grid system during their development. The grid was enforced even on uneven topography.

The 20th century brought the modernization of cities. Old and smaller houses are replaced by multi-storey buildings and great mansions. Even skyscrapers are built for both business and residence. Most of street frontage buildings change their utilities for commercial propose, and are decorated correspondingly. Besides, as essential functionalities of cities, facilities for medical service, educations etc. are massively constructed. By the way, all the buildings show a rich diversity of architecture styles.

In terms of transportation, streets were widening and newly paved by cement, equipped with bus stations, streetlamps, traffic lights etc. Due to the increasing amount of vehicles, new streets and park houses are built. At the same time, trams and subways are introduced in cities, as well as the matched installations such as railways and stations.

The latest technologies in the past 30 years and the consideration of environmental problems influence also the development of cities. Buildings are constructed with new materials. Their structures tend simple and environment concerned. Glass facades are preferred. Solar cells are widely equipped on roofs. The newest technologies have brought about not only new materials, ideas and designs, but also changed the functions of some city facilities. A remarkable example can be the change of TV masts and towers. They were built originally for telecommunications and broadcasting, including televisions. Nowadays, most of them are used only for radio broadcasting because of the rise of new communication techniques such as satellites and cable TVs.

At the time being, urbanization process is going on further and has speeded up. Real-estate economy becomes a more and more popular topic. Buildings including their flats can be freely rented, traded, and leased. Nevertheless, new districts are developed with full of buildings, because not only the number of the inhabitants increases, but also the number of migrants grows.

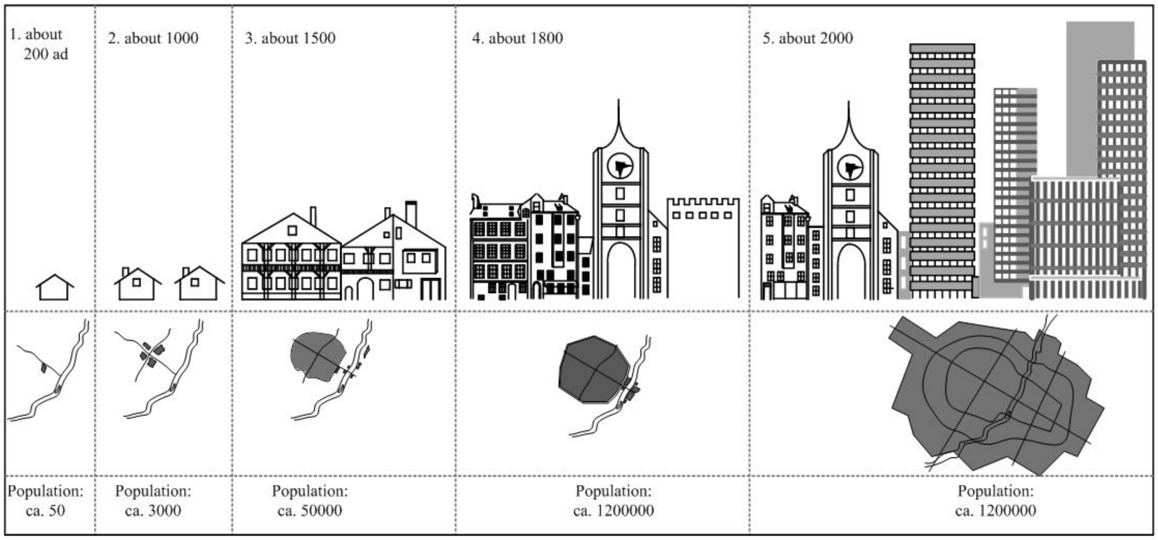


Fig.2.4. An example for the evolution of a typical European city[2] .

Figure 2.4 shows an example for the evolution of a typical European city by five snapshots. With the economic and technical development and social progress, the city grows: its population and area becomes larger and larger, the constructions especially buildings become bigger, higher, modern, as well as their appearances, materials, styles, functions, and other semantics change over the time.

The evolution of cities indicates the fact evidently, that changes exist in a city anywhere and any time. The materials for building construction change from mud and clay to polystyrene or polyurethane foam. Architecture evolved out of the dynamics between needs (shelter, security, worship, etc.) and means. Buildings changed in various terms, including shapes, functions, types, prices, ownerships and so on. Cities are developed from places for simple trading to prominent centers of inhabitation, trade, banking, innovation, and markets etc. Moreover, during the expansion and growth of cities, the frequencies of changes of city objects including buildings, city facilities and their attributes become higher and higher.

In general these changes can be categorized according to their geometry, topology and semantic. They may occur one after another or at the same time. All together, there might be eight different scenarios, as shown in Figure 2.5.

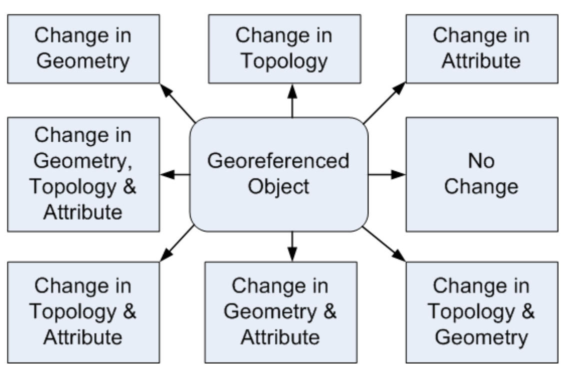


Fig.2.5. The eight possible changes of a geo-referenced object (Roshannejad & Kainz, 1995)

[2] The ideal of sketching this imaginary city is rooted onto the history of Munich, Germany, Erdmannsdorffer, 1972; München Stadtmuseum and Stadtarchiv München, 2008.

2.3 The state of the art of spatiotemporal modeling

Since the time aspect was considered in Geographic Information System for the first time by Gail Langran (Langran and Chrisman, 1988; Langran, 1988; Langran, 1992), numerous spatiotemporal data models dealing with data storage and management have been proposed in the past 20 years. As indicated in (Peuquet, 2005) and (Worboys, 2005), the development of spatiotemporal modeling has gone through three stages: the first stage with snapshot model, the second stage with object-based model and the third stage with event-based model.

Snapshot model

The snapshot model represents the spatiotemporal information as a series of time-stamped layers, whereby every layer is a collection of temporally homogeneous units of one theme, as shown in Figure 2.6. It shows the states of a geographic distribution at different points of time. Time functions as a temporal index in snapshot model. It is linear and discrete.

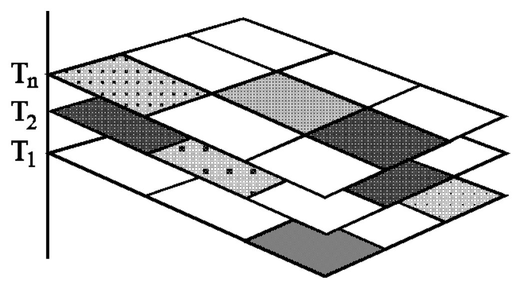


Fig.2.6. An example of the snapshot model (Armstrong, 1988)

This model is the most straightforward way to represent spatiotemporal information. Theoretically, it can be used to represent change of geographic features, however, only by exhaustively storing all versions of data. This is almost impossible in the practice. Besides, there are three major disadvantages when modeling with snapshots (Langran, 1992):

- To detect changes between snapshots, relatively expensive computation is required, because the two snapshots must be compared exhaustively.
- It is difficult to devise or enforce rules for internal logic or integrity because there is no understanding of the constraints upon temporal structure.
- Regardless the magnitude of changes, a complete snapshot is created at each time slice, which duplicates all the unchanged data and leads to the storage of redundant information.

Aiming to overcoming these disadvantages, differential snapshot model is proposed, which distinguishes the early work as static models. In the differential model, only the initial state is fully recorded. Changes are stored as increment for recording the differences from the previous state. Thus, the snapshot at any time equals to the addition of the former snapshot and the increment between them. This approach reduces storage requirements substantially, and makes the computation of changes between states less costly. However, to reinstate previous states of the current one, a series of increments must be added to the initial state, which makes this operation inefficient. Alternatively, the current state can be chosen to be stored in full, keeping decrement to track back to previous states. This can be preferable solution in case the current state is more frequently accessed than historic ones.

The snapshot model is by far most common in current temporal database models, and is linked directly to concepts such as timestamp, temporal granularity, and temporal indexing. The literature on this spatiotemporal model is extensive, and good accounts may be found in (Langran and Chrisman, 1988; Langran, 1992; Abraham & Roddick, 1999; Pelelis et al., 2004; Worboys, 2005; Li et al., 2007).

Object-based model

On the base of snapshot model, the development of spatiotemporal modeling later follows the idea of sequential updating, but drops the use of increment and retains unchanged components instead. In this scenario, when an object changes, its previous version is superseded but fully retained, accessible by temporal links and indexes, while a new object is created to describe the current state of the component. This is then the object-based model.

In contrast to snapshot model, the object-based model shifted the focus from the temporal sequences of object, their attributes and relationships, to the changes that can happen to objects, attributes and relationships.

Worboys is the first one who introduced the definition of the spatiotemporal object as a unified object with both spatial and bitemporal extents (Worboys, 1992, 1994). As shown in Figure 2.7, A spatial object is represented as simplicial complex composed of a set of simplexes (a single point, finite straight line segment or triangular area). A spatiotemporal simplex is defined intuitively as an elemental spatial object (simplex) to which a bitemporal reference is attached. The finite set of such spatiotemporal simplexes form then a spatiotemporal complex which is treated as the structure for representing a bitemporally-referenced spatial configuration. For this structure, query algebra is developed similarly to the relational algebra. A spatiotemporal complex traces changes in discrete steps, therefore is unable to represent continuous evolution, but is well suited for processes where mutations occur in sudden jumps (Abraham & Roddick, 1999).

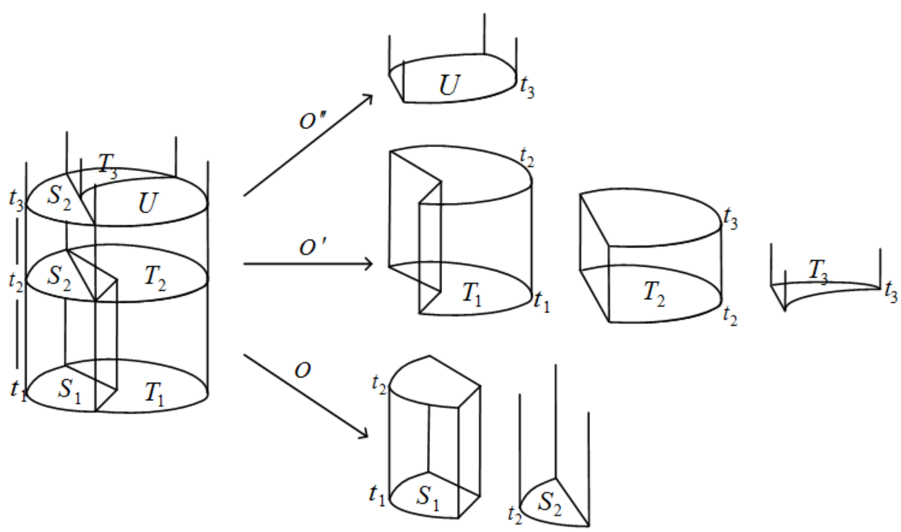


Fig.2.7. An example of an object-based model with spatiotemporal complex and simplexes (Worboys, 1992)

Although the simplicial complexes are the simplest forms of spaces and their manipulation is comparatively easier, their adoption from a set of simplexes results in voluminous data. Raza and Kainz (1999) proposed to represent a spatiotemporal object by using cell complexes instead of simplicial complexes. A cell complex has a sound mathematical basis which facilitates modeling in a more structured fashion, because every point, line and polygon can be mathematically mapped into cells of respective dimensions through homeomorphism (Kainz, 1995).

The models, so far, rarely considered the attributes or semantics of spatiotemporal objects. (Tryfona & Jensen, 1999, 2000) proposed a conceptual Spatial-Temporal Entity-Relationship (STER) model attempting to deal with complex geo-entity sets and interrelations of spatial and temporal semantics. In the STER model Entity-Relationship (ER) model (Chen, 1976) was used as the concrete context for presenting the constructs. For the spatiotemporal domain, an ontological foundation covering concepts such as objects, attributes, and relationships was defined. The STER model occurs as the outcome of applying the ontology to the modeling constructs of the ER model. It allows the description of attributes such as “ownership” and relationship among entity sets such as “reincarnation” and “splitting”. The STER model has been deployed in Land Information System (LIS) and showed its advantage in identifying temporal and spatial changes (Pelelis et al. 2004). However, it lacks the ability to capture the actual motion of the process of change and does not indicate whether a spatial object is dynamic or static.

In order to model complex changes, new approaches have been developed by using a collection of predefined combinators from change “primitives”, such as creation, destruction, appearance, disappearance, transmission, fission, and fusion (Hornsby and Egenhofer, 2000; Worboys, 2005). These terms themselves are events which might happen to object. In this context, modeling events is necessary to describe complex changes. The development following this concept marks then the third stage of spatiotemporal modeling: event-based model.

The object-based model seems natural, since it represents the world based on the evolution of objects through time, by retaining identity but changing spatial and other attributes. However, problems arise, particularly related to continuity of identity through time (Worboys, 2005), because ontology of physical object changes over time as well (Heller, 1990).

Event-based model

Both snapshot model and object-based model can be regarded as extensions of traditional vector and raster representations. They are good at performing either location-based or feature-based queries, but they not well suited for analyzing overall temporal relationships of events and patterns of events (Peuquet and Duan, 1995).

The first event-based model was proposed by (Peuquet and Duan, 1995), called the Event-based SpatioTemporal Data Model (ESTDM) (Figure 2.8).

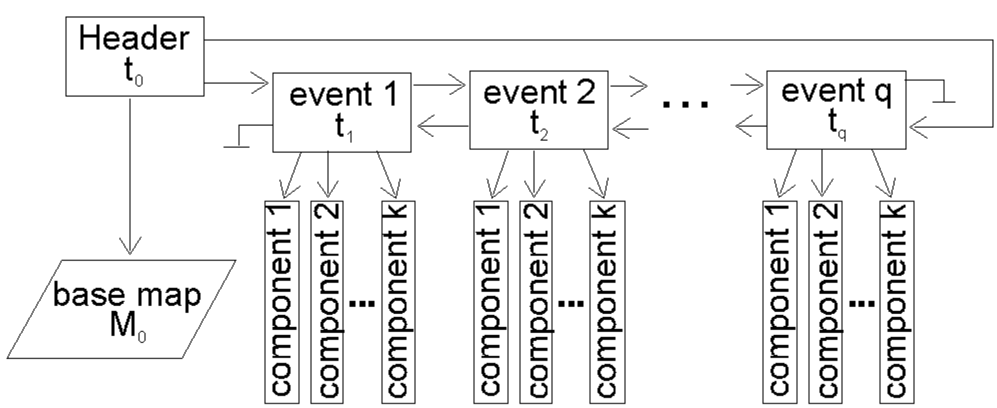


Fig.2.8. Primary elements and the pointer structure of an ESTDM (Peuquet and Duan, 1995)

ESTDM is similar to temporal map set. It is a raster-based model that uses a sequence of time-stamped components to represent an event, however, the model stores changes in relation to a previous state rather than a snapshot of instances. A base map is required as an initial snapshot of a single theme of interest. Every event is time-stamped and associated with a list of components which indicates changes to the previous state. And events are stored in an event list ordered by time. Additionally, a header is needed for storing the time-stamp of the initial time value associated with the base map, the name of the thematic domain, the name of the base map, the pointer to the base map, and pointers to the first and last elements of the event list. The ESTDM has shown its capabilities and efficiency to support both spatial and temporal queries. It reduces the computation cost when reinstating a state by using double pointers (backward pointer and forward pointer). Nevertheless, a complex process is inevitable. In addition, interpolation has to be done when the requested time does not meet any time-stamp in the event list.

Claramunt and Theriault (1995) proposed another event-based approach to model changes among a set of entities. In their model spatial entities and their temporal versions are associated through intermediary logical tables (past events, present events and future events), so that complex succession, production, reproduction, and transmission processes can be described (Claramunt and Theriault, 1995, 1996). These studies, however, have omitted certain kinds of change and so far no systematic treatments of change have been undertaken (Hornsby and Egenhofer, 2000).

In order to describe the relationships among events, Allen et al. (1995) suggests a generic model for explicitly representing casual links within a spatiotemporal GIS. A small number of elements were presented in that model via an extended Entity-Relationship formalism, including objects and their states, events, agents and conditions, as well as the relations. However, the causal connections between events were not addressed, nor the causal relationships between agents.

Following the idea of considering causal relationships between events, Chen and Jiang (1998, 2000) proposed an event-based approach for modeling the system’s behavior of the spatiotemporal process. In this model, events and the causal relationships between events and states (event-event, event-state, and state-state) were recorded. On this basis, a set of operators was introduced to describe the execution of events and their sequence. However, the model is much specified on processes in land subdivision.

The abovementioned researches so far were concentrated on modeling events and relationships between events. The time information is incorporated as an attribute for spatial object or an integral part of spatiotemporal object. On the other hand, semantics have not been much considered. A three domain model defines semantical, temporal, and spatial objects in three separate domains (Yuan, 1994, 1996, 1997 & 1999). Time is modeled as an independent concept in this model. Geospatial concepts and entities can then be represented with dynamic linkage of the objects in these three domains. This model is capable of handling spatial change. However, no operators are available to deal with the relationships among spatial objects and special mechanisms are required to calculate the change (Pelekis et al., 2004).

Furthermore, most of existing models are restricted to modeling two dimensional spatial data (Tryfona and Jensen, 1999; Wang et al., 2000; Camoss et al., 2003; Rezayan et al., 2007; Li et al., 2007; Jin et al., 2007; Wang et al., 2007), or they cover only partly the requirements by addressing either spatial or temporal modeling. Moreover, the existing spatiotemporal data models do not support the management of both discrete and continuous spatiotemporal changes (Jin et al. 2007).

In general, the existing approaches aim at the modeling of transition patterns among events in an effort to identify recurring spatiotemporal processes. They provide an explicit description of changes modeled in the geographic phenomena. However, they are restricted to model processes in which only one event with many objects or one object with many events (1 to N relationship) is involved. In other words, they are not capable of describing a scenario where a complex event can be composed of many smaller events and is coherent with a lot of objects (an N to N relationship). An initial study about the coherence between event and object can be found in (Worboys, 2005). Worboy discussed the distinctions and similarities between an event and an object and then emphasized that they might be equivalent at a certain ontological level. This leads to a novel idea like using the methods of object-orientation, designed for modeling static entities, to event modeling. This idea will be discussed in the next chapter and used for event modeling in this work. At the same time, the coherence between complex event and object will be considered in data model proposed by the dissertation.

Chapter 3

Event analysis for city objects

The changes of city objects including geometrical and semantic changes are termed as events. This chapter is devoted to the event analysis of city objects especially buildings. First of all, the notion of event is explained in general. Then the spatial aspects events are analyzed following a discussion about the similarities of events and objects. Finally, events which happen to city buildings are categorized by abstracting various changes in terms of geometry and semantics.

3.1 The concept of event

The Oxford English Dictionary (OED) defines an event as "anything that happens, or is contemplated as happening; an incident, occurrence … use chiefly restricted to occurrences of some importance". Broadly speaking, an event can be an arbitrary change of objects. If the existence of an object is viewed as a series of states or "snapshots", an event is then the change from one state to the next (Lansky, 1986).

Events are things that happen. They take place in time and space. It means that every event requires a reference to a location in time (Shipley, 2008). An event is generally said to have a life span or duration which is delimited by a start and an end point of time. Although some events appear to be instantaneous, they still can be decomposed if we examine them more closely (Allen, 1983). Sometimes an event has also a period if it occurs periodically. Normally, the end point lies after the start point on the time scalar, or in other words, the end point is greater than the end point, if they are observed and documented numerically. On certain occasions, these two points can be set to be equal to each other if an event appears to be instantaneous. In this case the duration indicates a zero-width.

Actually, whether an event has a zero-width duration or a non-zero-width duration depends on the way how we perceive it. For instance, the event "building a big church" has a duration of non-zero-width while the event "demolishing a family house" has a duration with zero-width. However, the duration of the event "building a big church" could be zero-width too, if we take hundred years or even thousand years as a temporal unit. This means whether an event has zero-width duration or not depends on its real duration in relation with the current used temporal resolution:

- If the real duration of an event is apparently longer than the currently used temporal resolution, this event has a non-zero-width duration.
- If the real duration of an event is shorter than or equal to the currently used temporal resolution, this event has a zero-width duration.

The temporal resolution means the time unit which is used or referred to the present time. Normally, it could range from second, minute, to century or longer.

Independent on the temporal resolution, some events especially the events which happened to objects' attributes have always zero-width duration, because they either appear or disappear. For example, the event "the owner of the building is changed" could have a long process because of the handling and obligation issues. But there is a key issue to this kind of events, i.e. sign a contract. If a contract is signed between the new owner and the former owner, we can say that the event "the owner of the building is changed" did happen. Otherwise, it can not be counted as an event, although there is a long process for preparation. And the day on which the contract is signed is normally regarded as the time point when the event happened.

Geometry of events

In line of geometry, event duration can be regarded as a segment line of time, like Zacks and Tversky (2001) define an event as "a segment of time at a given location that is conceived by an observer to have a beginning and an end". Then the duration of an event will be represented as a point, if it is zero-width.

As the same event might have nonzero-width duration for a used temporal resolution and zero-width duration for another used temporal resolution, it could be a line or a point. Thus, we can explain these relationships based on our real world experiences. A line observed at a near distance may vanish into a point with the increasing viewing distance. Similarly, a long-lasting, i.e. lengthy event at a fine temporal resolution would shrink to a point, if we observe it at a much coarser temporal resolution.

When an event is represented as a point on the temporal scalar, it occurs instantaneously. Then it is impossible to observe the process of the occurrence, as it either appears or disappears. On the contrast, snapshot can be captured during the process of an event, if it is represented as a line on the temporal scalar. And a decomposition of such events is possible and sometimes makes sense.

Structure of events

In general, an event is caused by an action. Allen and Ferguson (1994) emphasized that for every action there is a corresponding event consisting of an agent performing that action. On the other hand, action may be arbitrarily complex activities, and can be decomposed into less complex actions, which themselves are decomposable until a certain basic level of action is attained. In other words, an event that might be a child event of another event can be composed of many smaller events.

In fact, decomposition of events can be explained in a universal manner, as Hanson and Hirst (1989) write,

> *These boundaries may be fuzzy; the exact moment of transition between one event and the next may not be clear. As one reaches for a piece of toast and picks it up, the transition between reaching and picking up may be smooth, but the lack of a precise boundary does not imply that reaching and picking up are not discrete events. Events, like object categories, can have fuzzy boundaries and yet still be distinguished from one another. (p.136).*

As indicated above, events can be nested to other events and can be divided into smaller events which proceed according to a temporal-causal sequence (Hanson and Hirst, 1989). They can be seen to be hierarchically organized, in that embedded within the whole event are smaller segments of activities, each of which may be analyzed as an event in its own right (Nelson, 1986; Slackman et al., 1986).

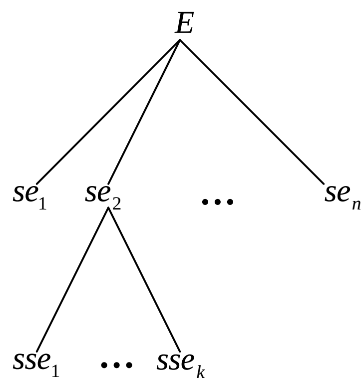


Fig.3.1. Tree structure of Events

Figure 3.1 shows a tree structure of events, where the root node E is a mother-event, the child node se represents sub-event and sse stand for sub-event of se ; $n = 1,2,3,...$ and $k = 1,2,3,...$. At the leaf level, the events are triggered by single actions that will not be split further (or a further split seems trivial or nonsense.). Taking 3D buildings as example, the event “creating a big shopping centre” could have sub-events such as follows: “building the groundwork”, “building the mainstay”, “equipping the doors and windows”, “coating the façade”, etc. Whereby, the sub-event “building the groundwork” could be represented by two events embedded within its duration, namely “digging the pit according to the ground plan” and “establishing the foundation”.

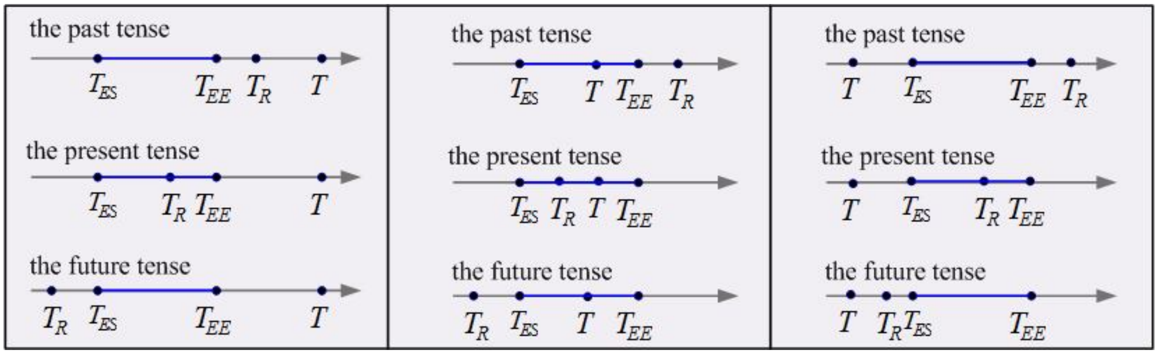
Tenses of event

In the English dictionary (Dictionary of Contemporary English, Fourth edition with Writing Assistant 2005, Longman) tense in natural languages is explained as one of the forms of a verb that show the time, continuance, or completion of an action or state that is expressed by the verb. Normally, there are more than three tenses in the most natural languages all over the world. And the tenses take the speech time as reference datum.

In the case of events of 3D buildings only three tenses would be meaningful, namely the past, present and future tense. Mani etc. illustrated the tenses in the English language with graphics (Mani et al., 2005). In a similar way we can also explain the tenses for events of 3D buildings. The difference is that our reference datum is the observation time. Thus we come to the following graphics (Figure 3.2), in which the initials ‘T_{ES}’, ‘T_{EE}’, ‘T_R’, ‘T’ stand respectively for ‘start point of event’, ‘end point of event’, ‘point of reference (observation time point)’, ‘the point of time at which the event is captured’. The direction of time is represented from left to right.

Whereby, the start point T_{ES} and end point T_{EE} form the duration T_D of an event as following:

$$T_D = [T_{ES}, T_{EE}] \qquad (3\text{-}1)$$



(a) the event is past at time T (b) the event is happening at time T (c) the event will happen in the predictable future

Fig.3.2. Tenses of events

Figure 3.2a demonstrates the majority case for the events for 3D buildings, because the most 3D buildings events are referred to as the histories of the buildings in a 4D City environment.

Nowadays, many big cities are changing very fast and there are many buildings under constructing at every time. This is to say many events are taking place at the time when they are captured in the data base. The tenses of these events could be shown in figure 3.2b.

If we have the planning schemes for a city or parts of city, we can predict some events, analyze and visualize them in a 4D City environment. Additionally, we can also predict some events in the future according to a developing model. In such cases the tenses of the events are just as what the graphics on the right side show.

The reference time, so far, is regarded as a time point. In many cases, they are, however, a time duration, since events are often observed duration a certain time period. Then the situation turns complicated. The relationships can not be described with the tenses any more. Instead, they can be illustrated by the temporal relationships between two events when treating the reference duration as duration of an event.

Temporal relationships between events

The temporal relationships between events are commonly represented as the temporal topology of event duration. As event has duration delimited by a start point and an end point, the temporal relationships between two events could be derived by comparing their lifespan and terminating points.

As already indicated above, some events have durations with identical start and end points of time while others have nonzero-width durations. Therefore, there are three cases when describing the topological relationships between two events. In the first case, both events have zero-width duration. Then, there are three relationships between them: before, after, and equal (Figure 3.3).

Relation	Symbol	Symbol for inverse	Pictoral example
X before Y	>	<	
X equal Y	=	=	

Fig.3.3. Three possible relationships between events which have zero-width duration

In the second case, one event has zero-width duration while another has nonzero-width duration. Their possible topologies are similar to those topologies between a point and a line segment (Figure 3.4).

Relation	Symbol	Symbol for inverse	Pictoral example
X before Y	>	<	
X at start of Y	sa		
X in Y	ci		
X at end of Y	ea		

Fig.3.4. Five possible relationships between a zero-width-duration event and a nonzero-width-duration event

In the third case, both events have nonzero-width duration. Then, there are 13 basic temporal relationships between them as introduced by Allen (1983) (Figure 3.5).

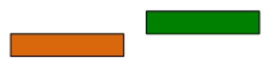
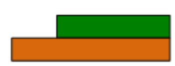
	Relation	**Symbol**	**Symbol for Inverse**	**Pictoral Example**
(1)-(2)	**X before Y**	<	>	
(3)	**X equal Y**	=	=	
(4)-(5)	**X meets Y**	**m**	$\bar{\mathbf{m}}$	
(6)-(7)	**X overlaps Y**	**o**	$\bar{\mathbf{o}}$	
(8)-(9)	**X contains Y**	**c**	$\bar{\mathbf{c}}$	
(10)-(11)	**X starts Y**	**s**	$\bar{\mathbf{s}}$	
(12)-(13)	**X finishes Y**	**f**	$\bar{\mathbf{f}}$	

Fig.3.5. The 13 possible relationships between events

The 13 basic temporal relationships are determined based on topological relationships in time and are denoted as *before* **(<)**, *after* **(>)**, *contain* **(c)**, *contained-by* **($\bar{\mathbf{c}}$)**, *overlaps* **(o)**, *overlapped-by* **($\bar{\mathbf{o}}$)**, *meets* **(m)**, *met-by* **($\bar{\mathbf{m}}$)**, *starts* **(s)**, *started-by* **($\bar{\mathbf{s}}$)**, *finishes* **(f)**, *finished-by* **($\bar{\mathbf{f}}$)**, *and equal* **(=)**. Based on transitivity of theses relationships, temporal reasoning can be performed among three events. For example, if *Event 1* meets *Event 2*, and *Event 2* contains *Event 3*, then *Event 1* must be before *Event 3* based on transitivity constraints. The temporal reasoning scheme serves as a foundation for the support of Temporal Relationship Query (Yuan and McIntosh, 2002).

Characteristics of events

In general events could be characterized by their duration, occurrence frequency as well as the consequences they have led to. Normally, their characters can be described with four variables - rate (speed), density (frequency), length of duration and amplitude. For each variable there are adjectives to describe the referred event, as follows:

Table.3.1. Variables to describe an event

Variable	Adjective
Rate (speed)	quick – slow, sudden - gradual
Density (frequency)	always, majorative, sporadic, unique
Length of duration	short – long
Amplitude	large – small

Where the variables in Table 3.1 could be understood as:

- **Rate:** the peed at which an event happens over time. For example, the rate of event that a building was destroyed in an earthquake is sudden while the rate of event that an air terminal was built is gradual.
- **Density (frequency):** the number of times that an event occurs within a particular period of time. For example, the density of the event “a wall of building becomes rougher because of erosion by wind and rain” is always; “the rent rises” is majorative, “the rent sinks” may be sporadic; and “a building is destroyed by explosion” occurs only one time in its life, therefore, its density is unique.
- **Length of duration:** the length of time in which an event happens. For example, the length of an event “putting up a marquee in Oktoberfest” is very short, while “building a skyscraper” lasts very long.
- **Amplitude:** the quantity of the consequences an event has led to, where the consequences could be the change of geometries, radiometry or attributes (e.g. the rent or the price of a building). For example, the amplitude of the event “building a car park” is larger than that of “building a garage” when the change in the geometry is regarded.

3.2 Spatial aspects in events

The major distinction between events and objects is that events require references both in location and time, while objects exist outside such a temporal reference. An object is not an event, because it does not require reference to time, it only endures over time. But the existence of an object is an event, because it requires reference to time, it extends over time. Generally speaking, events have temporal parts, while objects do not. In addition, tangible objects appear to be individually located in space – they appear to occupy their spatial location, whereas events seem to tolerate co-location (Casati and Varzi, 2008).

Although there is significant distinction between events and objects, they cannot simply be separated, as events always appear as changes of objects in attributes, appearance, locations, topologies or geometries. On one hand, an event may belong to a subsumption hierarchy and relationships to other events, so does an object. On the other hand, according to our experience, an event might involve many objects. At the same time, an object can be referred to many events. Thus, they are coherent in both hierarchies. In other words, they must be coherent (i.e. it must be ensured that they match and fit together), if both hierarchies exist for a specific object.

Besides, events and objects share some similarities at the ontological level. According to Worboys (2005) these similarities are listed in Tab.3.2.

Table 3.2. Object-event similarities according to (Worboys, 2005)

Objects	Events
Object instances	Event occurrences
Object attributes	Event attributes
Object taxonomy	Event taxonomy
Object partonomy	Event partonomy
Object relationships	Event relationships

Regarding to these similarities, the temporal parts of events can be analyzed similarly to their spatial parts, i.e. objects.

Structure of space

The spatial reality is perceived as a collection of objects which are constructed as three-dimensional entities. In GIS there are two basic models for storage of geographic objects: raster and vector model. In raster model, objects are structured as an array of cells, pixels or voxels for 2D or 3D representation respectively. Space is partitioned into grids where each cell is addressed by its position in the raster array. In vector model objects are composed of basic primitives like points, lines and polygons. Vector representations make more efficient use of computer storage as they utilize only useful data and not the entire plane.

Granularities

It is an inherent property of spatial objects that they may be viewed at different granularities and that the granularity affects LoDs required in the representation. Objects might be observed so closely that only a part of them can be seen. Then they show a high content density in geometry and texture. Or they might be viewed from a long distance. Then their detailed structures appear coarse and can finally converge to a point when the viewing distance is long enough. The models at different LoDs can be derived from the finest LoD by using generalization.

Attributes

Spatial objects have descriptive properties such as the owner’s name, the usage, object types, object id of a building.

Direction/Orientation

Direction can be used to describe the spatial relationships such as “on the left side of”, “to the right”, “north of”, “southeast of” etc.

Measurement/Metric

The measurements of spatial objects indicate the geometrical quantities such as length, perimeter, distance etc. On the other hand, it supports comparative operations such bigger, longer. And it also supports the description of spatial proximity like “500 meters away from”.

Topology

Topological relationships are the foundation for spatial overplays and network analysis such as navigation, routing, and facility location and allocation. Together with their transitivity rules, they provide a reasoning and computable base for queries about spatial relationships. The topological relationships in a 2D plane have been well studied in many literatures. Egenhofer and Al-Taha (1992) summarized eight basic topological relationships (Figure 3.6) between two objects by decomposing geospatial objects into three parts: boundary, interior, and exterior in a 2D plane.

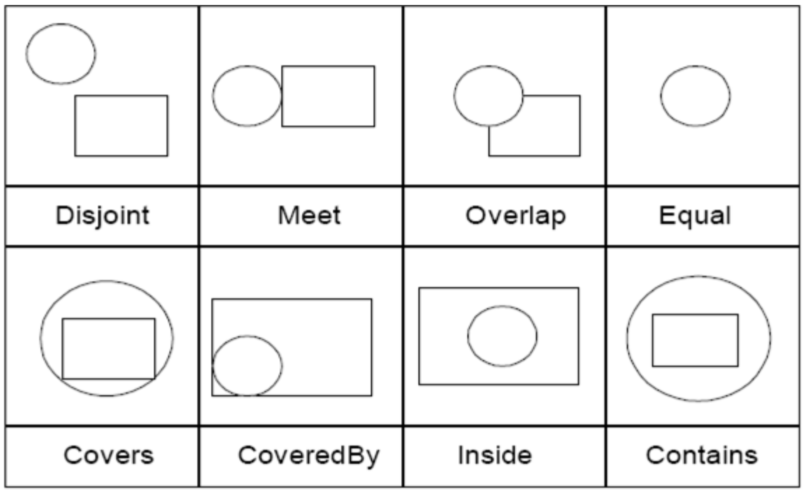


Fig.3.6. Topological relationships in 2D plane

Generally speaking, these eight topological relationships can also be deployed for 3D objects (Chen al et., 2008). In addition, topological relationships can be derived along the third dimension (the height) such as over, on, under, below, in the middle etc.

3.3 Event-induced changes for buildings

Events can be categorized based on salient perceptual attributes, or criterial features which have in common. As advised by (Stratulat et al., 2001) the notion of event type can be considered as an abstraction of a certain collection of events.

As already mentioned in Section 2.2, the geometry, topology and attributes of a spatiotemporal object may or may not change over time. Consequently, events can be categorized into three types: events of geometrical changes, events of topological changes, and events of attributive changes. For events belonging to each type, they can still be categorized further into many subtypes.

Events of geometrical changes: city objects e.g. buildings are changed in terms of shape and size. They might include:

- Construct a building: to make a new building e.g. a new house. For example: The family house on the Fellbachstr. 20 was built in 1976.
- Maintain a building: to protect a building from decaying or being damaged. For example: The roof structure of a building was strengthened.
- Restore a building: to make a building return to its former state, especially for the historic buildings, e.g. church, castle or palace. For example: The church was restored after the war.

- Renovate a building: to repair a building so that it is in good condition again. For example: The hotel was renovated two weeks ago.
- Destroy a building by unforeseen causes: to damage a building so badly that it no longer exists or cannot be used or repaired. For example: The clock tower was destroyed in the earthquake in 1950.
- Demolish a building in line with city planning: to make a building stop existing. For example: In order to establish a park, the old houses in the centre of the city were pulled down.
- Reconstruction, it might include:
    - Adding: to newly put up a part of building. For example: A chimney was added on the roof of our house.
    - Removing: to take away a part of building. For example: The balcony of a building was removed.
    - Expanding: to adjust a building or a part of it, so that it becomes larger. For example: The fence was moved outwards.
    - Shrinking: to adjust a building or a part of it, so that it becomes smaller. For example: The walls on the second floor were inwards reconstructed, and then there is a new terrace for the family.
    - Moving: to transport a whole building from one place to another, while its structure and textures remain unchanged. For example: The temple was moved five meters to the left in the earthquake.

Events of topological changes: The topologies between two objects may change due to their geometrical changes. Additionally, the topologies between two objects might change in accordance with the changes of their functionalities. Examples might include:

- The subway station is topologically met by the shopping center, because one of its exits is located directly adjacent to the main door of the shopping center. After the reconstruction of the subway station, one more exit is built on the first floor of the shopping center. Then the subway station is overlapped by the shopping center.
- The twin buildings become disjoint due to the land subsidence.
- The City Hall is moved from the link side of the square to the south of the city park.
- The factory is moved from the city center to the suburban area.
- The McDonalds is moved from the inside of the shopping center to the building opposite of the shopping center.

Events of attributive changes: The attributive changes of city objects may include:

- The ownership change that might take place in three different situations:
    - A building has been transferred: the ownership is changed because of selling or presentation.
    - A building has been inherited: the ownership is changed because of blood relationship.
    - A building has been nationalized: the building is confiscated by government after someone has died and he has on relationships to inherit his properties.
- The usage/function change, for instance, from private residence to commercial area or inverse.
- The price change, e.g. the marketing value of the building is more than the time before.
- The tenant change: the old tenant moved out and a new tenant moved in.
- The address change: a building can be assigned to a new address.

Besides, there are events represented as changes in the textures of a building. Textural events may be caused by geometric changes of the façade elements, such as:

- The pattern on the façade was changed.
- There was a new advertisement board on the façade.
- A new door was installed.
- A door was changed:
    - The door became bigger or smaller.
    - The door was replaced by another one with different type, e.g. a sliding door was installed in the place of an old swinging door.
- A new window was installed.
- A window was changed:
    - The window became bigger.
    - The window became smaller.
    - The window was replaced by another one with different type, e.g. a picture window was installed in the place of an old bow window.

Textual events may also be caused by changes that occur in material, image pattern, illumination/temperature or shadows of objects in the surrounding area, some typical events from this group are as follows:

- The material of the façade has been changed, e.g. the old red brick wall has been equipped with glass wall.
- The slogan on the wall was changed.
- A new door was installed.
- The door has been replaced by a new one with new material, e.g. the wood door has been replaced by a glass one.
- The window has been replaced by a one with new material, e.g. the wood window has been replaced by a steel one.
- The façade has been repainted with new color.
- The intensity or contrast of the surface turns lower because of environment pollution.
- The roughness of the façade is becoming always because of erosion by wind and rain.
- The illumination and temperature of the surface changes because of the sun shinning.
- The shadow of trees in front of the wall changes with the stand of the sun.

The lists above are not exhaustive. But it includes the most common events if textural events are concerned. Some events happen quickly, majoratively; the duration of them is very short; and the changes of textures resulted by them are large. And they indicate the change of the culture and society in a city and the development of the technology with the time. Some events happen all the time but very slowly, and they are very useful for us to analyze the degree of environment pollution and the speed of environmental degradation. Some are interesting for simulation and visualization of 3D buildings in the real world or a 4D scene in computer games. Moreover, events can be categorized according to the number of objects involved in the events: (i) events happened to single object, and (ii) events happened to a couple of objects.

In fact, events are sometimes complicated geospatial phenomena. There might be many possibilities to categorize them. Nevertheless, such categorization can provide guidelines, with which a knowledge base can be used to model spatiotemporal data.

Chapter 4

An object-oriented event-state spatiotemporal data model for a 4D city environment

This chapter presents a double indexed hierarchical spatiotemporal data model for storage and management of both semantic and geometric changes of 3D city objects. The data model is mainly composed of two parts: an object-oriented event structure for modeling events which happened to 3D city objects; and a hierarchical data model for modeling 3D city objects including geometries and semantics. The term "double indexed" is used, because the spatiotemporal data can be indexed both by events happened to objects and by objects involved in events. In this work, 3D city objects are referred to buildings including their components, since they are in the majority of city objects. This chapter is structured as following: the first section introduces the object-oriented event model; the second section presents the hierarchical 3D city model; the third section explains how the time factor i.e. event model is integrated with the spatial data model. Finally, the fourth section lists and describes the spatiotemporal algebras.

4.1 Object-oriented event modeling

This section begins with a general description of events. On this basis, several attributes can be derived, which can be then treated as variables if events are modeled using the object-oriented concept in the computer programming languages like c++ or Java.

4.1.1 Description of events with five Ws

In the field of journalism, there is a concept of Six Ws (Five Ws and one H) for news style, research and police investigations that are regarded as fundamentals in information-gathering. This concept can be deployed for events which happened to city objects. The difference is merely that the Why-Question is normally not required when events of city objects are concerned, since it can be implied by their attributes such as usages. Therefore, when talking about an event of a city object, for instance, a building, most people may raise five typical questions as: (i) what happened? (ii) where did it happen? (iii) when did it happen? (iv) how did it happen and (v) who was involved? These five questions can be answered respectively using the attributes of events, as following:

- What happened? -- To answer this question, the things happened have to be abstracted to a higher level, i.e. a term of type or class to which the event belongs. From this point of view, the question is asking the semantic of event.

- Where did it happen? -- This question concerns about the location of the occurrence. In our everyday life, an approximate position such as “the cross near the university main building” or “south of the TUM canteen” may help much more than a precise coordinate. In a city model, the location can be the addresses of the objects involved in the event.
- When did it happen? -- Depending on for what propose the question is asked, it can be answered with exact moment/period like “at 3:00 p.m. of last Friday”, “last July” etc. or using fuzzy concept as “ten years ago”, “during last century” etc.
- Who was involved? -- The interrogative pronoun “who” here is not restricted to human being, but rather all the physical objects which were involved in the event in question. In a city model, this question can be answered using the names or identities of the objects (buildings, or building parts etc.) which were involved.
- How did it happen? -- With this question the process or the way how the event happened is concerned. In our data model we propose to use event modes to describe the process. For instance, event modes for construction of a new building can be derived from the statistics of the construction works for the same type of buildings.

The above descriptions for events can be summarized and illustrated in Figure 4.1.

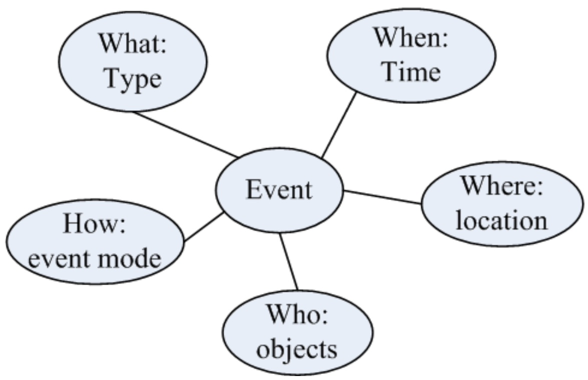


Fig.4.1. Five general attributes of an event

4.1.2 Structure and variables for event model

Taking the Five Ws as general attributes of events, then events can be modeled using a class structure as below. It is a void class that defines basic attributes and behaviors of any event that happened to city objects.

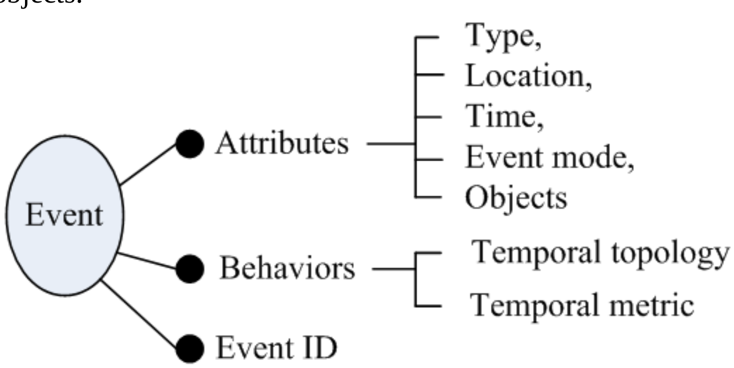


Fig.4.2. Class structure of the event model

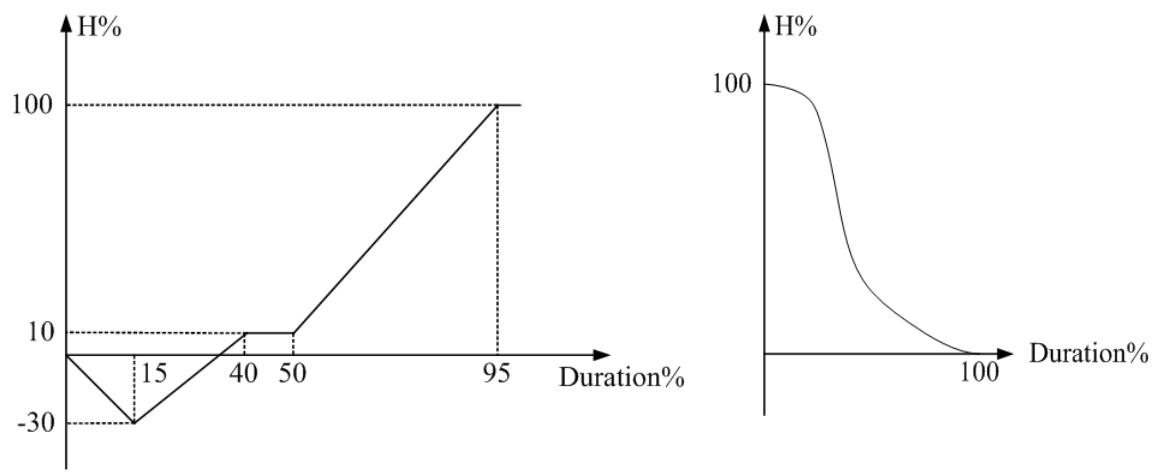
Structure of Attributes

The attribute “Type” records the information of “what happened?”. As indicated in Chapter 3, a type can be “construction”, or “destruction” etc. in terms of geometrical changes; or it can be “function change”, or “ownership change” etc. in terms of attribute changes.

The attribute “Location”can be represented by the addresses of objects involved in the event. In order to reduce the information redundancy, the involved addresses have to be abstracted as one address semantically. In case of interest, the individual address can be obtained by tracking the objects’ IDs that are stored as the object attributes.

The attribute “Time”is represented through the duration of the event. If there is geometrical change referred to the event, the duration should have a start point and end point of time, because the process of the change is normally meaningful. On the contrary, the duration of events that happen to objects’ attributes is zero-width and is therefore represented as a time point.

The attribute “Event mode”illustrates how the event happened. There are two cases depending on whether the event is geometric change or semantic change. In case of semantic change, the mode is represented by an attached document which records the semantic change. For example, an official document from the city hall can be treated as the best material for the event “the address of the building is changed”. In case of geometric changes, the mode can be a mathematical model which can approximate the process of the change. In a city model, several mathematical models can be derived by means of statistical analysis for, for instance, numerous construction works. Actually, a mathematical model for a construction work could be derived directly from the project plan. In addition, such models can also be found in the field of project management in civil engineering. Figure 4.3 shows an example model for a construction and destruction process of a building. According to this model, the height of the pre-building can be obtained by means of interpolation at any time during the duration of the construction or destruction.



a. an example model for the process of construction of a building

b. an example model for the process of destruction of a building

Fig.4.3. Example modes for events of buildings

The attribute “Object”records the IDs of the involved objects instead of their geometries which are stored in a hierarchical city model separately. With such an approach, data redundancy can be avoided. Otherwise, the geometry of an object has to be multiply stored, since it might be involved in many events.

Behaviors

The object-oriented event model defines two kinds of behaviors, namely, temporal topology and metric relationship. Whereby, temporal topologies cover the 21 possible relationships between two events (see Section 3.1). And the metric relationship includes temporal distance and temporal proximity.

4.2 The object-based hierarchical 3D city model

Broadly speaking, there are two types of models of geographical information, namely, field-based model and object-based model (Couclelis, 1992; Worboys, 1994b). The field-based model sees the world as continuous surface over which features vary (e.g. terrain, temperatures etc.), while the object-based model treats the world as surface littered with discrete objects (e.g. streets, buildings etc.). In our case, object-based model will be most appropriate, since we decompose a city into numerous city objects with distinct locations and boundaries for each. This section is devoted to an object-based hierarchical 3D of buildings. First of all, the hierarchical characteristics of building structures are discussed. Then the data structure for modeling building objects will be described.

4.2.1 The hierarchical characteristics of city structures

In our real world a city[3] is normally divided into several districts for administrative and geographical management. In every district there are a number of building blocks which exist at more or less the same times and reveal similar architectural style. In a block there might be many groups of buildings located very closely to each other. The individual buildings in a group usually share most of locational and functional properties. Further, a building can be segregated into walls, roof and floor; walls into even finer elements such as windows and doors, windows into a number of smaller components such as frame, glasses, grip etc., and so on. According to this decomposition a 3D building block could be semantically abstracted as a tree structure.

The hierarchies within the building block indicate “part-of” relationships. A roof object, for example, is usually part of a building object and should therefore appear as a child object of a building feature. Actually, this kind of hierarchy exists both in the semantic domain and spatial domain. In the semantic domain, the hierarchy stands for the structural subdivision of semantic information. Analogously, the hierarchy in the spatial domain denotes the levels of the structural subdivision of geometry into meaningful parts. At the same time, the hierarchy indicates the topological relations among objects.

According to (Stadler and Koble, 2007) these two hierarchies are considered coherent and structurally isomorphic, because they show the same structure. In other words, all semantic components should be correlated to geometric components on the same level of the hierarchy. The reflection of this characteristic on a concrete object is that it has geometry at a certain level and semantic at the same level (Figure 4.4). In this way, semantic and spatial information is conjunct on an object. For example, if a wall of a building has four windows and a door on the

[3] In this work, the term city is used as an aggregation of all the buildings located in the city. Other objects like streets, water, meadows etc. are not included in the current city model. Therefore, the term city object in this thesis refers, in general, to object like building block, or one of its components.

semantic level, then the geometry representing the wall must contain also the geometry parts of both the four windows and the door.

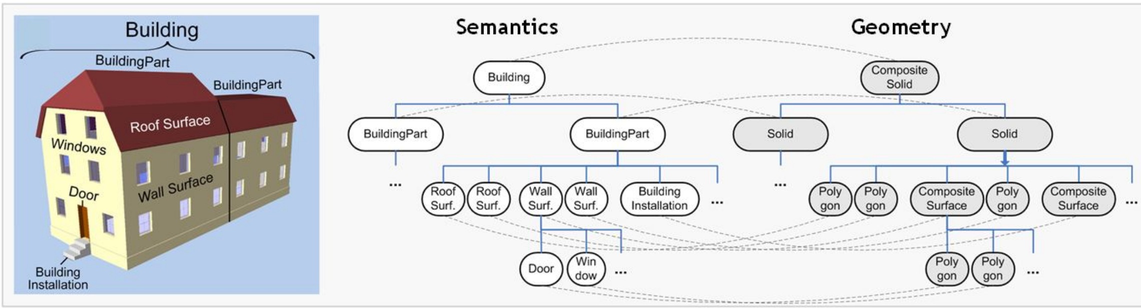


Fig.4.4. Coherent hierarchies in semantics and geometry of city object (i.e. building) (Stadler and Koble, 2007)

According to (Mayr, 1982; Gibson et al. 2000), the structure in which the building objects are nested is a constitutive hierarchy, because the lower level (roof, walls, windows, doors etc.) can be combined into new unit (a building) that has new functions and eminent properties. In terms of geometry, the new unit is a direct aggregation of its child units. However, the characteristics of the new unit are not the simple combination of attributes of its child units, but may show new collective functions and behaviors[4], when semantics are concerned.

Furthermore, the depth of the hierarchy reflects the complexity of the corresponding city object. On the other hand, building objects at a certain level in the hierarchy is identifiable only in a certain range of spatial resolution, because their patterns that appear at some levels of resolution or extent may be lost at lower or higher levels.

4.2.2 The spatial object

Figure 4.4 indicates that semantics and geometries are coupled for every distinguishable and meaningful part of a building. Treating them as attributes, their corresponding object can be modeled by a class using object-oriented concept (Figure 4.5).

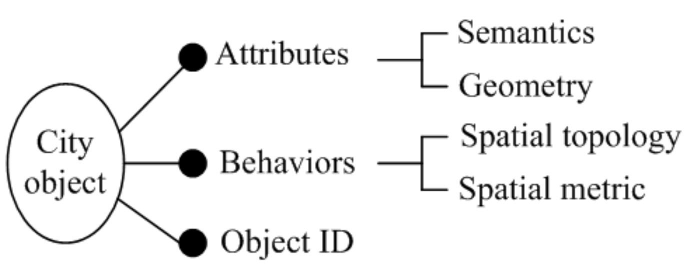


Fig.4.5. Superclass for city objects

The superclass in Figure 4.5 defines objectID, basic attributes and behaviors of any city object. For modeling objects at a certain level in the hierarchy, the superclass can be inherited, specified and extended according to the properties of their object class.

[4] Morgan (1894) was among the early scholars to point out that ‚at various grades of organization, material configurations display new and unexpected phenomena and that these include the most striking features of adaptive machinery’ (cited in Mayr, 1982, p.63).

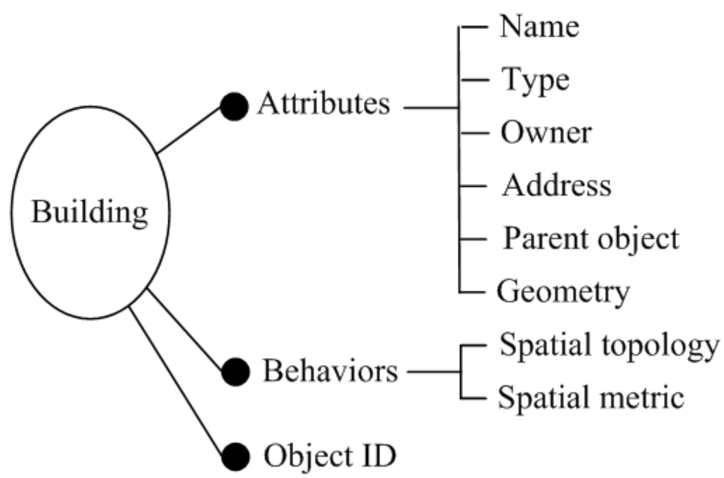


Fig.4.6. Class structure of buildings

Figure 4.6 shows the class structure of buildings. First of all a building should have a unique ID to be identified. At the same time, the behaviors defined for the building class can be deployed for all of its members of their own accord. In general, the behaviors defined in the model are spatial relationships between two objects which include topological relationships (such as meet, intersection, inside, contain, equal, and etc.) and metric relationships (such as distance, comparative operations etc.).

The attributes of the building class contain semantics and geometries. Name, type, owner, and address are obviously common semantics of a building, while geometries are spatial attributes. The attributes "parent object" and "child objects" can be regarded as pointers that indicate the hierarchical relationships between the building and a building group, and its components (such as roof, walls, chimney etc.) at a lower level respectively. From this point of view, they can be regarded both as semantic and geometrical attributes.

In principle, the Geometry attributes should record geometrical information of the building. Depending on at which level a city object is located in the hierarchy, it will be represented in different ways. In case that it is a leaf node, its geometry will be represented directly by polygons or multi-polygons. Otherwise, if the object is an internal node, it will be represented as a geometrical aggregation of its child objects. In this case, the geometries of the child objects will not be stored directly; instead they will be referenced by their IDs. Analog to the Geometry attributes the attributes of "child objects" may be empty or record the IDs of the child objects. In this sense, the attributes of "child objects" are equivalent to the Geometry attributes. Therefore, they must not be included in the class structure of buildings.

Apart from the building class, an object class will be defined for every city object which is semantically distinguishable and can be recorded meaningfully in the city model. In general, all the city objects may have similar behaviors. But they need different attributes to describe themselves. For some object classes, for instance roofs, the attribute of name is not needed, while the type attribute is necessary, since they are normally identified according their types.

The geometry of a building is a subtree in the hierarchy (see 4.2.1). Theoretically, a city object, for instance, a building can be semantically decomposed into very tiny objects, such as a handle on a door, or even a screw of it. Depending on how detail the object needs to be modeled, the object may be a leaf node[5] in the building hierarchy or may be decomposed up to a certain level in the semantic domain. Correspondingly, the geometry of the object needs to be represented using different level of detail (LoD).

4.2.3 The implementation of the data model

The above described city object class and the hierarchical city model can be implemented either using conventional database or Extensible Markup Language (XML) or other XML-based markup languages such as GML3 and CityGML.

In a conventional database, an object class is the equivalent of a table; each individual record is represented as a row and an attribute as a column. Object classes are related to each other using *1:N* mapping which is also know as one-to-many relationships. The child-mother-relationships are indicated by storing the corresponding IDs as attributes of “parent object” and “child objects”. This kind of approach is conceptually simple. Object class including the attributes and behaviors can be flexibly defined according to particular applications. However, the data exchange among different users is almost impossible due to the lack of standardization.

Another technology that can manage hierarchical data model effectively is the XML. Specified for city model, CityGML is an XML-based OGC (Open Geospatial Consortium) standard format for the storage and exchange of 3D data of city objects. It defines the classes and relations for the most relevant topographic objects in cities with respect to their geometrical, topological, semantical and appearance properties. Included are generalization hierarchies between thematic classes, aggregations, relations between objects, and spatial properties (Gröger et al., 2008).

On the base of XML-schema, CityGML represents topologies between city objects explicitly. Taking the building model as an example, every part of a building may be modeled only once and then referenced by all features which include the same geometry. For instance, if there is a window sealed in one side of a gabled roof, the window will be modeled only one time, the geometry will be referenced to that side of roof in which the window is installed, and then to the gabled roof, and to the building in further. In this way, redundancy can be avoided and explicit topological relations between objects are maintained (Stadler and Kolbe, 2007).

In the view of these advantages, CityGML is adoted in this thesis for representing the geometries and semantics of city objects. In addition, it is extended for our special purpose of integrating time-dependent features within 3D city model.

4.3 The integration of temporal information within 3D city model

Using the data model described in Section 4.2, a city including its entire objects can be modeled with complex geometries and rich semantics. Nevertheless, it is frozen at a certain time point, since it does not support any temporal information for indicating both geometrical and semantical changes. In this section, the spatial data model will be extended by (i) adding lifespan to the corresponding semantical and geometrical attributes to indicate their existence, (ii) combining the event model with the object model to represent the process of changes happening to the attributes, and (iii) defining new behaviors for spatiotemporal changes.

[5] The leaf node here means the object will be not semantically decomposed into smaller objects.

4.3.1 Time-stamped city objects

The notion of lifespan is introduced to represent the period of time for which an object exists or is expected to exist. It is formed by two points of time which sign respectively the instants when the object appears and disappears. Besides, the semantics of an object may also have lifespan, as they may be valid for a certain period during the existence of the object. Therefore, a straightforward approach of integrating time features within city model would be stamping their lifespan on city objects including the semantics and geometries.

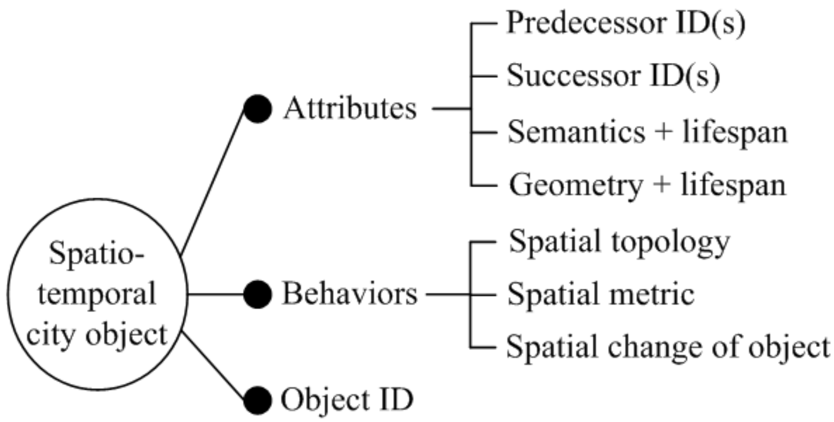


Fig.4.7. Class structure of time-stamped city objects

Figure 4.7 shows the class structure of time-stamped city objects. In comparison to (spatial) city objects, spatiotemporal city objects at the leaf level of the city hierarchy have lifespan for their geometries and semantics respectively. With respect to geometry, the objects at the leaf level of the city hierarchy will be modeled, in this work, as unchanged during their lifespan, although they might be changed geometrically in the reality. They behave only as appearing and disappearing. Therefore, their geometries will be modeled only one time for their whole life. On the contrary, their semantics may change during their lifespan. For example, the color of a window or door may be changed several times due to renovations. In this case, the semantics belonging the same type of attribute will be listed together with their lifespan (valid time) in a time sequence according to the order when they were valid.

For the objects corresponding to the internal nodes in the city hierarchy, their semantical attributes will be recorded in the same manner as for objects located at the leaf level. However, their geometries contain not only those of their current[6] child objects, but also the geometries of those objects which had been their child objects in the history (can also be called previous child objects). Normally, there are relations between the current child objects and the previous child objects of a city object. Some of the previous child objects might be predecessors or predecessors of predecessors of the current child objects, when they almost occupy the same space and posses similar functionalities. The predecessor-successor relationships are indicated by recording the corresponding object IDs in the attribute “predecessorID” and “successorID”.

In fact, if we treat the geometry of an object represented by its current child objects as the initial state of the object, the previous child objects of this object can be regarded as the changed parts of the initial state. In this way the redundancy is avoided. On the other hand, the approach of

[6] The word current here means at the moment when the referred city object is captured in the data base.

recording "predecessorID" and "successorID" as attributes ensures that changes such as replacement, splitting, and merging can be represented.

Moreover, the spatiotemporal object class defines behaviors with consideration of temporal factor. In comparison to those in the city object class, the spatial topologies, metric, as well as the comparative operations between two objects might be changed when their geometries change. When an object is compared with itself at different time or from different states, there could be spatial changes as follows: (i) appear/disappear, (ii) increase/decrease in height, area, volume etc., and (iii) splitting. These will be included in the behaviors of spatial change of the spatiotemporal object class.

4.3.2 Event as transition from one state to another

By using the spatiotemporal object class (section 4.3.1) the historical states of an object can be represented. One can see the change of an object from one state to another. But one can never know how it changes because there is no instruction about the process of the change. On the other hand, change takes time. Therefore, there will be gap between two states in the time dimension, if they are observed closely. In other words, the life of an object can not be represented continuously. In this work, the change from one state to another is treated as an event. Then the life of an object can be represented by a time-ordered sequence of states and events.

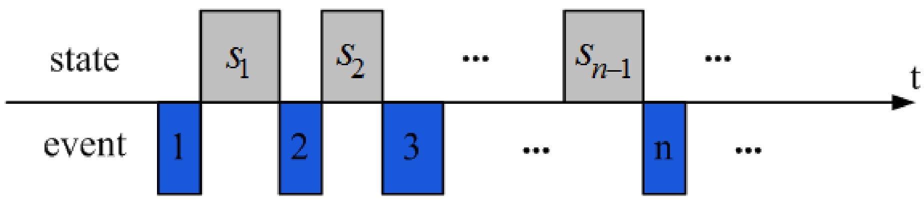


Fig.4.8. Event-state representation for the life of an object

Figure 4.8 shows an example of an event-state representation for the life of an object, whereby the gray boxes stands for the states of the objects in different periods and the blue boxes denotes the events that happen between states. In general, even a simple object has at least two events and one state, since it appears at a certain time, stays for a while and will disappear finally. Besides, their semantics may be changed during their lifespan.

There are two kinds of situations for an event happened to an object that can be semantically decomposed of a few smaller objects: (i) the entire object is involved in the event, and (ii) only one part of the object is involved in the event. In this work, the former is defined as the "1st-level" event, while the latter is called the "2nd-level" event.

Definition

1st-level event: an event of 1st-level to an object is an event in which the whole object is referred or involved.

Examples of "1st-level" events are: construction, destruction of a building; the change of the ownership, address, function etc. of a building. All these events are the "1st-level" events to the building.

2nd-level event: an event of 2nd-level to an object is an event in which only some of its child or grandchild objects are involved.

In fact, whether an event is constituted as a 1st-level event or a 2nd-level event depends on the relation of the referred object O_c and the object at the root node of the sub-tree whose all nodes are involved in the event O_i :

- In case of $O_c = O_i$, the event is a 1st-level event.
- In case of $O_c \subset O_i$, the event is a 1st-level event, only when the event is semantically decomposable.
- In case of $O_c \supset O_i$, the event is a 2nd-level event.
- Only the semantically decomposable objects have 2nd-level events.
- For a leaf-node object, there are only 1st-level events.

From this point of view, every 2nd-level event to an object is a 1st-level event to its child, and further offspring. Therefore, in order to avoid the problem of multiple-store of events, all the events in the event model are connected with their objects as the 1st-level events.

For the semantically decomposable events, there is certainly the coherent semantic structure for the corresponding spatial objects. When decomposing such events and their corresponding objects simultaneously, many 1st-level events arise. Depending on to which LoD the spatial objects are decomposed, the events can be represented at different LoDs with respect to how precisely the dynamic processes can be represented.

Example: event E_a - construction of a building is a 1st-level event to the building. This event can be decomposed of event E_{a11} ”construction of the roof” and E_{a12} ”construction of walls”. And they can be further decomposed of event E_{a21} ”construction of wall elements”, event E_{a22} ”installation of windows”, and event E_{a23}” installation of doors” and so on. In the event model, event E_a is connected with the ID of the building. For representing this event, the new construction works of roof and walls are not differentiated. In other words, the connected building can be represented as block model. But if the event is decomposed of E_{a11} and E_{a12}, the entityIDs of the corresponding roof and walls have to be given as the connection for the event model and the city model. Then the event E_a is represented as an aggregation of the event E_{a11} and E_{a12}. For further decomposition, the process of smaller changes i.e. construction of wall elements can be represented.

So far, a city object is modeled with an ObjectID, its semantics, geometries and general behaviors. A new ObjectID is created when an object is totally new constructed. This ObjectID will be eliminated when the object including all its components is eliminated or destroyed. After an object is eliminated, a new object may be created with similar functionalities and located in the same place. It will be treated as the successor to the objet eliminated before. They will be referenced to each other using successorID and predecessorID. They reflect two states of their parent object. Events of construction and destruction will be used to describe how it changes from one state to another.

4.4 Spatiotemporal operations in the data model

In the traditional DBMS there are five primitive operations for relational algebra according to Codd's theorem: the set union, the set difference, the Cartesian product, the selection, the projection (Chin and Tarski, 1948). These operations are widely used and extended according various specifications. In this section, extend them for the spatiotemporal data model designed in the previous sections and present them in the following, whereby Φ_1 and Φ_2 are two entities with spatiotemporal properties.

(1) Spatiotemporal union
$\Phi_1 \cup \Phi_2 \underline{\underline{\Delta}} \{\tau | \tau \in \Phi_1 \vee \tau \in \Phi_2\}$, where τ is a variable of spatiotemporal attribute. This operation is similar to the logical operation "or".

Actually, the spatiotemporal union can be composed of two operations: aggregation and abstraction. At first the involved objects will be directly aggregated both in spatial and temporal domain. In many cases, the result of the aggregation has to be generalized, since for the identification of new geometry smaller scale might be required and some detail may not be differentiated any more under the new scale. Then their semantics will be abstracted up to a higher level.

(2) Spatiotemporal difference
$\Phi_1 - \Phi_2 \underline{\underline{\Delta}} \{\tau | \tau \in \Phi_1 \wedge \tau \notin \Phi_2\}$, spatiotemporal difference gives spatiotemporal attributes in Φ_1 but not in Φ_2.

(3) Spatiotemporal intersection
$\Phi_1 \cap \Phi_2 \underline{\underline{\Delta}} \{\tau | \tau \in \Phi_1 \wedge \tau \in \Phi_2\}$, spatiotemporal intersection gives spatiotemporal attributes both in Φ_1 and in Φ_2.

(4) Spatiotemporal selection
The spatiotemporal selection severs for the spatiotemporal queries. It makes a horizontal partition on a spatiotemporal relation based on the given conditions. Let the given condition be represented by a predication formula F, the selected spatiotemporal entity should satisfy the equation: $\sigma_F(\Phi) \underline{\underline{\Delta}} \{\Phi | \tau \in \Phi \wedge F(\tau) = \text{true}\}$.

(5) Spatiotemporal projection
In contrast to the spatiotemporal selection, the spatiotemporal projection makes a vertical partition on a spatiotemporal relation. Suppose Φ has attributes $(a_1, \ldots, a_m)$, $(a_i, \ldots, a_n)$ are a set of attributes and $(a_i, \ldots, a_n) \in (a_1, \ldots, a_m)$, whereby $1 \le i \le n \le m$. Then the result of spatiotemporal projection is defined as $\pi_{a_i, \ldots, a_n}(\Phi)$, whose attributes are restricted to the set $(a_i, \ldots, a_n)$ – it discards (or excludes) the other attributes.

Chapter 5

Spatiotemporal queries and visualization in the 4D City environment

Every spatiotemporal object can be described with three components: space (where), time (when) and object (what). Accordingly three basic kinds of questions are possible when querying an entity, if one takes two components as constraints while inquiring the third one (Peuquet. 1994; Andrienko et al. 2003). Concerning the complexity of spatiotemporal processes, Yuan (2000) added the fourth key aspect in spatiotemporal queries: event or process (how). Corresponding to these four key aspects, the most frequently asked questions in a spatiotemporal system can be categorized into four types of spatiotemporal queries (Yuan, 1999): (i) queries about attributes of entities, (ii) queries about location, spatial properties, and spatial relationships, (iii) queries about time, temporal properties, and temporal relationships, and (iv) queries about spatiotemporal behaviors and relationships. These four types of queries are further synthesized in: one type of attribute query, three types of spatial queries, three types of temporal queries, and four types of spatiotemporal queries (Yuan and McIntosh, 2002).

Yuan's typology including 11 query types serves as foundation for inquiring information from a spatiotemporal system regardless how the spatiotemporal information is modeled. Depending on spatiotemporal data models, these 11 types of query can be posed in different ways: object-based or event-based.

In our case, the 4D City environment is modeled in an object-oriented event-state spatiotemporal data structure (Chapter 4): states of city objects including their attributes are represented using versions at different time and events are used to describe how they change from one state to another. In this specification, we categorize spatiotemporal queries into two types according to the structure of spatiotemporal entities: object-based queries and event-based queries. At the same time, the complexity and characteristic of spatiotemporal entities are considered when categorizing queries in this way.

The retrievals of the spatiotemporal queries can be classified in two groups with respect to their temporal attributes: information freezing at a point of time and information lasting over a time interval. The former is actually a snapshot while the latter contains changes or motions. Accordingly, the retrieved information can be visualized statically or dynamically.

In this chapter, spatiotemporal queries are explained in line with specifications of our data model. Then the concepts for visualizing retrievals of queries are introduced.

5.1 Spatiotemporal queries

As indicated in the introduction, spatiotemporal questions can be either object-based or event-based. In this section these two types of queries are addressed respectively.

5.1.1 Object-based queries

Object-based query means that objects have to be identified at first in order to answer the query. With respect to an object, not only its geometry and attributes can be asked, but also the spatial and temporal relationships (comparisons) between this object and others can be inquired.
In terms of geometry, the queries distinguish between geometrical properties of the referred object and geometrical relationships to other objects, whereby geometrical properties include area, volume, height, length, width, shape, and sinuosity etc., and geometrical relationships to other objects could be distances, directions and differences in their geometrical properties (difference in height) etc. Examples of queries about geometry might be:

- What is the height of the Olympia Tower in Munich?
- What does the facades of TUM main building look like in 2006?
- What is the average height of the buildings in the district of Moosach in 1980s?
- How many meters is the TUM main building higher than the building of its left wing in 1940?

In terms of attribute, information regarding attributes of objects and their relationships can be inquired. Moreover, the temporal attributes are often queried. Examples of attribute queries include:

- Where was the BMW headquarters in 1955?
- Who is the owner of Hotel Orly since 2005?
- What is the color of the facades of TUM main building in 2006?
- What is the function the building on the Lauterbachstr. 8 in Munich?
- How many buildings in Munich belonged to TUM in 1960s?
- When was the new Pinakothek built?
- How long did the watch tower exist in the east center station?

In terms of spatial relationship, spatial topologies between objects, comparative spatial proximities, and orientation-related information will be asked. In addition, many geometrical and attribute queries are conditioned by spatial relationships between objects. Examples of such queries are:

- Where is the Lidl market located in the Mira shopping center?
- What is the name of the nearest hotel to the Rathaus?
- How many hotels are there on the street directly south of the center station?
- How many family houses are there within 500 meters radius around the St. Helena church?

In terms of temporal relationship, temporal topologies as well as the comparative temporal proximities may be interested:

- Which buildings on the Loth street were built after 1990?
- What is the average lifespan of buildings in the district of Laim?
- How many buildings are older than the old Pinakothek?

5.1.2 Event-based queries

Event-based queries seek information about events; therefore an event has to be identified at first. In most cases, such queries can be answered by retrieving data records only in the event lists without any operations in the spatial data base.

As an event is normally modeled to answer five questions: what, where, when, who, and how (see Chapter 4), queries about events are specifications of these five kinds of questions.

Queries about what: question about what happened on a time slice or over a time range is one of the most common questions in spatiotemporal data models (Langran 1992; Langran 1993). Examples include:

- What happened to the facades of TUM main building since last year?
- How many times has this building changed ownership since 1980?
- How many buildings were constructed in the district of Sendling from 1990 to 2000?

Queries about where: this kind of query asks where the change occurred. Unlike the events of moving objects, the locations of events which happened to city objects including their geometries and attributes are coupled with city objects and normally kept unchanged during the events. Therefore, this kind query can be answered by retrieving addresses of objects that are involved in the event inquired in the query. Examples of such kind of query include:

- Where did the latest fire accident take place in Munich?
- In which district of Munich are the most of new constructions located?
- Where are the café restaurants whose businesses were started in 1995?
- Where was the city theater moved from?

Queries about when: the when questions ask the temporal information about events. The temporal information inquired in the queries is distinguished between the time including its temporal properties of the event and temporal relationships among events. Whereby, temporal properties include the length of duration as well as the frequency of the occurrences. The temporal relationships can be temporal topologies between events, or temporal distance. Examples include:

- When was the TV tower built?
- When was the BMW museum moved to the current building?
- How long did the construction work of the Oskar-von-Miller Forum last?
- How often is the restoration undertaken for the Frauenkirche?
- When was the second change of the ownership after the fire incident of that building?

Queries about who: this kind of query deals with objects which are involved in the event. Examples include:

- How many buildings changed their usages in 2000?
- Which buildings were flooded during the flood in the summer of 2004?
- How many buildings were destroyed during the earthquake?
- Which buildings were constructed with a time of more than two years?
- Which buildings on the Loth street were renovated last year?

Queries about how: this kind of query explores information in the process of an event. Regarding the events of attribute changes, the query can be answered by listing the attributes before and after the change and giving additionally the document which describes the attribute change in detail. Regarding the events of geometrical changes, the query is normally interested in the dynamic process of the changes, for instance, how is the geometry at a certain time during the dynamic process. In this case, interpolation is required. The geometry at every point of time can be interpolated by using the mathematical model for describing the change process (see 4.1.2). Examples of this kind of query include:

- Who is the owner after the third ownership change of that building?
- How many meters in height on the average were added to the building during the construction process?
- How did the usage of this building change last time?

In this section, spatiotemporal queries are categorized into two types: object-based and event-based, depending on which one has to be identified at first after the input of a query. Actually, in many cases these two types of queries can be distinguished quite obviously. But some object-based queries are also event-based at the same time. For example, which building was constructed in 1995? Such queries can be treated either as object-based or event-based in the practice.

5.2 Spatiotemporal visualization

The retrieval of the queries can be visualized not only statically, but also dynamically. In this section these two types of representation will be illustrated.

5.2.1 Static visualization

In the most common case, the static mode is used to represent a snapshot of geospatial changes. However, sometimes geospatial changes that occurred at different point of time or even over different durations can also be visualized statically. In this case, the temporal factor can be represented implicitly by using different colors, forms of lines or different forms of polygons etc. Similar performance could also be found in our everyday life.

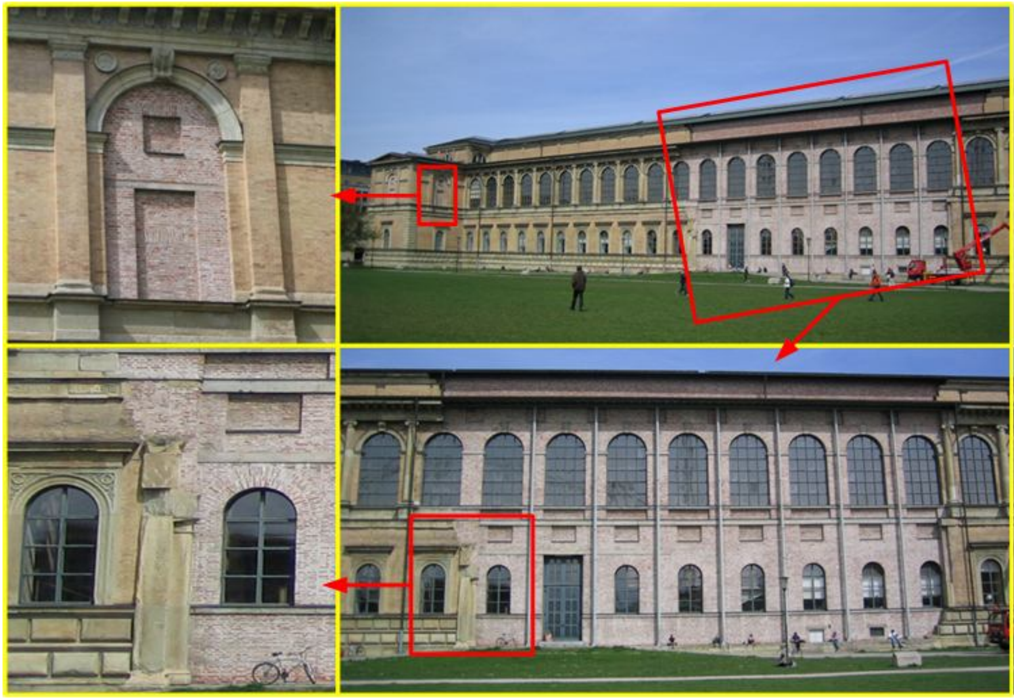
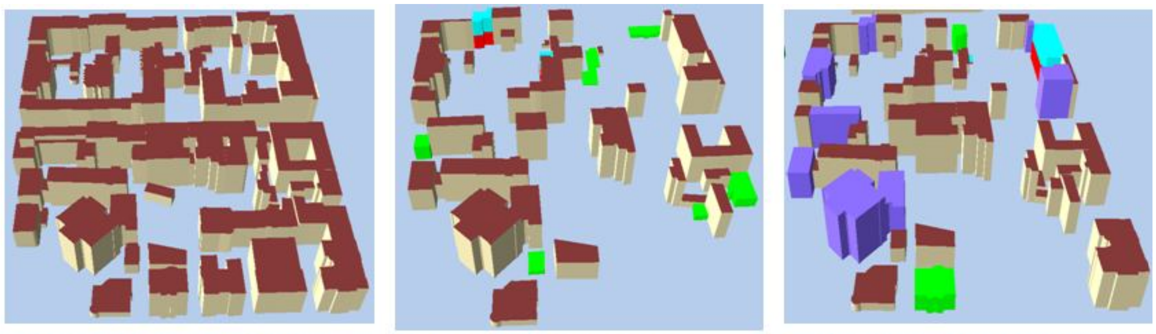


Fig.5.1. Alte Pinakothek in Munich

Figure 5.1 shows the "*Alte Pinakothek*" in Munich. The building was erected as gallery for painting collection in 1826. But it was considerably damaged in the World War II. Then the building was rebuilt by Hans Döllgast in 1957. However, rather than merely reconstructing the missing parts of the walls and windows, these areas were replaced with bare brickwork, in order to remain as visible "wounds" (website of Alte Pinakothek). If we observe this building from the time point of view, the building could be regarded as a visualization of different parts of buildings by using different colors and patterns to represent different time of construction.

In the following, an example will be given, in order to show how to represent temporal information using different colors[7]. Figure 6a gives an overview of all the buildings available in a block. Figure 5.2b shows the results of a query at a certain time. Whereby, the buildings represented same as those in Figure 5.2a existed at the time in the query. The buildings highlighted with green color were being constructed at that time. And the buildings represented with two colors were being destroyed, with light blue for the part already destroyed and red color for the part remained. Figure 5.2c shows the results of a query over a time interval, where the buildings in green were being constructed at the end of the querying time interval; the building in purple existed at the beginning of the querying time interval, but had been already destroyed at the end of the querying time interval; the building in two colors were being destroyed at the end of the querying time interval.

[7] Note: the selection of colors in this example is randomly done for the sake of differentiation, not more. In the future, for a more elaborated visualization tool, color design may become an issue.



a. An overview of all the buildings in the block

b. Results of query at a time point

c. Results of query over a period

Fig5.2. Static visualization of the querying results

Furthermore, the temporal factor might be explicitly represented by indicating the temporal information in different ways. In this case, temporal information can be indicated directly using indicators like “placemark”, sign board or writings on the facades of buildings (Figure 5.3). The approach using “placemark” for indicating temporal information is similar to the “placemark” used in Google map and Google Earth. As shown in Figure 5.3a, the construction time of a TUM building is denoted using a “placemark”. Figure 5.3c shows a sign board recording the historical moment when the fortress of Suomenlinna in Helsinki was included in the word heritage list. As a common phenomenon in Germany, for many historical buildings, some of their events such as brief information about construction (Figure 5.3b), the authorization of a certificate (Figure 5.4b), or change of ownership (Figure 5.4c) are carved on their facades.

a b c

Fig.5.3. Representation of temporal information using “placemark”, sign board or writings on the facade

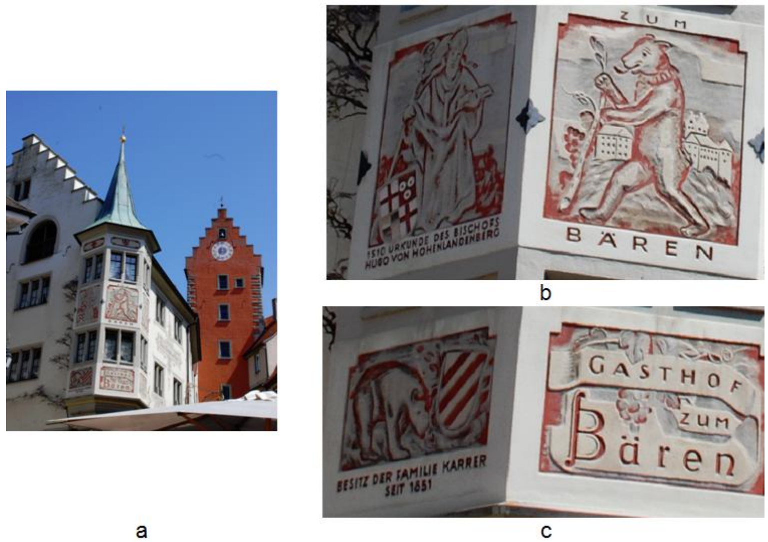


Fig.5.4. Some events of a building were carved on its facade

5.2.2 Dynamic visualization

The modern computer technologies enable the visualization of highly detailed 3D city objects. In recent years different applications have been made available for visualizing city objects in different data formats. However, most of the visualization systems today can not handle temporal changes of the underlying models in an efficient way, except that these systems can show such time stamps as textual annotations (Hertel et al., 2009).

In this work we proposed a framework for establishing a system to visualize 3D geometrical changes dynamically. Therefore, a number of issues and challenges must be addressed to develop and manage the operations and processes in the system.

First of all, a mechanism of spatiotemporal indexing has to be developed, in order to accelerate the process of queries and visualize the retrieval in further. Although there are many types of indexing methods which have been broadly used, such as the R-tree (Guttman 1984), the K-D-B tree (Robinson 1981), the K-D tree (Glassner 1984; Fussell and Subramanian 1988), and the Z-ordering (Orenstein 1986) etc., they can not satisfy the requirements of very high performance for high dimensionalities e.g. 3D or even 4D. In our project, we attempted to extend 3D K-D tree to 4D K-D tree with the intension of searching time incorporated 3D data more efficiently (Stefan et al., 2009). But the performance still needs to be evaluated.

Secondly, a set of constraints and modus have to be defined for visualizing temporal information and spatial changes over time. They include: (i) integration of various algorithms for interpolation of geometry, (ii) arbitrarily setting spatial and temporal grains for the visualization environment, (iii) functionality and capability of visualizing geometrical changes with different temporal grains at the same time[8], and (iv) switching among different LoDs smoothly while increasing or decreasing spatial scale. Finally, user-friendly interface and tools have to be developed, so that users can interact with the spatiotemporal data.

[8] Different changes are interpolated using different temporal grains. As a result, events which occurred slowly might be visualized with acceleration, while the processes of events which happened quickly might be visualized in slow motion so that their processed can be observed in detail.

Chapter 6

Spatiotemporal generalization

6.1 Introduction

In comparison to traditional spatial data models, dynamic processes can be stored and queried in the spatiotemporal data model. As presented in Chapter 4, the states and the event-induced changes of city objects are stored respectively in a hierarchical city model and event model. The input of an explicit query triggers the corresponding operations on the spatiotemporal data model. The states and/or changes satisfying the conditions defined in the query will be retrieved as results of the query.

For example, a terraced house on a street in Munich is composed of eight family houses which are numbered from h1 to h8 in the following. In the year of 2009 the following things happened to this terraced house:

(1) A shutter window of the house h1 was uninstalled, enlarged and replaced by a French window.

(2) The house h4 was transferred from Wolfgang Ullmann to Daniel Maier.

(3) A small green house was built on the roof of the house h4.

(4) The yellow coat of h4 was painted with light blue color.

(5) A half-glazed window of h4 was replaced by a casement window.

(6) An extension for porch was built for h5.

(7) A side door was installed for the entrance of the garden of h8.

Certainly, there should be a lot of changes happened to these eight family houses individually in their histories, i.e. their ownerships might be changed, some of them might be damaged by fire or other disasters and maintained thereafter, etc. An example query might be “what happened to h4 in 2009?” In order to answer this question, the above listed changes with the number (2), (3), (4) and (5) will be retrieved and treated as events happened to h4 in 2009.

The results might be different if we change the temporal and spatial scales:

- Elaborating the temporal scale causes an increasing number of the retrieved changes, while coarsening temporal scale leads to reduce of the number of the retrieved changes. For example, for the query “what happened to h4 since its construction?”, the changes of (4) and (5) will also be retrieved, but probably neglected, because they might not be significant or enough important in the entire history of h4.

- Changing the spatial scale may also cause an increase or decrease of the number of involved objects. For example, "what happened to the terraced house in 2009?" For such a query, all the changes happened in 2009 to the components of the terraced house will be retrieved. But some of the retrieved changes should be neglected if only a certain number of changes (significant changes) can be treated as events.
- On the other hand, some objects become invisible or unidentifiable with the coarsening spatial scale. In consequence, their 1st-level and 2nd-level events have to be neglected, since they are too small to be noticed.

The abovementioned problems can be explained as a matter of scale. Spatiotemporal objects or geospatial phenomenon are scale-dependent (MacEachren & Kraak, 2001). They can only be observed or identified within a certain range of spatiotemporal scales. In terms of the spatial part of the geospatial phenomenon, they can be modeled with very detailed geometries, for instance, a building can be modeled with detailed facade and roof installations or even with interior objects. But these details might be invisible or can not be differentiated when observing the building at its table-top or even smaller scale. Concerning the temporal part of the geospatial phenomenon, there might be many event-induced changes to a building. Some of them are 1st-level events to the building itself while some are 2nd-level events (i.e. 1st-level events to its child or grandchild objects). Some of these changes might appear as significant or insignificant, depending on the spatiotemporal environment for the observation. In order to find out the significant changes to the spatiotemporal objects at different spatiotemporal scales, spatiotemporal generalization has to be conducted.

In this chapter, a concept for the spatiotemporal generalization is proposed. First of all the issue of scale will be addressed in terms of "grain" and "extent". Then the concept of the spatiotemporal generalization is presented which is composed of two parts: event generalization (temporal generalization) and 3D generalization (spatial generalization). And the algorithm of event generalization is explained in the last section. The algorithms of 3D generalization will be presented in Chapter 6.

6.2 The issue of spatiotemporal scale

Every geographic phenomenon is scale-dependent. In broad sense, this scale-dependence is not only in spatial sense but also in temporal sense, when the time factor is also concerned. In fact, scientists have been aware of the spatiotemporal scale-dependence of geospatial phenomenon already over a long time (Gibson et al. 2000).

In this thesis, we categorize scale as is commonly done in everyday life (e.g. fine scale refers to minute resolution or small area, and broad scale refers to coarse resolution or large area) rather than use the cartographic scales (i.e., large scale refers to small resolution).

The word scale is used in many contexts and often connotes different aspects of space and time. In the field of geography scale is normally described using the terms "grain" and "extent" (Weins 1989; Turner et al. 1989; Allen and Hoekstra 1991; King 1998). There are two sets of forms for extent and grain when observing a geospatial phenomenon. One is for the geospatial phenomenon itself, whereby grain is the finest level of spatial or temporal resolution by which the geospatial phenomenon is measured or described, and extent refers to the spatial size or temporal duration of the phenomenon. The other is for the observation environment, whereby grain is the spatial or temporal resolution by which one observes the geospatial phenomenon, and extent refers to the spatial boundary or temporal range in which the geospatial phenomenon is observed.

Therefore, grain and extent may be intrinsic characteristics of the phenomena being observed on the one hand and extrinsic characteristics of the observation on the other hand. The observation grain and extent have to be matched with those of the geospatial phenomenon being observed. In other words, a geospatial phenomenon can only be observed within a certain range of spatiotemporal scale which is also called valid scales.

In general, there are two ways to guarantee that a geospatial phenomenon can be observed in an appropriate environment (with appropriate grain and extent). One way is to give valid intervals of spatiotemporal scales for the geospatial phenomenon (Fan et al. 2009b). The other way is to calculate the event significance value of the geospatial phenomenon under a given spatiotemporal environment of observation (Fan and Meng, 2008; Fan et al. 2009b). In the first case the valid intervals of spatiotemporal scales can be pre-calculated and treated as attributes of the geospatial phenomenon. In the second case, the calculation will be conducted on the fly, namely, after the parameters of the observation environment are given along with operations in the spatiotemporal data model.

The second way is more appropriate for spatiotemporal query. Often the grain and extent of the observation environment is implicitly given in a query. For example, “what happened to the terraced house in 2009?”, the extent of the observation environment is a spatial range in which the terraced house can be viewed at least in an overview, and the grain can be deduced from the display used for the visualization (Neudeck 2001) or by setting a minimum length which is still visible at the current spatial scale. The temporal extent is one year (the year 2009) and the temporal grain can be indicated implicitly or explicitly in the observation environment.

While the spatiotemporal extent may be indicated implicitly or explicitly in the query, the spatiotemporal grain is given or determined by the visualization system. By increasing the spatiotemporal extent, the number of the events that satisfy the condition in the query will be increased. As a result, some events should be aggregated, selected or typified according to the temporal and causal relationships among them. In other words, event generalization has to be conducted. By enlarging the temporal grain, some dynamic processes might become instantaneous and insignificant and should be therefore neglected in the dynamic visualization. By enlarging the spatial grain, some detailed geometries become invisible (lose their patterns) and the spatial generalization is therefore required.

As mentioned above, the event generalization can only be conducted when a spatiotemporal query is raised. Therefore, events will be generalized on the fly. In the contrast, the spatial generalization will be conducted in the pre-process, because it is triggered only with the increasing spatial grain. Moreover, it is relatively computing intensive, therefore, should be better conducted in a pre-process.

6.3 Event generalization

Similar to the news ranking in the media society, for event generalization in a 4D City environment the spatial scale as well as the temporal scale has to be defined at first. In the practice, the spatial scale is normally implied by the depth of the sub-tree whose root node is the referred object in the generalization. And the temporal scale is normally given as a period of time.

A common question which triggers the event generalization can be: “what happened to the object O_c during T ?”, whereby T stands for the time duration with the start point T_a and end point T_b .

Prior to the generalization, two types of events to the object O_c have to be retrieved from the event model:

- The "1st-level" events to O_c
- The "2nd-level" events to O_c

All these events should have occurred or be happening during the duration T. In case that the events were happening at the time point T_a or/and T_b, the changed part within the duration T can be derived.

After all the events to the object O_c are gathered, a ranking will be made according to their significances. In the specification of events to city objects, the following values can be used to evaluate their significances:

- Δ_m : the financial cost that the event led to
- Δ_f : the changed functional value (reduced or added) in the event
- Δ_s : the area of the geometrical change in the event
- Δ_v : the volume of the geometrical change in the event

These values can be used individually or in combination depending on the types of the events to be ranked. Normally, the financial cost Δ_m that the event led to is selected as the default value for ranking significance, since it can be used for describing events of both geometrical changes and semantic changes.

After the ranking, the generalization can be conducted by: (i) selecting a certain number of events or (ii) setting a threshold value for the significance.

Chapter 7

Spatial generalization

Spatial generalization is required when the generalized events (Chapter 6) and the states of city entities have to be visualized or represented with coarse spatial grain. It aims at reducing storage space, speeding up network transmission and geometric computation, hence improving rendering performance. With spatial generalization, objects at a coarser LoD are derived from those modeled at a finer LoD.

In this chapter, existing generalization algorithms for 3D buildings are analyzed, starting with a literature review of 3D generalization. Several LoDs are introduced by extending the LoD framework defined in CityGML. Accordingly, new algorithms are proposed to derive building models at different LoDs from the most detailed model. Finally, the implementation and some selected results are demonstrated.

7.1 Generalization for 3D buildings – the state of the art

In addition to the reasons in the context of this work, the necessity of generalization for 3D buildings can be found in many literatures (Thiemann 2002), (Meng & Forberg 2007), (Sester 2007) and (Glander & Döllner 2009). Since Staufenbiel (1973) proposed a rule-based approach for the simplification of 2D building ground plan, a number of algorithms have been made available for the generalization of building models. Early works were focused on developing techniques for generalizing 2D buildings (e.g. Lamy et al. 1999, Rainsford & Mackaness 2001, Regnauld 2001, Van Kreveld 2001). Using these techniques the amount of detail in the ground plan can be reduced by removing line segments underneath some minimum dimensions.

During recent years, a number of algorithms have been specified for the generalization of 3D building models. Kada (2002) presented an early approach for 3D generalization by extending Sester's approach (Sester 2000). He developed rules to remove too small structures in the 3D polyhedron at first; then the simplified building will be adjusted to its original shape. Lal and Meng (2001) defined some rules and constraints for 3D generalization. However, the generalization was restricted on one operation, namely, aggregation. Moreover, their work stayed at a conceptual level.

On the basis of the polyhedron segmentation proposed by Ribelles et al. (2001), Thiemann and Sester (2004) proposed an approach to decompose a building into many meaningful features that are stored as a cell complex and a CSG tree. The CSG tree presents a hierarchical subdivision which can be regarded as a necessary preparation for generalization (Sester 2007).

Kada (2006, 2007) proposed a similar approach. He defined parts of simplified buildings as intersections of half-planes and applied cell decomposition and primitive instancing. Moreover he segregated the original roof geometry into cells and activated a matching process with a pre-defined set of primitives instead of generalizing the roof structure (Kada 2007).

Meanwhile, some scholars tried to simplify 3D building structures 3D building models by elaborating operators of image processing and mathematical morphology. For example, opening, closing, erosion and dilation can be applied to eliminate small structures and smooth the outline of 3D buildings (Forberg 2004), (Mayer 2005) and (Forberg 2007).

The existing approaches for 3D generalization focus mainly on simplification of wall elements. Less attention has been given to the simplification of complicated roof structures. Moreover, the semantic information associated with geometrical objects of buildings is not considered. This could lead to elimination of some features which are important for visual impression or merge of features which belong to different objects. In order to make use of semantic information, the individual parts such as chimney, windows, and doors should be recognized. Thiemann and Sester (2005) interpreted object parts of a building with meaningful features using generic knowledge about the structure of buildings. Besides, Ripperda and Brenner (2006) proposed to use formal grammars to describe the structure of a building, especially facades.

CityGML is capable of representing 3D buildings at various degrees of complexity with respect to geometry as well as semantics (Gröger et al. 2008). In CityGML, even small components of a building such as windows, doors etc. are documented with high geometrical and semantic precision, where the semantic information indicates the meanings of the individual object components. More important, CityGML is implemented as an application schema of the extendible international standard for data exchange and encoding of 3D city models issued by the Open Geospatial Consortium (OGC).

These advantages of CityGML are important for generalization of 3D buildings (Fan et al. 2009; Fan and Meng, 2009). On this basis, a set of algorithms have been developed for deriving building models at different LoDs from the most detailed models.

7.2 The concept of LoD for 3D buildings

Level of detail, or LoD, is a technique used to improve the performance and quality for 3D object representation in computer graphics. It follows a simple fundamental rationale: when 3D scene is rendered, it is optically sufficient and computationally efficient to use a less detailed representation for small, distant, or unimportant portions of the scene (Luebke et al. 2003). James H. Clark introduced this concept for the first time in an issue of Communications of the ACM (Clark, 1976). Thereafter numerous frameworks have been proposed. Among others, discrete LoD, continuous LoD, view-dependent LoD and hierarchical LoD have been widely applied for various purposes (Heok and Daman, 2004), whereby the last three frameworks are tailored for run-time rendering, while the discrete LoD requires a preprocessing stage for creating individual LoD models. Their advantages and disadvantages make them useful for different applications.

For our purpose, we exploit the concept of LoD from virtual reality literature to represent the different level of knowledge. The LoD is used for storing a number of representations of a building, where the complexity of each representation is appropriate for some geospatial changes which can only be observed at a certain range of spatial scales.

Although LoD is uniformly defined as a number of milestones along the scale space when taking the scale space of 3D buildings as a linear continuum, there are no agreed LoDs for 3D buildings (Meng and Forberg, 2007) because LoD frameworks are normally established according to the spatial accuracy, the semantic precision, and the complexity of buildings required in different applications.

Lee and Nevatia (2003) defined the representation of 3D building models in three LoDs:

- LoD1 = structural information of building
- LoD2 = Façade texture information
- LoD3 = Detailed geometry of building façade

In the OGC standard CityGML, four LoDs are defined for 3D buildings (Gröger et al. 2008):

- LoD1 = Building in block model without any roof structures or textures.
- LoD2 = LoD1 enhanced with differentiated roof structures and textures.
- LoD3 = Architectural model with detailed wall and roof structures, balconies, bays and projections. High-resolution textures can be mapped onto these structures.
- LoD4 = LoD3 model completed with interior structures like rooms, interior doors, stairs, and furniture.

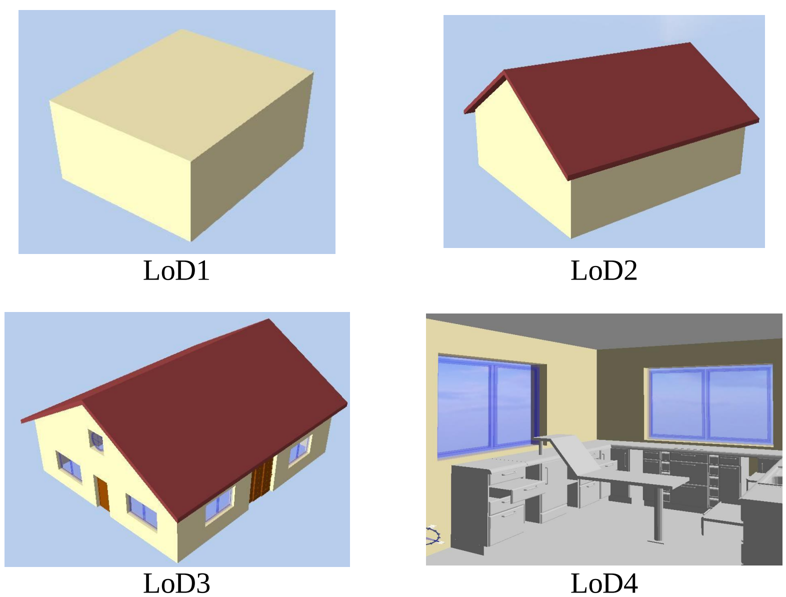


Fig. 7.1. The four levels of detail (LoD) defined by CityGML for building model (FZK-House modeled by Forschungszentrum Karlsruhe, Institute for Applied Computer Science), visualized in Autodesk LandXplorer CityGML Viewer

The LoDs in CityGML are characterised by differing accuracies and minimum dimensions of objects. As shown in Table 7.1, the positional and height accuracy of points at LoD1 must be 5m or less, while all objects with a footprint of at least 6m by 6m have to be considered. The positional and height accuracy of LoD2 has to be 2m or less. And at this LoD all objects with a footprint of at least 4m by 4m must be considered. In the detailed model at LoD3, both types of accuracies are 0.5m, and the minimum footprint is 2m by 2m. Finally, the positional and height accuracy of LoD4 must be 0.2m or less.

Table 7.1: LoD1-4 for buildings in CityGML with its accuracy requirements (source: Albert et al. 2003)

	LoD1	LoD2	LoD3	LoD4
Model scale description	City, region	City districts	Architectural models (out-side), landmark	Architectural models (interior)
Class of accuracy	low	middle	high	very high
Accuracy of position and height	5m	2m	0.5m	0.2m
Generalization	Object blocks as generalized features; >6*6m	Objects as generalized features; >4*4m	Object as real features; >2*2m	Constructive elements and openings are represented
Building installations	-	-	Representative exterior effects	Real object form
Roof form/structure	flat	Roof type and orientation	Real object form	Real object form

Originally, the LoDs in CityGML are defined according to varying requirements of applications on the one hand and the diversity of data sources and registration techniques on the other hand (Kolbe et al. 2009). Therefore, they might not include all representative milestones in the scale space of 3D buildings. For example, windows, doors, balcony etc. are vital architectural features which are important for visual impression. But they are removed during the transition from LoD3 to LoD2 model. It means the leap from LoD3 to LoD2 is too much and there should be at least some LoDs between them, so that a gentle transition along the scale space can be reached with respect to the visual impression of users.

In our work, two additional LoDs are introduced between LoD3 and LoD2 defined in CityGML. They are termed as sub LoDs or SLoDs:

- SLoD3 = exterior shell of the LoD3 model, whereby the opening objects like windows, doors as well as smaller façade objects are projected onto walls.
- SLoD2 = exterior shell after generalization of features on roof and walls.

Corresponding to the two SLoDs, the accuracies required in the SLoDs are defined, as shown in Table 7.2. The positional and height of points at SLoD3 must be less than one meter, while the minimal footprint of object is as large as required at LoD3, so that the exterior shell can give user almost the same visual impression as LoD3 does. At SLoD2, both types of accuracies are 1.5 meters. At the same time all objects with a footprint of at least 3m by 3m must be considered. Giving a building at a certain LoD, the models at less detailed LoDs can be derived by generalization procedures.

Table 7.2. SLoDs for 3D building model and their accuracy requirements

	Model scale description	Class of accuracy	Accuracy of position and height	Generalization	Building installations	Roof form/ structure
SLoD3	Landmark	High	1m	Object as real features; >2*2 m	Representative exterior effects	Real object form
SLoD2	On street view	High	1.5m	Objects as generalized features; >3*3m	Representative exterior effects	Roof shape form

The accuracies in position and height can be used as threshold of minimum length of a line segment during the process of generalization, since points below the corresponding accuracies at a certain LoD can not be differentiated any more. And the minimum size of the footprint can be used as threshold for eliminating/retraining isolated objects (polygonal features).

7.3 Deriving building models at different LoDs using generalization

In this section the algorithms of deriving 3D buildings at various coarser LoDs from their neighboring finer LoDs will be explained and described. The approach starts from the most detailed 3D building models at LoD4. The neighboring LoD3 can be obtained by transferring the entire geometries and semantics from LoD4 while neglecting the interior structures, since the LoD4 model is identical to that of LoD3 with respect to the exterior building shell (Gröger et al. 2008).

SLoD3 can be derived from LoD3 by extracting the exterior shell of the building model while keeping the semantics of the object components. Then SLoD2 model can be obtained by generalizing façade objects of SLoD3. A subsequent process is the transition from SLoD2 to LoD2. The obtained model can be further simplified and abstracted until the most abstract model LoD1 is reached.

7.3.1 Extraction of SLoD3 from LoD3

First of all, the walls in CityGML are converted into point clouds. For each wall its centroid $M_i = [M_x, M_y, M_z]_i$ is calculated and an adjusting plane F_i can be found by using the following equation:

$$F_i: \quad A_i x + B_i y + C_i z + D_i = 0 \tag{7-1}$$

where $\vec{n}_i = [A_i, B_i, C_i]$ is the normal vector of the plane F_i and D_i the closest distance of the plane to the origin of the coordinate system.

The average point $M = [X_m, Y_m, Z_m]$ of all centroids of the walls is treated as the centroid of the building. The point obtained in this way can guarantee that it lies more probably in the middle of the building than the point obtained by averaging all boundary points of walls.

For each polygon which belongs to the same wall, a plane F_{ij} can be computed by inputting the coordinates of all its vertices in the equation 7-1. Thereafter the angle θ_{ij} between the plan F_{ij} and F_i can be derived by

$$\theta_{ij} = \arccos\left(\frac{A_i A_{ij} + B_i B_{ij} + C_i C_{ij}}{\sqrt{A_i^2 + B_i^2 + C_i^2} \cdot \sqrt{A_{ij}^2 + B_{ij}^2 + C_{ij}^2}}\right) \tag{7-2}$$

If the angle θ_{ij} is close to 90°, the two planes are assumed to be orthogonal. The corresponding polygon should be deleted. If the angle θ_{ij} is close to 0° or 180° the two planes can be either coplanar or parallel. Therefore, the corresponding polygon should be preserved. After this process all the remaining polygons which belong to the same wall should be either coplanar or parallel. The distances from the centroid M to the planes of these polygons are calculated by

$$d_{m,ij} = \frac{\left| A_{ij} \cdot X_m + B_{ij} \cdot Y_m + C_{ij} \cdot Z_m \right|}{\sqrt{A_{ij}^2 + B_{ij}^2 + C_{ij}^2}} \tag{7-3}$$

The maximum value of the distances indicates the polygons which represent the exterior shell of the wall. The coefficients of the plane for these coplanar polygons $[A_i^{ex}, B_i^{ex}, C_i^{ex}, D_i^{ex}]$ will be used for the subsequent stage.

In order to preserve the spatial details of windows and doors within the wall, it is necessary to project them onto the exterior shell according to:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{projection} = \begin{pmatrix} x_{wd} \\ y_{wd} \\ z_{wd} \end{pmatrix} + \left(-D_i^{ex} + \begin{pmatrix} A_i^{ex} & B_i^{ex} & C_i^{ex} \end{pmatrix} \cdot \begin{pmatrix} x_{wd} \\ y_{wd} \\ z_{wd} \end{pmatrix} \right) \cdot \begin{pmatrix} A_i^{ex} \\ B_i^{ex} \\ C_i^{ex} \end{pmatrix} \tag{7-4}$$

where $\begin{pmatrix} x_{wd} & y_{wd} & z_{wd} \end{pmatrix}^T$ is a point of a window or a door, ex stands for the plane of exterior shell and i for the currently involved wall .

The polygons projected as lines on the exterior shell should be deleted because they lie perpendicular to the exterior shell. On the other hand if some polygons are identical to one another, only one of them should remain. As a result windows and doors are reduced to planes on the exterior shell.

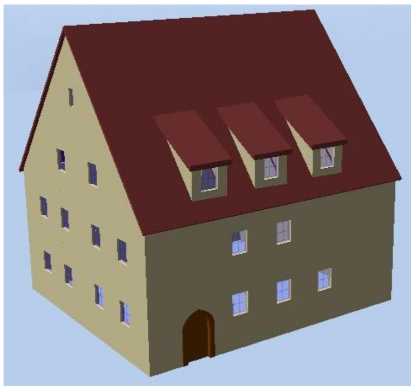
For the roof of the building there are two cases of modelling in CityGML:

(1) The roof is modelled as several *RoofSurfaces* and each *RoofSurface* is a polyhedron with at least six faces.

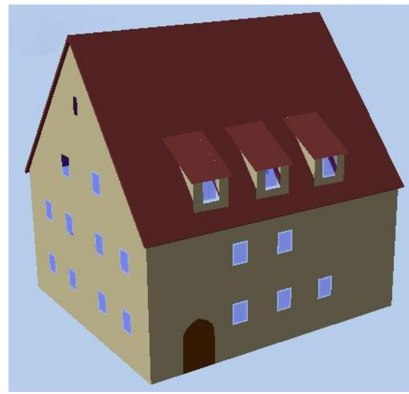
(2) The roof is modelled as one single *RoofSurface.*

In the first case the exterior shell of roof will be extracted using the same method as for the wall. In the second case the situation turns out relatively complex. Firstly, for each roof polygon its plane is computed by using equation (7-1). If the third coefficient of the plane is close to zero ($C \approx 0$), the polygon is orthogonal to the xy-plane, i.e. the ground plan. This polygon will be deleted. The remaining polygons will be classified into several clusters according to their orientations: the normal vectors of the planes for the polygons in the same cluster are the same and the absolute difference between their D values is smaller than 0.3 meter (since the thickness of roofs is normally less than 0.3m). In each cluster the distances from the centroid M to the planes of the polygons are computed by using the equation (7-3). The maximum distance indicates then the polygons on the exterior shell of roof.

Now the overall exterior shell of the building has been extracted. Figure 7.2 shows the original model (left) of a house in LoD3 and its exterior shell model (right). In comparison to the original model the exterior shell preserves almost all the details (features) of the house, but it needs only around 1/8 of polygons for the modelling.



Freihof in CityGML LoD3 with 2429 polygons
(from the city model of Ettenheim in Germany)



Exterior shell of Freihof
with 301 polygons

Fig. 7.2. An example house in LoD3 and its exterior shell

As a matter of fact, this kind of optical illusion could also be found in our real world. In order to conduct the renovation work for a building without influencing its exterior appearance, its facade is painted on a planar cloth which hangs in front of the building. As shown in Figure 7.3, the painted facade can be viewed as the exterior shell of the facade. It keeps all the spatial details, thus gives pedestrians a similar visual impression.

Fig. 7.3. Painted facades give pedestrians similar visual impression (the left building is located on the Neuhauserstr.6 and the right one in Kaufingerstr.26, Munich, Germany. The photos were taken on the 30th, November, 2008)

7.3.2 Transition from SLoD3 to SLoD2

In comparison to SLoD3, SLoD2 has lower geometrical accuracies and the smallest differentiable features at SLoD2 are larger than those at SLoD3. Therefore, they can be derived from their corresponding SLoD3 models by using generalization operations such as simplification, aggregation and typification for features on building facades and roofs.

Simplification: this operation is used to eliminate the sides of polygon that are smaller than the given threshold for line segments or eliminate isolated features whose sizes are smaller than the given threshold for polygonal features.

Aggregation: neighboring polygons with the distances between them smaller than the given threshold are merged to form one larger polygon. However, aggregation is only allowed when the involved polygons belong to the same object class.

Typification: typification denotes the process of replacing the originally large number of similar sized and shaped objects by a smaller number of uniform shaped objects.

Although typification is used in many literatures for generalization and various results of typification are presented in (Regnauld 2001; van Kreveld 2001; Thiemann 2002; Sester & Brenner 2004; Li 2007; Burghardt and Cecconi 2007 and etc.), it is not discussed whether and why their results are reasonable. In our early work (Fan et al. 2009; Jahnke et al. 2009) a user survey has been conducted, in order to find out what kind of representation after typification can be best associated to the original dataset with respect to visual impression. The results show that preserving the shape of the feature elements is the most important constraint for a reasonable typification process, which has also been verified quantitatively by calculating the similarities between the typified façades and the original façade using ARG and NEMD (Kim et al. 2010) algorithms (Mao et al. 2010).

On the base of our user survey, an algorithm is developed to generate perceivably reasonable representation for regularly distributed features on facades at a courser LoD (Fan et al. 2010). Since the most common regularly distributed features on facades are windows, the approach is proposed mainly for typifying windows. It includes several steps as follows:

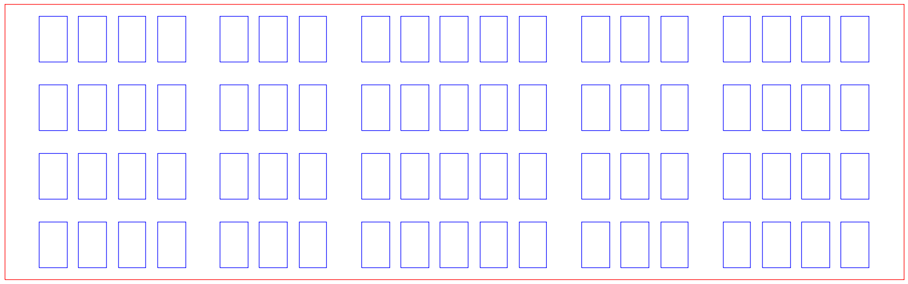
(1) Parameterize the distribution of the facade segment in which windows are distributed regularly.

(2) Establish an equation system according to the relations between the original distribution and the expected distribution of the typification in line with the result of our user survey.

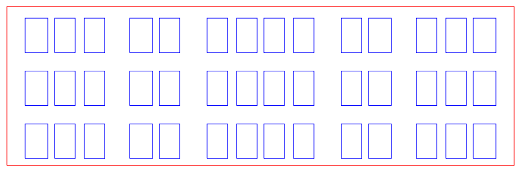
(3) Calculate the parameters of the expected distribution by recursively solving the equation system.

(4) Reconstruct the distribution of windows using the above calculated parameters.

The algorithm has been implemented and tested on a number of facades. The following example shows the result of typification for windows on a façade. Figure 7.4 shows a common façade with five segments of well-aligned and regularly distributed windows. In this case the whole façade had to be partitioned into several segments at first. Then the process of typification was carried out for every segment. At the same time the results of typification between neighbouring segments are aligned to reflect the original distribution characteristics. Figure 7.4b demonstrates the result of typification.



a. original facade



b. typified for scale reduction by 2

Fig.7.4. A facade before and after typification

Furthermore, the operation of typification can be used for generalization of roof structures, for example, uniform parallel hipped roof elements (Kada, 2007) or roofs constructed with terracotta tiles that are normally arranged in parallel strips. Furthermore, other uniform shaped and regularly distributed features like columns of landmarks can also be generalized using typification.

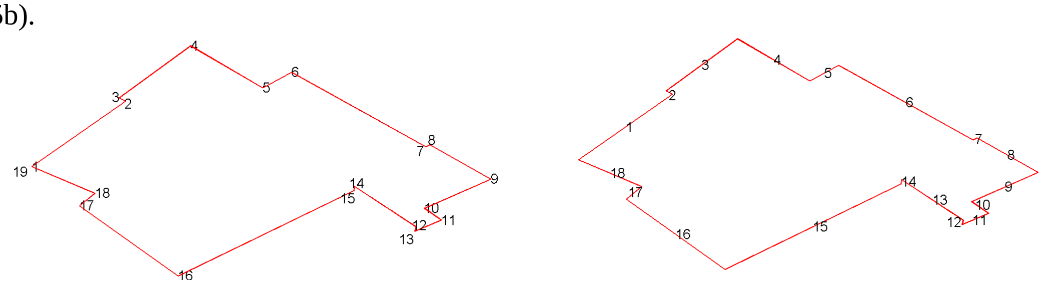
In comparison to SLoD3, SLoD2 obtained using the abovementioned operations contain a smaller number of objects, while their distribution patterns are well-preserved.

7.3.3 Transition from SLoD2 to LoD2

First of all, the opening objects at SLoD2 such as windows and doors have to be removed. Then LoD2 models can be derived from the remaining parts. The process is composed of three stages: (i) simplification of ground plan, (ii) generalization of roof structure and (iii) reconstruction of 3D building by combining simplified ground plan and roof structure.

Stage 1: Simplification of ground plan

The ground plan of a building can be derived by projecting the exterior shell model of the building on the ground and connecting the footprints into a closed polygon. The obtained polygon of the ground plan is represented as a list of points (Figure 7.5a). First of all, it has to be transformed into a list of line segments with an ID number and their start and end points (Figure 7.5b).



a b

Fig.7.5. Two types of representation of the ground plan: a. the polygon is represented as a list of points while b. the polygon is represented by a list of line segments

Concerning the simplification of ground plan, the main objective is the preservation of the characteristics (Sester 2005). Sester and her colleagues proposed effective algorithm for this issue (Sester 2000, Sester & Brenner 2004, Sester 2005). However, in their approach so far, only the rectangular structure is considered. In the reality, the structure of ground plan reveals a vast diversity and may contain many non-rectangular shapes. Even long narrow angles can appear in the structure of a ground plan. In this work, we developed a generic algorithm to handle these complicated shapes.

Before the simplification is conducted, the shortest side S_n of the ground plan must be identified. If S_n is smaller than the given threshold T_s just visible at a given scale, the simplification operation will be triggered and the two immediate neighbors S_{n-1} and S_{n+1} of S_n are checked in terms of the following two cases:

Case 1: S_{n-1} and S_{n+1} are parallel

In this case, their lengths are compared at first. If S_{n-1} is shorter than S_{n+1}, the side S_{n-2} should be then intersected with S_{n+1}, thus introduce a new vertex P_{in} which becomes the new end point of the side S_{n-2} and start point of S_{n+1}. At the same time, S_n and S_{n-1} have to be removed. Figure 7.6 illustrates the above described operation. Figure 7.6a gives a simple example for a part of a ground plan. The intersection point P_{in} between S_{n+1} and S_{n-2} is shown in Figure 7.6b, where the newly created line segment is highlighted in blue. And the red line segment will be removed. A bit more complicated ground plan and the corresponding simplification result are depicted in Figure 7.6c-d.

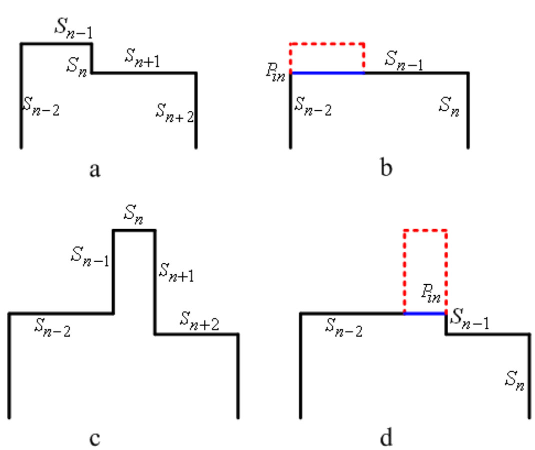


Fig.7.6. Different parts of a ground plan a) and c) and their simplification results b) and d) after the first iteration

If S_{n-1} is longer than the side S_{n+1}, the new vertex P_{in} is introduced by intersecting S_{n-1} with S_{n+2}, which becomes the new end point of S_{n-1} and start point of S_{n+2}. At the same time S_{n+1} and S_n have to be removed. Figure 7.7 shows the process with two different examples.

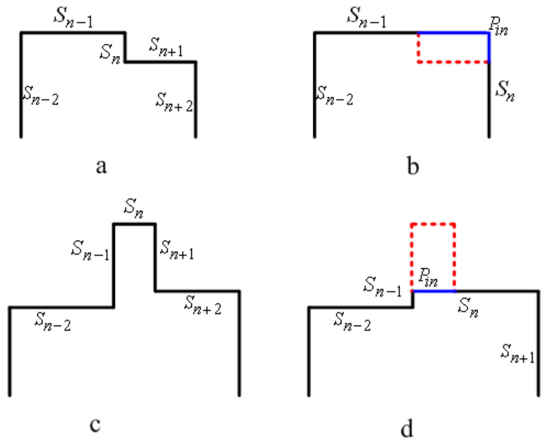


Fig.7.7. Different parts of a ground plan a) and c) and their simplification results b) and d) after the first iteration

Case 2: S_{n-1} and S_{n+1} are not parallel

In this case S_{n-1} and S_{n+1} must be intersected at P_{in}. The operation to be deployed depends on the topology of P_{in} in relation to S_{n-1} and S_{n+1}:

- If P_{in} lies on S_{n-1} or S_{n+1}, it becomes the new end point of S_{n-1} and start point of S_{n+1}. S_n has to be removed (see Figure 7.8a -d).

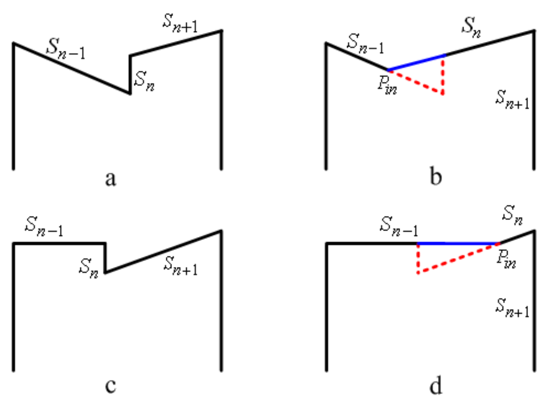


Fig.7.8. The shortest side is removed by intersecting its flanking neighbors

If P_{in} lies neither on S_{n-1} nor on S_{n+1}, the resulted angle α_n with the vertex P_{in} has to be compared with a threshold w_s .

- In the case of $\alpha_n > w_s$, the intersection point P_{in} is the new end point of S_{n-1} and start point of S_{n+1} (Figure 5a and 5b). S_n should be removed. (see also Figure 7.9)

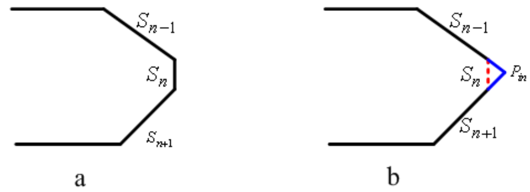


Fig.7.9. Wide angle as the simplification result

- In the case of $\alpha_n < w_s$, the side S_n will be shifted inward in the polygon while keeping the new line segment should be parallel to the original side as shown in Figure 7.10. Let the intersection of the new line with side S_{n-1} and S_{n+1} be P_s and P_e. The shifting will terminate when the new side is longer than the given minimum length T_s. However, sometimes, the new side is shorter than T_s even when one of its endpoints meets the start point of S_{n-1} or the end point of S_{n+1}. In this case, the other endpoint will be moved until the new side is longer than T_s or it meets the endpoint of the side S_{n+1} (or S_{n-1}). The flow chart in Figure 7.11 embraces all cases.

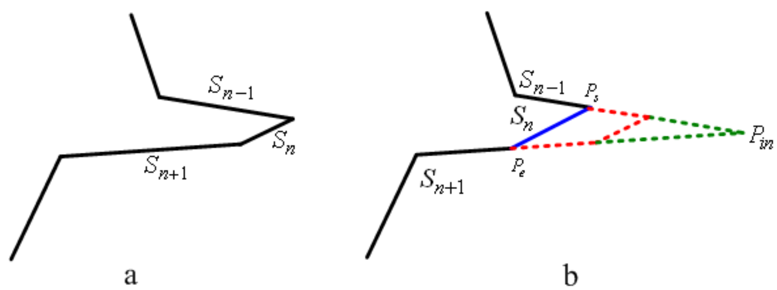


Fig.7.10. Removal of a sharp angle caused by removal of a short side

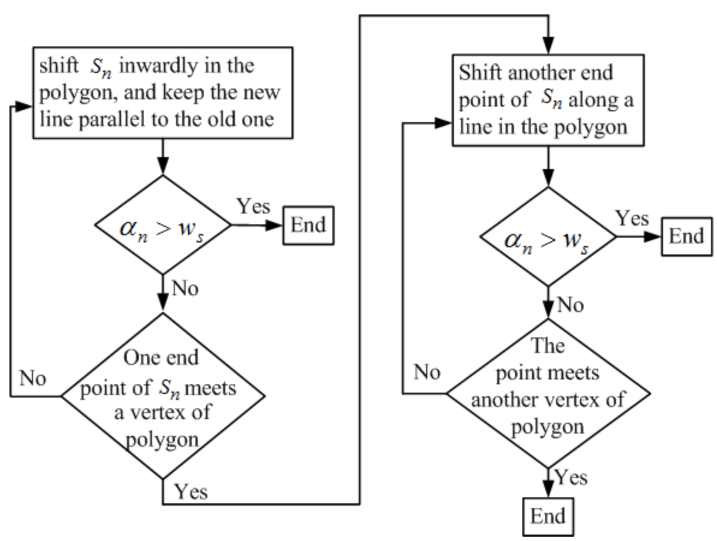


Fig.7.11 Flow chart for handling sharp angles

As mentioned above, the process of eliminating a corner is triggered by an angle threshold w_s. This threshold can be derived from the given minimum length T_s at a certain spatial scale as following:

Assume that there is a corner as a part of a ground plan which is, for instance, highlighted with blue color in Figure 7.12. The corner is composed of three sides: s_{11} , s and s_{12}. Whereby the side s starts at point p_s, ends at point p_e ; and its length d is equal the given minimum length T_s exactly ($d = T_s$). Extend both sides of the corner and let them intersect at point p_1. An angle α_1 is created at the same time. The distance between the intersection point and the side s is d_1.

If the intersection point is moved along the perpendicular bisector of side s outwardly while keeping p_s and p_e on the other two sides of the corner, a series of angles can be obtained. According to our everyday experience, the two extended parts (dash lines) can still be differentiated in the context of their original parts, as long as d_1 is shorter than T_s ($d_1 < T_s$). Therefore, the resulted angle at the point where the distance is equal to T_s can be treated as the angle threshold w_s. In Figure 7.12, the green lines indicate the case where the intersection angle is equal to w_s ($w_s = \alpha_2$ and $d_2 = d$). In this case the two dashed lines in green are easily distinctive. Further shifting the intersection point outwardly may cause the visual merging of the extended lines as indicated by red dash lines.

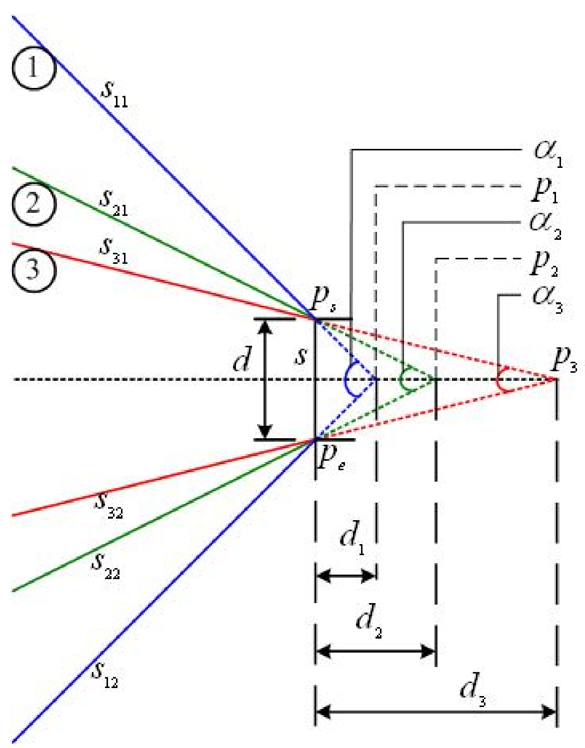


Fig.7.12. Derivation of the angle threshold according to the visual perception

The angle threshold w_s for the above case can be deduced as:

$$\begin{cases} w_s = \arctan\left(\dfrac{d/2}{T_s}\right) \\ d = T_s \end{cases} \Rightarrow w_s = 53°$$

After the operation for the current shortest side of the ground plan the polygon will be newly arranged for the next iteration will be conducted. The simplification process will terminate when all sides are larger than the given threshold.

Stage 2: Generalization of roof structure

Unlike the approach of Kada (2007), we prefer to conduct the generalization directly on the roof geometry because many roofs fall beyond the eight types of roof. On the other hand, many roofs in Western Europe are composed of several polygons with different normal vectors and slopes. Firstly, these polygons will be distorted when being projected onto the ground. Their sides as well as angles might be changed disproportionately. Some sides which are longer than the given threshold become shorter or even much shorter than the threshold after the projection. As a result, they will be removed. In extreme cases, some objects that are larger than threshold but steeply oriented might be eliminated completely. Secondly, the angle between two neighboring polygons can not be measured when they are projected onto the ground plan. Thirdly, the process of reconstruction (re-projection) of roof polygons is required after the generalization, in case that the roof polygons are projected onto the ground plan prior to the generalization. This requires additional computation. On this account, roof structures will be generalized without projecting onto 2D plan in case that the roof is not flat. Instead, all the calculations during the generalization process are conducted in 3D environment.

The process is composed of three steps: eliminating smaller roof features, simplifying roof polygons, and merging the simplified polygons. In the first step, the minimum bounding rectangle (MBR) of every polygon of the roof structure is calculated. If both sides of the MBR are smaller than the given threshold, its corresponding polygon has to be eliminated. In this way, smaller polygons that represent features, for instance, dormers are removed. The remaining polygons will be simplified individually in the second step, following the similar working principle for the ground plan. But the calculations are somehow complicated than those for simplifying ground plan, since they have to be done in 3D space. For example,

- Judging whether two lines are parallel

  Assume that there are two sides of a polygon with their end points $P_1(x_{p1}, y_{p1}, z_{p1})$, $P_2(x_{p2}, y_{p2}, z_{p2})$ and $Q_1(x_{q1}, y_{q1}, z_{q1})$, $Q_2(x_{q2}, y_{q2}, z_{q2})$ respectively. Let L_p be the line determined by P_1, P_2 and L_q be the line determined by Q_1, Q_2. Then these two lines can be expressed parametrically as: $L_p(t) = P_1 + \mathbf{V_p} \cdot t$ and $L_q(s) = Q_1 + \mathbf{V_q} \cdot s$, whereby $\mathbf{V_k}$ ($k = p, q.$) is the normalized (unit) direction vector of each line. After Goldman (1990), if there is $\left|\mathbf{V_p} \times \mathbf{V_p}\right| = 0$, the two lines are parallel. Otherwise, they intersect.

- Calculating the intersection point of two lines

  The intersection occurs when $L_p(t) = L_q(s)$. The solution gives the intersection point as: $t = \frac{Det\{(Q_1 - P_1), \mathbf{V_q}, \mathbf{V_p} \times \mathbf{V_q}\}}{|\mathbf{V_p} \times \mathbf{V_q}|^2}$ or $s = \frac{Det\{(Q_1 - P_1), \mathbf{V_p}, \mathbf{V_p} \times \mathbf{V_q}\}}{|\mathbf{V_p} \times \mathbf{V_q}|^2}$.

In comparison to the abovementioned calculation, the calculation on 2D plan is relatively easy: (i) write the two equations for the lines into a slope-intercept form; (ii) the two lines are parallel, if their slopes are same; (iii) if they are not parallel, there intersection point can be calculated by solving the equation system composed of the two equations for the lines.

After the simplification of the polygons of the roof structure, the obtained polygons and their neighbors can be merged with each other when they are coplanar or quasi coplanar (their intersecting angle is almost flat). The resulted polygons should be simplified using the same algorithm as that in the second step. Figure 7.13 shows an example of the abovementioned process.

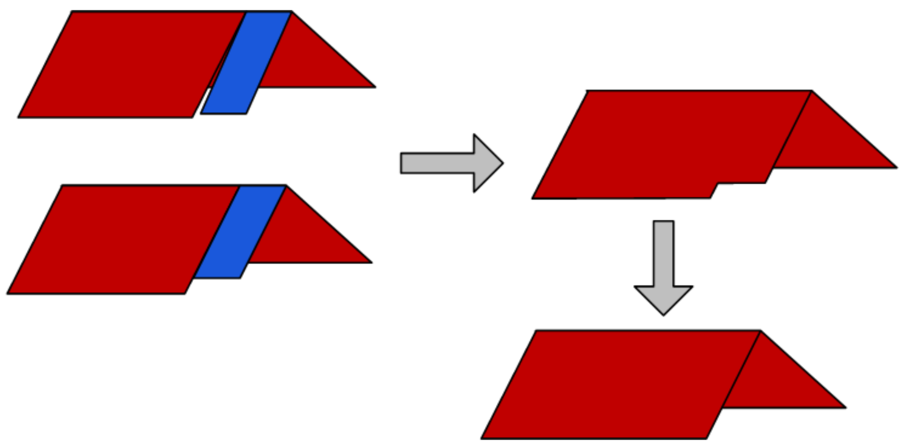


Fig.7.13. The simplified roof polygons can be merged with each other and then simplified again

It should be noted our current approach has not considered more complicated or elaborated roof structures such as those of an Asian temple or layered roof of a pagoda composed of a huge number of small polygons. For this special case, the generalization operators i.e. aggregation and typification should be deployed before the polygons are individually simplified.

Stage 3: Reconstruction of 3D building

After the simplification of ground plan and generalization of roof structure, walls will be derived from the new polygon of simplified ground plan. Each wall can be regarded as a polygon with four vertices and its heights will be increased until they touch the roof polygons. The polygon of a new wall can be acquired by removing the line segments above the intersection with roof. The procedure works as following:

1. Two intersecting points of a wall with roof polygons can be obtained by intersecting the two vertical line segments with the planes of roof polygons. In fact, a vertical line segment is intersected with all planes of roof polygons. If the intersection point on a plane is higher than all points of the corresponding roof polygon, it will be neglected. The

remaining points will be used to calculate their distance to the corresponding polygons. The minimum distance indicates the right intersection point on the roof.

Note: Normally, the obtained intersection point lies inside of the roof polygon or on one side of the polygon. But sometimes it can be located outside of the polygon, because the wall and roof structure are separately simplified. In this case the point is called near the polygon in order to differ from the relationship between this point and other polygons.

2. The calculation of the intersection line is dependent on the topological relationships between the intersection points and roof polygons:

    a. If the two points are inside of one roof polygon, they are obviously the endpoints of the intersecting line segments (Figure 7.14a).

    b. If one of the intersection point is located inside of a roof polygon and the other is near the polygon, or both points are near to the polygon, the polygon side which is closest to the two intersection points will be shifted until both of them are surrounded (including the case that a point on one side of the polygon) by the polygon (Figure 7.14b).

    c. If the two points are located in two neighboring roof polygons, the shared (overlapped) line segment must intersect with the wall plane. The three intersection points form two intersection lines on the roof (Figure 7.14c).

    d. If one point is near a polygon and the other is inside or near a neighboring polygon, two intersection lines on the roof will be detected according to case c. Then there are two cases of b or one case a and one case b. (Figure 7.14d).

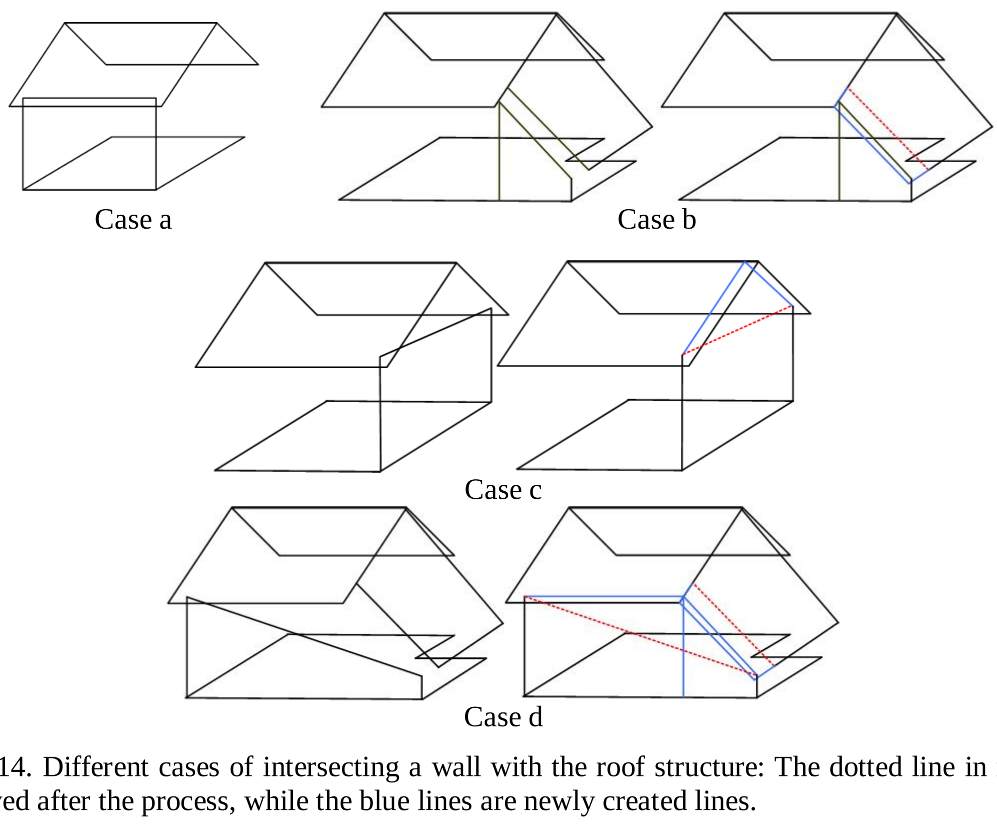


Fig.7.14. Different cases of intersecting a wall with the roof structure: The dotted line in red are removed after the process, while the blue lines are newly created lines.

Note: the topologies between the intersection points and roof polygons in case b may cause gap or hole when reconstructing the 3D building, therefore the involved roof polygon has to be extended in order to cover the ground plan completely.

After all the intersection lines between walls and roof structure are calculated, both the walls and the roof structure have been changed and they can be well adapted to each other. Finally, a simplified 3D building is completely reconstructed. And the models at LoD2 are obtained.

7.3.4 Transition from LoD2 to LoD1

At the coarsest LoD1 in CityGML, different structural entities of a building are aggregated to simple blocks and not differentiated in detail. Therefore, the entities modelled for *outerBuildingInstallation* like chimneys, stairs, antennas, balconies or attached roofs above stairs will be eliminated at first. Then the derivation of LoD1 model from LoD2 model can actually be regarded as a process of aggregation and simplification for the building geometries at LoD2.

Similar to the second step of the transition from SLoD2 to LoD2, the wall elements have to be projected onto ground prior to the aggregation and simplification. However, the threshold for triggering the simplification of ground plan should be larger than that in the case of deriving LoD2 from SLoD2 (see Table 5.1).

Since LoD1 is the well-known block model without any roof structures, the remaining walls after the simplification of ground plan will be covered with a flat plane. In the simplest case all walls are equally high. The flat roof plane might be identical with the ground plan, however with the same height as the walls. Otherwise, the height of the highest wall (or the height of the boundary box) will be taken for the flat roof plane. The lower walls will be stretched to the height of the flat roof plane.

Finally, all the remaining polygons will be modelled as one solid (or *MultiSurface*) at LoD1, resulting a complete block model.

7.4 Summary

This chapter presents algorithms for deriving 3D building models at different LoDs defined in CityGML. In comparison to the LoDs for buildings, two sub-LoDs are introduced between LoD3 and LoD2 models in CityGML. A series of algorithms is developed. The approach starts from the most detailed 3D building models at LoD4 which is progressively transferred to coarser building models.

The most remarkable steps in the generalization process are: (i) deriving SLoD3 building models from the LoD3 models and (ii) deriving LoD2 building models from the SLoD2 models. The former aims to extract the exterior shell of buildings at LoD3 where each building component is typically represented by a cuboid. In an exterior shell, however, the cuboid will be reduced to a single plane. For this reason, SLoD3 can reduce more than 80% of the storage space of 3D buildings without have to influence the visual impression. The latter is a comprehensive simplification of 3D building models. Unlike the previous approaches, the wall structure and the roof structure are separately simplified in this work. The algorithm for the ground plan simplification not only considers the buildings' characteristics such as rectangles, collinearity and parallelity, but also the characteristic of non-rectangular turning of walls.

The algorithms are specified for building models defined in CityGML. CityGML not only represents the shape and graphical appearance of 3D buildings but specifically addresses the object semantics and the representation of thematic properties, taxonomies and aggregations. The presented approach takes this advantage into account. In other words, the process of the generalization and simplification deals with the 3D geometries of buildings while considering the corresponding semantic information.

Chapter 8

Implementation and experimental results

As indicated in Chapter 4, the spatiotemporal data model of a city is composed of an event model and a time-stamped city model. For the management and storage of city entities CityGML is a preferred standard model due to its capability of representing 3D city objects with rich semantics and geometries. In order to embed the event-induced changes of city entities as well as their histories in CityGML, an extention is necessary, which results in the CityGML ADE (Application Domain Extension).

Considering the scale-dependencies of geospatial phenomenon, the geospatial entities and their event-induced changes have to be generalized with increasing spatiotemporal scales. In our approach, events are generalized on the fly according to the algorithms presented in Chapter 5, while the 3D buildings are generalized in a pre-process.

This chapter addresses the extension of CityGML for spatiotemporal city model and implementation of 3D generalization. Moreover, a prototype of the software environment for interactive visualization of dynamic 3D buildings is presented.

8.1 Extension of CityGML for spatiotemporal city model

CityGML generally provides two different ways of extension (Kolbe, 2008). One is to embed the CityGML objects into a broader application framework and establish a connection between application data and CityGML within the framework. The other is to incorporate application-specific information into the CityGML instance documents (Gröger et al. 2008). This concept is termed as Application Domain Extensions (ADE). The difference between these two extension concepts is that using ADE additional attributes (properties) will be introduced only for existing CityGML feature types, while using the first concept new feature types and attributes can be defined. In other words, the second concept is application-specific and introduces extra properties for objects like buildings. In our case, the first concept is deployed for the event model because events are neither application nor properties of objects in conventional sense. They happen to objects like buildings and have to be modeled as new features in CityGML.

In the extension of CityGML, events are modeled similarly to generic objects. According to the definition of OO (Object-Oriented) event in Chapter 4, an event has the attributes *class, mode, location, eventdate,* and *entityID*. *class* allows an event classification within the thematic area such as construction of a new building, change of ownership etc. (see 3.3). *mode* describes to how the event happened (see 4.1.2). *location* indicates where the event happens. *eventdate* records when the event happens. And *entityID* is used as the connection with the spatial city model. It answers the question which entities are involved in the event.

Non-decomposable events happened to a building such as “change of ownership”, “change of function”, “change of usage” etc., are connected with the spatial entities so that they are the 1st-level events in the data model. There is no requirement of the LoD modeling of the building, because this kind of event has nothing to do with the geometries of the building.

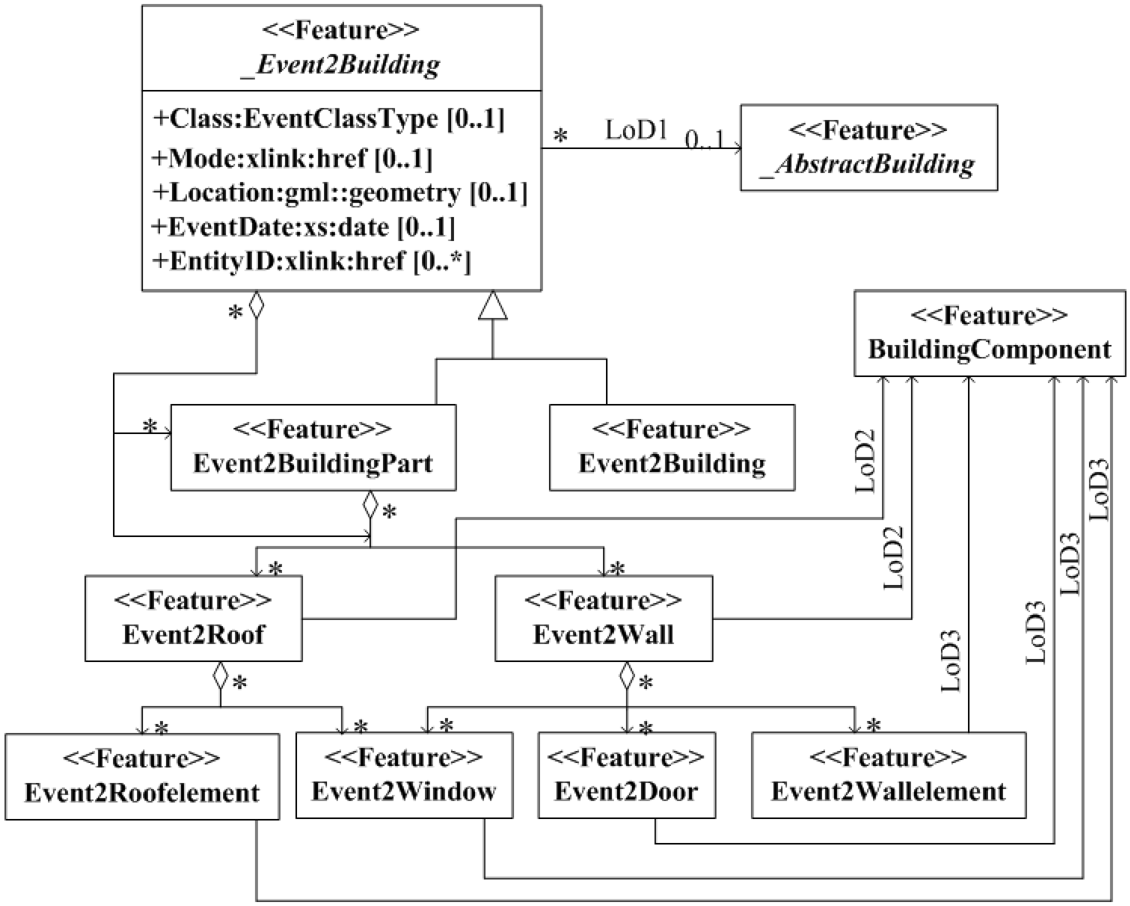


Fig. 8.1. UML diagram of Event2Building model of CityGML

A decomposable event can be decomposed along the semantic structure of the corresponding spatial entity for representing a causal and dynamic process. In the spatiotemporal domain, such an event can be represented at different LoDs. The UML diagram in Figure 8.2 depicts an event complex to a building, which consists of a number of distinct events happened to the components of the building respectively. All the classes inherit the attributes of *_Event* (Figure 8.1). The event model is successively refined and, therefore, can be represented at different LoDs. In the current approach, a decomposable event happened to a building can be represented at three LoDs corresponding to those in the spatial data model. At LoD1, the event is treated as the 1st-level event of the building. At LoD2, the event is divided into smaller events which happened to roofs and walls respectively: Event2Roof and Event2Wall. At LoD3, the event is represented by the changes of the smaller components of the building such as windows, doors, and other building elements.

In contrast to the event model, in a time-stamped city model new attributes such as *predecessor*, *successor* and *lifespan* of buildings, as well as of their components should be introduced (see Figure 4.7, Chapter 4). In this case, the second concept is adopted. Within the framework of our project, the extension[9] is conducted according to the ADE mechanism illustrated in (Gröger et al. 2008).

8.2 Preparation of spatial data at different LoDs using 3D generalization

The generalization algorithms for 3D buildings have been implemented using Matlab (version Matlab 7.4). The platform is a PC with Inter(R) 3.33GHz Xeon(R) CPU, 4.00GB RAM (3.49GB usable), and Microsoft Windows 7 Professional x86 (32bit).

8.2.1 The implementation of deriving LoD2 models from SLoD2 models

The program of deriving LoD2 building models from SLoD2 models consists of six modules including the input and output as illustrated in Figure 8.2. The simplified exterior shell (SLoD2) of a 3D building imported from CityGML is divided into two parts: roof structures and wall elements. They are separately simplified at first. Then the 3D building is reconstructed by intersecting wall elements with roof structures. Finally, the simplified building is exported to CityGML.

The algorithm for simplifying building ground plan has been tested for large dataset of city Munich with 64698 ground plans. The computing time for the whole dataset varies depending on the defined threshold. Table 1 lists the thresholds (in meter) in our test and their corresponding computing time (in second). The average computation time for a single ground plan is about 0.006 second.

Table 8.1. Thresholds and the required computation time of simplifying ground plans

Threshold	1m	2m	3m	5m	10m	15m	20m
Computing time	336.59 s	361.33 s	367.31 s	373.83 s	395.67 s	392.04 s	389.82 s

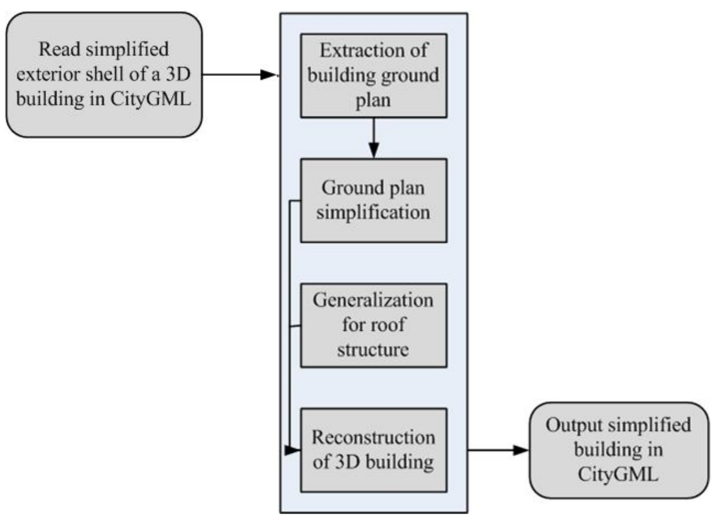


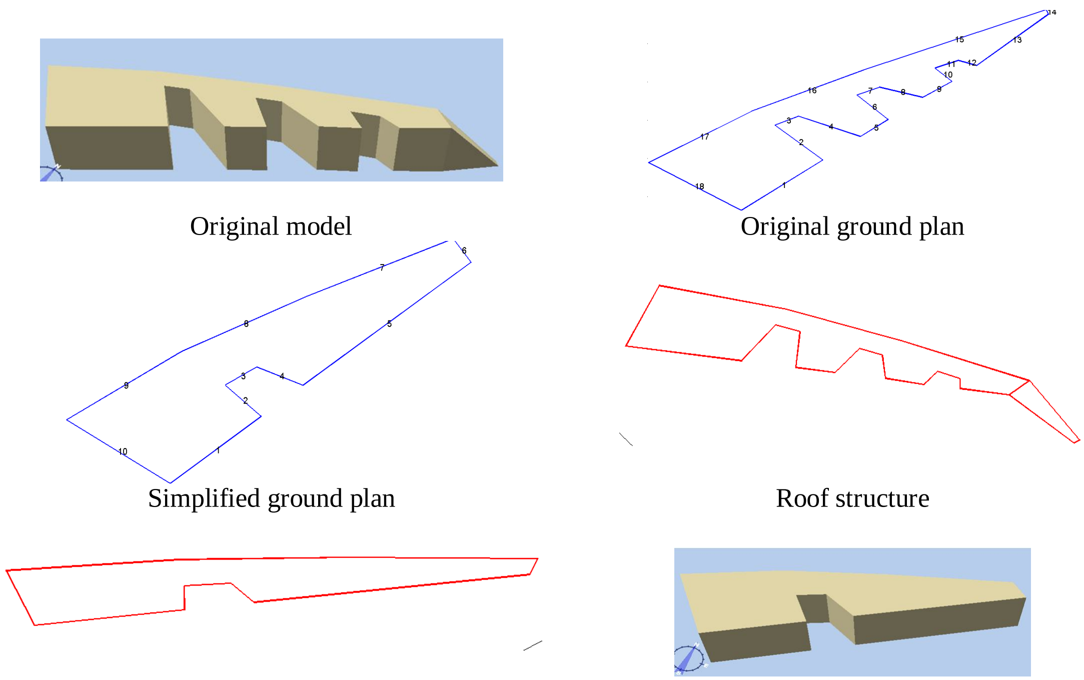
Fig. 8.2. Workflow and Matlab modules for deriving building models in LoD2 from SLoD2

[9] The detail of the extension is attached in the annex behind this thesis. The CityGML *temporal* ADE is defined within the XML Schema definition file temporal_ADE.xsd. On the other hand, the CityGML *Event* module is defined within the XML Schema definition file *event.xsd*.

On the other hand, the algorithm has been tested on some buildings from the city model of Ettenheim in Germany (free data in http://www.citygml.org/1539/). Moreover, a building in Copenhagen is used as a special example with complicated corners for the ground plan simplification. Further, a church in Berlin is used to demonstrate how a steep roof will be retained when the generalization of roof structures is conducted in 3D space. Our tests so far show that the computing time for a single building is about 0.08 second to 0.3 second depending on how complex the building is modeled. In next section, some experimental results will be presented.

8.2.2 The experimental results of deriving LoD2 models from SLoD2 models

In the first example, simplification of a flat-roof building is represented (see Figure 8.3). There are three trapezoid shapes on the long axis of the building and a bevel body as the extension of the building. In the process of the simplification of ground plan, long narrow angle will appear when extending the two sides of the trapezoid. Using our algorithm (see section 7.3.3), the short side (base) of the trapezoid is shifted outwards until it becomes larger than the given threshold. Figure 8.3 shows that two of the trapezoids are removed after the simplification while the trapezoid at the end of the building is enlarged. The too small bevel side in the roof structure is removed. The remaining part of the roof structure reveals sufficient similarity to the ground plan.



Original model | Original ground plan

Simplified ground plan | Roof structure

Simplified roof structure | Simplified model

Fig.8.3. A flat-roof building with several corners which can result long narrow angles. The building serves for Dansk Folkemindesamling (Danish Folklore Archives) and is located in Christians Brygge 8, Copenhagen, Denmark.

In the following, five examples are shown for the buildings with non-flat roofs that are quite common in Western Europe. For every example eight figures are used, corresponding to the three stages of the simplification process.

a. The original building model is presented by LandXPlorer CityGML Viewer 2009.
b. The ground plan is derived from the exterior shell of the 3D building model
c. The ground plan is simplified
d. The roof structure is the exterior shell of the roof
e. The roof structure is generalized in the 3D space
f. The generalized walls are increased in height
g. The walls are intersected with roof. Part of the roof is adapted to the structure of walls.
h. The simplified geometries are modeled by CityGML (or by other model languages) and shown in LandXPlorer CityGML Viewer 2009.

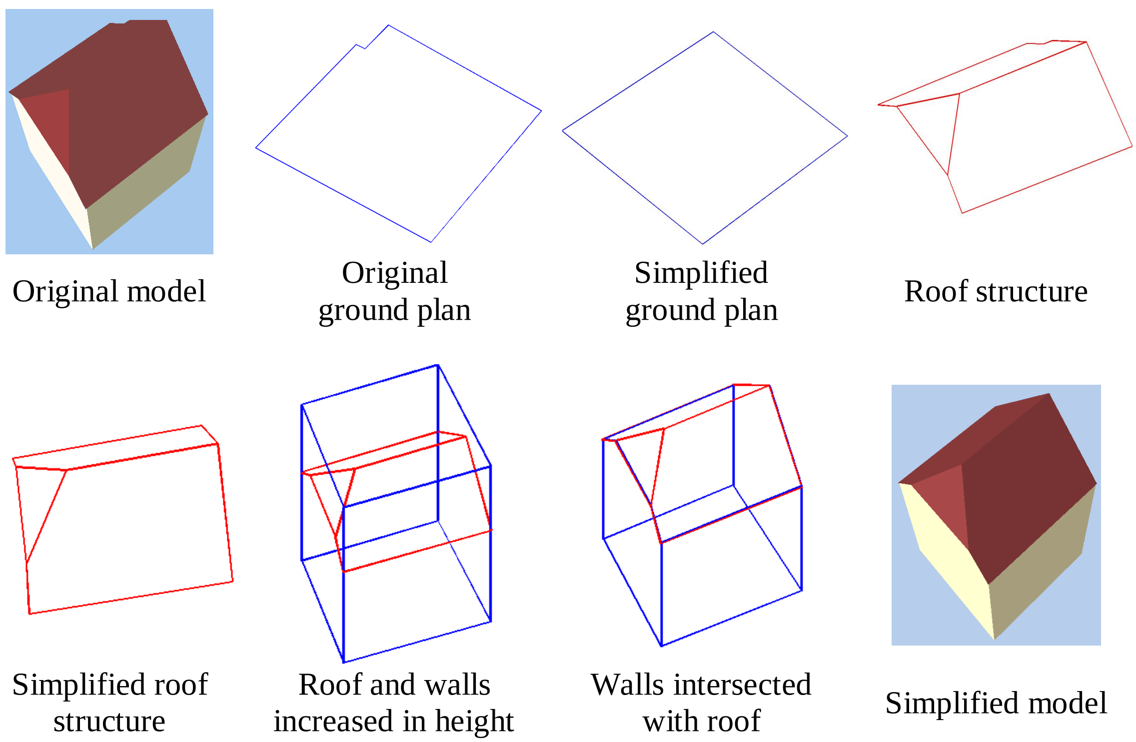


Fig.8.4. A building with a large asymmetric hipped roof which is preserved after the simplification

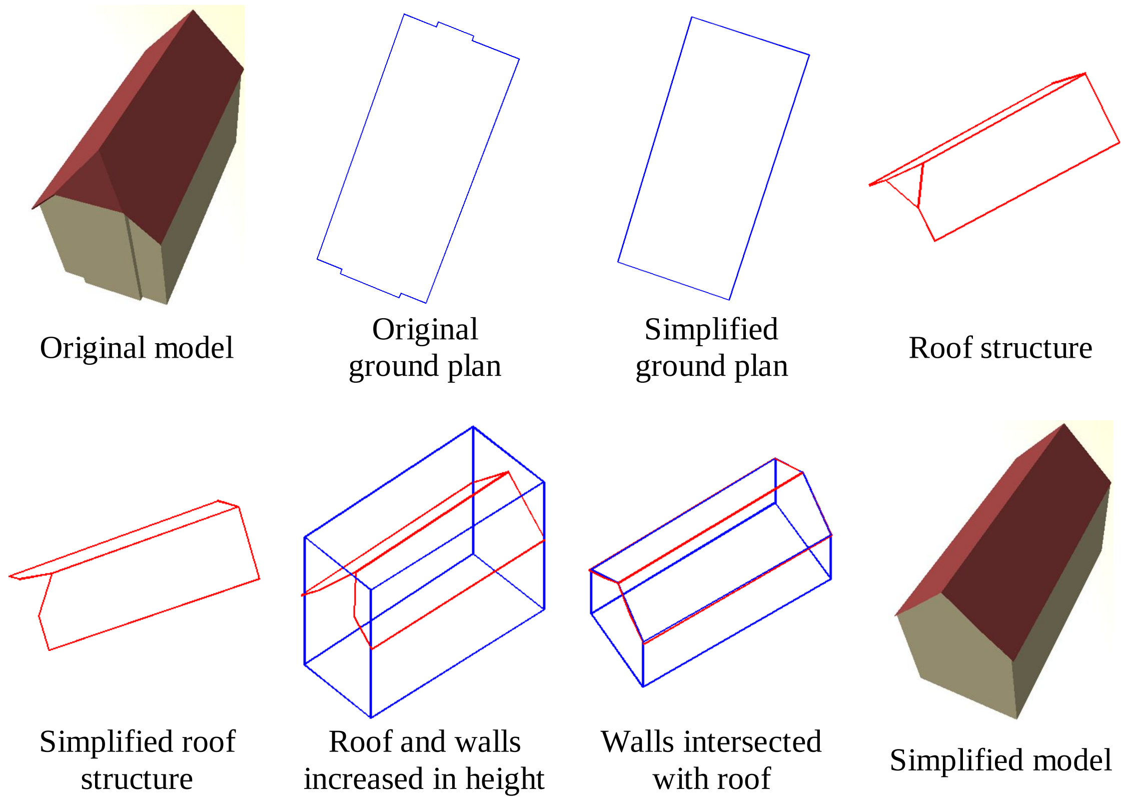


Fig.8.5. A simple building with a small asymmetric hipped roof which is removed after the simplification

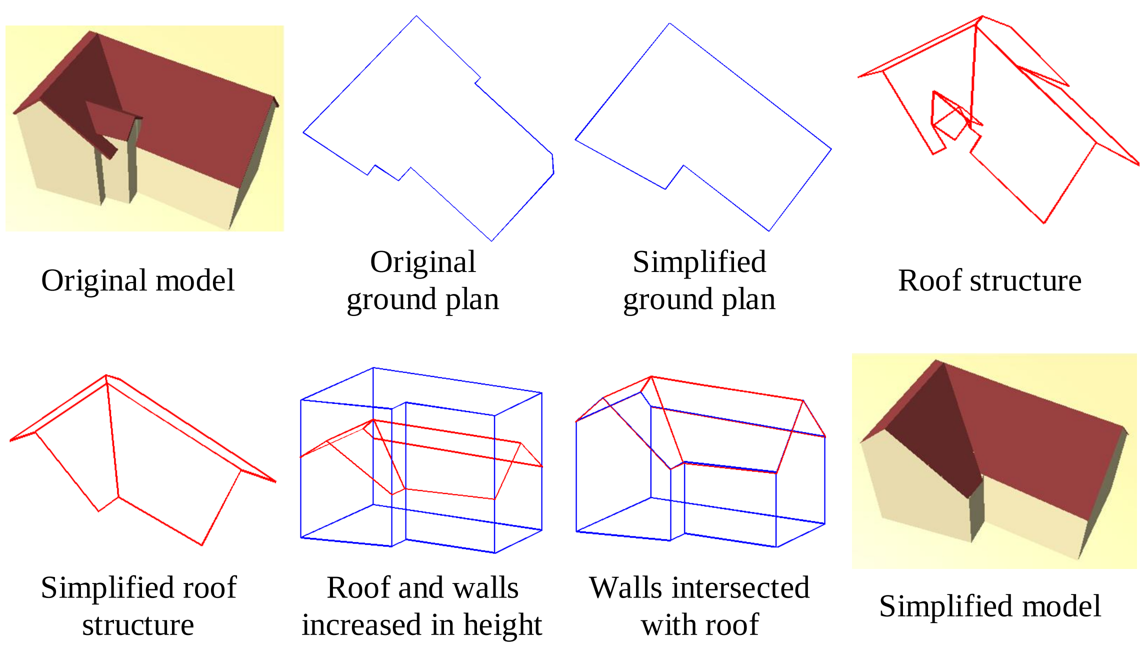


Fig.8.6. A complex building with a corner roof and an additional T-element which is removed after the simplification

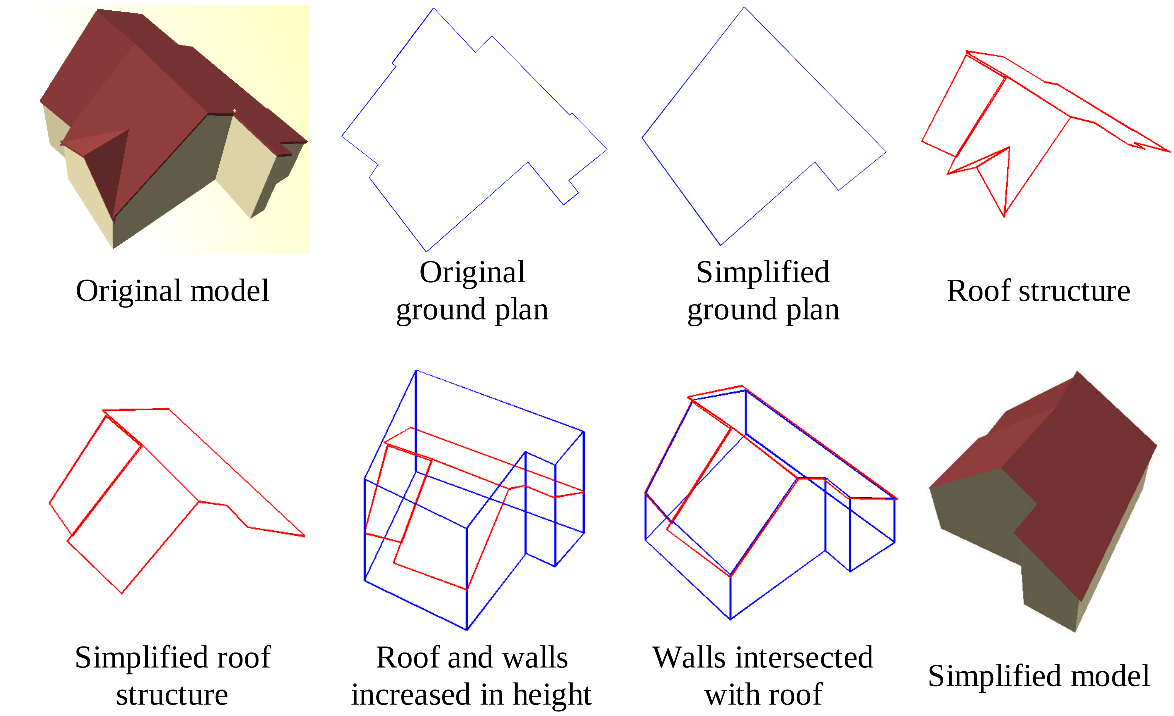


Fig.8.7. A building with a complicated ground plan and a small T-element which is removed after the simplification

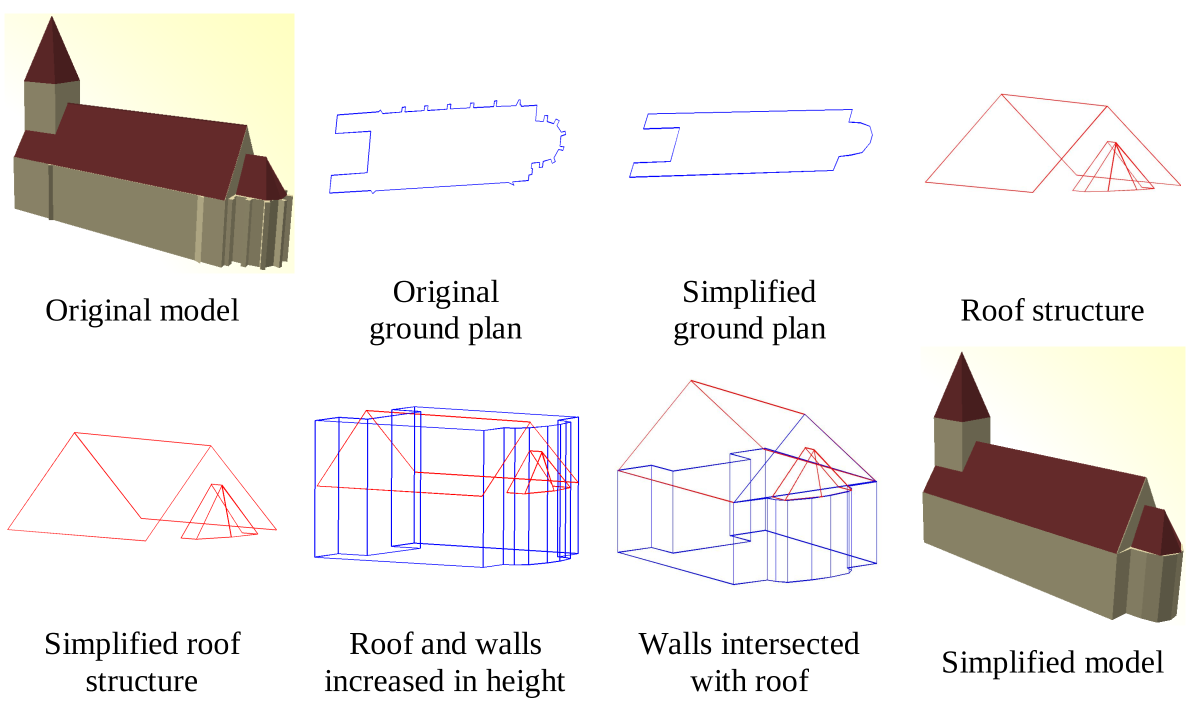


Fig.8.9. A building with a complicated ground plan and steep roof elements which is retained during the simplification

All these above-mentioned examples show that the ground plans and roof structures are simplified to a great extent while their basic shapes are preserved. Besides, an additional test for assessing the quality of polygon simplification has been conducted using the shape similarity measure (Arlin et al. 1991, Lee et al. 2003).

8.2.3 Quality assessment for ground plan simplification

Traditionally, there are two ways to represent a closed polygon: (i) by giving a list of vertices or (ii) by giving a list of line segments. Alternatively, a polygon can be represented using a list of angle-length pairs, whereby the angle at a vertex is accumulated tangent angle at this point while the corresponding length is the normalized accumulated length of the polygon sides up to this point. Let C be the polygon on the left of Figure 8.10. The tangent angle at a vertex is $\theta_1 = \varphi_1$. Then $\theta_{i,(i>1)}$ can be calculated as $\theta_i = \theta_{i-1} + \varphi_i$. The right of Figure 8.10 shows the change of tangent angles (y-axis) along the normalized accumulated length of the polygon sides (x-axis). From this point of view, the tangent angle can be treated as a function of the normalized accumulated length $T_C(l)$. It can be called tangent function or turning function (Arkin et al., 1991).

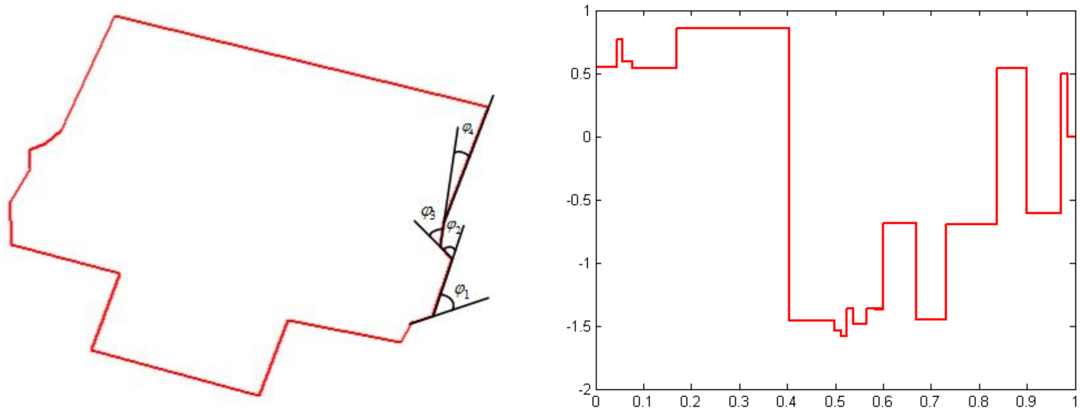


Fig.8.10 Illustration of tangent space representation of polygon

The function $T_C(l)$ measures the angle of the counter-clockwise tangent as a function of the normalized accumulated length l. The cumulative angle increases with left hand turns and decreases with right hand turns. This kind of representation is invariant to rotation, because it contains no orientation information. Furthermore, it is invariant to scaling, since the normalized length makes it independent to the polygon size.

Similarity measures can be derived based on the $L_2 - norm$ of the shape features. In this paper, the similarity of two polygons is defined as the distance between their tangent functions.

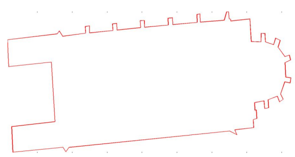
$$d(A,B) = \left\| T_A - T_B \right\|_2 = \left(\int_0^1 (T_A(s) - T_B(s))^2 ds \right)^{\frac{1}{2}} \quad (8\text{-}1)$$

As abovementioned, the tangent function is invariant to rotation and scaling. In other words, the rotation and scaling are not considered in Equation 8.1. However, for the quality assessment in ground plan simplification, whether the size can be preserved is an important factor. For this reason, a factor is added using the ratio of the perimeters between the two polygons.

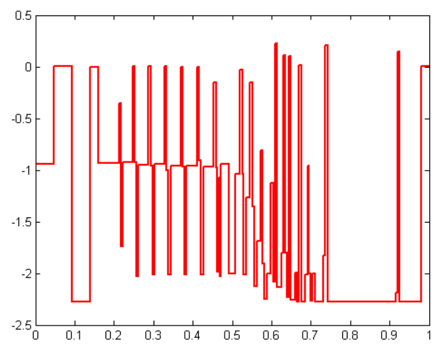
$$S(A,B) = \frac{\max(L_A, L_B)}{\min(L_A, L_B)} \cdot d(A,B) \tag{8-2}$$

where L_A and L_B are the perimeters of polygon A and B respectively. In order to avoid the translation of the tangent angle in relation to the other one, an identical point pair of the two polygons has to be found out and set as reference point for the calculation of the tangent angles. Note that $S(A,B)$ denotes actually the dissimilarity between A and B. The smaller $S(A,B)$ is, the more similar are the two polygons. In the case A is identical to B, there is $S(A,B)=0$.

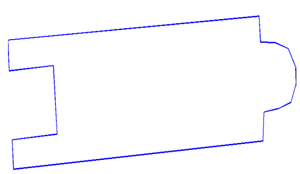
The above described methodology has been deployed in our current work for similarity measure of simplified ground plans with their original ones. Figure 8.11a shows the same ground plan as in Figure 8.9. It is simplified using our algorithm (Figure 8.11c). In order to compare the similarity, the original ground plan is simplified manually for an alternative result (Figure 8.11e).



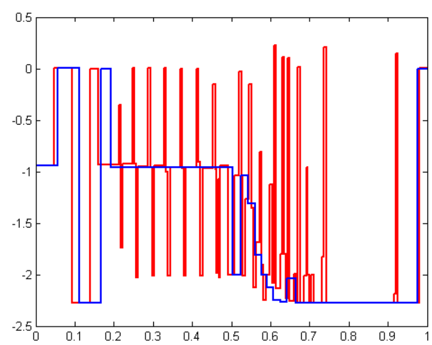
a. Original ground plan



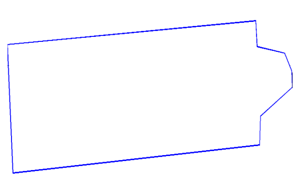
b. tangent space of ground plan in a.



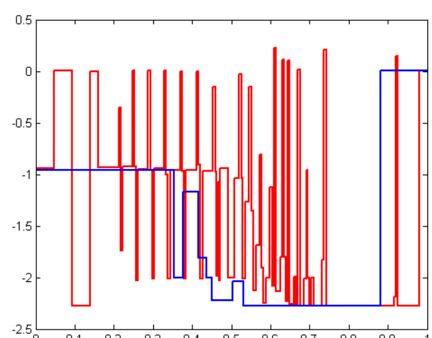
c. similarity $S = 0.47$



d. overlap of the tangent angles



e. similarity $S = 0.72$



f. overlap of the tangent angles

Fig. 8.11. Similarities of a ground plan with two varieties of simplification

The test in Figure 8.11 shows that the simplified polygons using our algorithm are more similar to the original ones. Therefore, the simplified building model can be easily associated with the original model.

8.3 Visualization environment

With the objective to visualize spatiotemporal buildings including their geometric and semantic changes, an interactive system was developed by (Hertel et al. 2009) and can be adopted in our context as a tool for user to navigate in a spatiotemporal city model and show query results.

In this system, time is interpreted as the fourth independent dimension to the 3D spatial data. In comparison to the conventional visualization systems, a 4D (time + 3D spatial data) data structure is introduced in the system. The data structure extends 3D KD-trees by adapting the Surface Area Heuristic methods (MacDonald & Booth, 1990) towards its use for partitioning spatiotemporal geometry models. By using this novel data structure, the system is able to render large-scale city models interactively.

Figure 8.12 shows the German city Ettenheim at four snapshots captured during the visualization of the city in a time sequence.

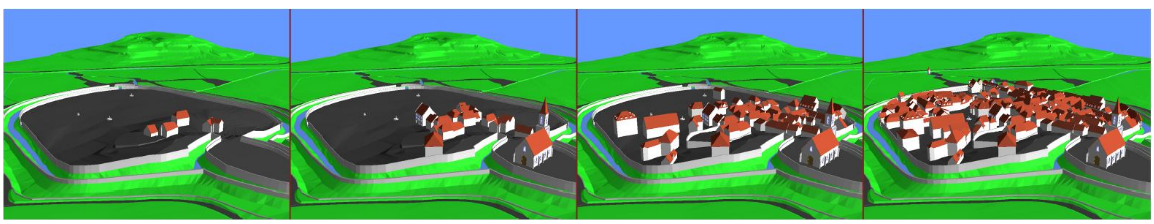


Fig.8.12. Visualizing a time-dependent city model in a time sequence.

The system can handle city models given in CityGML format, with an application domain extension (ADE) for the time stamps of building objects. Every building object in the dataset can have a date of construction and a date of destruction. If one of the dates is not given, it is assumed that the object was constructed an infinite time ago or will never be destructed. The CityGML dataset is converted to an internal format at the system startup by interpreting events of construction and destruction as the fourth dimension of the underlying domain.

The current system is, in fact, only a proof of concept for the visualization of time-dependent city model. It is restricted to two simple cases: an object can appear at a given time point and disappear from the model at an another given time. Dynamic geometrical changes can not yet be visualized. Moreover, other types of time-dependencies, for instance, movements of objects, are not modeled in the system.

Chapter 9

Conclusion and outlook

9.1 Conclusion

The main findings of this thesis can be better described as answers to the five research questions raised at the beginning of this thesis (Section 1.2).

Question 1: How to define a suitable data structure that supports not only spatial queries, but also queries of events, processes as well as temporal topological relationships?

In this work, an object-oriented event-state spatiotemporal data model for 4D city environment is presented with the focus on 3D building objects that change over time. The data model is composed of an event model and a city model (Figure 9.1). They both are constructed according to the principle of object-oriented modeling. The events are modeled with five attributes describing what, where, when, how, and who – with "what" for the type or class of events, "where" for location of events, "when" for time point or duration of events, "how" for the changing modes of events, and "who" for the objects involved in the events. Buildings and their components are modeled with attributes like geometries, functions, and names etc. Furthermore, every attribute is stamped with its lifespan that indicates the duration when it is valid.

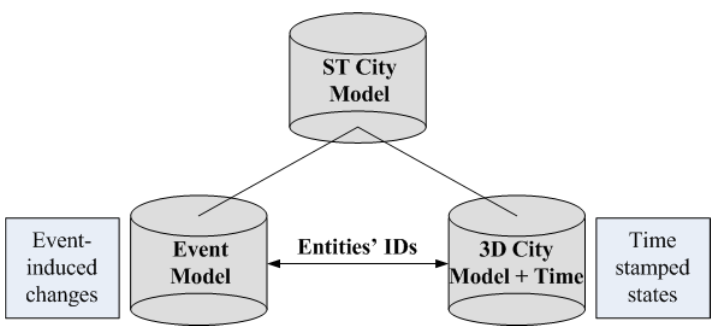


Fig.9.1. Two components of the spatiotemporal city model, whereby ST stands for "spatiotemporal"

In this way, histories of city objects are modeled. While the states of city objects are represented in the time-stamped city model, the events which indicate how the city objects change from one state to another are represented in the event model. The spatiotemporal data can be "double indexed", namely by events happened to objects and by objects involved in events.

Question 2: How to contemplate a data model that is able to handle temporal information associated with 3D city objects at different granularity levels?

In the proposed spatiotemporal data model, the coherent relations between events and objects are considered. Events are bound to objects and can be described with respect to the hierarchy of the involved objects. There are two types of events: events happened to geometries of objects and events happened to attributes of objects. An event of the first type can be decomposed into a number of smaller events when the involved objects can be semantically decomposed, because such an event is normally a dynamic process and can be represented by a sequence in which the parts of objects may get involved one after another. An event of the second type cannot be decomposed. It happens or it does not happen and there is a straightforward relation between an attribute and its object.

For the first type of events, there is an *n*-to-*m* relation between event components and object components. If the entire object is involved in the event, then the event is termed as a 1^{st}-level event. If only one part of the object is involved in the event, then the event is a 2^{nd}-level event. In fact, every 2^{nd}-level event can be decomposed into several 1^{st}-level events to its object child, and further offspring. To avoid multiple storages of events, all the events are connected with their objects as the 1^{st}-level events.

For the semantically decomposable events[10], there is a coherent semantic structure for the corresponding spatial objects. When decomposing such events and their corresponding objects simultaneously, many 1^{st}-level events occur. Depending on to which LoD the spatial objects are decomposed, the events can be represented at different LoDs with respect to how precisely the dynamic processes can/should be represented.

Question 3: How to develop mechanisms of compressing spatiotemporal information for fast rendering?

In this work, fast rendering is achieved by reducing the number of events and objects for the visualization, and by preparing 3D buildings at different LoDs.

The input of an explicit query triggers the corresponding operations on the spatiotemporal data model. The states or (and) changes satisfying the conditions defined in the query are exhaustively retrieved. The query results can be visualized if necessary. However, some of the states or/and events might not be significant enough to be noticed at the current spatiotemporal scale of visualization environment because the geospatial phenomenon are scale-dependent. Consequently, these insignificance states and events will be neglected during the visualization.

In this thesis, algorithms for spatiotemporal generalization are also proposed which is composed of event generalization and 3D spatial generalization. Event generalization is similar to the news-ranking in the media society. After ranking of events according to different criterion like changed area, volume, cost, etc, the non-significant events will be neglected. As a result, the number of events and their corresponding objects are reduced.

The most efficient way for the fast rendering is to reduce the memory usage and speed up geometric computation. For this purpose, algorithms for 3D generalization are developed. In

[10] A semantically decomposable event here is an event whose bearer object can be semantically decomposed. Normally, it is decomposed into several smaller events following the same principle of decomposing its bearer object.

advance of the generalization, specific LoDs are defined for building model by: (i) deploying the concept of LoDs in CityGML, and (ii) adding two sub-LoDs between LoD3 and LoD2 of CityGML. The approach starts from the most detailed 3D building models at LoD4 which is progressively transferred to coarse building models. The most remarkable steps in the generalization process are: (i) deriving SLoD3 building models from the LoD3 models and (ii) deriving LoD2 building models from the SLoD2 models, where SLoD3 and SLoD2 indicate the exterior shell of buildings at LoD3 and LoD2 respectively. After these two steps, the required storage of a building is reduced substantially.

Question 4: What can be operated and queried spatiotemporally in the 4D city environment?

The spatiotemporal data model defines a set of spatiotemporal operations by extending five basic operations in the traditional DBMS: set union, set difference, Cartesian product, selection, and projection to spatiotemporal union, spatiotemporal difference, spatiotemporal intersection, spatiotemporal selection, and spatiotemporal projection.

Regarding the spatiotemporal queries, two types of queries are possible with our spatiotemporal data model: object-based queries and event-based queries. Object-based query means that objects have to be identified at first in order to answer the query. With respect to an object, not only its geometry and attributes, but also the spatial and temporal relationships (comparisons) between this object and others can be inquired. Event-based queries seek information about events; therefore an event has to be identified at first, whereby the attributes of events like time, location, objects involved in them are asked. Moreover, the spatiotemporal data model is double-indexed. Events and 3D objects can be queried independently, which allows efficient queries.

Question 5: How to visualize the results of spatiotemporal queries?

The current work suggests two ways to visualize the results of spatiotemporal queries: static visualization and dynamic visualization. The static mode is mostly used to represent a snapshot of geospatial changes. However, geospatial changes that occurred at different points of time or over different durations can also be statically visualized, for instance, by using different colors, forms of lines or different forms of polygons etc.

For the dynamic visualization of geospatial processes, a system for visualizing spatiotemporal datasets using a four dimensional KD-tree as acceleration structure can be adopted (Hertel et al., 2009). The system allows the user to interactively explore large scale time-dependent city models

9.2 On-going research work and the works in the future

This work has initiated two concrete research tasks – spatiotemporal generalization and visualization. The first results are promising and will stimulate further improvement and extension.

Section 7.3.1 presents the algorithm to extract the exterior shell of 3D buildings from their LoD3 models. Theoretically, it can reduce more than 80% of the storage space of 3D buildings which are modelled as cuboids in LoD3. Optically one single plane is sufficient to represent a building component such as window and door. Our own experiments show that the approach can reduce about 90% of the storage space of 3D buildings (Fan et al. 2009). The exterior shell is utilized as it could preserve almost all the details of the original LoD3 model. For this reason, shell model is defined as the alternative representation of LoD3 model in CityGML.

Typificaiton is used in this work as an operator to generalize uniformly shaped features on building facades (7.3.2). In order to find out what kind of representation after typification can be best associated to the original dataset with respect to the human visual impression, a user survey has been conducted. The results show that preserving the shape of the feature elements is the most important constraint for a reasonable typification process, which has also been verified quantitatively by calculating the similarities between the typified façades and the original façade using ARG and NEMD algorithms. Based on that, an algorithm is developed to generate perceivably reasonable scaled-down representation from the original distribution. However, the developed approach can be only used for typifying facade with rectangular windows. For windows with complicated structure e.g. church window more factors have to be introduced. In the nearest future, the so called contextual typification will be investigated, which means that the typification for windows on a facade should consider the distributions of windows on its neighboring facades. Moreover, the developed approach will be tested on ground plans of buildings.

So far, our approach of 3D generalization can only handle 3D buildings individually. For a larger change of spatial scale, buildings have to be generalized in group. Some approaches on aggregating building blocks have been reported (Anders 2005, Glander & Döllner 2009). In fact, human eyes can perceive (differentiate or identify) many important features of building groups within a large range of scale until they can be viewed as a block. These features include patterns formed by windows on façades of row houses, patterns formed colors of façade elements, the height differences of the neighboring buildings, the roof structures of row houses, and etc. The generalization of such features can be called "Generalization for row houses". Algorithms can be developed by extracting features of the row houses in different directions, whereas filters can be defined for extracting 3D features.

With regard to the visualization environment, the currently available system allows visualization of large datasets of spatiotemporal city model and user navigation. In the future, this system will be extended to (i) visualize dynamic changes of a city such as the representation of its history, (ii) query 3D buildings and events happened to them, (iii) conduct spatiotemporal simulations and so on.

List of figures

Fig.2.1 The historical-geographic work of Constantinople 16
Fig.2.2 Panoramic painting "Along the River During the Qingming Festival". 16
Fig.2.3 Parametric building models 18
Fig.2.4 An example for the evolution of a typical European city 22
Fig.2.5 The eight possible changes of a geo-referenced object 22
Fig.2.6 An example of the snapshot model 23
Fig.2.7 An example of an object-based model with spatiotemporal complex and simplexes 24
Fig.2.8 Primary elements and the pointer structure of an ESTDM 25

Fig.3.1 Tree structure of Events 31
Fig.3.2 Tenses of events 32
Fig.3.3 Three possible relationships between events which have zero-width duration 32
Fig.3.4 Five possible relationships between a zero-width-duration event and a nonzero-width-duration event 33
Fig.3.5. The 13 possible relationships between events 33
Fig.3.6. Topological relationships in 2D plane 36

Fig.4.1. Five general attributes of an event 40
Fig.4.2. Class structure of the event model 40
Fig.4.3. Example modes for events of buildings 41
Fig.4.4. Coherent hierarchies in semantics and geometry of city object (i.e. building) 43
Fig.4.5. Superclass for city objects 43
Fig.4.6. Class structure of buildings 44
Fig.4.7. Class structure of time-stamped city objects 46
Fig.4.8. Event-state representation for the life of an object 47

Fig.5.1. Alte Pinakothek in Munich 55
Fig.5.2. Static visualization of the querying results 56
Fig.5.3. Representation of temporal information on the facade 56
Fig.5.4. Some events of a building were carved on its facade 57

Fig.7.1. The four levels of detail (LoD) defined by CityGML for building model 65
Fig. 7.2. An example house in LoD3 and its exterior shell 69
Fig. 7.3. Painted facades give pedestrians similar visual impression 69

Fig.7.4. Typifying a facade 71
Fig.7.5. Two types of representation of the ground plan........ 71
Fig.7.6. Different parts of a ground plan a) and c) and their simplification results b) and d) after the first iteration........ 72
Fig.7.7. Different parts of a ground plan a) and c) and their simplification results b) and d) after the first iteration........ 73
Fig.7.8. The shortest side is removed by intersecting its flanking neighbors 73
Fig.7.9. Wide angle as the simplification result 74
Fig.7.10. Removal of a sharp angle caused by removal of a short side........ 74
Fig.7.11 Flow chart for operating the sharp angle 74
Fig.7.12. Derivation of the angle threshold according to the visual perception........ 75
Fig.7.13. The simplified roof polygons can be merged with each other and then simplified again........ 77
Fig.7.14. Different cases of intersecting a wall with the roof structure........ 78

Fig. 8.1. UML diagram of CityGML's Event2Building model 82
Fig. 8.2. Workflow and Matlab modules for deriving building models in LoD2 from SLoD2 83
Fig.8.3. A flat-roof building with several corners which can result long narrow angles........ 84
Fig.8.4. A building with a large asymmetric hipped roof which is preserved after the simplification........ 85
Fig.8.5. A simple building with a small asymmetric hipped roof 86
Fig.8.6. A complex building with a corner roof and an additional T-element........ 86
Fig.8.7. A building with a complicated ground plan and a small T-element........ 87
Fig.8.9. A building with a complicated ground plan and steep roof elements........ 87
Fig.8.10 Illustration of tangent space representation of polygon........ 88
Fig.8.11. Similarities of a ground plan with two varieties of simplification 89
Fig.8.12. Visualizing a time-dependent city model in a time sequence. 90

Fig.9.1. Two components of the spatiotemporal city model 91

Bibliography

Abraham T. and Roddick J. 1999. Survey of spatio-temporal databases. In: GeoInformatica, 1999 (3): 61-69.

Albert J., Bachmann M. and Hellmeier A. 2003. Zielgruppen und Anwendungen für Digitale Stadtmodelle und Digitale Geländemodelle. Erhebung im Rahmen der SIG 3D der GDI NRW. http://www.ikg.uni-bonn.de/fileadmin/sig3d/pdf/Tablelle_Anwendungen_Zielgruppen.pdf

Allen A., Edwards G. and Bedard Y. 1995. Qualitative causal modeling in temporal GIS. In: Spatial Information Theory: A theoretic basis for GIS. Springer-Verlag, No. 988, pp. 397-412.

Allen J.C. 1983. Maintaining knowledge about temporal intervals. CACM 26 (11) 832-843.

Allen J.C. & Ferguson, G., 1994. Actions and Events in Interval Temporal Logic, Journal of Logic and Computation. 1994 (5) 531-579.

Allen T.F.H. and Hoekstra T.W. 1991. Role of Heterogeneity in Scaling of Ecological Systems under Analysis. In: Ecological Heterogeneity, edited by Kolasa J. and Pickett S. New York: Springer. Pp. 47-68.

Anders K.H. 2005. Level of detail generalization of 3D building groups by aggregation and typification. In: XXII International Cartographic Conference (ICC2005). A Coruna, Spain, 11-16 July 2005.

Andrienko N., Andrienko G. and Gatalsky P. 2003. Exploratory spatio-temporal visualization: an analytical review. In: Journal of Visual Languages and Computing 14(2003) 503-541.

Armstrong, M.P. 1988. Temporality in spatial databases. In: Proceedings GIS/LIS'88, 2:880-889.

Bairoch P. 1988. Cities and economic development: from the dawn of history to the present. Chicago: University of Chicago Press, c1988. XXII, 574p. ISBN: 0226034658 0226034666

Bagrow L. and Skelton R.A. 1963. Meister der Kartographie. Safari-Verlag, Berlin. 4., neu bearbeitete und erweiterte Auflage.

Berry B., 1964. Approaches to regional analysis: a synthesis. In: Annals of the Association of American Geographers, 54, pp. 2-11.

Brunn A., Gülch E., Lang F. and Förstner W. 1998. A hybrid concept for 3D building acquisition. In: ISPRS Journal of Photogrammetry & Remote Sensing 53 (1998). pp. 119-129.

Burghardt D. and Cecconi A. (2007). Mesh simplification for building typification. In: International Journal of Geographical Information Science, Vol. 21, No. 3, March 2007. pp. 283-298.

Camoss E., Bertolotto M., Bertino E. and Guerrini G. 2003. A Multigranular Spatotemporal Data Model. In: Proceedings of the 11th ACM international symposium on Advances in geographic information systems. New Orleans, Louisiana, USA. November 7-8, 2003. pp. 94-101

Caati R. and Varzi A. 2008. Event Concepts. In: Understanding events: from perception to action, Shipley T.F. & Zacks J.M. (eds.), Oxford University Press, Inc., pp. 31-53.

Chen J. and Jiang J. 1998 Event-based Spatio-temporal Database Design. In: Fritsch D., English M. & Sester M., eds, IAPRS, Vol.32/4, pp. 105-109.

Chen J. and Jiang J. 2000. An Event-Based Approach to Spatio-Temporal Data Modeling in Land Subdivision Systems. In: GeoInformatica 4:4, 387-402.

Chen P. 1976. The Entity-Relationship Model: Toward a unified view of Data. In: ACM TODS, Vol. 1(1): 9-36.

Chen T.K., Abdul-Rahman A. and Zlatanova S. 2008. 3D Solids and Their Management in DBMS. In: Advances in 3D Geoinformation Systems, P. van Oosterom, S. Zlatanova, F. Penninga and E. Fendel (Eds.), Lecture notes in geoinformation and cartography, Springer Verlag Berlin Heidelberg 2008. pp. 279-311.

Chin L.H. and Tarski A. 1948. Remarks on Projective Algebras. In: Bulletin of the AMS, 54:80-81.

Claramunt C. and Thériault M. 1995. Managing time in GIS: an event-oriented approach. In: Recent Advances in Temporal Databases, Springer Verlag, pp. 23-43.

Claramunt C. and Thériault M. 1996. Toward semantics for modeling spatio-temporal processes within GIS. In: Proceedings of International Symposium on Spatial Data Handling (SDH'96), Netherlands, pp. 27-43.

Clark J.H. 1976. Hierarchical Geometric Models for Visible Surface Algorithms. In: Communications of the ACM, October 1976 Volume 19(10): 547-554.

Couclelis H. 1992. People manipulate objects (but cultivate fields). In: Proceedings of International Conference on GIS. Lecture Notes, Springer-Verlag. Pisa, Italy. Pp. 65-77.

Egenhofer M.J. and Al-Taha K. 1992. Reasoning about gradual changes of topological relationships. In: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. A. Frank, I. Campari and U. Formentin (eds.), New York, Springer-Verlag, pp: 196-219.

Fan H. and Meng L. 2008. Analysis of events in 3D building models. In: Proceedings of SPIE, the 16th International Conference on Geoinformatics and the Joint Conference on GIS and Built Environment, Guangzhou, China, June 28-29, 2008.

Fan H., Meng L. and Stilla U. 2008. A concept for integrating time-dependent features in 3D building models. In: International Archives of Photogrammetry, Remote Sensing and Spatial Geoinformation Science, Vol 37(B2): 45-50. Beijing, China. 03-11. July 2008

Fan H., Meng L. and Jahnke M. 2009. Generalization of 3D Buildings modeled by CityGML. In: AGILE2009, Advances in GIScience. Lecture Notes in Geoinformation and Cartography, Springer Dordrecht Heidelberg London New York, ISSN 1863-2246, pp. 387-405. AGILE2009, 02-05 June 2009. Hannover, Germany.

Fan H., Liu L. and Meng L. 2009b. A mathematical model for event detection in spatiotemporal city environment. In: the Proceedings of ISPRS workshop on Quality, Scale and Analysis Aspects of Urban City models, Lund, Sweden, 3-4 Dec. 2009.

Fan H. and Meng L. 2009. Automatic derivation of different levels of detail for 3D buildings modelled by CityGML. In: Proceedings of the 24th International Cartographic Conference (ICC), Santiago, Chile, 15-22 Nov. 2009.

Fan H., Mao B. and Meng L. 2010. An automatic approach for the typification of façade structures. Submitted for ISPRS Journal of Photogrammetry and Remote Sensing.

Forberg A. 2004. Generalization of 3D Building Data based on a Scale-Space Approach. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, Istanbul, Turkey, Vol. XXXV, Part B.

Forberg, A. 2007. Generalization of 3D building data based on scale-space approach. In: ISPRS Journal of Photogrammetry and Remote Sensing 62 (2007), pp. 104-111.

Förstner W. 1999. 3D City models: Automatic and Semiautomatic Acquisition Methods. In: Photogrammetric Week 99. Fritsch D. & Spiller R. Eds., Wichmann Verlag, Heidelberg, 1999. pp. 291-303

Fussell D.S. and Subramanian K.R. 1988. Fast ray tracing using k-d trees. Technical report. University of Texas at Austin, Austin, TX, USA.

Gibson C.C., Ostrom E. and Ahn T.K. 2000. The concept of scale and the human dimensions of global change: a survey. In: Ecological Economics 32(2000) 217-239.

Glander T. and Döllner J. 2009. Abstract Representation for Interactive Visualization of Virtual 3D City Model. In: Computers, Environment and Urban Systems, Volume 33, Issue 5, September 2009, pp. 375-387.

Glassner A.S. 1984. Space subdivision for fast ray tracing. In: IEEE Computer Graphics and Applications, vol. 4, no. 10, pp. 15-22.

Gröger G., Kolbe TH., Czerwinski A. and Nagel C. 2008. OpenGIS® City Geography Markup Language (CityGML). Implementation Specification. http://www.opengeospatial.org/legal/.

Guttman A. 1984. R-trees: A dynamic index structure for spatial searching. In: Proceedings of ACM SIGMOD, International Conference on Management of Data, Boston, MA, USA, pp. 47-57.

Haala N., Becker S. & Kada M. 2007. Cell Decomposition for Building Model Generation at Different Scales. In: Proceedings of Urban Remote Sensing Joint Event 2007, Paris, France, April, 11-13, 2007.

Harley J.B. and Woodward D., 1987. Eds. Cartography in Prehistoric, Ancient, and Medieval Europe and the Mediterranean. Volume 1 of The History of Cartography. Chicago and London: University of Chicago Press, 1987.

Hanson C. and Hirst W. 1989. On the representation of events: A study of orientation, recall, and recongnition. In: Journal of Experimental Psychology: General, 118, pp. 136-147.

Heller M. 1990. The Ontology of Physical Objects: Four-dimensional Hunks of Matter. Cambridgs Studies in Philosophy. Cambridge University Press, UK.

Heok T.K. and Daman D. 2004. A review on Level of Detail. In: Proceedings of the international Conference on Computer Graphics, Imaging and Visualization (CGIV'04).

Hertel S., Wagner M. and Westermann R. 2009. A system for visualizing spatiotemporal urban datasets. In: the Proceedings of the 17th International Conference on Geoinformatics, August 12-14, 2009 Fairfax VA, USA.

Hoffmann C.M. 1989. Geometric and Solid Modeling: an introducation. Morgan Kaufman Publishers Ins. San Francisco, CA. USA. ISBN: 1-55860-067-1.

Hornsby K. and Egenhofer M.J. 2000. Identity-based change: a foundation for spatiotemporal knowledge representation. In: International Journal of Geographical Information Science, 14:3, 207-224.

Jahnke M., Fan H. and Meng L. 2009. A user perspective on typification of façade features of 3D building models. In: Proceedings of the 24th International Cartographic Conference (ICC), Santiago, Chile, 15-22 Nov. 2009.

Jin P., Yue, L. and Gong Y. 2007. Design and implementation of a unified spatio-temporal data model. In: ISPRS Book Series: Advances in Spatio-Temporal Analysis. Taylor & Francis Ltd. Pp. 67-75.

Kada M. 2002. Automatic generalization of 3D building models. In: Proceedings of the Joint International Symposium on Geospatial Theory, Processing and Applications, Ottawa 2002.

Kada, M. 2006. 3D Building Generalization based on Half-Space Modeling. In: Proceedings of the ISPRS Workshop on Multiple Representation and Interoperability of Spatial Data, Hannover.

Kada M. 2007. Generalization of 3D building models by Cell Decomposition and Primitive Instancing. In: Proceedings of the Joint ISPRS Workshop on “Visualization and Exploration of Geospatial Data”. Stuttgart, Germany.

Kada M. 2007. Scale-dependent simplification of 3d building models based on cell decomposition and primitive instancing. In: Winter S., Duckham M., Kulik L. and Kuipers B. (Eds.) Spatial Information Theory: 8th International Conference, COSIT 2007, LNCS 4736, Springer-Verlag Berlin Heidelberg, pp.222-237.

Kainz, W. 1995. Logical Consistency. In: Elements of Spatial Data Quality, Guptill C.S. & Morrison L.J. (eds.), Elsevier Publication.

Kim D.H., Yun D. amd Lee S. 2010. Attributed relational graph matching based on the nested assignment structure. In: Pattern Recognition, Volume 43, Issue 3, March 2010, Pages 914-928.

King A.W. 1998. Hierarchy Theory: A Guide to System Structure for Wildlife Biologists. In: Wildlife and Landscape Ecology, Effect of Pattern and Scale, edited by Bissonette J.A. Springer New York. Pp. 3-22.

Kolbe T.H. and Gröger G. 2003. Towards unified 3D city models. In: Schiewe J., Hahn M., Madden M. And Sester M. (eds): Challenges in Geospatial Analysis, Integration and Visualization II. Proceedings of Joint ISPRS Workshop, Stuttgart.

Kolbe T.H., Gröger G. and Plümer L. 2005. CityGML – Interoperable Access to 3D City Models. In: Proceedings of the Int. Symposium on Geo-information for Disaster Management on 21-23. March 2005 in Delft. Oosterom, Zlatanova and Fendel (eds.) Springer Verlag.

Koble T.H. 2008. Representing and Exchanging 3D City Models with CityGML. In: LNGC 3D Geo-Information Sciences. Springer Berlin Heidelberg Verlag. Pp. 15-31.

Kolbe T.H., Nagel C. and Stadler T. 2009. CityGML – OGC Standard for Photogrammetry? In: Photogrammetric Week 09, Fritsch D. (Ed.). Wichmann Verlag. 07-11 September, 2009: 265-277. Stuttgart Germany.

Lal J. and Meng L. 2001. Rules and constraints for 3D generalization of urban area. In: Journal of Geographical Science, Vol. 11 Supplement: 17-28.

Lamy S., Ruas A., Demazeau Y., Jackson M., Mackaness W. and Weibel R. 1999. The Application of Agents in Automated Map Generalization. In: Proceedings of the 19th International Cartographic Conference of the ICA, Ottawa, Canada.

Langran G. and Chrisman N.R. 1988. A framework for temporal geographic information. In: Cartographica, 25(3): 1-14.

Langran G. 1988. Temporal design tradeoffs. In: Proceedings of GIS/LIS'88, Volume 2 (Falls Church, VA: ACSM), pp. 890-899.

Langran G. 1992. Time in Geographical Information Systems, ed. Taylor & Francis, London, 1992.

Langran G. 1993. Manipulation and analysis of temporal geographic information. In: the Proceedings of the Canadian conference on GIS, Ottawa.

Lansky A.L. 1986. A Representation of Parallel Activity based on Events, Structure, and Causality. Technical Note 401, SRI International, Menlo Park.

Lee S.C. and Nevatia R. 2003. Interactive 3D building modeling using a hierarchical representation. In: the 1st IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis (HLK'03).

Li L., Yin Z. and Zhu H. 2007. Spatio-temporal indexing mechanism based on snapshot-increment. In: Advances in Spatio-Temporal Analysis, Tang et al. (eds). Taylor & Francis Group, London. Pp. 77-84.

Li Z. (2007). Algorithm foundation of multi-scale spatial representation. CRC Press of Taylor & Francis Group. Boca Rotan London New York. 2007.

Liu J. 1998. An ancient cadastre – Tongguan Tu. In : Bulletin of Surveying and Mapping. 1998(7). (In Chinese)

Luebke D., Reddy M., Cohen J.D., Varshney A., Watson B. and Huebner R. 2003. Level of Detail for 3D Graphics. The Morgan Kaufmann series in computer graphics and geometric modelling. Morgan Kaufmann Publishers, San Francisco, 2003. ISBN: 1-55860-838-9.

MacDonald D.J. and Booth K.S. 1990. Heuristics for ray tracing using space subdivision. In: The Visual Computer: International Journal of Computer Graphics, vol. 6, no. 3, pp. 153-166.

MacEachren A.M. and Kraak M.-J. 2001. Research Challenges in Geovisualization. In: Cartography and Geographic Information Science, 28(1) 3-12.

Mani I., Pustejovsky J. and Gaizauskas R. 2005. The language of time: a reader, Oxford University Press.

Mayer, H. 2005. Scale-spaces for generalization of 3D buildings. In: International Journal of Geographical Information Science. Vol. 19, No. 8-9, September-October 2005, pp. 975-997.

Mayr E. 1982. The Growth of Biological Thought: Diversity, Evolution, and Inheritance. Belknap Press, Cambridge, MA.

Meng L. and Forberg A. 2007. 3D building generalization. In: Mackaness, W., Raus, A. and Sarjakoski, T. (Eds): Generalization of Geographic Information: Cartographic Modelling and Applications, Elsevier, 2007.

Müncher Stadtmuseum and Stadtarchive München. 2008. Stadtatlas München Digital: Karten und Modellen von 1570 bis heute. Franz Schiermeier Verlag.

Nelson K. 1986. Event Knowledge and cognitive development. In: Event Knowledge: Structure and Function in Development, Nelson K. (eds.) Psychology Press; 1 edition (April 1, 1986).

Neudeck S. 2001. Zur Gestaltung topografischer Karten für die Bildschirmvisualisierung. Dissertation at the Universität der Bundeswehr München. (in German)

Orenstein J. 1986. Spatial query processing in an object-oriented database system. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp.326-336

Pelekis N., Theodoulidis B., Kopanakis I. and Theodoridis Y. 2004. Literature Review of Spatio-Temporal Database Models. In: The Knowledge Engineering Review, 19(3): 235-274.

Peuquet D.J. 1994. It's about time: a conceptual framework for the representation of temporal dynamics in geographic information system. In: Annals of the Association of American Geographers 84(3) 441-461.

Peuquet D.J. 2001. Making Space for Time: Issues in Space-Time Data Representation. In: GeoInformatica 5:1, pp. 11-32.

Peuquet D.J. 2005. Theme section on advances in spatio-temporal analysis and representation. In: ISPRS Journal of Photogrammetry & Remote Sensing 60 (2005) 1-2.

Rainsford D. and Mackaness W. 2001. Template Matching in Support of Generalization of Rural Buildings. In: Proceedings of the Joint International Symposium on "Geospatial Theory, Processing and Applications" (ISPRS Commission IV SDH2002), Ottawa, Canada, July 8-12, 2002.

Raza A. and Kainz W. 1999. Cell Tuple Based Spatio-Temporal Data Model: An Object Oriented Approach. In: Proceedings of the 7th ACM international symposium on Advances in geographic information systems. Kansas City, Missouri, US. Pp. 20-25.

Regnauld N. 2001. Contextual Building Typification in Automated Map Generalization. In: Algorithmica, 30(2), pp. 312-333.

Rezayan H., Karimipour F., Delavar M.R. and Frank A.U. 2007. Temporal topological relationships of convex spaces in Space Syntax theory. In: ISPRS Book Series: Advances in Spatio-Temporal Analysis. Taylor & Francis Ltd. Pp. 85-100.

Ribelles J., Hechbert P.S., Garland M. and Stahovich T.F. 2001. Finding and removing features from polyhedra. In: Proceedings of DETC'01, ASME Design Engineering Technical Conference, Pittsburgh, Pennsylvania, USA.

Ripperda N. and Brenner C. 2006. Reconstruction of Facade Structures Using a Formal Grammar and RjMCMC. In: Franke K., Müller K.R., Nickolay B. and Schäfer R. (Eds.), Pattern Recognition, Proceedings of the 28th DAGM Symposium, September 2006. Berlin, Germany. Pp. 750-759.

Robinson J. 1981. The K-D-B-tree: A search structure for large multidimensional dynamic indexes. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 10-18.

Roshannejad A.A. and Kainz W. 1995. Handling Identities in Spatio-Temporal Databases. In: Proceedings of ACSM/ASPRS 1995 Annual Convention and Exposition Tech. Papers.

Sester M. 2000. Generalization based on Least Squares Adjustment. In: International Archives of Photogrammetry and Remote Sensing, Vol.33, ISPRS, Amsterdam 2000.

Sester M. & Brenner, C. 2004. Continuous Generalization for Visualization on Small Mobile Devices. In: Proceedings of Spatial Data Handing 2004, Springer Berlin Heidelberg, pp.355-368.

Sester M. 2005. Optimization approaches for generalization and data abstraction. In: International Journal of Geographical Information Science. Vol. 19, No. 8-9, September-October 2005, pp. 871-897

Sester M. 2007. 3D Visualization and Generalization. In: Photogrammetric Week '07, Fritsch D. (Ed.) Herbert Wichmann Verlag Heidelberg, pp. 285-295. September 3-9, 2007, Stuttgart Germany.

Shipley T.F. 2008. An Invitation to an Event. In: Understanding events: from perception to action, Shipley T.F. & Zacks J.M. (eds.), Oxford University Press, Inc., pp. 3-30.

Slackman E.A., Hudson J.A. and Fivush R. 1986. Actions, Actors, Links, and Goals: the structure of children's event representation. In: Event Knowledge: Structure and Function in Development, Nelson K. (eds.) Psychology Press; 1 edition (April 1, 1986).

Stadler A. and Kolbe T.H. 2007. Spatio-Semantic coherence in the integration of 3D city models. In: Proceedings of the 5th International Symposium on Spatial Data Quality. Enschede 2007.

Suveg I. & Vosselman G. 2004. Reconstruction of 3D building models from aerial images and maps. In: ISPRS Journal of Photogrammetry & Remote Sensing 58(2004) 202-224.

Staufenbiel W. 1973. Zur Automation der Generalisierung topographischer Karten mit besonderer Berücksichtigung großmaßstäbiger Gebäudedarstellungen, Wissenschaftliche Arbeiten der Fachrichtung Vermessungswesen der Universität Hannover (51), Ph.D Thesis.

Stratulat T., Clèrin-Debart F. and Enjalbert P. Norms und Time in Agent-Based Systems. In: Proceedings of the 8th International Conference on Artificial Intelligence and Law, St. Louis, Missouri, 178-185 (2001).

Thiemann F. (2002). Generalization of 3D Building Data. In: the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, Volume XXXIV, Part 4.

Thiemann F. and Sester M. 2004. Segmentation of Buildings for 3D Generalization. In: Working Paper of the ICA Workshop on Generalization and Multiple Representation, Leicester, UK.

Thiemann F. and Sester M. 2006. 3D-Symbolization Using Adaptive Templates. In: ISPRS Technical Commission II Symposium 2006, Vienna, Austria 12-14 July 2006, pp. 49-54.

Tobler W., 1959. Automation and Cartography. In: Geographical Review, vol. 49, pp. 526-534.

Tryfona N. and Jensen C.S. 1999. Conceptual Data Modeling for Spatiotemporal Applications. In: GeoInformatica, Vol.3:273-300.

Tryfona N. and Jensen C.S. 2000. Using Abstractions for Spatio-Temporal Conceptual Modeling. Proceedings of the 2000 ACM Symposium on Applied Computing, Como, Italy, 2000.

Tsao H., 2003. Unraveling the mystery of the handscroll 'Qingming Shanghe Tu'. In: Journal of Sung-Yuan Studies 33.

Turner M.G., O'Neill R.V., Gardner R.H. and Milne B.T. 1989. Effects of Changing Spatial Scale on the Analysis of Landscape Pattern. In: Landscape Ecology 3, 153-162.

Van Kreveld M. 2001. Smooth Generalization for Continuous Zooming. In: Proceedings of the ICA, 4th Workshop on Progress in Automated Map Generalization, Beijing China, August 2001.

Wang X., Zhou X. and Lu S. 2000. Spatiotemporal Data Modeling and Management: a Survey. In: Proceedings of 36th International Conference on Technology of Object-Oriented Languages and Systems. October 30 – November 4, 2000. Xi'an, China. Pp. 202-211.

Wang H., Tang X., Lei B. and Zhai L. 2007. Spatio-temporal data model: An approach based on version-difference. In: ISPRS Book Series: Advances in Spatio-Temporal Analysis. Taylor & Francis Ltd. Pp. 67-110.

Weins J.A. 1989. Spatial Scaling in Ecology. In: Functional Ecology 3, 385-397.

Worboys M. 1992. A model for spatio-temporal information. In: Proceedings: the 5th International Symposium on Spatial Data Handling, 2:602-611.

Worboys M. 1994. A unified model of spatial and temporal information. In: Computer Journal, 37(1): 26-34.

Worboys M. 1994b. Object-oriented approaches to geo-referenced information. In: International Journal of Geographical Information Science, 8: 4, 385-399.

Worboys M. 2005. Event-oriented approaches to geographic phenomena. In: International Journal of Geographical Information Science, Volume 19, Issue 1 January 2005, pages 1-28.

Yuan M. 1994. Wildfire Conceptual Modeling for Building GIS Space-Time models. In: Proceedings of the GIS/LIS'94. pp. 860-869.

Yuan M. 1996. Modeling Semantical, Temporal, and Spatial Information in Geographic information Systems. M. Craglia and H. Couclelis eds. Geographic Informaiton Research: Bridging the Atlantic, Taylor & Francis. Pp. 334-347.

Yuan M., 1999. Use of a Three-Domain Representation to Enhance GIS Support for Complex Spatiotemporal Queries. In: Transactions in GIS, 1999, 3(2): 137-159.

Yuan M. 2000. Modeling geographic information to support spatiotemporal queries. In: Life and Motion of Socio-Economic Units. A.U.Frank, J.Raper and J.P. Cheyland. London, Taylor and Francis.

Yuan M. 2001. Representation Complex Geographic Phenomena in GIS. In: Cartography and Geographic Information Science, vol. 28, No. 2, 2001, pp. 83-96.

Yuan M. and McIntosh J. 2002. A Typology of Spatiotemporal Information Queries. In: Mining Spatiotemporal Information Systems. Shaw K., Ladner R., and Abdelguerfi M. (eds.). Kuwer Academic Publishers. Pp. 63-82.

Yuan M., 2008. Adding Time into Geographic Information System Databases. In: The handbook of Geographic Information Science, edited by John P. Wilson & A. Stewart Fotheringham. Blackwekk Publishing Ltd, 2008, Chapter 9: 169-184.

Zacks J.M. and Tversky B. 2001. Event structure in perception and conception. In: Psychological Bulletin, 127, pp. 3-21.

Internet links:

Architecture of Alte Pinakothek. http://www.pinakothek.de/alte-pinakothek/ (accessed: 03.05.2010)

Acknowledgments

I thank Prof. Dr.-Ing. Liqiu Meng for her excellent mentoring. She opened the door to the academic world for me. She encouraged my research work and offered me much appreciated freedom in research. Despite her busy diary she always managed to discuss emerging research questions and read drafts. She helped me to improve my English skills both in speaking and in writing. She invited me to many home parties, especially every Christmas Eve. It gave me the feels like at home.

I also express my sincere thanks to Prof. Dr.-Ing. Rüdiger Westermann for reviewing this thesis and acting as co-supervisor.

At the beginning of my research work, Prof. Dr.-Ing. Uwe Stilla was one of my first contact persons for any problems related to my tasks. He kindly offered me a working place for the first half year. His advice had much influence on my research.

Dr. May Yuan from the Centre for Spatial Analysis at the Oklahoma University shared her research experiences and gave me many valuable advices and comments for the part of spatiotemporal modeling in this thesis. With Dr. John McIntosh I had also many discussions about the spatiotemporal generalization.

Luise Fleißner helped me a lot for dealing with administrative staffs. She always kindly made a phone call for me whenever I needed her logistic support.

Many thanks to Lu Liu, Wei Yao and Hongbo Gong, with whom I had over thousand discussions about my research. I appreciate that they were always patient for discussions even when the topic was just a fantasy sometimes. I thank Lu Liu for sharing the flat with me, for spending time in the kitchen, for watching every football match together. Wei Yao is regarded as my moving library. Whenever I had trouble to get a reference, he could always help me.

For the most part, my dissertation presents results of the project “Integration and Visualization of time-dependent Features within 3D City Model”, which was funded by the International Graduate School of Science and Engineering (IGSSE). I gratefully acknowledge this support. Without it, this thesis would not have been possible.

I very much appreciate to work at Lehrstuhl für Kartographie (LfK) of Technische Universität München and do enjoy being a member of the research group for 3D visualization. Thanks to Dr. Jukka Krisp, Holger Kumke and Mathias Jahnke for their helps and collaborations. Thanks all the colleagues of the LfK for providing always a friendly and fun working atmosphere in the department.

Finally, I want to thank my parents, my two brothers and their families who have encouraged and supported me in following my research interests. And most of all, I thank my wife Qing Fu for being as she is, for her never ending understanding and supporting.

Curriculum Vitae

Fundamental information

Name： Hongchao FAN
Nationality: China
Date of Birth: 26 Oct. 1977
Place of Birth: Xiangfan, P.R. China

Education

1984/09-1990/07 Wangzhuang Elementary School in Xiangyang, Hubei Province

1990/09-1993/07 Huangji No.1 Middle School in Xiangyang, Hubei Province

1993/09-1996/07 Xiangyang No.1 High School, Hubei Province

1996/09-2000/07 Geoinformation system at Wuhan Univeristy（Bachelor for Engineering）

2001/09-2002/07 Language centre at the University of Stuttgart (DSH certification for German Language)

2002/09-2006/12 Geodesy and Geoinformatics at the University of Stuttgart (Dipl.-Ing.). Master thesis (Diplom Arbeit): Entwicklung der Verfahren und Vorrichtungen zur Erfassung eines störungsbehafteten Emfangssignals in einem videobasierten Regensensorsystem (in German). (Signal detection of a rain sensor system using the techniques of digital image processing). The thesis is conducted at Robert BOSCH GmbH in Stuttgart.

Experiences

2000/07-2001/09 Assistant engineer at the Hainan Bureau of Surveying and Mapping

2007/07-2010/09 Scientific assistant at the Department of Cartography of the Technische Universität München (Technical University of Munich)

Scholar exchange

2009/05 The State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS) at Wuhan University

2009/08-2009/10 Division of Geoinformatics at the Royal Institute of Technology (KTH), Stockholm, Sweden; Department of Surveying at the Helsinki University of Technology, Helsinki Finland

2010/05-2010/06 Center for Spatial Analysis at the University of Oklahoma, USA