6. Geographic Information Systems

Norbert Bartelme

Before addressing the diverse functionalities that can be found in Geographic Information Systems (GIS) today, a definition of a geographic information system, its purpose, and its general architecture shall be given. The different forms of GIS that are found in today's ever-expanding range of information technology tools will therefore be discussed in Sect. 6.1. The core of each GIS is constituted by analysis functions. They are the reason why a GIS is created in the first place. The last section in this chapter (Sect. 6.2) contains a list of the most common GIS functionality categories. A few typical examples of such analysis functions are described in more detail, to explain what geographic information (GI) is all about and how its digital form can be utilized to solve problems of geospatial nature efficiently, to gain insight into the processes of geospatial nature that influence many aspects of our life, and to arrive at decisions that are sound, explainable, and repeatable.

6.1 Architecture of a GIS

6.1.1 Information and Data

Before it is possible to provide a definition for a *Geographic Information System* (GIS) or, using a term that has become increasingly popular, a *geospatial information system*, we must first clarify what *information* means in the sense of information technology (IT) and how and to what degree it specializes into geographic/geospatial information. Basics for this discussion can be found in the introductory chapter of this book, where modeling and encoding are dealt with in detail and within the broad spectrum of information technology, therefore being also relevant for geographic information technology (Chaps. 1 and 4). The reader may wonder why we place such importance on the definition of the term *information*, since it is used widely in connection with – and sometimes identically to – digital data that are becoming ubiquitous in today's world. The Internet, for example, can be seen as an ever-expanding resource of digital data, much of which is geographical data. From this pool of data, we seek to retrieve information that will answer the questions or satisfy the criteria that made us search. The crucial *difference between data and information* is already visible. Any source on the Internet, any database, can only contain digital data – much of which is geographic data. Information results in our brains, when we interpret the data for the sake of our questions or problems to be solved. The Latin root of the word is *informare*, which literally means *to give form or shape* and this is usually extended into *to give form to the mind* as in education, instruction, or training. In contrast to data, information can be seen as an answer to a question (even if it has only been implicitly posed) that heightens the level of understanding of the inquirers or makes them capable of reaching a goal.

Figure 6.1 illustrates data, being pixels or points, lines, and areas, from which we retrieve the information that these data describe part of a city, with streets, a river, and other topographic landmarks. If we utilize a web tool for route planning, wanting to go from street A to square B, such services usually answer with maps and textual descriptions of the recommended route. Example data are the characters in a string of such text, or paragraphs, or the whole text. Data can also be individual pixels, or chunks of pixels, or a rectangular array of pixels constituting the map. Information is what we retrieve from the text, including its meaning for us: the recommended roads, turns, and routes. Information is also what we see on the map image. Our eyes register pixels (data) but our mind registers the best route

Fig. 6.1 Data (points, lines, areas) and information (streets, rivers, etc.)

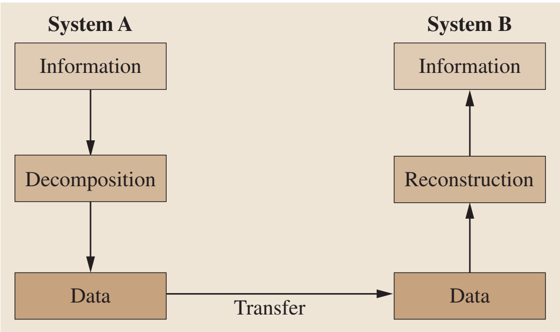

Fig. 6.2 Transfer of data and (implicitly) of information (Fig. 4.1 in Chap. 4)

(information). A third and still higher level on top of data and information would be knowledge: a concept that extends the concept of information into comparing, remembering, and learning. If one recognizes the city shown in Fig. 6.1, one has crossed from information to knowledge. Also, in the routing example just given, a knowledge engineer would combine the collective memory of typical route users, such as taxi drivers, and use their experience, which is sometimes difficult to model, about typical recommendations for times of day, seasons, or vehicle types to arrive at a complex knowledge system that becomes smarter with every use. Knowledge is dealt with in Chap. 5.

Returning to the difference between and blending of data and information, we arrive at a point that is crucial for information technology, for any information system, and therefore also for *geographic/geospatial information technology* being implemented in *GIS* or – in a web-enabled environment – in *GIS services*. One of the things that information systems have to do is the transport of information from system A to system B or from user C to user D, yet all that can really be done is transport of data (Fig. 6.2, Fig. 4.1 Chap. 4). This means that information in the sending system has to be decomposed into chunks of data that can be – and must be – transported to the receiving system, where they can be reconstructed into information. The last and most relevant, reconstruction step has to be taken by the user sitting in front of the screen in reading and interpreting text or a map. It is clear that this process, going in opposite directions at both the sending and receiving end, can only in theory result in equivalent information content on both sides. In all practical cases, information loss has to be taken into account. The more elaborate the information construct, the greater the danger of information loss. Information that is highly interconnected in a network, information that is highly structured, and information carrying strong application-dependent semantics will be especially vulnerable in such a sequential stream of basic data entities. One of the most relevant consequences for the design of any information system is therefore the necessity to place great importance on the ways in which users are assisted when performing this reconstruction of data into information they need. Of course, data may also show a structure, for example, a sequence of value pairs carrying some common attributes. Reading the attribute values *River* and *Danube* and noting that the value pairs are given by real numbers, we interpret this data as the River Danube. Likewise, a linear concatenation of blue pixels on the screen may spawn the same interpretation, leading us from data to information.

To sum up, the difference between information and data relevant for GIS is characterized as follows:

- *Data* is what is stored and transported, such as strings of characters or pixels, or defined structures thereof.
- *Information* is a result of the interpretation when visualizing or analyzing data.

6.1.2 Geographic (Geospatial) Information

Having clarified the concepts of and differences between data and information, we can proceed to geographic information, or geo-information, or geospatial information. The term *geo-information* is mostly used in German-speaking countries. *Geospatial* is a term that successfully tries to bridge the gap between geographical information in the strict sense, for example, digital terrains, and spatial information, for example, a model for a three-dimensional (3-D) structure, a building, or a bridge. In a widened perspective, this term denotes not only things that exist (or are being planned) on some location of the Earth's surface, but also events such as traffic congestions, floods, and yes, also events in everyday language, such as an open-air festival. For the rest of this chapter and for the sake of simplicity, let us assume that the terms mentioned are all synonyms. All these examples share one basic aspect: they exist or happen somewhere on the Earth's surface and they have a spatial extent; in many cases they also have a temporal aspect, a position in time, and a temporal extent. We can therefore speak of a four-dimensional (4-D) continuum (three spatial dimensions and one temporal dimension) that characterizes geographic data and geographic/geospatial information.

At this point of the discussion, we may stop and consider what types of information are nonspatial or nongeographic, in the sense that they are independent of where and when they are relevant. Strictly speaking, there is hardly any such information that is totally void of any geospatial context. This context may not always appear in the model of such information, but it is inherently there. In cases of land use, traffic and transport, climate, agriculture, and economics, the spatial aspect is evident and in many cases also modeled. In other cases such as linguistics, philosophy, and literature it may be less evident and it is seldom modeled. At the end of the range are laws of mathematics, physics, and chemistry, which are almost independent of the location where their validity is tested. This seemingly philosophical discussion about the extent of geospatially relevant application domains has – high-brow as it seems at first sight – some very practical consequences for the design of a GIS. The more straightforward the geospatial interpretation, the clearer the modeling of the geometric and topological properties of such information. For land use and cadastre, well-known geometric entities exist, and this is also the case for roads and intersections in traffic and transport. However, the geospatial extent of a dialect in linguistic GIS applications requires a different approach, since this extent is fuzzy.

To sum up, geographic (or geospatial) information is characterized as follows:

- *Location, extent, and coverage* are aspects of prime importance for such information.
- However, *geometrical concepts* may – depending on the application – be concise or fuzzy.

6.1.3 Geographic Information System Definitions

Let us now cite some definitions for the essence of a GIS. There are scores of definitions from which to draw. We prefer some of the older definitions, because they pertain to the still-existing traditional GIS (in contrast to newer paradigms such as geospatial web services). The references to older publications can be found in the references of this chapter, followed by more recent publications of the mentioned authors. Also, the choice of definitions used here reflects the fact that each of the following citations carries a flavor that distinguishes it to some extent from the other definitions.

- It serves for capturing, storing, analysis, and visualization of data that describe a part of the Earth's surface, the technical and administrative entities, as well as findings of geoscience, economics, and ecological applications [6.1, 2].
- It is an information system with a database of observables of spatially distributed objects, activities, or events, which can be described by points, lines, or surfaces [6.3].
- It is a comprehensive collection of tools for capturing, storing, retrieval, transformation, and visualization of spatial data of the real world for special applications [6.4].
- It is an information system containing all spatial data of the atmosphere, the Earth's surface, and the lithosphere, allowing the systematic capture, update, manipulation, and analysis of such data, based on a standardized reference frame [6.5].

- It is a system for decision support which integrates spatial data in a problem-solving environment [6.6].

Further definitions of GIS may be found in [6.7]. Depending on the point of view, a GIS can be seen as

- A collection of spatial data plus the corresponding functions for storage and retrieval
- A collection of algorithmic and functional tools
- A set of hardware and software components needed for the handling of geospatial data
- A special type of information technology
- A gold mine for answers to geospatial questions
- A model of spatial relationships and spatial reconnaissance.

6.1.4 Classical GIS and Recent Modifications

In the classical sense, an information system consists of a database representing the inner core of the system, being managed by a database management system (DBMS), and of an outer shell of tools that can be utilized by the user for manipulating and analyzing this data (Fig. 6.3). This definition, taken from [6.8], is one of the earliest and best known. Chapter 3 is devoted to databases; the reader is invited to check further details there. Here, we concentrate on the impact of databases on GIS. Taking the definition just given, traditional GIS conform to this concept. While in the early years of GIS technology the database usually followed a standard relational database approach, this concept has in the sequel been expanded to an *object-relational database management system* (ORDBMS) to enable geospatial data types and geospatial predicates, as well as the means to deal with the ever-growing volume of geospatial data collections by supplying methods for organizing, searching, and retrieving large datasets.

There is no such thing as a general-purpose information system. Data are abstractions of reality. An abstraction process can lead to different results, depending on the view of the abstractor. Therefore, each information system is *application dependent*. The degree of dependence varies for the different components of an information system. The data will often lend themselves to more than one purpose. This is also advisable for reasons of practicality, efficiency, and cost. The functions will depend more heavily on the application that is envisioned. Typically, a GIS offers functions for storage and retrieval, query, visualization, transformation, geometrical and thematic analysis, and more (see also the next section for more details). Even if these functions are more or less available in every GIS, their specific form depends to a large extent on the application. As an example, visualization in a cadastral application will have prerequisites that are quite different from those for a 3-D city model. The same holds also for the other types of functions.

The traditional setup of a GIS has been modified in several ways, due to the arrival of new technologies and new concepts. The arrival of the Internet, of *web-based service* approaches, tools, and applications, has greatly influenced and modified the whole IT arena. The second boost has been initiated by *mobile technology* and the miniaturization of hardware components. Therefore, also the paradigms of GIS have changed, and the architecture of a GIS nowadays is quite different from what it was a few years ago (Fig. 6.4). Data are no longer restricted to the user's primary domain of interest and control, but they can in principle be imported from everywhere, anytime, and to any device. Often, a normal browser can perform GIS tasks, even if this is nowadays restricted to simple GIS functions such as displaying, zooming, and panning.

We talk about *ubiquitous geographic/geospatial information* as the universal availability of geographic information as seen on mobile devices such as cellphones, where maps, satellite images, positioning, routing services, and even 3-D simulations are gaining an ever-larger segment of the consumer market. (For location-based services on mobile devices see Chap. 21; for ubiquitous geographic information see Chap. 18.) Also, in contrast to earlier times when a GIS consisted

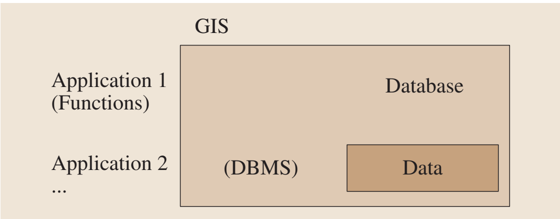

Fig. 6.3 GIS in the classical sense

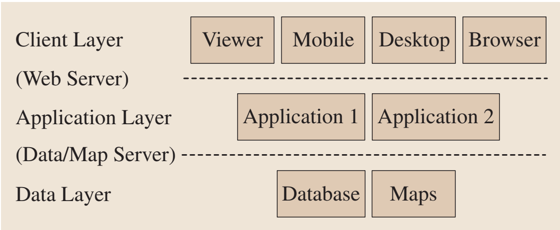

Fig. 6.4 GIS in a client–server architecture

of a well-balanced combination of hardware, software, and data components, today the borderlines between the functionality of an ordinary Internet browser and GIS functionality is often fuzzy. Likewise, content is displayed that may, via web services such as web map services (WMS), be composed on the fly, coming from different sources but having the appearance of a combined dataset, whereas in many cases the data themselves are not transported but are rather visualized on the fly. The data remain at their various home localities, which is an asset as far as currency and lack of redundancy is concerned.

The question of whether a cellphone can be seen as a GIS is a rather theoretical one and certainly does not interest the general public. However, these new developments have opened new user segments. They do not replace traditional GIS users in the domains of public administration, utility companies, and communities in the diverse domains of science that are deeply rooted in models of geospatial nature. However, these new web-based and mobile amenities have become important additions to the core elements of GIS technology, especially since they lend themselves to more user-friendly handling that is less dependent on office hours and strict work protocols. To be able to work from home, to perform part of the job requirements in different places, or to integrate via web services the strengths of other experts or other systems greatly enhances the value of geographic information and the associated technology as a whole.

Another change to the concept of a GIS, less spectacular than the web-based and mobile aspects but still very important, was introduced with *object-oriented modeling* and the corresponding methods in programming and database structures. Object-oriented methods bridge the gap between data and functions by the definition phase, prior to the insertion of data. Also, for each class of objects, the appropriate functions are defined. Object-oriented methods have bridged the gap between data and functions. For each class of objects an individual and particularly suitable function may be defined in order to serve for data capture, storage, and retrieval. This relieves the burden on the functional shell that is built around the core database. One step in this direction has already been explained in the previous paragraphs. Object-relational database management systems relieve the outer shell of many typical definitions of data types as they are needed in geospatial applications, as well as many typical operations that conform to such data types. However, they are still based on the relational concept of tables, with rows (for the objects) and columns (for the attributes of such objects). In addition to the pure relational elements of a table, a geometry column comes into effect, which takes care of the geospatial aspects mentioned. Object-oriented databases go one decisive step further: they are no longer based on relational tables. Instead, object classes take over the role of principle building blocks, adhering to principles of data abstraction, encapsulation, modularity, polymorphism, and inheritance. Object-oriented GIS have up to this point gained no wide acceptance in the arena of GIS products; however, they carry great promise since GIS – in contrast to many other information technology domains – characterize structures that are often more complicated and less unified. As an example let us look at the way a building as an object can be constructed of many different components (base area, walls, roof) that can again be decomposed into still simpler objects. Each level of this decomposition can be seen as a different object class. In contrast to this GIS domain, the books in a library constitute a rather flat structure with components that are uniform. An information system dealing with entities from this domain lends itself more easily to a strictly relational database management approach.

To sum up, the modifications that have in recent years been transforming classical GIS into newer forms of geospatial analysis tools are as follows.

- *Web-based and service-oriented approaches* have led to a client–server architecture.
- *Mobile technology* brings ubiquitous GIS to hand-held devices, opening a whole new market.
- *Object-oriented concepts* have partly entered via object-relational databases.
- Traditional GIS also use benefits from *mobile and service-oriented technology.*

6.2 GIS Functionality

6.2.1 Categories

The list of GIS functionality categories can be arranged in many ways. There is no ideal arrangement, since we eventually have to arrive at a sequential setup, whereas many functions interact with each other in a manner that corresponds to a network rather than to a sequential list. However, a list is a way of ordering chunks of any matter that is easy to grasp for our minds. Our list, being simple, follows the lifecycle of data from creation via structuring and storage and eventually to analysis and presentation. Each GIS will contain representatives of the groups listed, although in different quantity and quality, depending on the respective application range. In fact, GIS can be categorized by the degree of importance they allot to the different groups, giving rise to categories such as

- Data capture systems
- Administration systems
- Analysis systems
- Presentation systems

A *data capture system* will put greatest emphasis on the first phase in the lifecycle of geographic data. As an example, let us consider a topographic mapping authority in any given country that is in the process of converting large quantities of paper maps into digital form, and not only in a graphical pixel-based form such as an electronic image. For instance, road features shall be extracted from this image and their geometry and several semantic attributes are to be stored in the database. Such a job usually requires semiautomated preprocessing including scanning and image processing, and subsequent interactive editing. This system, even though data capture is its prime purpose, will also need a certain amount of administration for the data that have been captured. It will need some analysis function (for example, to find out whether all road segments indeed connect in a topological way, and if not, to perform the creation of topology). Certainly it will also need some presentation and visualization tools to assist the operators in their job, giving them feedback on what has already been done and what still needs to be done. So, these other three categories of administration, analysis, and presentation play a role, albeit minor compared with the data capture role.

An *administration system* puts the main focus on long-term storage of data, keeping them in shape and up to date. Here, examples include land information systems with cadastre that have to keep such data available for years and decades. They administer the data. Occasionally, though, some minor data capture is necessary for updating purposes. Analysis functions and presentation functions help in the maintenance of such administrative systems. So again here, we have a main focus and three subordinate interests.

In an *analysis system*, there is a corresponding situation. These systems put the main emphasis on one or several of the types of analysis that are discussed in the following sections; for example, a utility company providing power supply to households throughout a province may periodically need to analyze the total power consumption, the statistical distribution of peaks in consumption, the typical breakdown rate of power supplies during storms, the capacities of long-range connections, the overall flow capacity of a network, and others. Also, such an analysis system needs to some extent tools from the data capture, administrative, and presentation domains.

Finally, a *presentation system*, as we know it from many Internet applications that provide only the final graphical visualization of a given setup, conforms to what has been said about the four categories of systems. Certainly, normal Internet users cannot interact with it any more than zooming in or out, panning, and selecting and deselecting layers. However, behind the web barrier, also such a system needs some data capturing for editing geospatial data, and it needs administrative and analysis functions to a certain extent.

6.2.2 Data Capture Functions

Chapter 9 is exclusively devoted to data capture in its different forms, describing for each category of capturing techniques the requirements, the current state of the art, and the results. Also, data capture depends essentially on the model of reality to be reached. Modeling is extensively discussed in Chap. 1. Therefore, at this point, we can restrict considerations to the conceptual characteristics pertaining more or less to all data capturing techniques, as well as their impact on the essence of a GIS and on the expectations that can be built on such a system.

Bringing geospatial data into a GIS is one of the most challenging tasks among all the functional categories explained in this section. First, it has to encompass a large number of capturing technologies such as

- Global positioning systems (GPS) and geodetic surveying
- Laser scanning
- Photogrammetry and satellite-based remote sensing
- Any other technology

Second, the quality of all subsequent processes to which these data are subject depends to a great extent on the amount of care and attention paid to the capturing process. To put it plainly, any output cannot be better than the original input. No matter how clever and intricate the subsequent editing functions, the transformation and structuring functions, or the analysis and visualization functions, they cannot undo a careless or sloppy way of getting the primary input. So, while all other functions described in this section can in principle be redone if it is decided that they do not work in the intended way or that they produce poor results, this easy way out does not apply to data capturing, for several reasons. It is one of the most costly tasks in the lifecycle of data, involving a large amount of manpower, organization, and preplanning. Therefore, in many cases the data are acquired once and for all. This is especially valid for data being captured by surveying methods, be it traditional, as for example using a theodolite, or be it by satellite-based methods. In any case, fieldwork is necessary. Second, the original setup may be lost. Consider a remote sensing campaign giving satellite imagery or also airborne techniques. Such capturing methods rely heavily on the season of the year because of foliage, draughts, precipitation, or weather conditions. Third, data may already be in use by several applications, so it may not be possible to simply renew the data without risk for these applications.

Another aspect that pertains to data capturing in general is the fact that it can be seen as a process that maps the real world to a digital representation of the real world (Fig. 6.5). Often it is argued that not the whole real world is mapped, but rather only a *Universe of Discourse* (UoD), constituting the subset of the real world around us that corresponds to a specific application in mind. As an example, in land cadastre applications, we talk about parcels, usage, property, survey points, etc., while in route planning applications we talk about start and end points, via points, routes, road segments, intersections, toll roads, traffic regulations, etc. A third example is given by a municipal authority that administers the situation, infrastructure, and make-up of streets, such as the number of lanes, the sidewalks, lampposts, gutters, paving, and sewage system serving this part of the street, etc. This example shows that any street of the real world may be considered in a multitude of UoDs, since it can be seen as a parcel of public property (geometrically an area), a connection in a traffic network (geometrically a line), or a combination of individual features which in their combination make up the publicly accessible spaces in a city. At this point in the discussion it becomes apparent that each UoD eventually needs to be mapped to a different *conceptual data model*, leading to different data structures in a GIS. This, in turn, tells us that there is no such thing as a universally applicable data model. Instead, each application requires its own data model, its own data structures, its own geometry, its own semantics, and its own accuracy requirements. Of course, this will not be practical and also not financially feasible, let alone the problem that many future applications may not be foreseeable at the time when data are being captured. So, in practice, a compromise must be reached between theoretical prerequisites and practical restrictions. However, it is important to stress this fact that basically every application needs – before it can be successfully used – a lot of special considerations with respect to the data available and the techniques used during their capturing phase. This, by the way, is an essential ingredient in all arguments about the importance of *metadata* (Chap. 12), since metadata can be used to render – among other things – a detailed *report* on the lifecycle of data, especially the conditions under which they were born (i. e., captured).

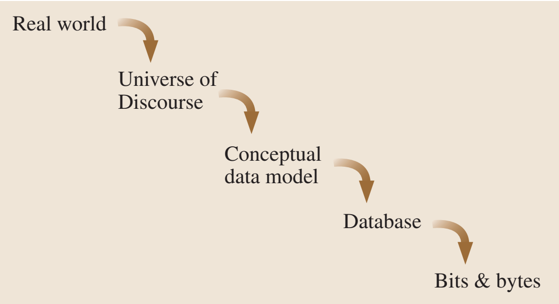

Fig. 6.5 Modeling

Data capturing will create digital data constituting a mapping or partial replica for an application-dependent subset of the real world. It cannot be a replica in the strict sense. It is rather an *approximation to reality*. The real world is characterized by an infinite number of aspects, only a finite number of which can be taken into account. The quality of the resulting data will greatly depend on the way this subset is chosen. This pertains to all aspects relevant for the information involved.

For an example, let us consider a mountain road that shall be mapped into GIS data (Fig. 6.6). Apart from the fact that an information system database is finite, several other choices need to be made. The data geometry will in most cases be 2.5-dimensional, observing the fact that north and east have far more importance than height, even for a mountain road. While the former are mapped to a two-dimensional (2-D) coordinate system, the height is often carried as an attribute. This disparity between the importance of the first two dimensions and the third one is signified by the value 2.5. Of course, such a strategy can only work if any given line in the resulting geometry does, for each north and east

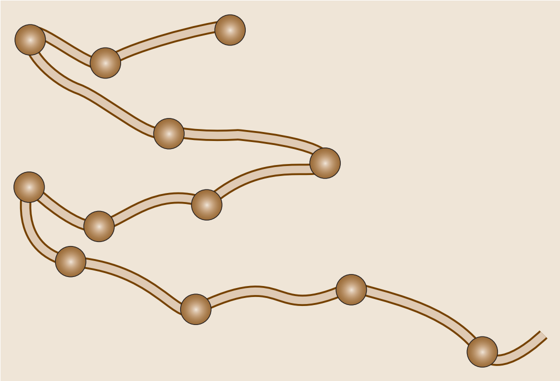

Fig. 6.6 Modeling the geometry of a mountain road

value, only have one height value, so that there can be a unique and invertible mapping between a 3-D surface and a 2-D plane. Such an assumption makes sense for mountain roads, but it does not make sense for subway lines, for example.

Having taken care of the third dimension, we still have to decide upon the remaining two dimensions. A road is a general surface in reality and now we have reduced it to a plane surface. In most cases, the modeling process will further reduce it to a line, being the center line or axis of the road, and again taking care of the second dimension by an attribute which is the width of the road at certain points. So, the second dimension has the same fate as the third dimension before: it ends up in attribute values, leaving the geometry of the mountain road as a one-dimensional line. Such methods can be seen as simplifications of the real world. The art of modeling consists of simplifying without overdoing this. The last simplification step would be the selection of certain points along the road axis to represent the road in its digital replica. It is impossible to choose infinitely many points of course, and it is not recommendable to choose too many points (because this would make the data volume explode), neither to go to the other extreme (if there are not enough points, the information about the cumulative presence or absence of turns, being a very poignant characteristic of a mountain road, will not be available). So, again here the art of modeling and the art of capturing just the right amount of data at just the right places will determine the success of any further applications built on that dataset.

The previous discussion provides a short insight only. However, it can easily be transferred to other domains such as sensors in photogrammetry and remote sensing and other air- and space-borne sensors as well as to mobile mapping systems in general.

To sum up, any data capturing technique must, for the maximum benefit of resulting data in a GIS, adhere to the following principles:

- *Suitability* for given or envisioned applications.
- *Reduction of the richness of a UoD* by just the appropriate amount of simplifications.
- Observing *best-practice quality criteria*.

6.2.3 Update Functions

Creation of data and their insertion into a GIS will many times be followed by amendments, corrections, updates, and deletions. Almost any information system and therefore also a GIS will be created with long-term usage in mind. The data capturing process is characterized by the requirements of high *geometric and semantic quality* as well as *stable and suitable structures*. This, for example, includes the structure for features and for topological networks. See the previous section for more details on data capturing and the following section for more details on structuring. The consequence of all this is the well-known fact that data capturing in general is a costly task. Costs arise partly due to the sheer volume of data and partly because of time-consuming capturing tasks that involve a considerable amount of manpower. If data are found out to be erroneous, ill-structured, or incomplete, it is seldom advisable to capture them anew; rather they should be subjected to amendment and update procedures.

This sounds simple, but if those amendments are done carefully and in a *user-friendly* way, this task may turn out to be quite difficult. Any primary input is far easier to handle than subsequent amendment processes. In the initial data capturing phase, data usually arrive en bloc in the database. Consider as an example the insertion of data that model road geometries (Fig. 6.7), since we have used this example also in the previous section. As outlined there, in most cases the geometry of a road segment going from one intersection to the next will consist of a sequence of points denoting the axis of the road, while the road width at each of these points is modeled by an appropriate attribute value. In the same way, the height value is handled. This procedure is straightforward if the whole road segment is inserted into the database in this way. Now consider the case that an additional point has to be inserted, or one point has to be eliminated from the sequence. While for the user it is clear what has to be done, this cannot be so easily achieved by the system, because any insertion in the middle of the sequence has to be handled differ-

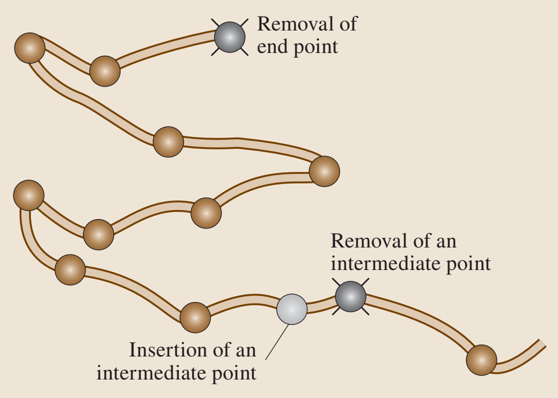

Fig. 6.7 Updating geometry and topology

ently from an insertion at the beginning or at the end. The problems can be overcome, but the challenge is far bigger, since the user interface shell of the GIS must reckon with a score of different situations that can arise due to erroneous inputs.

While such questions arise at the microscale, i.e., for individual features inserted or updated, there is also a macroscale problem that has to be analyzed so that eventually ways can be found how it can be dealt with. GIS have been designed to provide a long-term basis for geospatial analysis. GIS data will have a long lifespan. Cadastral data are a prominent example, but also many other geospatial data describe objects or phenomena in the real world that have existed for a long time, and whose existence is likely to continue into the future. These long lives of all the individual GIS features did not all start at a common *big bang*. Rather, their lifecycles meet and overlap on the real world axis of time in an unforeseeable manner. Additionally, we have to observe yet another time axis, corresponding to the time for a real-world object or phenomenon that its GIS data replica was inserted into the GIS. To add yet more complexity to the problem, we have to consider the fact that GIS data originate from many different sources at different times. This is a great bonus, a great strength of any GIS, but also one of the main causes of problems arising during the update process, due to incompatibility issues. When different data collections from different data sources have to be merged, this can be seen as an update process at the macroscale.

All the aspects mentioned will add to the complexity of GIS update functions. Let us again make use of the road example. This road may be part of a provincial road network that has been digitized at a certain point A in time and since then has been present in the GIS database. This road network has been checked for consistency, in both the geometric and topological sense. When another layer of GIS data is added to the system, containing roads that belong to another administrative hierarchy, say a road maintained by a local township administration, it is likely that the combined network of provincial and local roads will have a topology that is more detailed and will typically contain more intersections (topologically speaking, they are called *nodes*) than the provincial network. An update function in this general sense, merging two datasets that each had their own special lifecycle, quality measures, and application domains, and thus creating a new combined dataset where all these criteria need to be harmonized, requites quite a bit of effort in both the database shell of the GIS as well as the human–computer interface shell.

To sum up, update functions can be characterized in the following way:

- They are much more complex than the original input functions.
- This is true for the database shell, but even more so for the user interface shell.
- Update functions cover geometric, semantic, and structural corrections on the microscale (for any feature).
- They also cover the macroscale when different datasets need to be merged.
- Different time scales (real-world and GIS-insertion times) complicate the process further.

6.2.4 Structuring Functions

Data enter a GIS in a more or less basic structure, as points and lines for vector-based applications and as pixels and images for raster-based applications. Such basic data are the building blocks for more complex structures that are needed to model the real world in a way that can be put to use for typical GIS applications. These functions assemble basic entities into more complex structures. Let us again use the illustrative example of road geometries that has been used in previous sections. Starting out with individual points that are being captured by GPS methods, by digitizing points from the screen, or by some other method, it is necessary to connect two adjacent points, forming a line. This may seem self-evident at first glance, but it is not. Let us explain the difficulty by looking up at a starry sky on a cloudless night. What we see can be considered as points – the stars. Forming clusters that correspond to signs of the zodiac and then – for each sign – connecting the stars in an appropriate order is not an easy task. Should such a structuring process be done in even a semiautomated way, the difficulty becomes still greater by several orders of magnitude. Similar things can be said for the interactive and semiautomated structuring of GIS data. It takes a lot of experience and a concise *set of rules* – some of them pragmatic or based on experience only – to make this process work. The order of points forming the axis of our example road can in many cases only be guessed, unless we have additional information that can be used, for example, a digital terrain model. Ambiguities may be avoided by using some assumptions that make sense, for example, that a road is more likely to follow the terrain than to go up and down in a zigzag manner. Often, a decision can also be made easier if we consider *topological rules*: a road may not cross itself; a figure-of-eight shape is usually considered to be a topological error; undershoots and overshoots need to be corrected; ambiguities for nearby points need to be resolved by averaging them, etc. Structural deficiencies have to be eliminated (Figs. 6.8, 6.9).

Once the order of sequence has been determined, the whole road can be structured, like forming a string of pearls. What has not been decided yet is the geometric form of the connecting lines. They may be straight lines or curves in the real world. For the case of a mountain road, curves are more likely. If we consider streets in a US suburb, straight lines are more likely. If we create a model for rivers instead of roads, then straight lines between adjacent points are almost unthinkable. This shows the vast range of considerations that have to be included in any structuring process.

In a similar way to the one-dimensional structuring of roads, rivers, etc., two-dimensional (2-D) structures may be formed out of either points or lines. If the points from our starry sky example denote the corners of buildings that shall be modeled in a map-like two-dimensional way only, i.e., neglecting heights, storeys, etc., it will be necessary to arrange the corner points for each building in a closed ring sequence. The simplest form of a building in this two-dimensional projection will be a rectangle. Again there will be rules that have to be observed as in the previous case, where we did not allow figure-of-eight shapes. We also have to consider other topological rules of adjacency, nonoverlapping, etc., and there will again be restrictions based on experience. As an example, we can rule out buildings with an extremely elongated shape or buildings that would have an area of negligible size.

With each dimension added, the complexity of structuring increases. Three-dimensional structuring, as is the case, for example, in 3-D city models, represents the most complex form of geometric structuring. We can begin the structuring starting out from points in 3-D space as they originate from surveying or laser scanning. Those points will typically be the ground corner points as well as characteristic points of the building fronts and roof. Points can be structured into lines, for example, a roof line or a base line. Lines again may be used to form a two-dimensional structure such as a house front or part of the roof. Then all these components can finally be assembled into a 3-D structure. This structuring follows an approach that considers only the outer cover of the building. There exist several alternatives. For details on 3-D city modeling, we refer to Chap. 10.

Structuring is not restricted to *geometric* and *topological* aspects. Also the *semantics* as well as *time* will often give rise to structuring. An administrative area that is built up from several topologically independent patches is one example, as is the mainland of the USA together with Alaska and the Hawaii Islands (Fig. 6.10).

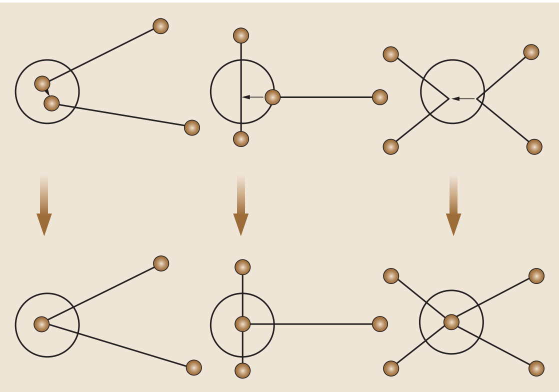

Fig. 6.8 Structural deficiencies in linear topology and how to amend them

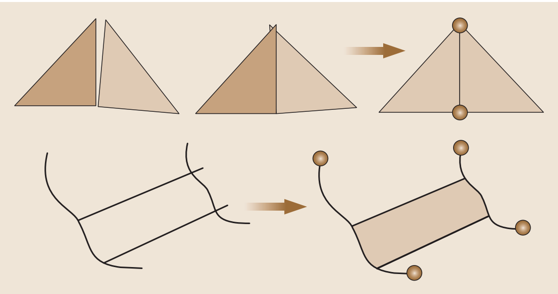

Fig. 6.9 Structural deficiencies in area topology and how to amend them

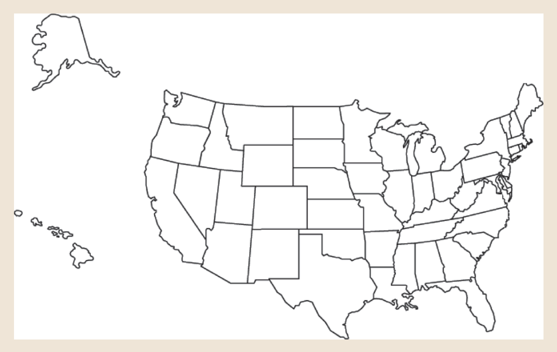

Fig. 6.10 Structuring according to semantic criteria (US states)

So far, we have only discussed structuring for vector-type GIS data. Considerations for raster GIS data are just as important. Let us, for example, consider an image stemming from an airborne campaign and let us assume that this photo has been subjected to rectification processes, yielding an orthophoto. The human eye can differentiate on this photo visible features such as rivers, roads, and built-up areas. If we now try to let a GIS structuring function do this, at least in a semiautomated way, many of the above-mentioned questions arise also in this context. Grouping the pixels of the photo into clusters that belong together can be seen as a classification process. There are many pixel-based and object-based methods and tools to perform such structuring (Chap. 10).

To sum up, structuring functions can be characterized as follows.

- They *assemble basic components* stemming from data capture into high-level application-friendly structures.
- *Geometry* and *topology* as well as *semantics* and *time* give rise to needs for structuring.
- Structuring must adhere to *application-dependent rules* and often to *informal knowledge* (experience).

6.2.5 Transformation Functions

The term *transformation* in a GIS environment is widely associated with *coordinate transformations* (Fig. 6.11). An essential benefit of using GIS is the opportunity to bring many different types of geospatial data together under one roof, enabling their comparison in terms of their identical, overlapping, or nearby locations and coming to conclusions about interactions of geospatial aspects. Bringing together different worlds of data is only possible if the spatial references of all data are resolved so that they can be transformed onto each other. This signifies that coordinate transformations are really at the core of any GIS. Mostly such processes are triggered implicitly, without waiting for an explicit command from the user. However, in order to be able to discuss the benefits and also risks of comparing or even combining disparate datasets, it is necessary to acquire a basic understanding of coordinate transformations and the reference systems and reference frames on which they are built. Geodesy deals with this, and consequently it is discussed in detail in Chap. 8. Coordinate transformations are not only necessary in the inner core of a system, where they largely go unnoticed by the user unless they fail to work properly. Also on the user interface level, any zoom or pan involves aspects of coordinate transformations, and therefore they are to be discussed in a comprehensive list of GIS functions, even if such zoom and pan functions are considered as very basic and hardly worth mentioning. However, it is not difficult to find an example where zooming becomes slightly more complicated: large-scale maps are typically presented in a plane coordinate system that is rectangular, without too much loss of accuracy. In contrast, regional or state maps usually have to take the Earth's curvature into account if they are not to look awkward. So it can very easily happen that GIS users notice the relevance of choosing a coordinate reference system that better suits the current zoom.

Coordinate transformations are not the only GIS processes that are modeled by transformation functions. Another important representative of such functions is the *conversion* between *raster data* and *vector data* (Fig. 6.12). These two modeling strategies leading to two different types of data structures have in the past often led to two different, and to some extent incompatible, worlds. Often, a GIS was seen mainly through the vector-type lens, the raster world only being present as a backdrop image. In recent times this has changed dramatically. Not only is raster imagery becoming increasingly available and – despite its volume – manageable, but also a large number of image-processing functions can be put to good use for GIS data. Let us just cite one example. Cartographic generalization is one of the

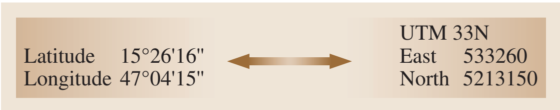

Fig. 6.11 Coordinate transformation

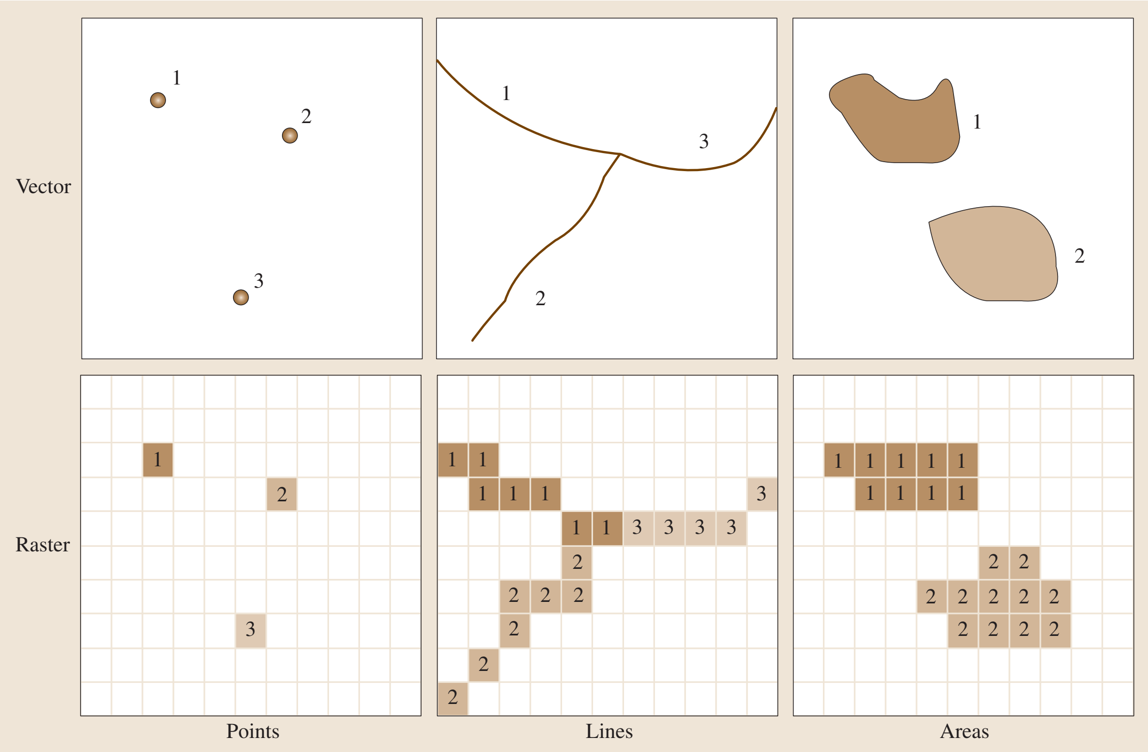

Fig. 6.12 Vector and raster data

key issues in the area of visualization. Image-processing techniques lend themselves easily to generalizations based on raster data. It is therefore desirable that – at least temporarily – vector data are transformed into their raster counterparts. Of course the inverse techniques are also needed. We will not go into detail here, because all this has been successfully used for many years in image processing; the reader can find ample coverage in the literature, for example: *Pavlidis* [6.9], *Rosenfeld* and *Kak* [6.10], and *Gonzalez* and *Woods* [6.11].

In this section dealing with transformation functions, it must also be noted that other GIS functionalities can also be brought into this discussion. This includes interpolation and approximation functions and adjustment theory, computer-aided design (CAD) construction functions, as well as transformations in image and raster data processing (Chap. 2).

To sum up, transformation functions can be characterized in the following way.

- In their form as *coordinate transformations*, they are ubiquitously present in GIS and of overall importance.
- *Transformations between vector- and raster-type data* are increasingly taken from image processing.
- Tools from *mathematics, approximation and interpolation theory, and adjustment theory* can be used.

6.2.6 Storage, Checking, Archiving, and Data Transfer Functions

In Sect. 6.1, we defined information as an answer to a (sometimes implicitly posed) question that heightens the level of understanding of the inquirers and/or makes it easier for them to achieve a certain goal. For a GIS, the metaphor of an engine or a vehicle that brings us nearer to the solution of a geospatial problem is therefore appropriate. Any vehicle needs an infrastructure it can run on, for example, roads, railroad tracks, waterways, etc. For a *geospatial problem solution engine*, geographic data provide such an infrastructure. Roads, railroads, and waterways are essential for the functioning of modern society and therefore considerable investments are put into their maintenance. Likewise, a *geospatial data infrastructure* needs to be well maintained, since it provides a basis for current and future GIS technology, for *geospatial solution engines* in a wide range of potential implementations, and therefore represents a considerable value. Effort, time, and cost have been invested in creating it (Chaps. 14 and 30). It will have to be available for a long time to come. Thus, proper maintenance is necessary. There are scores of functions to make data fit for long-time storage, for lifelong checking on consistency and other quality parameters, and for interoperable use of geographic data including transfer and services. The metaphor we have invoked has in recent times become widely used due to initiatives for geographic data infrastructures on a global (Global Spatial Data Infrastructure, GSDI), national (National Spatial Data Infrastructure, NSDI), and regional level. The European Infrastructure for Spatial Information in Europe (INSPIRE) initiative that is evolving into a framework of harmonized national laws supporting *interoperability* (Fig. 6.13) can be seen as falling in between the global and national levels. More information on geospatial data infrastructures and interoperability can be found in Chaps. 14 and 30.

The range of GIS functions that shall be mentioned in this context starts with all database operations that are available at the user interface level. Database insertions, updates, and deletions are often implicitly invoked without needing an explicit command from the user. This is the case, for example, when data capture or update functions are executed. When users manipulate geographic data on the screen, they can safely assume that all their actions at the user interface level are mirrored by appropriate functions at the lower level of database management. However, there are cases when users explicitly need to invoke storage and archiving functions. This is comparable to a desktop office environment where periodically the current state of progress is archived, while the user at certain stages of the process may explicitly want to trigger a save or even an archiving function. While the former simply saves a current state in order to be able to document it or to safeguard against system failures, the latter provides a means to keep track of several states in time, a *time machine* as it is called in some systems. Geospatial applications are often characterized by the need to keep track of past situations or situations that have become obsolete or historical. Think, for example, of cadastral applications, where not only the current ownership situation is of interest, but also the history of a parcel and how it has evolved over time, including all partitioning and merging operations in the past.

Another important category of functions to be mentioned in the context of this section is *checking and validating*. Let us again take up the metaphor that compares geographic data infrastructures to roads and railroads. Periodically, such an infrastructure must be scrutinized in all its aspects, for example, in terms of desired geometric quality, topological consistency, and semantic and temporal validity. Additionally, the structuring rules need to be checked for compliance. Such functions may include small corrections that shall be done automatically or under user surveillance. Consider, for example, a regional directory of emergency services. Any new entries and also deletions and updates call for interaction by the authority responsible for keeping and providing the directory. Changes in some attributes, such as the update of a telephone number, can be traced automatically and therefore also be corrected without needing interaction.

This leads us to the last group of functions that need to be discussed in the context of this section: those dealing with the *transfer of geographic data* and *services*. Historically, the transfer of geographic data from one system to another was one of the main challenges of GIS technology. The need for interface standards spawned interdisciplinary and international standardization initiatives, for example, on the CEN and ISO levels. CEN/TC 287 *Geographic information* and ISO/TC 211 *Geographic information/Geomatics* paved the way for many advances in standardization. The Open Geospatial Consortium (OGC) put the term *interoperability* on its banner. The ISO/CEN and OGC developments together provide some essential steps further, beyond the simple data exchange paradigm of the past. Interoperability denotes the capability of different systems to work to-

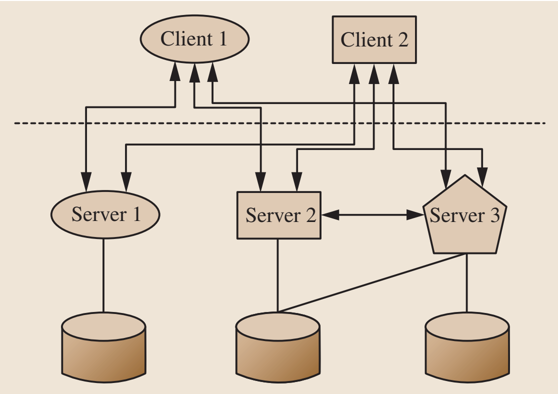

Fig. 6.13 Interoperability ensuring standardized access to different servers

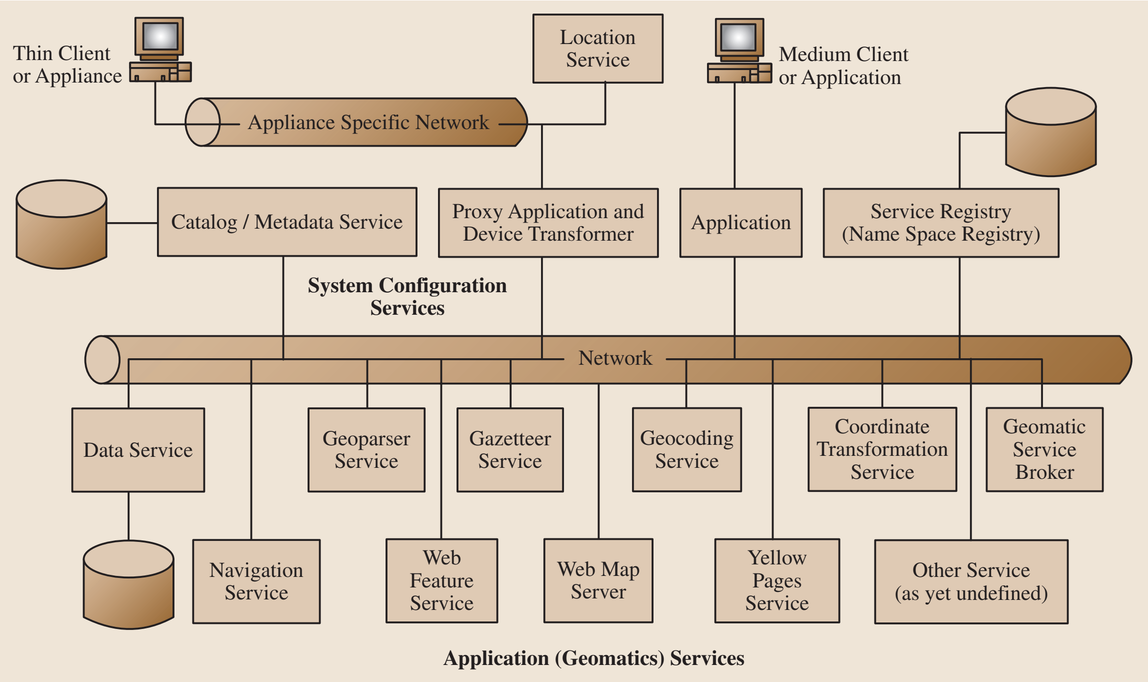

Fig. 6.14 Services according to ISO 19132 *Geographic information – Location-based services – Reference model*

gether, to make use of each other without interfering in internal procedures of the individual systems. This leads straight into the modern paradigms of GIS services that can be invoked on the Internet (Fig. 6.14). The traditional geographic data exchange is one special kind of service, but many other services (for example WMS) function without needing to exchange basic geographic data. A route finder is an Internet-based service that is even better known to the public. The result of such a route query is not achieved by transferring geographic data of roads to a local system and invoking a routing function there, but rather by transferring a map-like representation that can be visualized by a simple browser. For further reading on interoperability and GIS services, the reader is referred to Chaps. 13 and 14.

To sum up, data storage, archiving, checking, and transfer functions can be characterized in the following way.

- GIS data infrastructures with *long-term stability* call for best practice in data storage and checking.
- *Archiving of geographic data* meets the demand for time-machine-like capabilities of a GIS.
- *Standards on data transfer and interoperability* pave the way for web-based service architectures.

6.2.7 Data Request and Retrieval Functions

Information is an answer to a – possibly implicitly posed – question, as was outlined in the first sections of this chapter. Web-based search engines are information retrieval tools in wide use, giving answers to questions about content. For example, searching the Internet for information about clock towers, we may retrieve an answer that a clock tower is a tower showing one or more (often four) clock faces, usually being part of a church or municipal building such as a townhall, but that can also be free-standing. If we seek an answer not only to the *what* but with equivalent importance laid on the *where*, geographic/geospatial information will result. We may find out about the Westminster clock tower in London, the Zytglogge in Berne (Switzerland), and the Uhrturm in Graz (Austria), as well as many other examples worldwide. It would be nice to see a world map with pinpoints of clock tower occurrences or even find such examples that are close to home – provided that the search engine can (and should) be informed where *home* is. For planning a weekend trip, we may even want to search for clock towers that are within a distance of a 3 h drive, assuming average driving habits and average traffic flow. We soon get into details of a GIS search and query, combined with topological GIS analysis functions based on road networks and possibly including route planning facilities (Sect. 6.2.8). Knowing where things are in relation to our current location satisfies a basic need, since hardly any action we plan to take, hardly any decision we have to make, is independent of location. We humans try to place and pinpoint ourselves as well as our points of interest in space.

Computations often become an essential part of information requests, as the example of driving distances and driving time shows. Searching for a coffee shop that is not farther away than 5 min by bicycle requires some computing, as does the query function about sightseeing locations that are not farther than 5 km away from a vacation trip route or a mountain bike track that stays within given limits as far as horizontal distances as well as summed-up vertical distances are concerned. All these examples show that GIS functions for request and retrieval could also be listed as *query functions* under the heading of GIS analysis functions. However, typically such a geospatial search and query question and its answer decompose into two parts, one operating at or near the user interface level of a GIS (dealt with in this section) and one operating in the *analysis engine* of the GIS (dealt with in the next section). We first have to formulate our question, and this shall not only be done via character strings. Introducing search methods that are based on semantics as well as geospatial concepts brings considerable challenges. Searches based on ontology and semantic web concepts are becoming increasingly important. For example, an ontology-based search would not necessarily presuppose the user's ability to provide the string *clock tower*, but rather be able to use hints coming from the user, who may not have expertise in posing formalized questions. The search would – as in a detective's work – combine several parts of a puzzle and let the final form of the search query boil down from concepts such as building structures, historical landmarks, sightseeing, medieval city centers, etc. and what they have in common. This is also a typical task of a knowledge engineer leading to a *knowledge-based system*, as an advanced form of an information system (Chap. 5 for more elaborations on knowledge and Chap. 15 on the geospatial semantic web).

The geospatial aspect of the question we have in mind adds yet another challenge. The question *where* may be too fuzzy in terms of the capabilities of the system we intend to use. The geographic coordinates of any clock tower are of little use to the general public. A simple display on a map may be more informative, but if we really plan to visit the tower, the ways it can be accessed, the time it takes to go there, and the best route depending on the means of transportation are all necessary to give us the information of whether it is feasible and advisable to visit this sight – in the sense of the term *information* described in Sect. 6.1 – as an aid to help us make a decision. Up to now, we have discussed information retrieval for features that can be pinpointed, meaning that their location roughly corresponds to a point on the Earth's surface. This holds for our example of the clock tower, even though such a tower is not a point in the geometrical sense. However, its mapped replica in the GIS is a point feature, tied to the point location of its center point or its main entry. For line features, area features, and 3-D features, each category adding yet another geometry dimension, things become more complicated. Evaluating the distance to a line feature such as a mountain road will be more difficult than it would be for a point feature, even if we use the 2-D distance only and do not take into account any 3-D information or any obstacles on the way, such as a river or a steep slope that cannot be crossed. Projecting the current standpoint orthogonally onto all intermediate points of the road axis and also computing the distance to its start point and its end point and finally choosing the road point having the minimum distance may answer the question satisfactorily. However, there may be ambiguities if we ask for the nearest road point. Adding yet another dimension and asking about distances between 2-D features will naturally be even more complicated, as for example the distance between housing complexes with detailed geometries.

The distance between the Mississippi River and Colorado River is a question that cannot be answered satisfactorily. If we compute all possible distances from any point on the Mississippi River axis to any point on the Colorado River axis and choose the minimum, then the likewise procedure for the Mississippi River and the Missouri River would yield a distance zero, since at one point the two rivers join. But is this a satisfactory answer? The same problem arises when we ask for the distance between America and Europe. Yet another degree of difficulty enters the discussion if the delimiters of spatial features are not precisely defined. Examples are given by the question of the distance between Burgundy and the Alps or between the Rocky Mountains and the Great Plains. All these aspects influence the way we have to ask a question and therefore the system capabilities that should help us ask the question in an efficient way, as well as the way the GIS answers the question. Also for the latter, the GIS needs to be equipped with facilities that are far more intricate than a simple map output with pins and/or mileage numbers displayed on it. Due to the approximate character of typical GIS queries, methods and functions must be provided to assist the user in determining the appropriate answer out of several choices.

Performance issues are also part of the discussion about GIS request and retrieval functions. GIS typically have to handle very large amounts of geographic data. Efficient search algorithms have to be used to allow for fast access and retrieval, not only because of the large volume but also because of the imprecision and multidimensionality of query predicates, which describe the query condition. The time a system takes to answer a question should roughly correspond to the difficulty the user allots to it, on an intuitive scale, and not specifically to the data volume stored in the system. For a problem where the user has to put a larger effort into formulating the question and interpreting the answer, it is tolerable if the GIS query function needs more time. However, the size of the data volume stored in the system will not increase the user's patience. Care has to be taken that the answer does not exceed the time span of the user's attention. If this attention is diverted due to excessive waiting, the usability of the GIS suffers greatly.

To sum up, data request and retrieval functions can be characterized in the following way.

- They have a *big impact on the user interface*.
- *Semantic and geospatial aspects of the query* need to be formulated and to some extent formalized.
- Likewise, the *resulting information must be structured* in a way that suits the user's needs.
- *Performance criteria* are important.

6.2.8 Analysis Functions

This is by far the largest group of GIS functions and also the hardest one to confine or categorize. The number of special GIS applications is growing fast. Many such applications may appear as newly added features in information technology domains which can be quite distant from a GIS in the strict sense. A typical example is given by routing and guidance applications which nowadays appear in many web page presentations of companies and organizations. Typical subtypes of GIS analysis functions deal with *geometry* (for example, proximity analysis, overlays, and intersections) and *topology* (for example, network analysis as in routing and guidance). *Simulation and planning* is another quite essential topic, introducing a wide arena of models and analysis requirements ranging from urban planning and its effects on flood disaster management to agriculture and geomarketing simulation models. Another essential segment of GIS analysis is constituted by *statistical functions*, for example, ranking, regression, trend surfaces, prediction, and filtering. We will discuss a few line interpolation and surface interpolation techniques such as digital terrain models (DTM) which are standard in a GIS. For other statistical functions we refer to Chap. 2.

Data request and retrieval functions that were discussed in the previous section usually appear in combination with analysis functions, but deal with the way a typical query is handled and supported by the user interface of a GIS. The current section highlights the different aspects of the *analysis engine* that operates somewhere below the user interface level but interacts with it in a symbiotic mode.

Geometrical Analysis

For geospatial data, the geometry of features is of prime interest. Many – if not all – GIS analysis functions contain a significant portion of considerations on geometry. The location of a feature, also compared with the location of other features, is a basic ingredient of geospatial information. Two or more sets of features may have to be overlaid and their geometries intersected. Ranges, neighborhoods, and buffers will be evaluated. This metrical part of geometry, being tightly connected with measuring and rendering distance measures and eventually coordinates, will in many GIS applications have to be complemented by topological analysis rendering network-like connections, adjacency, and containment, which are, conceptually speaking, independent from distances and coordinates. These topological aspects and the corresponding analysis tools will be discussed in the next section.

Two-dimensional *Euclidian geometry* serves as a first intuitive approach to geometry. It is built on the concept of an idealized flat surface, which serves as a good approximation for many purposes, including the purpose of this section, being geometrical analysis of geospatial information. Another assumption that is reasonable will be an orthogonal coordinate system with axes pointing east and north. Extensions towards 3-D *Euclidian geometry* are straightforward. When we move on to still higher dimensions (as an example, taking time as the fourth dimension), our imagination – in terms of an image – will not stay with us, but mathematically it is possible and it is also being done. Non-Euclidian geometries are not that far-fetched. Consider, for example,

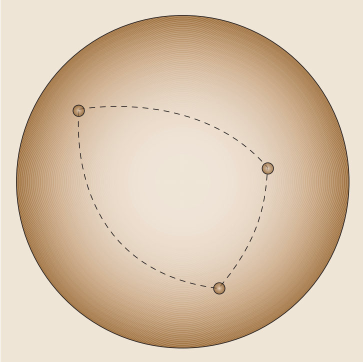

Fig. 6.15 Spherical triangle with a sum of angles greater than 180°

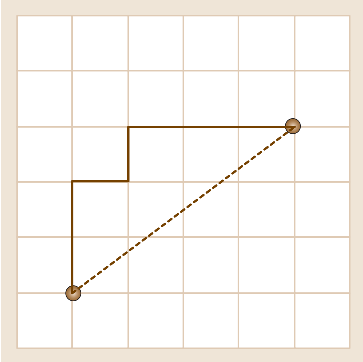

Fig. 6.16 Euclidian metric (distance = 5 units) and Manhattan metric (distance = 7 units)

the surface of a sphere, as it would be an approximation to the Earth surface. On a sphere, *in spherical geometry*, triangles can have angles whose sum is greater than 180° (Fig. 6.15), which is not possible in Euclidian geometry. Just think of going north from your current position until you arrive at the North Pole, then going south again on another meridian, traveling the same distance as before, and finally moving along the parallel of latitude towards your original position. This way, a triangle results that has at its base two angles of 90°, plus the angle at the North Pole. Spherical geometry is not the only alternative to Euclidian geometry. There are many others.

In 2-D Euclidian geometry, the distance between two points is given by the square root of the sum of squared coordinate differences. This is the famous formula of Pythagoras, since the distance line and the two coordinate difference lines form a right-angled triangle. The distance between a point and a line will be given by the length of the orthogonal projection of this point onto the line. The distance between a point and an area will be the minimum of all distances from the point to any of the circumference lines of the area. Line–line, line–area, and area–area distances are defined by appropriate extensions. Mathematically, a space allowing the measurement of distances between its elements is called a metric space. Any distance measure must conform to the rules of a *metric space*. The first rule says that a distance will always be greater than zero, except for two elements that are equal, in which case the distance will be zero. The second rule says that the distance must be symmetrical, i.e., it is irrelevant whether you measure the distance from the first to the second element or vice versa. The third rule, called the triangle inequality, says that, if you do not go directly from one element to another but take a detour through a third element, the total distance will not be shorter than the direct path.

Euclidian space and the Euclidian metric seem to be straightforward (in the double meaning of this word). However, in GIS this often needs to be modified. The *Manhattan metric* – even though, as a metric, complying with the three rules postulated – takes another approach (Fig. 6.16). The distance between two street intersections in a city with a rectangular network of streets can not only be measured in the Euclidian way. A more realistic measure in this case would be the number of blocks you have to wander east (or west), plus the number of blocks going north (or south). The Manhattan metric is important in raster-based applications where raster cells correspond to the city blocks in our example. However, also for traffic and transport applications that are vector based, this strategy is in certain cases nearer to real-life requirements for vehicles and pedestrians than the Euclidian distance, which would rather model a bird's flight. There are several other extensions and/or alternatives to Euclidian metric that serve a good purpose in GIS. Using time as a distance measure may in many cases be more realistic than mileage. Also, since geospatial data are always finite, even when using Euclidian metric in general, we must allow for modifications. For example, the fact that line features are finite means that distances cannot always be realized by measuring orthogonal projections, since the projection points may be outside the range of the line feature.

Having defined the framework upon which geometrical analysis needs to be built, we can now proceed to some protagonists of such functions. A good start to this list would be a *nearest-neighbor search* (Fig. 6.17). The question sounds simple and we might surmise that also the answer is. The problem arises in standard situations during interactive work. The user points the cursor to some feature on the screen, with some question about the feature in mind or with the intention to

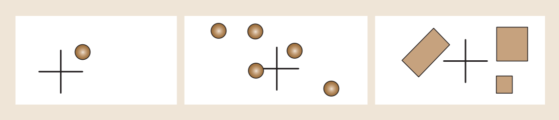

Fig. 6.17 Nearest-neighbor search for one or several points and for corresponding features

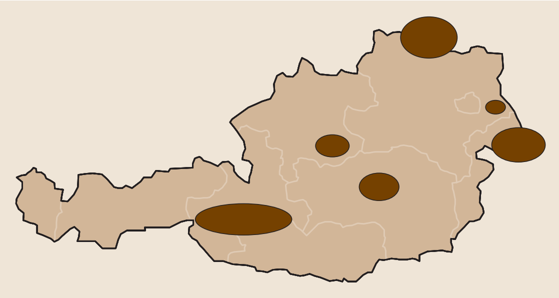

Fig. 6.18 Polygon overlay and intersection

use this feature in one of the next analysis steps. It will hardly be possible to hit any point stored in the database precisely. A search must therefore be initiated, starting with the cursor position and retrieving candidates that are nearby. This search must deliver a result very quickly, no matter how large the database is. Any delay would be criticized, since users intuitively equate the tolerable waiting time with the difficulty of the question posed. This expectation is of course not correct, but such an argument is not acceptable either. Another problem may arise due to ambiguities, when the computation of distances to candidate points does not render a unique result. In this case, a spiral search of nearest, second-to-nearest, etc. points is advisable where the user is asked to decide. This is not theoretically difficult to achieve, but it puts more burden on a user-friendly interface. Finally, let us mention a conceptual aspect that often causes another kind of complication. Even though points are at the base of geometric structures, and identifying structures on a higher level will often be achieved by identifying their points first, sometimes users consider another approach to be natural. Take a map of the USA. When users are asked to identify one state, they will place the cursor somewhere in the middle of a state and not to any border point (which would be ambiguous anyway). So, it depends on the given situation, and again a good user interface *should know* what is the appropriate action and reaction for this situation.

Polygon overlay and intersection is another protagonist of geometric analysis functions. Consider an application dealing with environmental protection zones for wildlife, vegetation, etc. They can geometrically be modeled by 2-D polygons consisting of points on the circumference of each polygon, adhering to a specified order and connected by straight lines. Typically, the polygon circumferences will follow some natural form of boundary, for example, topographic landmarks, rivers, or forest edges. In Fig. 6.18, we have used simpler shapes (ellipses) for their complex geometry. These polygons shall now be spatially compared with administrative areas such as counties or municipal districts, in order to assess the area percentage of protection zones per administrative unit. Two meshes of polygons have to be overlaid, the intersection points have to be found, and their coordinates have to be computed. The resulting mesh shall have all original points plus the new intersection points, and the new tessellation will often be finer than either of the original ones, and never coarser. The new intersection polygons must then again be correctly formed by putting their circumference points into the appropriate order. The polygon intersection answers the question of where environmental protection zones possibly interfere with the agenda of a municipality, and which municipality has to find a balance with which protection zone. Also, areas can be computed, both in km^2 and as percentages, and this can be done for each individual administrative unit, for each individual protection zone, or in total for the whole set of data.

The *overlay of lines and polygons* (Fig. 6.19) is done in a similar fashion. Consider again the protection zones and a road network being overlaid. The question is which road segments fall into which protection zones, where those road segments enter a zone and where they leave it, how long those road segments are, and what would be the total length of roads running through protection zones. It is important to note that there may be a number of special cases. A road segment running along part of the circumference of a protection zone leads to a discussion of whether it is to be

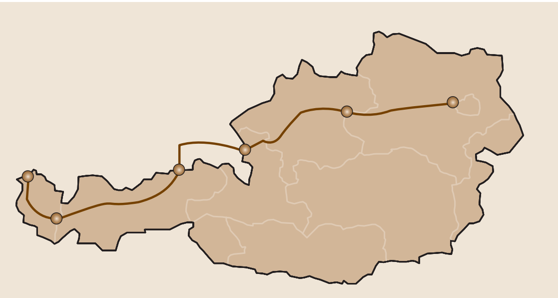

Fig. 6.19 Overlay of lines and polygons

counted or not. Even more special is the case when a road touches the circumference of a zone only at one point, say, a road junction. Such situations must be reckoned with and dealt with in a consistent manner. Also, the accuracy of computational geometry is of interest. Taking into account that the geometry of geospatial data depends to a large degree on the accuracy achievable during data capture (Sect. 6.2.2) and that geospatial data that are submitted to a geometrical analysis may originate from different sources with different accuracies, we are bound to run into a few geometrically unfavorable situations where imprecision and sometimes even ambiguities result. Consider a road that touches a protection zone at one point. Due to imprecision, the system may arrive at an answer that there are two intersection points (when the road, due to imprecise coordinates, erroneously runs through the zone, instead of only touching it), or no solution at all (when, for similar reasons, the road stays outside of the zone). Such inconsistencies may in most cases be overcome by setting fuzzy *tolerances*. This means that point locations being closer to each other than a set value are considered to be identical. In this case, we can, for example, compute an average or weighted average of the coordinate pairs. Note, however, that the problems discussed are inherent to information technology and GIS technology and cannot be totally eliminated, due to the fact that imprecision is *part of the game*. Each data capturing technology has limits on the accuracy achievable, and also data capturing costs would explode if we tried to push the limit of achievable accuracy higher and higher.

At this point, it is necessary to delve a little into performance issues. In a polygon overlay scenario where a few polygons from one dataset are compared with a few polygons from another dataset, the results will be available in almost no time at all. However, the volume of GIS data is huge and the tendency for further increases is steep. Therefore, comparing thousands of polygons, each possibly having thousands of edge points, will become noticeable, up to the point where it interrupts the workflow and the user's line of thoughts in a very annoying way. So, we have to think about means to alleviate the effects of large data volumes. In the case of polygon overlays, this can be done by a special kind of spatial filter, or rather a special combination of such filters, that is applied to the datasets involved. A filter – we can also call it a sieve – can be seen as a device that retains a *few* gems and lets go of a lot of unwanted *junk*, as if one were looking for gold nuggets. The simpler a filter and the more it can get rid of unwanted things,

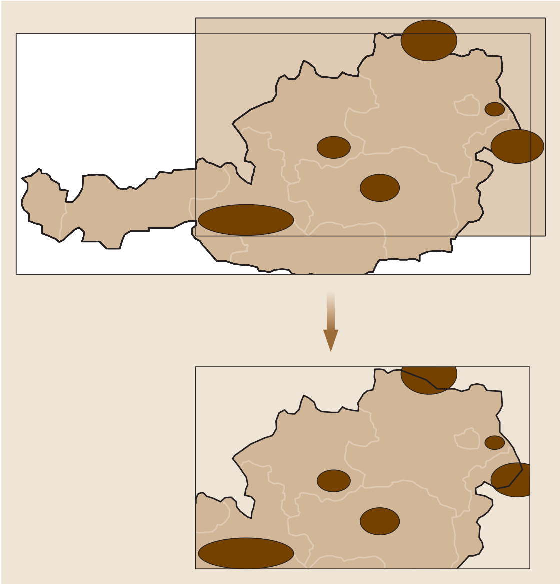

Fig. 6.20 Performance and recursive spatial filters

the more suitable it will be. Spatial operations in general and geometrical analysis tools specifically will gain considerably in their performance if several such filters are applied in an appropriate sequence (Fig. 6.20). The sequence of a coarse filter followed by a fine filter is a good example. A coarse filter would be given by a simple approximation of a feature or a collection of features, in most cases a rectangle ranging from the lowest to the highest coordinate appearing in the feature or the collection, in both east and north directions. This is called the minimum bounding rectangle (MBR). When comparing two datasets, as in the polygon overlay example given above, we could first compare the MBRs for each of the datasets and intersect them, arriving again at a (smaller) rectangle. The filter will discard all elements that are outside this rectangle, while the elements contained in the intersection will need further scrutiny through a second filter. This filter may be more complicated to apply, but it only has to deal with a small fraction of the original data. Such a strategy may be applied recursively. In this way, performance can be significantly improved, because rectangles are much easier to compare than the original geometries and because a lot of unwanted elements can be eliminated by this simple method.

A byproduct of these findings on the value of spatial filters is the concept of a range query. In GIS applications, we often need to compare a dataset against a rectangle. As an example, any display on a screen and any selection of geospatial data that should be drawn on a paper map will make use of a range query, since both are rectangles. Apart from this, range queries serve as quick and convenient tools to obtain preliminary results, and not only in geometrical analysis. Note, however, that a range query may contain *results* which in a second, more detailed geometric query may be discarded, known as false hits. A query for cities in Austria will in the first range query – among many correct answers such as Vienna – also retrieve the city of Munich, which must be discarded in a second, more detailed geometric analysis of a *point-in-polygon* test (Fig. 6.21).

From this arena of geometrical analysis tools, yet another type of spatial query, a *buffer analysis* (Fig. 6.22), is very common to a score of GIS applications. In contrast to the methods described earlier for setting a comparison filter, buffers typically use a line feature playing the role of an axis for a buffer whose width may be specified. Let us assume a line feature representing a road and that we want to define a buffer that extends for a given distance on both sides of the road axis. We may want to find all agricultural areas that are within a 500 m distance of the road axis or cities that are more than 10 km away from a railroad line. Of course, a buffer around an area feature or – much simpler – around a point can also be defined. In the latter case, it will be a circle. All these variations on buffering may be simple queries – or they may be part of an intersection overlay, aiming at a result that will reduce the original dataset to features (wholly or partially) inside the buffer zone.

As a final remark on these geometrical analysis tools, let us point out the possibility to solve any of the problems listed above by temporarily moving to another world – the raster world. Taking the first example of overlaying two sets of polygons, we can for both sets, i.e., the administrative areas and the environmental

Fig. 6.21 Range query including false hits to be discarded in a detailed point-in-polygon test

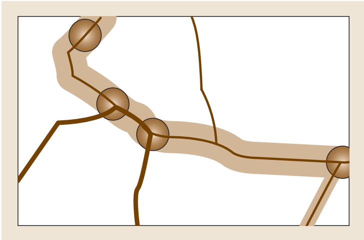

Fig. 6.22 Buffer analysis

protection areas, perform a transformation from vector to raster (Sect. 6.2.5). For example, each administrative cell receives a number signifying the administrative unit to which it belongs, and likewise this is done for the environmental cells. Depending on the accuracy to be achieved, these raster cells, being needed only temporarily, may be rather coarse in their spatial resolution. However, it is essential that both datasets result in the same resolution, having the same origin. Now, all the individual cells from both datasets can be compared, in a cell-by-cell fashion. This results in a large number of comparisons; however, they are extremely simple to perform. If necessary, the results can then be transformed back to the original vector world, although in many cases the answers that were sought can also be answered by counting and summing up the findings from all individual cells. Raster technology methods can also be utilized in many other areas of geometrical analysis. A buffer can, for example, be described by first transforming data into raster format and then inflating the central axis pixels by a given amount.

To sum up, geometrical analysis functions can be characterized in the following way.

- They are at the core of many analysis processes, since geometry is a basic aspect of geospatial data.
- Polygon overlay, intersection, range queries, buffering, and nearest-neighbor search are examples.
- Spatial filters in special arrangements help boost the performance of geometric operations.

Topological Analysis

Topology is a branch of mathematics dealing with spatial properties and relationships that are left intact by continuous deformations such as stretching. A *balloon metaphor* is often used. Imagine a balloon that has a map of Europe drawn on it, or a road network. If you inflate the balloon, the geometry changes while adjacent countries will remain adjacent, the tiny state of San Marino will still be inside Italy, and roads connected to each other via a junction will remain connected. The

Let us focus on another question that directly addresses the overlap and conflict between the two worlds mentioned: the question of visualized texts. Strictly speaking, any visualized text belongs exclusively to this second world of portrayal. Its counterpart in the first world, the geospatial world, is in most cases an attribute value. Consider, for example, a database of municipalities in a certain country. The class of municipalities shall have an attribute called *name*. The instances of this attribute will be *Alpha City*, *Betatown*, etc., and these instances will be visualized in the portrayal, displaying the text at a position near the reference point of a municipality, using a text font reserved for municipalities and choosing a text size that corresponds to the population size, which is another attribute value of this class of municipalities. Also, any GIS application could in principle refrain from showing text at all, since with a *mouse-over* effect we could always find the value of any attribute, in this case the name of the municipality under the mouse pointer. However, cartographic texts are something that we have become accustomed to, ever since we first learned to interpret maps in school. This is why cartographic texts are sometimes treated not only as aspects belonging solely to portrayal, but as a special kind of geospatial feature.

This leads us to a final discussion point here. The question arises of how much we should cling to old traditional ways of displaying geospatial data, the most appropriate example again being the topographic maps to which we have become used throughout our life. If we compare the traditional graphical appearance and the traditional legends of symbols, line types, hatch styles, etc. with the means of present-day computer graphics, including animation, pseudo 3-D, and the introduction of sound, it may seem advantageous to introduce some of these new technologies so that visualizations of geospatial data can benefit from them. However, occasionally there exist standards and rules for such symbol catalogues that cannot be neglected. A bigger argument still is the fact that human experience, human habits, and expectations will never change as fast as technology. The overall goal of visualization will always be to minimize the loss of information as data are transported from the person conceiving an application and its visualized form to the person receiving it and retrieving as much information from it as possible. Sometimes adhering to old but approved and familiar ways may be preferable. More on issues about visualization and cartography can be found in Chap. 11.

To sum up, design and presentation functions can be characterized in the following way.

- They deal with portrayal data different from their counterpart geospatial data.
- Preferably these two worlds are linked.
- Portrayal is often the end product in a lifecycle of geospatial information.
- However, visualization increasingly extends to the core of GIS analysis, offering a sandbox scenario.

References

6.1 N. Bartelme: *GIS-Technologie: Geoinformationssysteme, Landinformationssysteme und ihre Grundlagen* (Springer, Berlin Heidelberg 1989), in German

6.2 N. Bartelme: *Geoinformatik: Modelle, Strukturen, Funktionen* (Springer, Berlin Heidelberg 2005), in German

6.3 K.J. Dueker: Land resource information systems: A review of fifteen years of experience, Geo Process. **1**, 105–128 (1979)

6.4 P.A. Burrough: *Principles of Geographical Information Systems for Land Resources Assessment* (Clarendon, Wotton-under-Edge 1979)

6.5 W. Göpfert: *Raumbezogene Informationssysteme* (Wichmann, Heidelberg 1991), in German

6.6 D.J. Cowen: GIS versus CAD versus DBMS: What are the differences?, Photogramm. Eng. Remote Sens. **54**, 1551–1555 (1988)

6.7 D.J. Maguire, M. Goodchild, D. Rhind (Eds.): *Geographic Information Systems* (Longman Scientific, Harlow 1991)

6.8 C.A. Zehnder: *Informationssysteme und Datenbanken* (Teubner, Wiesbaden 1989), in German

6.9 T. Pavlidis: *Graphics and Image Processing* (Springer, Berlin Heidelberg 1982)

6.10 A. Rosenfeld, A.C. Kak: *Digital Picture Processing* (Academic, New York 1976)

6.11 R.C. Gonzalez, R.E. Woods: *Digital Image Processing* (Prentice Hall, New York 2007)

http://www.springer.com/978-3-540-72678-4

Springer Handbook of Geographic Information
Kresse, W.; Danko, D.M. (Eds.)
2012, XXXI, 1120 p., Hardcover
ISBN: 978-3-540-72678-4