# Springer Handbook of Geographic Information

*Kresse*

*Danko*

*Editors*

Springer

# Springer Handbook of Geographic Information

**Springer Handbook** provides a concise compilation of approved key information on methods of research, general principles, and functional relationships in physical and applied sciences. The world's leading experts in the fields of physics and engineering will be assigned by one or several renowned editors to write the chapters comprising each volume. The content is selected by these experts from Springer sources (books, journals, online content) and other systematic and approved recent publications of scientific and technical information.

The volumes are designed to be useful as readable desk reference book to give a fast and comprehensive overview and easy retrieval of essential reliable key information, including tables, graphs, and bibliographies. References to extensive sources are provided.

# Springer Handbook of Geographic Information

Kresse, Danko (Eds.)

With 688 Figures and 116 Tables

Springer

*Editors*
Wolfgang Kresse
University of Applied Sciences Neubrandenburg
Neubrandenburg
Germany

David M. Danko
Environmental Systems Research Institute, Inc.
Vienna, VA
USA

ISBN: 978-3-540-72678-4 e-ISBN: 978-3-540-72680-7
DOI 10.1007/978-3-540-72680-7
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2011941320

Production and typesetting: le-tex publishing services GmbH, Leipzig
Senior Manager Springer Handbook: Dr. W. Skolaut, Heidelberg
Typography and layout: schreiberVIS, Seeheim
Illustrations: le-tex publishing services GmbH, Leipzig; Hippmann GbR, Schwarzenbruck
Cover design: eStudio Calamar S.L., Spain/Germany
Cover production: WMXDesign GmbH, Heidelberg
Printing and binding: Stürtz GmbH, Würzburg

Printed on acid free paper

Springer is part of Springer Science+Business Media (www.springer.com)

30/3180/YL 5 4 3 2 1 0

# Preface

*I do not know what I may appear to the world, but to myself I seem to have been only like a boy playing on the sea-shore, and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay all undiscovered before me.*

Isaac Newton

In this sense, this handbook may help us to discover a small bay in the ocean of truth and may slightly improve our insight into geographic information.

It has been almost 50 years since the phrase *geographic information system* was coined in the early 1960s. Geographers recognized the need for automation of detail-oriented processing, and computers had matured enough to handle rudimentary models of geographic phenomena. In the early days, geographic information systems were operated mostly in the purview of national governments and universities. As processing power and model complexity grew and the cost of memory storage dropped, GIS spread slowly beyond governments into many disciplines and the private sector. Since the turn of the century, GIS has continued to grow rapidly, and with the advent of the Internet it is now used every day by the average person in all walks of life.

As a typical cross-sectional science, geographic information supports many other subject areas regarding their spatial component. The diversity of geographic information is often overlooked; one of the goals of this handbook is to demonstrate this variety of applications. They range from classical subjects such as cartography and photogrammetry through modern fields such as Internet-based Earth browsers to specialized systems for agriculture or health services, to mention just two.

Often the term *geoinformatics* is used in place of geographic information science. This is deliberate, because the international standardization committee considers geographic information a specialization of information technology. Consequently, formerly prevailing paradigms have been pushed back. Cartography is no longer primarily an *engineering art*. Data capture from aerial and satellite imagery is not just precision engineering, optics, and applied mathematics. Property cadastre is not exclusively measuring art and legislation. This new philosophy is also applied in information technology, in particular the Internet, in the form of static and mobile technologies, databases, and combinations of components from computer science. This handbook is concerned with explaining this common ground. ISO and Open Geospatial Consortium (OGC) standards are referenced in many chapters as an important foundation for geographic information applications.

**Wolfgang Kresse**

**David M. Danko**

The handbook is subdivided into three parts: Basics and Computer Science, Geographic Information, and Applications. Though the structure may be quite clear, not every topic can be indisputably allocated to one of these three parts.

As an introduction, the book begins in its first part with a chapter about modeling. In many other chapters, partial aspects of geographic information are linked to the basics of information technology, e.g., in the chapters on encoding, security, registries, and last but not least metadata. The chapters about data mining/knowledge discovery and the geospatial semantic web illuminate developments of information technology that are essential for spatial data, often characterized by huge data volumes as well as multicultural and multilinguistic environments, albeit not yet fully mature for application.

The second part addresses the specific foundations of geographic information, starting with Chap. 6, which gives a short textbook guide to geographic information. The subsequent chapters present geodesy and coordinate reference systems, data capture (from remote to indoor sensing), geometry, and cartography/portrayal.

Out of the multiplicity of applications, only a limited number of typical cases can be presented in the third part of the book. However, the selection focuses on the broad range of the field and stimulates the reader to reach a better understanding and perhaps some new ideas.

The authors, from all parts of the world, convey their distinctive perspectives on the same large field of geographic information. While in Europe geographic information is driven by the legislative and organizational framework, regarding property cadastre and planning in particular, in other parts of the world it is more technology driven, as seen in ubiquitous GIS.

The development of applications extends from proprietary systems to the open-source community. The handbook allows for both. The large software vendors continue to play a predominant role in governmental systems and/or demanding developments, as illustrated in the chapters on marine GIS and hydrography, energy suppliers, and defense. In contrast, the open-source concept carries a particular charm and has released a lot of new power, also in geographic information; the open-source concept is presented in its own chapter, as well as in the chapters on web mapping and environmental modeling. Geology, which has always been a driving force for development in cartography and geographic information, is relevant today to both administrative systems and the open-source world.

Access to the Internet via cellphone networks has widely abolished the distinction between static and mobile applications. Mobile applications mainly differ from static ones by their specific tasks. This topic is addressed in the chapters on location-based services and GIS in transportation.

Economically relevant applications are fully developed and in daily operation, but only because they were preceded by research activities such as those covered in the chapters on topics such as change detection, movement patterns, and marine GIS, with a focus on marine ecology.

What is the distinction between a textbook and a handbook? A handbook is like a collection of many short textbooks. Every chapter conveys a good and complete summary of a subject area. The authors have solved their tasks in different ways. Some of them have prepared the subject like a tutorial helping to understand a lecture. An example is Chap. 2 on positional accuracy improvement, which includes an introduction to adjustment theory. Other authors explain the basics, complemented with elaborated examples as, e.g., is done in the section about spatial databases in Chap. 3. Moreover, the handbook promotes harmonization of content and terminology, primarily in Chap. 13 about standards. The comprehensive glossary on all geo-relevant ISO terms also functions as a reference book.

As described, geographic information is a diversified subject that resists full documentation in a single handbook. We hope that our selection of topics reflects all important and many typical perspectives, and that the numerous references to other sources will help the reader to proceed where coverage by the handbook ends.

Dezember 2011
Wolfgang Kresse — Neubrandenburg
David M. Danko — Washington, DC

## List of Authors

**Daniel P. Ames**
Idaho State University
Geosciences and Civil & Environmental Engineering
995 University Blvd.
Idaho Falls, ID 83402, USA
e-mail: *dan.ames@isu.edu*

**Kristine Asch**
Bundesanstalt für Geowissenschaften und Rohstoffe
Stilleweg 2
30655 Hannover, Germany
e-mail: *kristine.asch@bgr.de*

**Norbert Bartelme**
Graz University of Technology
Institute for Geoinformation
Steyrergasse 30
8010 Graz, Austria
e-mail: *norbert.bartelme@tugraz.at*

**Matthias Becker**
Technische Universität Darmstadt
Institut für Geodäsie
Petersenstrasse 13
64287 Darmstadt, Germany
e-mail: *becker@ipg.tu-darmstadt.de*

**Ralf Bill**
Rostock University
Faculty for Agricultural and Environmental Sciences
Justus-von-Liebig-Weg 6
18059 Rostock, Germany
e-mail: *ralf.bill@uni-rostock.de*

**Thomas Brinkhoff**
Jade University of Applied Sciences
Institute for Applied Photogrammetry and Geoinformatics (IAPG)
Ofener Str. 16/19
26121 Oldenburg, Germany
e-mail: *thomas.brinkhoff@jade-hs.de*

**Jean Brodeur**
Natural Resources Canada
Centre for Topographic Information
2144-010 King West Street
Sherbrooke, Québec J1J 2E8, Canada
e-mail: *brodeur@rncan.gc.ca*

**Robert G. Brook**
Esri
380 New York Street
Redlands, CA 92373, USA
e-mail: *rbrook@esri.com*

**Thomas E. Burk**
University of Minnesota
Department of Forest Resources
1530 Cleveland Avenue North
St. Paul, MN 55108, USA
e-mail: *tburk@umn.edu*

**Keechoo Choi**
Ajou University
Department of Environment, Civil, and Transportation Engineering
Woncheon-dong, Yeongtong-gu
Suwon 443-749, Korea
e-mail: *keechoo@ajou.ac.kr*

**Michael Cramer**
Universität Stuttgart
Institut für Photogrammetrie (ifp)
Geschwister-Scholl-Str. 24 D
70174 Stuttgart, Germany
e-mail: *michael.cramer@ifp.uni-stuttgart.de*

**David M. Danko**
Esri
8615 Westwood Center Drive
Vienna, VA 22182-2214, USA
e-mail: *ddanko@esri.com*

**William F. Davenhall**
Esri
380 New York Street
Redlands, CA 92373, USA
e-mail: *bdavenhall@esri.com*

**Kian Fadaie**
Department of Fisheries and Oceans Canada
615 Booth Street
Ottawa, Ontario K1A 0E6, Canada
e-mail: *kian.fadaie@dfo-mpo.gc.ca*

**Kenneth Field**
Esri
380 New York Street
Redlands, CA 92373, USA
e-mail: *kfield@esri.com*

**Betsy George**
Oracle America, Inc.
Spatial and Location Technologies
One Oracle Drive
Nashua, NH 03062, USA
e-mail: *betsy.george@oracle.com*

**Frank Gielsdorf**
technet GmbH Berlin
Maassenstrasse 14
10777 Berlin, Germany
e-mail: *frank.gielsdorf@technet-gmbh.com*

**Görres Grenzdörffer**
Rostock University
Department Geodesy and Geoinformatics
Justus-von-Liebig Weg 6
18059 Rostock, Germany
e-mail: *goerres.grenzdoerffer@uni-rostock.de*

**Gerhard Gröger**
University of Bonn
Institute for Geodesy and Geoinformation
Meckenheimer Allee 172
53115 Bonn, Germany
e-mail: *groeger@uni-bonn.de*

**Joachim Gudmundsson**
University of Sydney
School of Information Technologies
1 Cleveland Street
Sydney NSW 2006, Australia
e-mail: *joachim.gudmundsson@sydney.edu.au*

**Norbert Haala**
University of Stuttgart
Department Institute for Photogrammetry
Geschwister-Scholl-Str. 24
70174 Stuttgart, Germany
e-mail: *norbert.haala@ifp.uni-stuttgart.de*

**Paul Hardy**
Esri
302 Science Park, Milton Road
Cambridge, CB4 0WG, UK
e-mail: *phardy@esri.com*

**Tobias Hillmann**
University of Applied Sciences
Department of Landscape Architecture,
Geoinformatics, Geodesy and Civil Engineering
Brodaer Str. 2
17033 Neubrandenburg, Germany
e-mail: *hillmann@hs-nb.de*

**Ned Horning**
American Museum of Natural History
Department Center for Biodiversity and
Conservation
Central Park West at 79th Street
New York, 10024, USA
e-mail: *horning@amnh.org*

**Marco Hugentobler**
Sourcepole AG
Churerstrasse 22
8808 Pfäffikon, Switzerland
e-mail: *marco@sourcepole.ch*

**Sung-Gheel Jang**
Cleveland State University
Urban Studies
1717 Euclid Ave.
Cleveland, OH 44115, USA
e-mail: *s.jang75@csuohio.edu*

**Ari Jolma**
Aalto University School of Engineering
Department of Civil and Environmental
Engineering
Niemenkatu 73
Lahti 15140, Finland
e-mail: *ari.jolma@aalto.fi*

**Mathias Jonas**
Bundesamt für Seeschifffahrt und Hydrographie (BSH)
Neptunallee 5
18057 Rostock, Germany
e-mail: *mathias.jonas@bsh.de*

**Gerhard Joos**
Technical University of Denmark
DTU Space
Juliane Maries Vej 30
2100 Copenhagen Ø, Denmark
e-mail: *gerhard.joos@dotGIS.de*

**Matthias M. Jöst**
Heidelberg mobil International GmbH
Head of Development
Schloss-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
e-mail: *matthias.joest@hdm-i.com*

**Holger Kessler**
British Geological Survey
Geological Modelling Systems
Kingsley Dunham Centre
Nottingham, NG25 0LT, UK
e-mail: *hke@bgs.ac.uk*

**Tschangho J. Kim**
University of Illinois at Urbana-Champaign
Urban and Regional Planning, Civil and Environmental Engineering
111 Temple Buell Hall, 611 Taft Dr.
Champaign, IL 61820, USA
e-mail: *tjohnkim@uiuc.edu*

**Christopher Kinabrew**
National Network of Public Health Institutes
1515 Poydras, Suite 1200
New Orleans, LA 70112, USA
e-mail: *ckinabrew@nnphi.org*

**Wolfgang Kresse**
University of Applied Sciences Neubrandenburg
Brodaer Str. 2
17033 Neubrandenburg, Germany
e-mail: *kresse@hs-nb.de*

**Patrick Laube**
University of Zurich
Department of Geography
Winterthurerstr. 190
8057 Zurich, Switzerland
e-mail: *patrick.laube@geo.uzh.ch*

**Steve Lime**
University of Minnesota
Geography Social Sciences
267 19th Avenue S
Minneapolis, MN 55455, USA
e-mail: *limex002@umn.edu*

**Roger Lott**
Ilkley, West Yorkshire , UK
e-mail: *rogerlott@btinternet.com*

**Stephen J. Mathers**
British Geological Survey
Keyworth
Nottingham, NG12 5GG, UK
e-mail: *sjma@bgs.ac.uk*

**Andreas Matheus**
Universität der Bundeswehr München
Institut für Technische Informatik
Werner-Heisenberg-Weg 39
85577 Neubiberg, Germany
e-mail: *andreas.matheus@unibw.de*

**William (Bill) Meehan**
Esri
Utility Solutions
380 New York Street
Redlands, CA 92373, USA
e-mail: *bmeehan@esri.com*

**Helena Mitasova**
North Carolina State University
Department of Marine, Earth, and Atmospheric Sciences
2800 Faucette Drive
Raleigh, NC 27695, USA
e-mail: *hmitaso@unity.ncsu.edu*

**Edward Nash**
DVZ Datenverarbeitungszentrum
Mecklenburg-Vorpommern GmbH
Lübecker Str. 283
19059 Schwerin, Germany
e-mail: *e.nash@dvz-mv.de*

**Markus Neteler**
Fondazione Edmund Mach
GIS and Remote Sensing Unit
Via E. Mach 1
38010 S. Michele all'Adige (TN), Italy
e-mail: *markus.neteler@iasma.it*

**Andreas Neumann**
City of Uster
Department of Construction
Oberlandstrasse 78
8610 Uster, Switzerland
e-mail: *andreas.neumann@stadt-uster.ch*

**Silvia Nittel**
University of Maine
Department of Spatial Information Science &
Engineering
5711 Boardman Hall
Orono, ME 04473, USA
e-mail: *nittel@spatial.maine.edu*

**C. Douglas O'Brien**
IDON Technologies, Inc.
1430 Prince of Wales Dr.
Ottawa, K2C 3Y7, Canada
e-mail: *cdobrien@idontech.ca*

**Roland Pesch**
University of Vechta
Landscape Ecology
49364 Vechta, Germany
e-mail: *rpesch@iuw.uni-vechta.de*

**Clemens Portele**
interactive instruments GmbH
Trierer Str. 70-72
53115 Bonn, Germany
e-mail: *portele@interactive-instruments.de*

**Aaron Racicot**
Z-Pulley Inc.
Langley, WA 98260, USA
e-mail: *aaronr@z-pulley.com*

**Charles Roswell**
10009 Hackberry Lane
Columbia, MD 21045, USA
e-mail: *charlesaroswelljr@verizon.net*

**Winfried Schröder**
University of Vechta
Landscape Ecology
49364 Vechta, Germany
e-mail: *wschroeder@iuw.uni-vechta.de*

**Markus Seifert**
Landesamt für Vermessung und Geoinformation
Alexandrastr. 4
80538 München, Germany
e-mail: *markus.seifert@lvg.bayern.de*

**Wenzhong Shi**
The Hong Kong Polytechnic University
Department of Land Surveying and
Geo-Informatics
Kowloon, Hong Kong
e-mail: *lswzshi@polyu.edu.hk*

**Jan Skaloud**
Swiss Federal Institute of Technology Lausanne
Station 18
1015 Lausanne, Switzerland
e-mail: *jan.skaloud@epfl.ch*

**Christian Strobl**
German Aerospace Center (DLR)
German Remote Sensing Data Center (DFD)
Münchner Str. 20
82234 Oberpfaffenhofen, Germany
e-mail: *christian.strobl@dlr.de*

**Tim Sutton**
Linfiniti Consulting CC.
3 Buirski Plein
Swellendam, Western Cape, 6740, South Africa
e-mail: *tim@linfiniti.com*

**Jérôme Théau**
Université de Sherbrooke
Département de géomatique appliquée
Sherbrooke, QC J1K 2R1, Canada
e-mail: *jerome.theau@usherbrooke.ca*

**Ranga R. Vatsavai**
Oak Ridge National Laboratory
Computational Sciences and Engineering Division
Oak Ridge, TN 37831, USA
e-mail: *vatsavairr@ornl.gov*

**Lutz Vetter**
University of Applied Sciences
Geoinformatics
Brodaer Str. 2
17033 Neubrandenburg, Germany
e-mail: *vetter@hs-nb.de*

**Jan O. Wallgrün**
University of Bremen
Department for Mathematics and Informatics
Enrique-Schmidt-Str. 5
28359 Bremen, Germany
e-mail: *wallgruen@informatik.uni-bremen.de*

**Shuliang Wang**
Wuhan University
International School of Software
129 Luoyu Road
Wuhan, Hubei 430079, China
e-mail: *slwang2005@whu.edu.cn;*
*slwang2005@gmail.com*

**Frank Wilke**
Fasanenweg 65
44269 Dortmund, Germany
e-mail: *frank@wilke.org*

**Thomas Wolle**
Arclight Sydney
Lvl7/89 York Str.
Sydney NSW 2000, Australia
e-mail: *thomas.wolle@gmail.com*

**Jessica Wyland**
Esri
Marketing Communication
380 New York Street
Redlands, CA 92373, USA
e-mail: *jwyland@esri.com*

**Alexander Zipf**
University of Heidelberg
Department of Geography
Berliner Str. 48
69120 Heidelberg, Germany
e-mail: *zipf@uni-heidelberg.de*

# Contents

## Part C Applications

# List of Abbreviations

| | |
|---|---|
| 0-D | zero-dimensional |
| 1-D | one-dimensional |
| 1G-E | OneGeology-Europe |
| 1NF | first normal form |
| 2-D | two-dimensional |
| 2NF | second normal form |
| 3-D | three-dimensional |
| 3DWEG | 3D Web Editeur Geologique |
| 3G | Third generation network technology |
| 3NF | third normal form |
| 4-D | four-dimensional |

## A

| | |
|---|---|
| AAA | AFIS–ALKIS–ATKIS |
| AAFIF | automated air facilities intelligence file |
| AAT | Automated aerial triangulation |
| ABJ60 | US Committee on Geographic Information Science and Applications |
| ACT | artemisinin-based combination therapy |
| ADO | ActiveX Data Objects |
| ADT | abstract data type |
| AFIS | Official Geodetic Control Station Information System |
| AFNOR | Association Française de Normalisation |
| AGC | US Army Geospatial Center |
| AGI | Association of Geographic Information |
| AIP | Architecture Implementation Pilot |
| AIS | Automatic Identification System |
| AIXM | Aeronautical Information Exchange Model |
| AI | artificial intelligence |
| AJAX | Asynchronous JavaScript and XML |
| ALB | Automatisiertes Liegenschaftsbuch (real estate book) |
| ALKIS | amtliches Liegenschaftskataster-Informationssystem |
| ALK | Automatisierte Liegenschaftskarte (real estate map) |
| ALPR | automatic license plate recognition |
| ALS | airborne laser scanning |
| AM/FM | automated mapping/facilities management |
| AMI | Advanced metering infrastructure |
| AMPS | auto map production system |
| ANSI | American National Standards Institute |
| APDM | ArcGIS Pipeline Data Model |
| API | application programming interface |
| APP | Allied Procedures Publication |
| APTS | Advanced Public Transportation System |
| ARID | Attribute Road Inventory Data |
| ARPA | Automatic Radar Plotting Aids |
| ART | anti-retroviral therapy |
| ASCII | American Standard Code for Information Interchange |
| ASP | Active Server Pages |
| ASV | Adobe SVG viewer |
| ATC | Air traffic control |
| ATIS | Advanced Traveler Information System |
| ATKIS | Amtliches Topographisch-Kartographisches Informationssystem |
| ATMS | Advanced Traveler Information System |
| ATM | Air tasking message |
| ATM | Automatic teller machine |
| ATO | air task order |
| ATSDR | Agency for Toxic Substances and Disease Registry |
| AT | aerial triangulation |
| AUV | Autonomous Underwater Vehicle |
| AVCS | Advanced Vehicle Control System |
| AVHRR | Advanced Very High Resolution Radiometer |
| AVL | Automatic Vehicle Location |
| AWACS | Airborne warning and control system |
| AWI | Alfred-Wegener-Institute |
| AdV | Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik der Bundesrepublik Deutschland |
| AmSI | ambient spatial intelligence |
| ArcIMS | ArcGIS Internet Map Server |

## B

| | |
|---|---|
| BASINS | better assessment science integrating point and nonpoint sources |
| BBOX | Bounding Box |
| BBR | Bundesamt für Bauwesen und Raumordnung, Federal Office for Building and Regional Planning |
| BBSR | Federal Institute for Research on Building, Urban Affairs, and Spatial Development |
| BGS | British Geological Survey |
| BIH | Bureau International de l'Heure |
| BIIF | Basic Imagery Interchange Format |
| BIL | band interleaved by line |
| BIP | band interleaved by pixel |
| BIS | Bodeninformationssystem |
| BI | business intelligence |
| BLM | US Bureau of Land Management |

| | |
|---|---|
| BLOB | Binary Large Object |
| BLUE | Best Linear Unbiased Estimation |
| BMVBS | Bundesministerium für Verkehr, Bau- und Stadtentwicklung (Federal Ministry of Transport, Building, and Urban Affairs) |
| BMZ | Baumassenzahl (cubic index) |
| BOD | Board of Directors |
| BP | Best Practices (paper) |
| BRFSS | Behavioral Risk Factor Surveillance System |
| BRGM | French Bureau de recherches géologiques et minières |
| BSD | Berkeley Software / Source Distribution |
| BSH | Bundesamt für Seeschifffahrt und Hydrographie, Federal Maritime and Hydrographic Agency |
| BSI | British Standards Institution |
| BSQ | band sequential |
| BTNT | Biotop- und Nutzungskartierung, biotope and land use type mapping |

## C

| | |
|---|---|
| C/A | coarse-acquisition |
| C2IS | Control Information Systems |
| C2 | Command and Control System |
| C3 | Communications, Command, and Control |
| C4ISR | Command, Control, Communication, Computers, Intelligence, Surveillance and Reconnaissance |
| C4I | Command, Control, Communications, Computer, and Intelligence |
| CADD | computer aided design and drafting |
| CAD | computer aided design |
| CAG | Colloquium on African Geology |
| CAM | Computer Aided Manufacturing |
| CARS | calibration and stabilization of LRS |
| CART | Classification and Regression Tree |
| CCD | Charge Coupled Device |
| CDC | US Centers for Disease Control and Prevention |
| CDRS | construction detour reporting system |
| CD | committee draft |
| CEMAT | Council of Europe Conference of Ministers Responsible for Spatial/Regional Planning |
| CEN | Comité Européen de Normalisation |
| CEOS/WGISS | Committee on Earth Observation Satellites/Working Group on Information Systems and Services |
| CEO | Chief Executive Officer |
| CEP | celestial ephemeris pole |
| CERL | US Army Corps of Engineers Construction Engineering Research Laboratory |
| CERN | Conseil Européen pour la Recherche Nucléaire |
| CGAL | Computational Geometry Algorithms Library |
| CGDI | Canadian Geospatial Data Infrastructure |
| CGI | Commission for the Management and Application of Geoscience Information |
| CGI | Common Gateway Interface |
| CGM | computer graphics metafile |
| CHAID | Chisquare Interaction Detection Method |
| CHAMP | Challenging Minisatellite Payload |
| CIMIC | Civil military cooperation |
| CIM | Common Information Model |
| CLI | common language infrastructure |
| CLOB | Character Large Object |
| CMAQ | congestion mitigation and air quality |
| CMYK | cyan, magenta, yellow, key (black) |
| CM | central meridian |
| CNS | car navigation system |
| COM | Component Object Model |
| COP | Common Operational Picture |
| CORBA | Common Object Request Broker Architecture (OMG) |
| CORINE | coordination of information on the environment |
| CORS | Continuously Operating Reference Stations |
| CP-DGPS | Carrier-Phase Differential GPS |
| CPAN | Comprehensive Perl Archive Network |
| CPU | central processing unit |
| CRAN | Comprehensive R Archive Network |
| CRF | coordinate reference frame |
| CRS | coordinate reference system |
| CSDGM | Content Standard for Digital Geospatial Metadata |
| CSG | constructive solid geometry |
| CSI | communications service interface |
| CSMF | conceptual schema modeling facilities |
| CSS | Cascading Style Sheets |
| CSV | comma-separated values |
| CSW | Catalogue Services for the Web |
| CS | coordinate system |
| CTD | conductivity, temperature, and density |
| CTS | computational transportation science |
| CT | Counterterrorist |
| CVO | Commercial Vehicle Operation |
| CVS | Concurrent Versions System |
| CW | Continuous wave |

## D

| | |
|---|---|
| DAML | DARPA Agent Markup Language |
| DAO | Data Access Object |
| DBMS | database management system |
| DBS | distributed spatial database system |
| DCMI | Dublin Core Metadata Initiative |

| | |
|---|---|
| DCOM | Distributed Component Object Model (Microsoft) |
| DCW | Digital Chart of the World |
| DE-9IM | Dimensionally Extended 9 Intersection Model |
| DEM | digital elevation model |
| DFDD | DGIWG Feature Data Dictionary |
| DGIWG | Digital/Defence Geographic Information Working Group |
| DGNSS | Differential Global Navigation Satellite System |
| DGPS | Differential Global Positioning System |
| DHHS | US Department of Health and Human Services |
| DHTML | dynamic HTML |
| DIAL | differential absorption LIDAR |
| DIGEST | Digital Geographic Information Exchange Standard |
| DIME | Dual Independent Map Encoding |
| DIN | Deutsches Institut für Normung e.V., German Institute for Standardization |
| DIS | draft international standard |
| DLG | Digital Line Graph |
| DLT | direct linear transform |
| DL | description logics |
| DMA | US Defense Mapping Agency |
| DMKD | data mining and knowledge discovery |
| DNC | Digital Nautical Chart |
| DN | digital numbers |
| DOM | Document Object Model |
| DOP | dilution of precision value |
| DOQQ | Digital Ortho Quarter Quad |
| DOQ | Digital Orthophoto Quadrangle |
| DORIS | Doppler Orbitography and Radio-positioning Integrated by Satellite |
| DOT | US Department of Transportation |
| DP | Discussion Paper |
| dpi | dots per inch |
| DRG | Digital Raster Graphics |
| DRM | Digital Rights Management |
| DR | dead reckoning |
| DSLR | digital single-lens reflex |
| DSM | digital surface model |
| DSS | decision support system |
| DSTL | U.K. Defence Science and Technology Laboratory |
| DSV | Deep Submergence Vehicle |
| DTD | Document Type Definition (of XML) |
| DTED | Digital Terrain Elevation Data |
| DTG | dynamically tuned gyros |
| DTM | digital terrain model |
| DTW | dynamic time warping |
| DT | Drafting Team |
| DVOF | digital vertical obstruction file |
| DWG | Domain Working Group |
| DXF | Drawing Exchange Format |
| DoD | US Department of Defense |
| DoG | difference of Gaussians |

## E

| | |
|---|---|
| E–R | entity–relationship |
| EBL | Electronic Bearing Line |
| ebRIM | e-business Registry Information Model |
| ebRS | Registry Services and Protocols |
| ECA | Economic Commission for Africa |
| ECDIS | Electronic Chart Display and Information System |
| ECEF | Earth-centered Earth-fixed |
| ECOSOC | Economic and Social Council |
| ECP | Enhanced Client or Proxy |
| ECa | apparent electrical conductivity |
| ED50 | European Datum 1950 |
| EDBS | Uniform Database Interface (Einheitliche Datenbankschnittstelle) |
| EDGE | Enhanced Data rates for GSM Evolution |
| EDMED | European Directory of Marine Environmental Datasets |
| EER | Extended Entity Relationship Model |
| EEZ | Exclusive Economic Zone |
| EGDB | Oracle Enterprise Geodatabase |
| EGM96 | Earth Gravitational Model 96 |
| EGNOS | European Geostationary Navigation Overlay Service |
| EKF | extended Kalman filter |
| EMODNET | European Marine Observation and Data Network |
| EMP | environmental modeling platform |
| EMRO | Regional Office for the Eastern Mediterranean |
| EMSA | European Maritime Safety Agency |
| EM | Electromagnetic |
| EM | expectation maximization |
| ENC | Electronic Navigational Chart |
| ENVISAT | Environmental Satellite |
| EOC | emergency operations center |
| EOSE-RM | extended open system environment reference model |
| EOSE | extended open system environment |
| EO | exterior orientation |
| EPA | US Environmental Protection Agency |
| EPHT | Environmental Public Health Tracking |
| EPSG | European Petroleum Survey Group |
| EROCIPS | Emergency Response to coastal Oil, Chemical and Inert Pollution from shipping |
| ERP | enterprise resource planning |
| ERS | emergency route service |
| ERT | Electrical Resistivity Tomography |
| ER | Engineering Report |
| ER | entity relationship |
| ESA | European Space Agency |

| | |
|---|---|
| ESDP | European Spatial Development Perspective |
| ESPON | European Spatial Planning Observation Network |
| Esri | Environmental Systems Research Institute |
| ETL | extraction, transformation, and loading |
| ETRS89 | European Terrestrial Reference System 1989 |
| EUNIS | European Nature Information System |
| EUROGI | European Umbrella Organization for Geographic Information |
| EW | Electronic warfare |
| EuroSDR | European Spatial Data Research |

## F

| | |
|---|---|
| FAA | US Federal Aviation Administration |
| FACC | Feature Attribute Coding Catalogue |
| FAO/UN | Food and Agriculture Organization of the United Nations |
| FAO | Food and Agriculture Organization |
| FCC | US Federal Communications Commission |
| FC | Feature Catalogue |
| FDIS | final draft international standard |
| FDOT | Florida Department of Transportation |
| FDO | feature data object |
| FDS | Formal Data Structure |
| FES | Filter Encoding Standard |
| FFH | Fauna Flora Habitat |
| FFI | foreign function interface |
| FGDC | US Federal Geographic Data Committee |
| FHWA | Federal Highway Administration (USA) |
| FIG | International Federation of Surveyors (Fédération International des Géomètres) |
| FK | Fundamentalkatalog |
| FME | feature manipulation engine |
| FMIS | farm management information system |
| FOG | fiber optical gyros |
| FOSS | free and open-source software |
| FOV | field of view |
| FTP | File Transfer Protocol |

## G

| | |
|---|---|
| G8 | Group of Eight |
| GAGAN | GPS Aided Geo Augmented Navigation |
| GALEON | Geo-interface for Atmosphere Land Earth, and Ocean netCDG |
| GA | Geoscience Australia |
| GBO | Land register (Grundbuchordnung) |
| GCP | ground control point |
| GDAL | Geospatial Data Abstraction Library |
| GDAS | Geographic Data Attribute Set |
| GDB | generic database |
| GDF | Geographic Data Files |
| GDOP | Geometrical Dilution of Precision |
| GEOINT | Geospatial Intelligence |
| GEOSS | Group on Earth Observation System of Systems |
| GEOS | Geometry Engine – Open Source |
| GEO | Global Environment Outlook |
| GEO | Group on Earth Observations |
| GFM | General Feature Model |
| GF | general feature |
| GGIC | Australian Government Geologists Information Committee |
| GIF | graphics interchange format |
| GIHS | geographic information human interaction service |
| GIMS | geographic information model management service |
| GIPS | geographic information processing service |
| GIRAF | Geoscience Information in Africa |
| GIS-T | GIS for Transportation |
| GIS | Geographic Information System |
| GI | geographic information |
| GKS-3D | Graphic Kernel System 3D |
| GKS | Graphical Kernel System |
| GLAS | geoscience laser altimeter |
| GLCNMO | Global Land Cover by National Mapping Organizations |
| GLONASS | Globalnaya Navigatsionnaya Sputnikovaya Sistema |
| GLPK | GNU linear programming kit |
| gmdXML | geographic metadata XML encoding |
| GML3 | Geography Markup Language version 3 |
| GMLJP2 | GML in JPEG 2000 for Geographic Imagery |
| GML | Geography Markup Language |
| GMS | Geo Mobility Server, GeoMobility Server |
| GNOME | GNU object model environment |
| GNSS | Global Navigation Satellite System |
| GOCE | Gravity field and steady-state Ocean Circulation Explorer |
| GPL | GNU General Public License |
| GPRS | General Packet Radio Service |
| GPR | Ground Penetrating Radar |
| GPST | Global Positioning System time |
| GPS | Global Positioning System |
| GPX | Global Positioning System Exchange Format |
| GRACE | Gravity Recovery and Climate Experiment |
| GRASS | Geographic Resources Analysis Support System |
| GRCH | George River Caribou Herd |
| GRIB | general regularly-distributed information in binary |

| | |
|---|---|
| GRIP | Geographical Resource Intranet Portal |
| GRS80 | Geodetic Reference System 80 |
| GRSS | IEEE Geoscience and Remote Sensing Society |
| GRZ | Grundflächenzahl (site occupancy index) |
| GSC | Geological Survey of Canada |
| GSDAS | geospatial database access system |
| GSDI | Global Spatial Data Infrastructure |
| GSD | Ground sample distance |
| GSM | Global System for Mobile communication |
| GSN | geosensor networks |
| GSO | geological survey organizations |
| GUI | graphical user interface |
| GeoDRM | Geospatial Digital Rights Management Reference Model |
| GeoRSS | Really Simple Syndication for geographic information |
| GeoSciML | GeoScience Markup Language |
| GeoTIFF | Geography TIFF 6.0 |
| GeoXACML | Geospatial eXtensible Access Control Markup Language |
| GiST | Generalized Search Tree |
| GsP | geosemantic proximity |

## H

| | |
|---|---|
| HDOP | horizontal dilution of precision |
| HHS | human health services |
| HIMSS | Healthcare Information and Management Systems Society |
| HMIS | health management information system |
| HMI | human–machine interface |
| HMMG | Harmonized Model Maintenance Group |
| HPMS | Highway Performance Monitoring System |
| HQ | Headquarter |
| HSCA | Human Service Coordination Alliance |
| HSCSD | High Speed Circuit Switched Data |
| HSV | hue, saturation, value |
| HTI | human-technology interface |
| HTML | Hypertext Markup Language |
| HTTPS | Hypertext Transfer Protocol Secure |
| HTTP | Hypertext Transfer Protocol |

## I

| | |
|---|---|
| I/O | input/output |
| IACS | EU Integrated Administration and Control System |
| IAG | International Association of Geodesy |
| IAU | International Astronomical Union |
| IA | integrated assessment |
| IC-ENC | International Centre for ENC |
| ICAO | International Civil Aviation Organization |
| ICA | International Cartographic Association |
| ICD | International Classification of Diseases |
| ICES | International Council for the Exploration of the Sea |
| ICRF | International Celestial Reference Frame |
| ICRS | International Celestial Reference System |
| ICSU | International Council for Science |
| ICS | International Stratigraphic Chart |
| ICT | information and communication technology |
| ICZM | Integrated Coastal Zone Management |
| IDL | Interactive Data Language |
| ID | identifier |
| IEC | International Electrotechnical Commission |
| IED | Improvised explosive device |
| iEMSs | International Environmental Modeling and Software Society |
| IERS | International Terrestrial Reference System |
| IETF | Internet Engineering Task Force |
| IE | Internet Explorer |
| IE | Interoperability experiment |
| IFOV | instantaneous field of view |
| ifp | Institute for Photogrammetry |
| IGC | International Geological Congress |
| IGS | International GNSS Service |
| IGW | International Geospatial Warehouse |
| IHO | International Hydrographic Organization |
| IIF | Image Interchange Format |
| IMAA-CNR | Institute of Methodologies for Environmental Analysis of the Italian National Research Council |
| IMINT | Imagery Intelligence |
| IMO | International Maritime Organization |
| IMS | Internet Map Server |
| IMU | inertial measurement unit |
| INSPIRE | Infrastructure for Spatial Information in the European Community |
| INS | Integrated Navigation System |
| INTERREG | Interregional co-operation in the EU |
| IO | interior orientation |
| IPC | interprocess communication |
| IPR | intellectual property rights |
| IPSec | Internet Protocol Security |
| IP | Interoperability Program |
| IRS | Indoor residual spraying |
| IR | Implementing Rules |
| ISA | International Federation of the National Standardizing Associations |
| ISCGM | International Steering Committee for Global Mapping |
| ISEO | integrated sensor orientation |
| ISI | Information service interface |
| ISO/PAS | ISO publicly available specification |
| ISO/TC | ISO technical committee |

| | |
|---|---|
| ISO/TR | ISO technical report |
| ISO/TS | ISO technical specification |
| ISO | International Organization for Standardization |
| ISPRS WG | ISPRS working group |
| ISPRS | International Society for Photogrammetry and Remote Sensing |
| ISP | international standardized profile |
| ISTEA | Intermodal Surface Transportation Efficiency Act (USA, 1991) |
| IS | Implementation Specification |
| IS | International Standard |
| ITAR | International Traffic in Arms Regulations |
| ITCS | communication service |
| ITHS | human interaction service |
| ITMS | Internet Traffic Monitoring System |
| ITN | Insecticide-treated bednet |
| ITPS | processing service |
| ITRF | International Terrestrial Reference Frame |
| ITRS | International Terrestrial Reference System |
| ITSS | system management service |
| ITS | Intelligent Transportation System |
| ITU-R | radiocommunication sector |
| ITU-T | telecommunication standardization |
| ITU | International Telecommunication Union |
| ITWS | workflow/task service |
| IT | information technology |
| IUGG | International Union of Geodesy and Geophysics |
| IUGS | International Union of Geological Sciences |
| IUT | implementation under test |
| IVHS | intelligent vehicle highway system |
| IWA | international workshop agreement |
| IWG | Interoperability Working Group |
| IfM | Institute of Marine Research |
| InSAR | Interferometric SAR |

## J

| | |
|---|---|
| JAG | Joint Advisory Group of ISO/TC 211 and OGC |
| JCS | JCS conflation suite |
| JDBC | Java Database Connectivity |
| JD | Julian date |
| JOG | joint operations graphics |
| JPEG | Joint Photographic Experts Group |
| JPL | Jet Propulsion Laboratory |
| JRC | European Commission Joint Research Centre |
| JSON | JavaScript Object Notation |
| JTS | Java Topology Suite |
| JUMP | Unified Mapping Platform |

## K

| | |
|---|---|
| KBES | knowledge-based expert system |
| KDE | K desktop environment |
| KGIS | King George Island GIS |
| KML | Keyhole Markup Language |
| KVP | Key–Value Pair |

## L

| | |
|---|---|
| l-ENU | east–north–up |
| l-NED | north–east–down |
| LADM | Land Administration Domain Model |
| LAI | leaf area index |
| LAN | Local Area Network |
| LBMS | location-based mobile services |
| LBS | location-based services |
| LCCS | land cover classification system |
| LCD | Liquid Crystal Display |
| LCML | land cover metalanguage |
| LCSS | longest common subsequence |
| LDAP | Lightweight Directory Access Protocol |
| LGPL | GNU Lesser General Public License |
| LIDAR | light detection and ranging, Laser Scanning |
| LIS | land information system |
| LLR | lunar laser ranging |
| LMM | linear mixing model |
| LMO | Legally Mandated Organization |
| LOB | large object |
| LOR | Lebensweltlich orientierter Raum (life-worldly oriented area) |
| LOS | Lines-of-sight |
| LPIS | Land Parcel Information System |
| LRM | location referencing management system |
| LRS | Linear Referencing System |
| LSR | Local space rectangular |
| LSU | Louisiana State University |

## M

| | |
|---|---|
| MAF | master address file |
| MBR | Minimum Bounding Rectangle |
| MCC | material composition category |
| MDA | Model driven architecture |
| MDM | Marine Data Model |
| MD | Oracle Multi-Dimension |
| MEASURE | monitoring and evaluation to assess and use results |
| MEMS | microelectromechanical system |
| ME | mean error |
| MGCP | multinational geospatial coproduction program |
| MGRS | Military Grid Reference System |
| MIF | MapInfo Interchange Format |
| MIME | Multipurpose Internet Mail Extensions |

| | |
|---|---|
| MIP | Multilateral Interoperability Programme |
| MIT | Massachusetts Institute of Technology |
| MJD | modified Julian date |
| MKRO | Ministerkonferenz für Raumordnung (Standing Conference of Ministers Responsible for Spatial Planning) |
| MLRS | Multilevel linear referencing system |
| MMS | Multimedia Messaging Service |
| MOH | Ministry of Health |
| MOI | Ministry of the Interior (Taiwan) |
| MPE | median percentile error |
| MSA | metropolitan statistical area |
| MSC | Maritime Safety Committee |
| MSDI | Marine Spatial Data Infrastructure |
| MSL | mean sea level |
| MSN | Microsoft Network |
| MSP | Maritime Spatial Planning |
| MSSQL | Microsoft SQL |
| MSS | Multi-Spectral Scanner |
| MS | multispectral |
| MTSAT | Meteorological Satellite |
| MUDAB | Marine Environmental Data Base |
| MUG | MultiUser Geodatabase |
| MVR | multi version R-tree |
| MZ | management zone |

## N

| | |
|---|---|
| NAC | National AIDS Commissions |
| NASA | US National Aeronautical and Space Administration |
| NAS | standard-based data exchange interface (Normenbasierte Austauschschnittstelle) |
| NATO | North Atlantic Treaty Organization |
| NAVEX | naval exercise area |
| NBA | User-specific updating of secondary database (Nutzerbezogene Bestandsdatenaktualisierung) |
| NCC | normalized cross-correlation |
| NCHS | National Center for Health Statistics |
| NDVI | normalized difference vegetation index |
| NECTAR | Nebraska enterprise centerline transportation attribute resource |
| NEPAD | New Partnership for Africa's Development |
| netCDF | network Common Data Form |
| NGA | US National Geospatial-Intelligence Agency |
| NGN | next generation networks |
| NGO | non-governmental organization |
| NGS | National Geodetic Survey |
| NHD | National Hydrography Dataset |
| NIMA | US National Imagery and Mapping Agency |
| NIR | Near Infrared |
| NISO | National Information Standards Organization |
| NIS | network information systems |
| NMCA | National Mapping and Cadastral Agencies |
| NMCP | national malaria control program |
| NMEA | National Marine Electronics Association |
| NM | nautical miles |
| NNQ | nearest-neighbor query |
| NOAA | US National Oceanic and Atmospheric Administration |
| NRAMS | natural resource analysis and mapping system |
| NSDI | National Spatial Data Infrastructure |
| NSIF | NATO Secondary Imagery Format |
| NSI | native space indexing methods |
| NTDB | National Topographic Data Base |
| NUTS | Nomenclature d'unités territoriales statistiques |
| NWIP | new work item proposal |

## O

| | |
|---|---|
| O & M | observations and measurements |
| OAB | OGC Architecture Board |
| OAGS | Organization of African Geological Surveys |
| OASIS | Organization for the Advancement of Structured Information Standards |
| OCAP | Outreach and Community Adoption Program |
| OCL | Object Constraint Language |
| ODBC | Open Database Connectivity |
| ODMG | Object Data Management Group |
| ODRL | Open Digital Rights Language |
| OEEPE | Organisation Européenne Photogrammétriques Expérimentales |
| OGC | Open Geospatial Consortium |
| OGF | Open GRASS Foundation |
| OGP | International Association of Oil and Gas Producers |
| OGR | OpenGIS Simple Features Reference Implementation |
| OIL | Ontology Inference Layer |
| OLAP | Online analytical processing |
| OLE DB | Object Linking and Embedding, Database |
| OLGP | online geospatial processing |
| OLTP | online transactional processing |
| OMA | Outlook Mobile Access |
| OMG | Object Management Group |
| OOA | object-oriented analysis |
| OODBS | Object Oriented database system |
| OOD | object-oriented design |
| OOPL | object-oriented programming |
| OO | object-oriented |
| OPAREA | operating areas |

| | |
|---|---|
| OQL | object query language |
| ORDBMS | Object Relational Database Management System |
| OSCE | Organization for Security and Co-operation in Europe |
| OSE-RM | ISO/IEC Open systems environment reference model |
| OSGeo | Open Source Geospatial Foundation |
| OSI | Open Systems Interconnection |
| OSOW | oversize overweight |
| OS | Ordnance Survey |
| OWL | Web Ontology Language |
| OpenLS | OpenGIS Location Services |

## P

| | |
|---|---|
| PAIGH | Panamerican Institute of Geography and History |
| PAI | Positional Accuracy Improvement |
| PAOS | reverse SOAP |
| PAP | policy administration point |
| PCA | Principle Component Analysis |
| PCGIAP | Permanent Committee on GIS Infrastructure for Asia and the Pacific |
| PCIDEA | Permanent Committee on Spatial Data Infrastructure for the Americas |
| PCL | printer command language |
| PC | Personal Computer |
| PDA | personal digital assistant |
| PDES | Pulse Doppler elevation scan |
| PDF | portable document format (Adobe) |
| PDL | Perl Data Language |
| PDNES | Pulse Doppler nonelevation scan |
| PDOP | position dilution of precision |
| PDP | policy decision point |
| PD | principal distance |
| PEP | policy enforcement point |
| PHIGS | programmer's hierarchical interactive graphic system |
| PIANC | Permanent International Association of Navigation Congresses |
| PIG | Pipeline Inspection Gauge |
| PIM | platform-independent model |
| PIP | policy information point |
| PI | place identifier |
| PLACE | priorities for local aids efforts |
| PLSS | Public Land Survey System |
| PLTS | production line tool set |
| PMG | programme maintenance group |
| PMTCT | prevention of mother-to-child transmission |
| PNA | personal navigational assistant |
| PNG | portable network graphics |
| PODS | Pipeline Open Data Standards |
| PODS | points of dispensing |
| POI | Point-of-Interest |
| POV-Ray | Persistence of Vision Raytracer |
| PPDM | Professional Petroleum Data Management Association |
| ppi | pixels per inch |
| PPP | Precise Point Positioning |
| PP | principal point |
| PRF | Pulse repetition frequency |
| PR | point region |
| PSI | parametric space indexing method |
| PSM | platform-specific model |
| PoP | Point of Purchase |
| PyPI | Python Package Index |

## Q

| | |
|---|---|
| QC | quality control |
| QDA | Quadratic Discriminant Analysis |
| QGIS | Quantum GIS |
| QUEST | Quick, Unbiased, Efficient, Statistical Tree |
| QoE | quality of experience |
| QoS | quality of service |

## R

| | |
|---|---|
| RADAR | Radio Detection and Ranging |
| RAM | random-access memory |
| RANSAC | random sample consensus |
| RAR | real aperture RADAR |
| RBAC | role-based access control |
| RCRS | road condition reporting system |
| RDBMS | relational database management system |
| RDBS | relational database systems |
| RDF-S | RDF Schema |
| RDF | Resource Description Framework |
| RDO | Remote Data Objects |
| RDT | rapid diagnostics test |
| REL | |
| RENC | Regional Electronic Chart Coordinating Centre |
| REP | Recognized Environmental Picture |
| REST | Representational State Transfer |
| RFC | Request for Comment |
| RFID | Radio Frequency Identification |
| RGB | red, green, blue |
| RIA | Rich Internet Applications |
| RIF | rapid inquiry facility tool |
| RIF | Road Impact Fees |
| RIF | Rules Interchange Format |
| RIMS | Roadway Information Management System |
| RLG | ring laser gyros |
| RM-ODP | Reference Model for Open Distributed Processing |
| RMI | Remote Method Invocation |
| RMSE | root-mean-square error |

| | |
|---|---|
| ROI | region of interest |
| ROV | Remotely Operated Vehicle |
| RPC | Remote Procedure Call |
| RSS | Really Simple Syndication |
| RS | remote sensing |
| RTCA | Radio Technical Commission for Aeronautics |
| RTCM | Radio Technical Commission for the Maritime Services |
| RTK | Real Time Kinematic |

## S

| | |
|---|---|
| SAE | Society of Automobile Engineers |
| SAGA | System for Automated Geoscientific Analyses |
| SAML | Security Assertion Markup Language |
| SAMS | Safety analysis management system |
| SAPOS | Satellitenpositionierungsdienst (German Satellite Positioning Service) |
| SARS | severe acute respiratory syndrome |
| SAR | synthetic aperture Radar |
| SBAS | satellite-based augmentation system |
| SBA | Societal Benefit Area |
| SCADA | supervisory control and data acquisition |
| SCAR | Scientific Committee on Antarctic Research |
| SCC | Standardisation Council of Canada |
| SCSI | small computer system interface |
| SC | subcommittee |
| SDBMS | spatial database management system |
| SDE | spatial data engine |
| SDF | AutoDesk Spatial Data Files |
| SDIC | Spatial Data Interest Community |
| SDI | spatial data infrastructure |
| SDMKD | spatial data mining and knowledge discovery |
| SDO | Oracle Spatial Data Option |
| SDSS | spatial decision support system |
| SDTS | Spatial Data Transfer Standards |
| SDV | Spatial Data Viewer |
| SEAMIC | Southern and Eastern African Mineral Centre |
| SE | Symbology Encoding |
| SFA | Simple Features Access |
| SGML | Standard Generalized Markup Language |
| SGS | sequential Gaussian simulation |
| SGU | Sveriges Geologiska Undersökning |
| SIFT | scale invariant feature transform |
| SI | Système International |
| SLAM | simultaneous localization and mapping |
| SLA | shuttle laser altimeter |
| SLD | Styled Layer Descriptor |
| SLEWS | Sensorbased Landslide Early Warning System |
| SLR | Satellite laser ranging |
| SLR | Side Looking RADAR |
| SME | spatial modeling environment |
| SMIL | Synchronized Multimedia Integration Language |
| SNR | Signal-to-Noise Ratio |
| SOAP | Simple Object Access Protocol |
| SOA | Service Oriented Architecture |
| SONAR | Sound Navigation and Ranging |
| SOS | Sensor Observation Service |
| SOTDMA | Self-Organizing Time Division Multiple Access |
| SPARC | Standards Planning and Requirements Committee |
| SPARQL | SPARQL Protocol and RDF Query Language |
| SPOT | Satellite pour l'Observation de la Terre |
| SPP | Single Point Positioning |
| SPS | Sensor Planning Service |
| SQL | Structured Query Language |
| SRID | spatial reference identifier |
| SRS | Spatial Reference System |
| SRTM | Shuttle RADAR Topography Mission |
| SSO | Single-Sign-On |
| STANAG | NATO Standardization Agreement |
| STAR | spatiotemporal association rules |
| STS | secure token service |
| SVG | Scalable Vector Graphics |
| SWAT | soil and water assessment tool |
| SWE | Sensor Web Enablement |
| SWF | Shockwave Flash |
| SWG | Standards Working Group |
| SWIG | simplified wrapper and interface generator |
| SWMM5 | storm water management model |
| SWT | Standard Widget Toolkit |
| SW | swath width |

## T

| | |
|---|---|
| TAI | International Atomic Time |
| TCM | terrain compensation module |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| TC | Technical Committee |
| TDT | temps dynamique terrestre |
| TEA-21 | Transportation Equity Act for the 21st Century (USA) |
| TEN | tetrahedral network |
| TICS | transport information and control system |
| TIFF | tagged image file format |
| TIGER | Topologically Integrated Geographic Encoding and Referencing system |
| TIN | Triangulated Irregular Network |
| TIS | Transportation Information System |
| TJS | Table Joining Service |
| TLM | Topographic Line Map |

| | |
|---|---|
| TLS/SSL | Transport Layer Security/Secure Sockets Layer |
| TLS | terrestrial laser scanning |
| TMG | terminology maintenance group |
| TMP | transverse Mercator projection |
| TOC | table of contents |
| TOD0 | Tactical Ocean Data – level 0 |
| TP-MBR | time-parameterized minimum bounding rectangle |
| TPC | Tactical Pilotage Chart |
| TPR-tree | time parameterized R-tree |
| TR | technical report |
| TSS | Traffic separation scheme |
| TV | Television |
| TWG | thematic working group |

## U

| | |
|---|---|
| UAV | unmanned aerial vehicle |
| UBA | Umweltbundesamt (German Federal Environmental Agency) |
| UBGI | Ubiquitous geographic information |
| UCAR | University Corporation for Atmospheric Research |
| UCB | University of California at Berkeley |
| UCGIS | University Consortium for Geographic Information Science |
| UCS | Universal Multiple-Octet Coded Character Set |
| UDM | urban data model |
| UDT | user defined types |
| UI | User Interface |
| UML | Unified Modeling Language |
| UMN | University of Minnesota |
| UMTS | Universal Mobile Telecommunication System |
| UN DPKO | United Nations Department of Peacekeeping Operations |
| UND | University of North Dakota |
| UNECA | United Nations Economic Commission for Africa |
| UNECE | United Nations Economical Commission for Europe |
| UNEP | United Nations Environment Programme |
| UNESCO | United Nations Educational, Scientific and Cultural Organization |
| UNGEGN | United Nations Group of Experts on Geographical Names |
| UNGIWG | United Nations Geographic Information Working Group |
| UNSDI | United Nations Spatial Data Infrastructure |
| UN | United Nations |
| UPA | Ubiquitous public access |
| UPS | Universal Polar Stereographic |
| UPU | Universal Postal Union |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| URN | Uniform Resource Name |
| USACE | US Army Corps of Engineers |
| USAID | US Agency for International Development |
| USGS | United States Geological Survey |
| UTC | universal time coordinated |
| UTM | Universal Transversal Mercator |
| UT | universal time |
| UVMAP | urban vector map |
| UoD | Universe of Discourse |

## V

| | |
|---|---|
| VASAB | Visions and Strategies around the Baltic Sea |
| VB | Visual Basic |
| VCT | voluntary counseling and testing |
| VDOP | vertical dilution of precision |
| VGI | Volunteered geographic information |
| VHF/UHF | Very-high-frequency/ ultrahigh-frequency |
| VHF | Very-high-frequency |
| VII | vehicle–infrastructure integration |
| VLBI | Very Long Baseline Interferometry |
| VMAP | vector map |
| VML | Vector Markup Language |
| VM | virtual machine |
| VNIR | Visible/near infrared |
| VPF | vector product format |
| VPS | Vorsorgeplanungssystem |
| VRF | vector relational format |
| VRM | Variable Range Marker |
| VRT | variable rate technology |

## W

| | |
|---|---|
| W3C | World Wide Web Consortium |
| WAAS | Wide Area Augmentation System |
| WASP7 | water quality analysis simulation program |
| WCS | Web Coverage Service |
| WD | working draft |
| WEND | World-wide Electronic Navigational Chart Data Base |
| WFD | Water Framework Directive |
| WFP | United Nations World Food Programme |
| WFS-T | Web Feature Service – Transactional |
| WFS | Web Feature Service |
| WGS84 | World Geodetic System 1984 |
| WG | working group |
| WHO | World Health Organization |
| WKB | well known binary |
| WKT | well known text |

| | |
|---|---|
| WLAN | wireless local area networks |
| WMO | World Meteorological Organization |
| WMS | Web Map Server |
| WMTS | Web Map Tile Service |
| WNS | Web Notification Service |
| WOCE | World Ocean Circulation Experiment |
| WPS | Web Processing Service |
| WP | White Paper |
| WSDL | Web Services Description Language |
| WSMO | Web Service Modeling Ontology |
| WSN | wireless sensor network |
| WS | web services |
| WXXM | Weather Information Exchange Model |
| WebCGM | Web Computer Graphics Metafile |
| Wimax | Worldwide Interoperability for Microwave Access |

## X

| | |
|---|---|
| X3D | XML-based file format for representing 3D computer graphics |
| XACML | Extensible Access Control Markup Language |
| XAML | Extensible Application Markup Language |
| XMI | XML metadata language |
| XML MG | XML maintenance group |
| XML | Extensible Markup Language |
| XMML | Exploration and Mining Markup Language |
| XSD | XML schema document |
| XSL-FO | Extensible Stylesheet Language – Formatting Objects |
| XrML | extensible rights markup language |

Part A

Basics an

# Part A Basics and Computer Science

# 1. Modeling of Geographic Information

**Charles Roswell**

This chapter describes the methodology for conceptual modeling of geographic information that is specified in the International Standards developed by ISO/TC 211. It begins with a presentation of some general concepts involved in data modeling. This is followed by a high-level description of the Unified Modeling Language developed by the Object Management Group. Next there is a description of the General Feature Model developed by ISO/TC 211 as a metamodel for the development of application schemas for geographic information. The chapter ends the a brief example of such an application schema.

## 1.1 Background

### 1.1.1 Fundamental Standards

The approach to geographic information modeling described in this chapter is that taken by ISO Technical Committee 211 (ISO/TC 211), *Geographic information/Geomatics*, as described in three international standards:

- ISO 19101 *Geographic information – Reference model* [1.1]
- ISO/TS 19103 *Geographic information – Conceptual schema language* [1.2]
- ISO 19109 *Geographic information – Rules for application schema* [1.3]

This approach was conditioned by the assumption that geographic information is fundamentally no different from other kinds of information, so that methods applied across the entire field of information technology are equally valid for geographic information.

### 1.1.2 The Modeling Approach

#### Modeling

ISO/TC 211 defines a model as an abstraction of some aspects of reality. This chapter concerns the conceptual data modeling of geographic information. Conceptual data modeling is a method for understanding concepts and the relationships between them as well as the way in which they are represented in data. It is concerned with representation, however, only at a rather abstract level. A conceptual data model reaches its limit in specifying that a concept is represented by an integer or character string, for example, unlike a physical data model which would specify how such things as integers or character strings are represented by the bits and bytes internal to a specific computing platform. Conceptual data modeling is considered to be a critical tool for developing a common understanding of the meaning of geographic information. Thus, a discussion of conceptual data modeling is a fundamental element of this Handbook.

### Model-Driven Architecture

The TC 211 approach to modeling makes use of some of the ideas concerning model-driven architecture (MDA) that have been described in detail by the Object Management Group [1.4]. The basic MDA concept is that models can be used not only to understand requirements and thus assist the process of system design; they can also drive the implementation, operation, and maintenance of computational systems (see later).

A fundamental concept of MDA is that an application can be described independently of the platform on which it runs. A platform is a set of subsystems and technologies that provide functionality to support an application with no concern as to how that functionality is implemented. There are three levels of platform independence:

- A computation-independent model ignores the details of system structure in order to focus on domain requirements. For this reason, it is sometimes called a domain model. ISO 19101 [1.1] provides a domain model for geographic information.
- A platform-independent model ignores platform-specific implementation details.
- A platform-specific model combines the specifications of the platform-independent model with the details of how they are implemented on a particular platform; this may include code for running the system.

ISO 19119 [1.5] applies the concepts of platform-independent and platform-specific models to geographic services.

The MDA concept includes the idea of transforming models, for example, from a platform-independent model to a platform-specific model. As described in the next section, ISO 19118 [1.6] applies this to the process of encoding data for transmission from one system to another (Chap. 4).

### Conceptual Modeling

Conceptual modeling provides a formal structure for understanding and using data about entities in the real world. A conceptual model is an abstract description of some aspect of the real world and a set of related concepts. It is a computation-independent model. All aspects of physical data representation should be excluded.

A conceptual model describes the content of a universe of discourse which includes everything of interest from the perspective of a specific view of the world. Thus, the universe of discourse for geographic information includes all real-world entities that can be associated with locations on or near the surface of the Earth.

The 100% principle requires that a conceptual schema describe all of the relevant structural and behavioral rules concerning the universe of discourse. On the other hand, the conceptualization principle limits a conceptual schema to only those structural and behavioral aspects that are relevant to the universe of discourse.

A conceptual schema should be constructed and interpreted using an agreed set of semantic and syntactic rules, preferably those defined in a formally defined conceptual schema language such as UML.

A conceptual model has several levels of abstraction [1.1].

- The meta-metamodel level contains the defining schema that identifies the principles to be used in organizing information about the concepts of interest. It is usually expressed in a natural language. The defining schema for geographic information is described in the next paragraph.
- The metamodel level includes a specification of the language to be used in describing the concepts. A conceptual schema language provides the semantic and syntactic elements used in a rigorous description of the conceptual model consistent with the defining schema. A number of conceptual schema languages have been developed over the years. ISO/TC 211 elected to use the UML as described below. The metamodel level also contains a normative schema that identifies the basic types to be used in modeling at a lower level. ISO/TC 211 [1.3] developed the General Feature Model (GFM) described below for this purpose.
- The application level contains application schemas that describe the specific concepts that are instantiated to produce a data set.
- The lowest level is a data set that instantiates the concepts defined by an application schema.

The defining schema (meta-metamodel) for geographic information contains several key principles:

- The real-word entities of interest for geographic information are described as features.
- Feature characteristics are known as feature attributes. Spatial characteristics, including position and configuration (geometry), are of particular importance.
- Features may be related to each other in many ways, but relationships dependent upon spatial position are especially important.
- Features may perform a variety of functions in the natural or cultural environment in which they occur.
- A set of common characteristics, relationships, and functions common to many features define a feature type.

For example, a feature type named "Highway" may be defined to include linear features constructed as part of a network of similar features intended to support the movement of vehicles.

At the metamodel level, ISO/TC 211 opted to specify the object model as the normative schema for geographic information. The UML described in ISO/IEC 19501-1 was selected as a conceptual schema language with minor restrictions specified in ISO/TS 19103 [1.2]. Although several conceptual schema languages have been used to model geographic information, UML was chosen because it supports object-oriented modeling of entities with characteristics, relationships, and behaviors. An overview of UML is presented in Sect. 1.2.

ISO/TC 211 developed the GFM described in ISO 19109 [1.3] as the metamodel for the development of application schemas for geographic information (Sect. 1.3 for more details).

### Object-Oriented Modeling

Conceptual modeling done under the auspices of ISO/TC 211 is based on an object-oriented paradigm. In this approach, an object represents an entity in the real world. An object contains information that describes the properties and behavior of the entity that it represents. An individual object is distinguished by the elements of information in which it differs from other objects.

The state of an object is the condition of the object (i. e., the values of its properties) at a given point in time. Change in the state of an object can only occur because of an internal action of the object or due to an interaction between the object and its environment. According to the principle of encapsulation, objects can only interact at interfaces. Furthermore, the information contained in an object is accessible only through interactions at the interfaces supported by the object.

The behavior of an object is a set of actions in which the object may take part, together with possible constraints on those actions. The state of the object is one of the factors that determine the actions in which the object can participate. Since actions can cause changes in state, the state at any given time may depend upon past actions.

An interface represents part of an object's behavior – a particular subset of its possible interactions. An object can interact with itself as well as with other objects. An object can support several interfaces in order to separate distinct functions of the object.

## 1.2 Unified Modeling Language

The UML is a graphical language for specifying, constructing, and documenting systems. UML supports the construction of a variety of kinds of diagrams for modeling both structure and behavior of systems. Conceptual modeling of information makes use primarily of structural diagrams, especially package diagrams and class diagrams, which use elements that come from the Kernel package at the core of the UML specification. A simplified description of the principal elements used in package and class diagrams is presented below. For a detailed description, see [1.7].

### 1.2.1 Package

A package identifies a namespace for a group of elements contained within the package. A package is represented by a rectangle with a smaller rectangular tab at its upper left corner (Fig. 1.1). Elements contained in the package may be shown within the rectangle, in which case the package name is placed in the tab. If contained elements are not shown, the package name is placed in the larger rectangle.

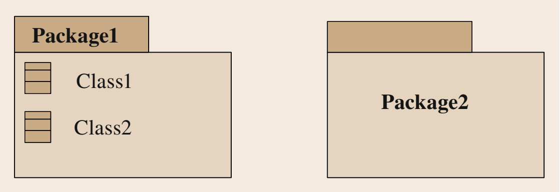

**Fig. 1.1** Examples of UML packages

## 1.2.2 Classifier

The fundamental element of a UML class diagram is the classifier. A classifier represents a concept within the system being modeled. It describes a set of objects that have common characteristics. Each such object is an instance of the class.

A classifier is represented by a solid rectangle containing the name of the classifier and, with the exception of the subtype *class*, a «keyword» that identifies its subtype. The rectangle may be divided by horizontal lines into compartments that contain features of the classifier. There are several types of classifiers.

### Class

A UML class (Fig. 1.2) is a kind of classifier that has attributes and operations. A class represents a set of objects that share the same set of specifications of semantics, features, and constraints. A class may represent any set of objects, whether or not those object have physical existence. In the case of geographic information, classes are used most commonly to represent feature types, but they may represent feature properties as well. A class is instantiated as an object, that is, as a set of attribute values and operations that describes a specific instance of the class. A class may represent lampposts or entire cities (feature types). It also may represent the ownership of parcels or the endangerment of certain species (feature properties).

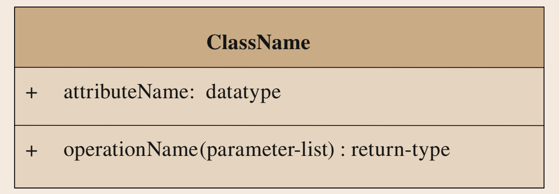
ClassName
+ attributeName: datatype
+ operationName(parameter-list) : return-type

**Fig. 1.2** Example of a UML class

The rectangle representing a class is divided into three compartments. The top compartment holds the class name and other general properties of the class; the middle compartment holds a list of attributes; the bottom compartment holds a list of operations. The attribute and operation compartments may be suppressed to simplify a diagram. Suppression does not indicate that there are no attributes or operations.

### Interface

An interface describes a service offered by instances of any class that implements the interface. An interface is not instantiated in itself, nor does it specify how it is to be implemented in a class that realizes it. Rather, it describes the public behavior of a class that implements it. An interface may be implemented by several classes, and a class may implement more than one interface.

As an example, consider an interface called AreaOfVisibility. AreaOfVisibility is defined as a two-dimensional shape that outlines the area on the ground surface from which an object can be seen. The interface might be implemented by any class that represents a physical object. It could be implemented either as an operation that derives the shape from the geometric characteristics of the object and the surrounding terrain, or as an attribute that contains the shape.

An interface may be represented as a classifier diagram with the keyword «interface» in the topmost compartment of the diagram. The interface diagram may be attached to the diagram representing an implementing class by a realization symbol, which is a dashed line with an open triangle at the end attached to the interface diagram (Fig. 1.3). The dependency of an implementing class upon the interface that it implements may also be shown by attaching a circle containing the interface name to the implementing class by a solid line.

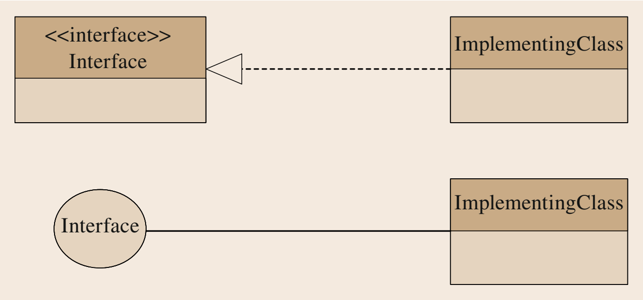

**Fig. 1.3** Alternative representations of a UML interface

### Datatype

A datatype is a kind of classifier that differs from a class only in that instances of a datatype are identified only by their value. An instance of a datatype cannot exist independently of the property whose value it provides. Datatypes include primitive predefined types and user-definable types. A datatype is identified by the keyword «dataType» in the topmost compartment of the classifier diagram.

Primitive datatypes include such things as integers and real numbers. An example of a user-defined datatype might be a combination of alphanumeric characters used to identify route numbers within national highway systems.

#### Enumeration

An enumeration is a kind of datatype whose instances form a list of named literal values. Both the enumeration name and its literal values are declared. An enumeration is a short list of well-understood potential values within a class. An example might be a list of the four points of the compass: east, west, north, and south.

#### Codelist

A codelist is a flexible enumeration specified in ISO/TS 19103 [1.2]. Code lists are useful for expressing a long list of potential values. An enumeration should be used when the elements of the list are completely known; a codelist should be used when only a set of likely values of the elements is known. A codelist may be extended in an application schema. The BuildingUse and LandUse codelists in Fig. 1.12 are examples. The set of codes in a code list may be specified by a standard, such as the ISO 639-2 list of codes for identifying languages.

### 1.2.3 Class Features

#### Attribute

An attribute represents a characteristic common to the objects of a class. Examples for attributes may be the footprint and the height of a building. An attribute is described by a string composed of elements that specify the properties of the attribute

$$\textit{visibility name: prop-type[multiplicity]} = \textit{default}\ ,$$

where

- *visibility* may be public (indicated by $+$) or private (indicated by $-$). [A private element is accessible (visible) only from elements within the namespace that owns it; a public elements is visible to all elements that can access the namespace that owns it.]
- *name* is a character string that identifies the attribute. A slash (/) preceding the name indicates that the value of the attribute is derivable from the values of other model elements.
- *prop-type* identifies the datatype of the attribute.
- *multiplicity* specifies the number of values that an instance of a class may have for a given attribute. Notation for multiplicity is explained later. When multiplicity is not shown, its value is 1.
- *default* is an optional field that specifies the initial value of the attribute.

#### Operation

An operation represents an action that can be performed by an object. An example for operations may be the write and the read function that sets and gets the above-mentioned attributes footprint and height. An operation is described by a string composed of elements that specify the properties of the operation

$$\textit{visibility name (parameter-list): return-type}\ \{\textit{oper-property}\}\ ,$$

where

- *visibility* may be public (indicated by $+$) or private (indicated by $-$).
- *name* is a character string that identifies the operation.
- *parameter-list* is a list of parameters, each in the form

$$\textit{direction parameter-name: type-expression}\ ,$$

where *direction* may be *in*, *out* or *inout* and defaults to *in* if the field is omitted, *parameter-name* is the name of the parameter, and *type-expression* identifies the datatype of the parameter.

- *return-type* identifies the datatype of the value returned by the operation.
- {*property-string*} is an optional field containing a list of property values that apply to the operation.

#### Association

An association (Fig. 1.4) specifies connections between instances of classes. An association is drawn as a solid line connecting the class rectangles. An association may have a name, represented as a character string placed near the line, but not close to either end. Each end of an association carries information pertaining to the class at that end of the association, including its role name,

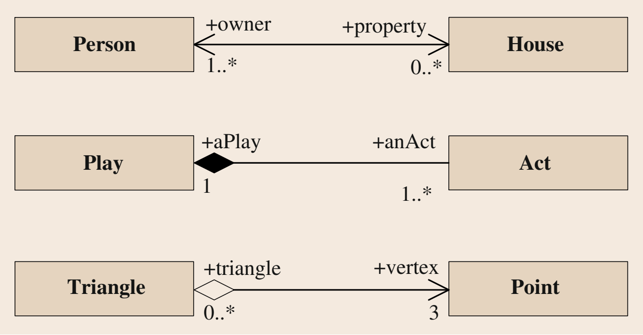

**Fig. 1.4** Examples of UML associations

its multiplicity, and its navigability. An association between two instances of the same class is represented by a line that has both ends attached to the same class rectangle.

*Role Name.* A role name at an association end specifies the behavior of the class at that end with respect to the class at the other end of the association. A role name is represented as a string beginning with a lowercase letter. In Fig. 1.4, the role name "owner" indicates that an instance of Person is an owner of an instance of House while the role name "property" indicates that an instance of House is the property of an instance of Person. In data processing, the role name is used to identify the subset of instances of a class that are involved in an association. It might be used, for example, to select the owners from the set of instances of Person included in a database.

*Multiplicity.* Multiplicity specifies the number of instances of a class that may be associated with a class at the other end of the association. Multiplicity is expressed as a pair of integers that identify the end points of a range of values, with two periods separating the integers. An asterisk replaces the second integer if the range is unlimited. If only a single value is allowed, it may be represented by repeating that value at both ends of the range or by using a single integer in place of the range. Figure 1.4 provides the following examples.

- An instance of House may be the property of one to many owners.
- A Person may be the owner of zero to many houses.
- A Play is composed of at least one and possibly of many instances of Act.
- An instance of Act belongs to one and only one instance of Play.
- An instance of Triangle aggregates three and only three points as its vertices.
- An instance of Point serves as a vertex for zero to many Triangles.

The same notation is used for multiplicity of attributes.

*Navigability.* Navigability describes the ability of one element to use information contained within another element. An arrowhead attached to the end of an association path indicates that navigation is allowed in the direction of the class attached at the arrowhead. In other words, information contained in that class is accessible from the class at the other end of the association. For example, Fig. 1.4 shows that it is possible to navigate from an instance of Triangle to obtain information about the instances of Point that form its vertices, but it is not possible to navigate from an instance of Point to identify the instances of Triangle for which it serves as a vertex.

*Aggregation.* Associations may be used to show aggregation or composition relationships between classes. An open diamond on an association end indicates that the class at that end of the association is an aggregate of instances of the class at the other end of the association. For example, an instance of the class named Triangle in Fig. 1.4 is an aggregate of three instances of the class named Point. Aggregation is considered a weak form of composition. The members of an aggregation can exist independently of the aggregation and can be members of more than one aggregation.

*Composition.* A closed diamond on an association end indicates that the class at that end of the association is composed of instances of the class at the other end of the association. For example, the class named Play in Fig. 1.4 is composed of one or more instances of the class named Act. Members of a composite cannot exit independently of the composite class, nor can they be members of more than one composite class.

### Association Class

An association class combines the semantics of an association with those of a class. In other words, it is an association that has properties in its own right. It is represented as a class symbol attached to an association path by a dashed line (Fig. 1.5, in which a Ford is modeled as an association between a Road and a River).

### Generalization

A generalization is a taxonomic relationship between a more general element and a more specific element.

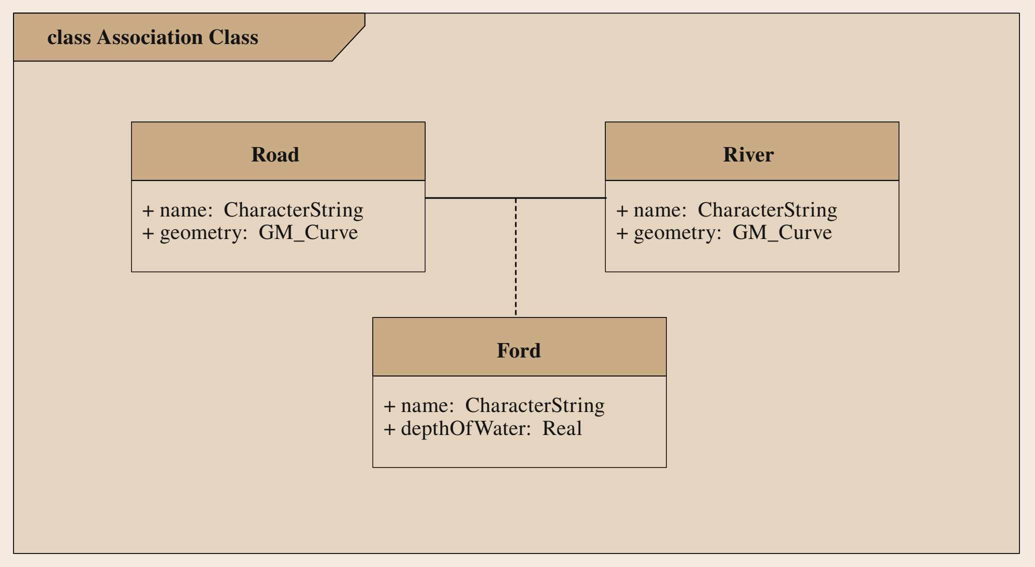

**Fig. 1.5** Example of an association class

The more specific element is fully consistent with the more general element and contains additional information. An instance of the more specific element may be used where the more general element is allowed. Generalization is shown as a solid line connecting the child (the more specific element, such as a subclass) to the parent (the more general element, such as a superclass), with a large hollow triangle where the line meets the more general element. An abstract class, which has its name in italics, can only be instantiated as instances of its subclasses. Sometimes a superclass is defined as an abstract class if it has no attributes but is rather created in order to establish a clear hierarchy of the model. The resulting dataset has no objects that are named according to the abstract class. Figure 1.6 shows an example of two generalization relationships in which each of the two subclasses of Vehicle has a third attribute in addition to the two that are inherited from Vehicle.

Individual generalization relationships may be grouped into generalization sets, each of which represents one possible way of partitioning a superclass. For example, a superclass Person might be subdivided into one generalization set on the basis of gender and into another generalization set on the basis of level of education. Each generalization set has a name that is attached to the lines connecting the subclasses in that set to the parent class.

### Stereotype

Stereotypes extend the semantics of pre-existing elements, but do not affect their structure. A stereotype is identified by a name enclosed in guillemets («…»). An example is the stereotype «Metaclass», which identifies a classifier whose instances are themselves classes. A metaclass represents a concept at a higher level of abstraction than do the classes that instantiate it.

### Note

A note contains textual information. It is shown as a rectangle with the upper right corner *bent*, attached to zero or more model elements by a dashed line. Notes may be used to contain comments or constraints. The example in Fig. 1.6 contains a constraint.

### Constraint

A constraint specifies a semantic condition or restriction. Although the UML specification includes an Object Constraint Language for writing constraints, a constraint may be written using any formal notation, or a natural language. A constraint is shown as a text string in braces "{ }". It is contained in a note or placed near the element to which it applies. See the example in Fig. 1.6, which constrains the value of the numberOfAxles attribute that PassengerCar inherits from Vehicle.

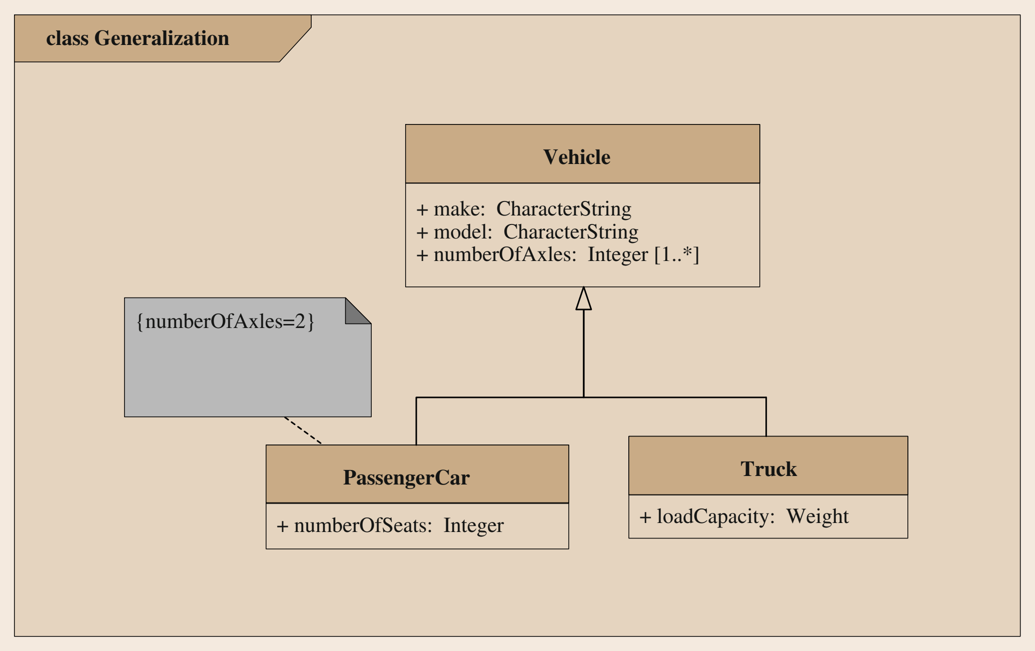

**Fig. 1.6** Example of a generalization

If the notation for an element is a text string (e.g., an attribute), the braces containing the constraint may follow the element text string. A constraint included as an element in a list applies to all subsequent elements in the list, down to the next constraint element or the end of the list.

### Dependency

A dependency indicates that the implementation or functioning of one or more elements requires the presence of one or more other elements. It relates the model elements themselves and does not require a set of instances for its meaning. A dependency is shown as a dashed arrow between two model elements. The model element at the tail of the arrow (the client) depends on the model element at the arrowhead (the supplier). The kind of dependency may be indicated by a keyword in guillemets, such as «import», «refine», or «use». In the example of Fig. 1.7, Package1 has a «use» dependency upon Package2, meaning that one or more elements in Package1 use elements specified in Package2. For example, an attribute specified in Package1 might use datatypes specified in Package 2.

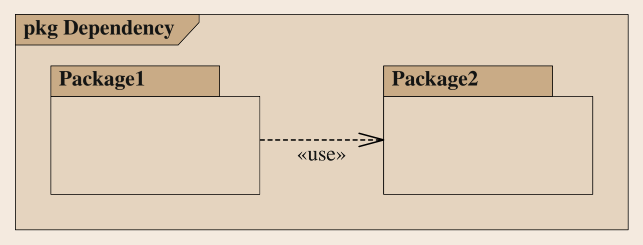

**Fig. 1.7** Example of a dependency

## 1.2.4 Naming of UML Elements

### Namespaces

Names of UML elements are required to be unique within the namespaces within which they reside. Thus, class names must be unique within the package within which they are specified. Attribute and operation names must be unique within the class within which they are defined. Role names must be unique within the context of the using class.

### Naming Style

The UML specification provides a nonnormative set of guidelines for naming elements of UML models. They are exemplified in Figs. 1.5–1.7.

Names of classes are expressed in boldface and centered in the name compartment of the class. Each word in the class name begins with an upper-case letter, but there are no spaces between words. The name is italicized if the class is abstract.

Keywords and stereotypes are in plain face, placed within guillemets, and centered within the name compartment above the class name.

The first word in names of attributes, operations, parameters, and role names begins with a lower-case letter; the initial letters of subsequent words in the name begin with upper-case letters. Names of attributes and operations are left-justified within their name compartments with no spaces between words.

## 1.3 The General Feature Model

ISO/TC 211 developed the GFM described in ISO 19109 as a metamodel for representing features in application schemas. The elements of the metamodel may be represented as elements of a UML model of the application schema. Those that cannot be represented graphically should be included in the documentation of the model, which may be in the form of a data dictionary for the application schema.

### 1.3.1 GF_FeatureType

The principal element of the GFM is the metaclass GF_FeatureType (Fig. 1.8). An instance of GF_FeatureType is a UML class named for a specific feature type, such as bridge, factory, or road. GF_FeatureType has three attributes that are intended to be included in each instance:

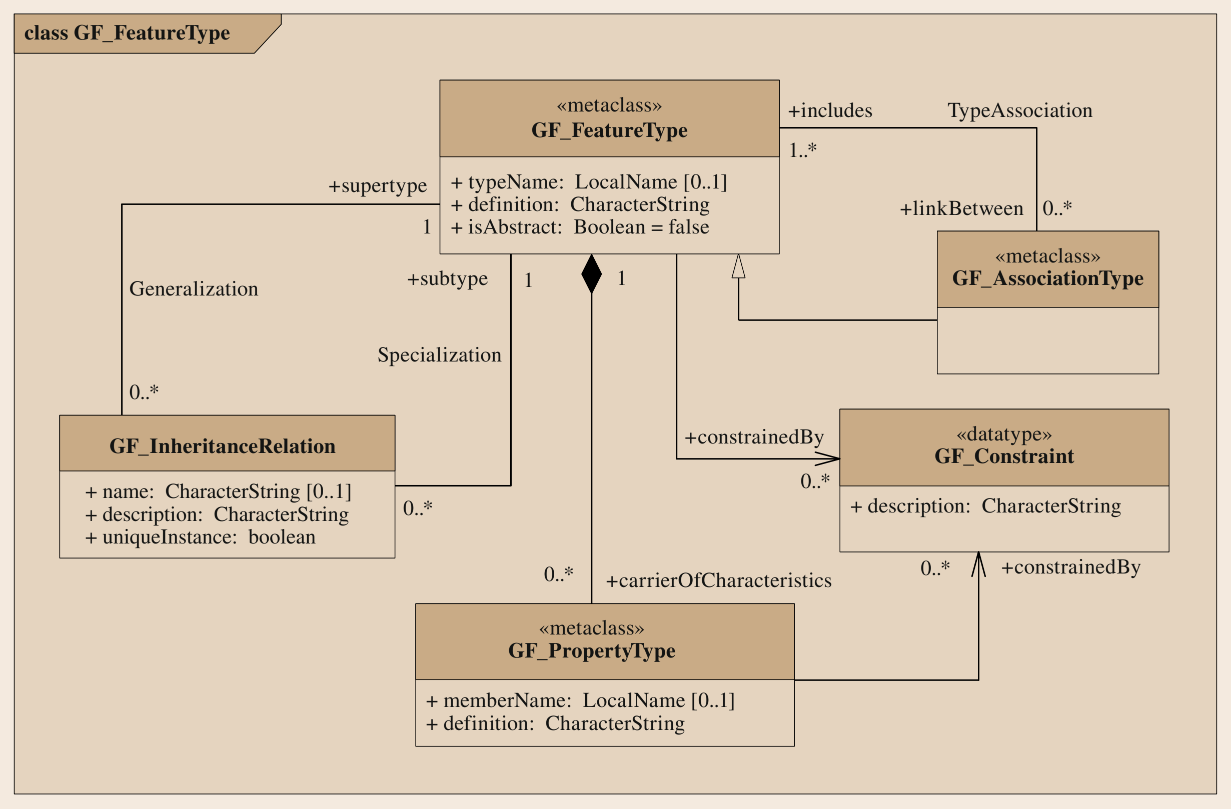

**Fig. 1.8** GFM: GF_FeatureType

- *typeName* contains the name of the specific feature type represented by the instantiating class and used as the name of that class;
- *definition* contains the definition of that feature type, and is carried as an attribute of the instantiating class;
- *isAbstract* contains a Boolean value indicating whether the instantiating class is or is not abstract. If this attribute is TRUE the name of the UML class is placed in italics to indicate that it is abstract.

### 1.3.2 GF_InheritanceRelation

The two associations generalization and specialization connect GF_FeatureType to the class GF_InheritanceRelation. This structure shows that two instances of GF_FeatureType may be related to each other as a superclass and a subclass in a generalization hierarchy.

While the *name* attribute of GF_GeneralizationRelation is optional, the attributes *description* and *uniqueInstance* are required. The attribute *uniqueInstance* has a Boolean value, such that TRUE indicates that an instance of the superclass may be an instance of only one of its subclasses, while FALSE indicates that an instance of the superclass may be an instance of more than one of the subclasses.

An instance of GF_InheritanceRelation is modeled as a UML generalization relationship. The optional attribute *name* may be attached to the generalization relationship if it is part of a generalization set, in which case every member of the set must have the same name. Otherwise, this attribute should not be implemented. The attribute *description* is included in the model documentation. A generalization set defaults to the situation where the attribute *uniqueInstance* is TRUE. The constraint {overlapping} should be applied to the generalization set if *uniqueInstance* is FALSE.

Consider the example in Fig. 1.9. The feature type Building has two subclasses: School and Hospital. The Building feature type has three feature attributes: length, width, and numberOfStories that are inherited by both of its subclasses. Each of its subclasses has specific attributes: for School, educationalLevel and numberOfTeachers; for Hospital, numberOfBeds. The two generalization relationships belong to a generalization set named Function.

All three classes – Building, Hospital, and School – are instances of the metaclass GF_FeatureType from the

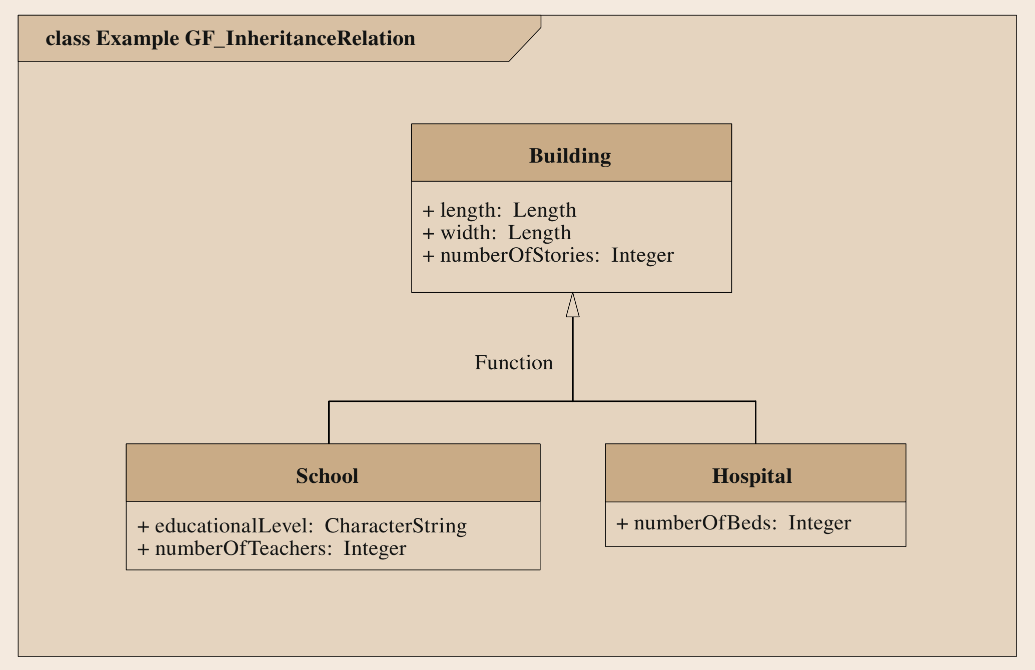
class Example GF_InheritanceRelation
Building
+ length: Length
+ width: Length
+ numberOfStories: Integer
Function
School
+ educationalLevel: CharacterString
+ numberOfTeachers: Integer
Hospital
+ numberOfBeds: Integer

**Fig. 1.9** Example implementation of GF_InheritanceRelation

GFM (Fig. 1.9). The hierarchical relationship in which Building is a generalization of Hospital and School is an instantiation of the class GF_InheritanceRelation.

### 1.3.3 GF_AssociationType

There are two relationships between GF_FeatureType and GF_AssociationType.

The association named TypeAssociation in Fig. 1.8 indicates that an instance of GF_FeatureType (i. e., a class that represents a specific feature type) may be included in an instance of GF_AssociationType. Generally, GF_AssociationType is implemented in an application schema as a simple UML association between the two classes that represent different feature types. However, the second relationship between GF_FeatureType and GF_AssociationType is an inheritance relationship that makes GF_AssociationType a subclass of GF_FeatureType.

As a subclass of GF_FeatureType, GF_AssociationType inherits the three attributes of GF_FeatureType, although the *name* is optional in the case of GF_AssociationType. GF_AssociationType also inherits the associations of GF_FeatureType. This means that it can contain properties, in which case it is modeled in an application schema as a UML association class rather than as a UML association. This also means that a feature association can be treated as a feature in its own right; the Ford shown in Fig. 1.5 is an example. Another typical example is a bridge, which may be modeled as a road segment or as a feature on its own with special overpass information. In the first case the bridge may simply associate two other road segments. In the latter case the bridge is an instance of GF_FeatureType.

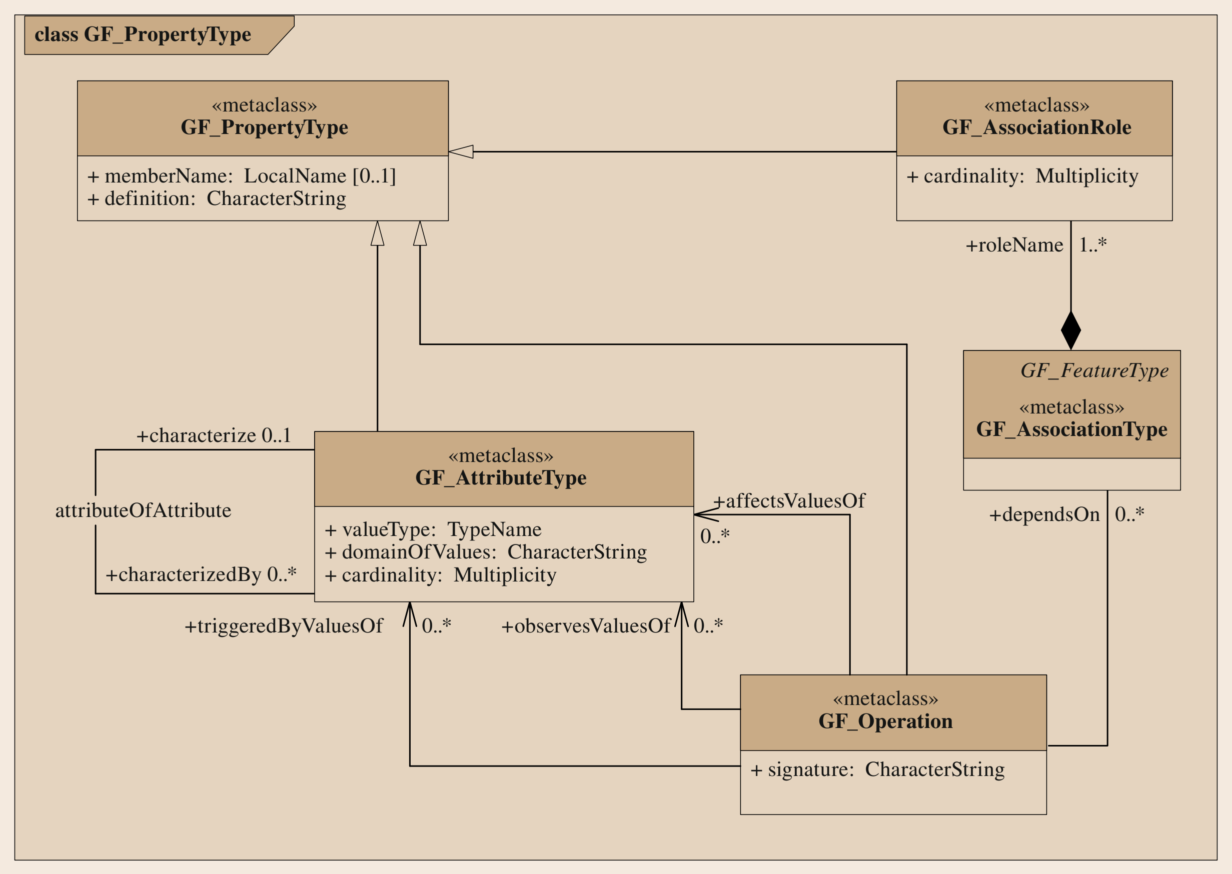

Fig. 1.10 GFM: GF_PropertyType

### 1.3.4 GF_Constraint

Both GF_FeatureType and GF_PropertyType have associations to GF_Constraint; GF_AssociationType inherits this association from GF_FeatureType. GF_Constraint represents a description of a constraint that may be applied to an instance of any of its associated metaclasses. An example is a road which is open only to vehicles traveling faster than 60 km/h.

### 1.3.5 GF_PropertyType

As shown by the composition association between GF_FeatureType and GF_PropertyType (Fig. 1.8), a feature type class can contain zero to many instances of GF_PropertyType. GF_PropertyType is an abstract metaclass with three subclasses – GF_AssociationRole, GF_AttributeType, and GF_Operation – each representing a different kind of property type that may be assigned to a feature type class. The class GF_PropertyType contains only the name and the description of the property type (the attributes memberName and description). All other details are in the subclasses.

GF_PropertyType (Fig. 1.10) has two attributes that are intended to be included in each instance.

- *memberName* contains the name of the property type represented by the instance;
- *definition* contains the definition of that property type.

### 1.3.6 GF_AssociationRole

GF_AssociationRole describes a role that an instance of a feature type may have in an association with another instance of the same or another feature type. GF_AssociationRole inherits the *memberName* and *definition* attributes of GF_PropertyType. The additional attribute *cardinality* specifies the multiplicity for instances of the feature type acting in this role. GF_AssociationRole inherits an association to GF_Constraint from GF_PropertyType, but also has an association from GF_AssociationType which specifies that it is a component of an instance of GF_AssociationType.

In a UML diagram, the attributes *memberName* and *cardinality* are represented by the role name and multiplicity at the end of the line representing the association. The attribute *definition* is included in the documentation of the model.

Consider the example in Fig. 1.4, where the role name *property* indicates that an instance of a House feature may be the property of one or more instances of Person with the role name *owner*.

### 1.3.7 GF_AttributeType

An instance of GF_AttributeType describes a type of attribute that belongs to a feature type. GF_AttributeType has three attributes in addition to the *memberName* and *definition* attributes that it inherits from GF_PropertyType. The attribute *valueType* specifies the datatype of the attribute. The attribute *domainOfValues* identifies the set of values that the attribute may take on. The attribute *cardinality* specifies the multiplicity of the attribute of the feature type.

GF_AttributeType inherits an association to GF_Constraint from GF_PropertyType, but also has an association to itself called *attributeOfAttribute*. This indicates that an attribute may itself have attributes. In this case, the attribute is modeled as a UML class which may be used as the data type for the attribute or connected to the feature type by an association.

The UML representation of a feature attribute includes the *memberName*, the *datatype*, and the *cardinality*. The *definition* and *domainOfValues* need to be specified in a data dictionary.

Figure 1.11 illustrates the three ways in which GF_FeatureAttribute may be instantiated in an application schema. The *routeNumber* feature attribute of the RoadSegment feature type is specified within the attribute compartment of the class that represents RoadSegment. The *surfaceMaterial* feature attribute has attributes itself; it is specified within the attribute compartment of the class diagram, but it uses a separately defined class as its datatype. Finally, the feature attribute *maintenanceHistory*, which also has attributes, is modeled as a separate class connected to the feature type by an association. Note that the role name at the end of a UML association is analogous to an attribute name for the class at the other end of the association.

### 1.3.8 GF_Operation

An instance of GF_Operation describes an operation of the feature type to which it belongs. In addition to the *memberName* and *definition* attributes that it inherits from GF_PropertyType, GF_Operation has one attribute, *signature*, that provides the name, arguments, and return types.

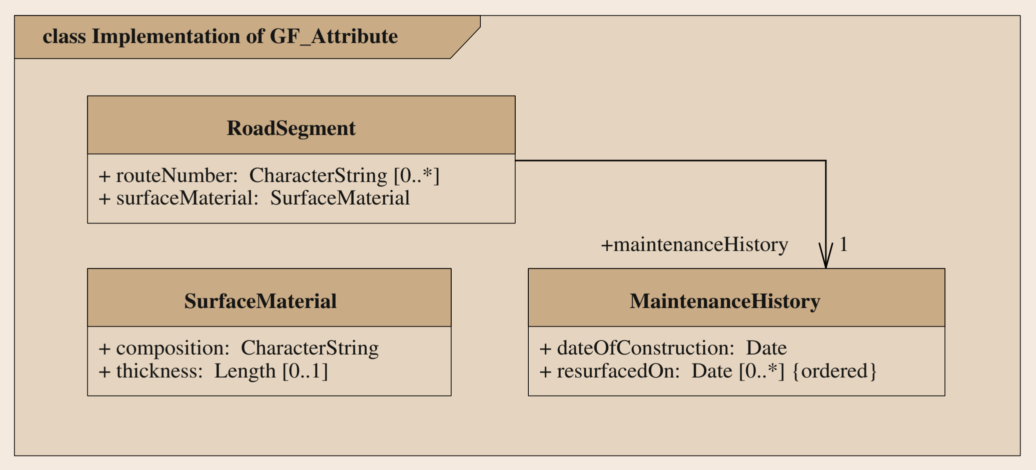
class Implementation of GF_Attribute
RoadSegment
+ routeNumber: CharacterString [0..*]
+ surfaceMaterial: SurfaceMaterial
+maintenanceHistory
1
SurfaceMaterial
+ composition: CharacterString
+ thickness: Length [0..1]
MaintenanceHistory
+ dateOfConstruction: Date
+ resurfacedOn: Date [0..*] {ordered}

**Fig. 1.11** Examples of instantiation of GF_FeatureAttribute

As an example consider an operation called *trafficFlow* that serves as a property of a Highway feature type. Such an operation might have a signature such as

trafficFlow(dayOfWeek:DayName,
timeOfDay:Time):vehiclesPerHour:Integer .

This signature indicates that the operation accepts a day of the week and a time of day as input parameters and returns an integer number of vehicles per hour.

GF_Operation inherits an association to GF_Constraint from GF_PropertyType, but has four other associations as well. Three associations to GF_AttributeType describe the way in which an instance of GF_Operation may use one or more values of a feature attribute. The role name *triggeredByValuesOf* identifies attribute types that serve as triggers for an operation in that execution of an operation will be initiated by a change in the value of the attribute. The role name *observesValuesOf* identifies attribute types that are used as input parameters for the operation. The role name *affectsValuesOf* identifies attribute types that are the output parameters of the operation. The association to GF_AssociationType identifies the pathways through which the instance of GF_Operation obtains the information about the feature attributes that it uses.

## 1.4 Application Schema Example

This section presents an example of a simple application schema developed according to the principles described in this chapter. The schema conforms to the GFM and is described in UML.

The example is an application schema for describing the spatial characteristics of a single farm. Its content includes those features that can be displayed on a map (Fig. 1.12). The features are provided with sufficient attribution to distinguish one instance from another. The schema makes use of three of the geometric object classes specified in ISO 19107 to describe the spatial position and geometry of these feature types (Chap. 13). The objects selected are those appropriate for the two-dimensional coordinate space of a map.

In this schema, a Farm has three attributes: name, owner, and an area represented as a GM_Surface. The multiplicity of the *land* attribute indicates that a Farm may consist of noncontiguous land areas. A Farm aggregates instances of two other types of features: SubAreas and Buildings. A Farm must have at least one SubArea but need not have any Buildings. The plain language constraint on Farm indicates that the entire area of the farm must be included in its aggregated instances of SubArea. The navigability designations show that it is possible to query a Farm object in order to determine which instances of SubArea and Building are contained in the Farm, but it is not possible to query a Building or SubArea object to determine what instance of Farm it belongs to.

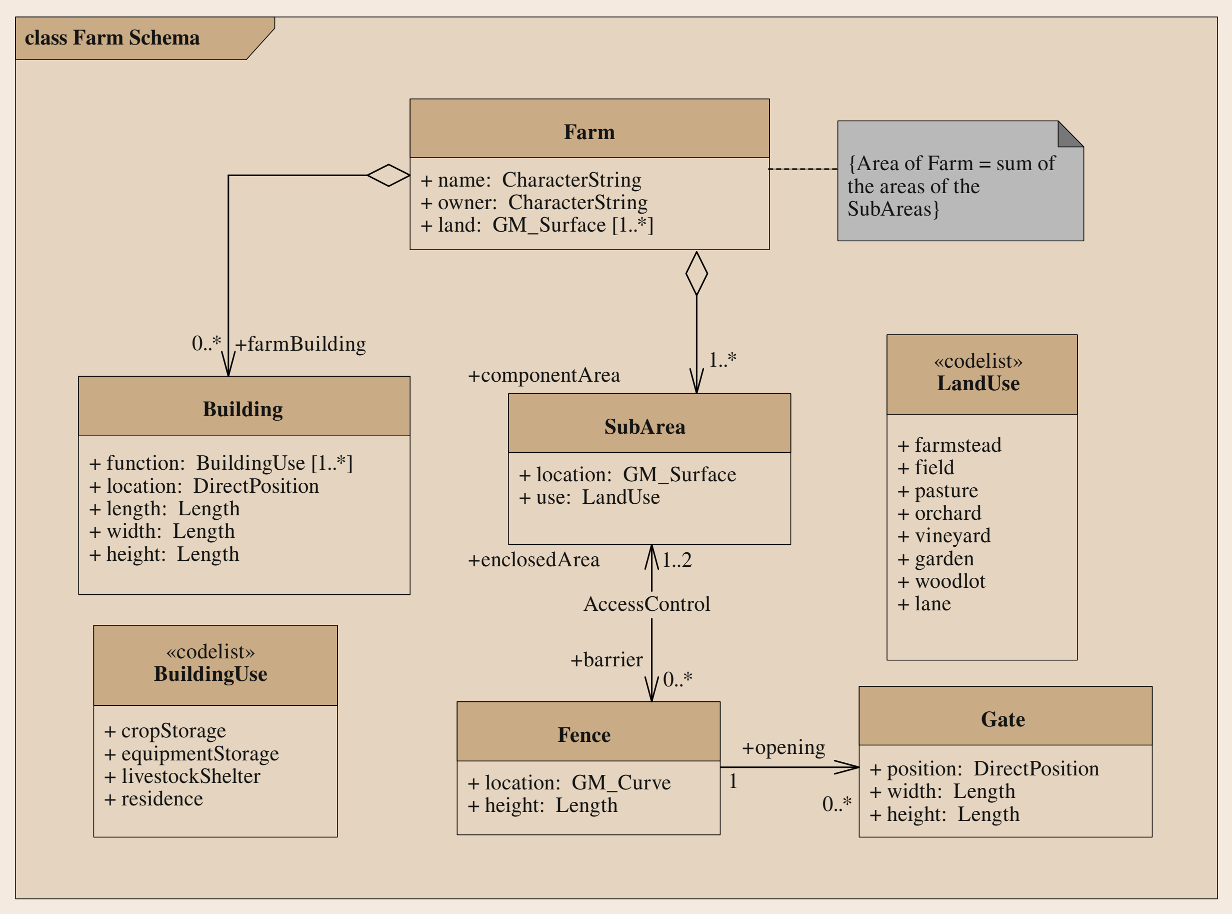

**Fig. 1.12** Simple Farm schema

The SubArea and Building feature types have properties that describe their location, geometry, and use. Note that neither feature type has an identifier attribute; instances can be distinguished from each other by their locations. The spatial relationships between SubAreas and Buildings are not described by associations because they can be derived from their *location* attributes.

A Building is treated as a point object, so its location is described as a DirectPosition. Its size is described by three required attributes that provide its dimensions using the Length measure specified in ISO/TS 19103 [1.2] as their datatype. Its function is identified by one of the values from the code list BuildingUse; a Building may have more than one function.

A SubArea is treated as a two-dimensional object. Both its location and its geometry are described by the geometric object GM_Surface. Its use is identified by one of the values from the code list LandUse.

The named association AccessControl shows that access to a SubArea may be restricted by a set of Fences. The multiplicity at the *enclosedArea* end of the association reflects the fact that every instance of Fence separates two areas, but one of these areas may be excluded from the data set because it is outside the boundaries of the Farm. The multiplicity at the other end of the association supports unfenced instances of SubArea but is unlimited because each of the contiguous instances of SubArea is considered to be separated from this instance by a unique instance of Fence. This association is navigable in both directions, meaning that it is possible to determine the characteristics of any instance of Fence that bounds an instance of SubArea, and to determine the characteristics of the instances of SubArea that are bounded by any instance of Fence.

A Fence is treated as a one-dimensional object whose geometry is described by a GM_Curve, but it is also given a height attribute. An instance of Fence may have a set of zero to many openings, each controlled by an instance of Gate. Each instance of Gate is associated with only one instance of Fence. The navigability of the association means that a Fence object knows about the associated instances of Gate but not vice versa.

A Gate is treated as a point object so its location is described as a DirectPosition, but it is also given height and width attributes.

## 1.5 Conclusion

To sum up, this chapter has presented some basic principles for conceptual data modeling in the field of geographic information. In particular it describes the use of the unified modeling language and the GFM from ISO/TC 211 for developing application schemas for various uses of geographic information. Description of the need to develop conceptual data models for specific applications of GIS is contained in the chapters of Part C.

### References

1.1 ISO 19101:2002 Geographic information: Reference Model (ISO, Geneva 2002)

1.2 ISO/TS 19103:2005 Geographic information: Conceptual schema language (ISO, Geneva 2005)

1.3 ISO 19109:2005 Geographic information: Rules for application schema (ISO, Geneva 2005)

1.4 Object Management Group, Inc.: MDA Guide Version 1.0.1 (Object Management Group, Needham 2003) http://www.omg.org/cgi-bin/doc?omg/03-06-01

1.5 ISO 19119:2005 Geographic information: Services (ISO, Geneva 2005)

1.6 ISO 19118:2005 Geographic information: Encoding (ISO, Geneva 2005)

1.7 OMG: OMG Unified Modeling Language (OMG UML), Superstructure Version 2.2 (Object Management Group, Needham 2009) http://www.omg.org/spec/UML/2.2/Superstructure/PDF/

# 2. Mathematics and Statistics

Frank Gielsdorf, Tobias Hillmann

This chapter consists of two main parts, an introduction to adjustment techniques (Sects. 2.1, 2.2) and an overview of geostatistical methods (Sect. 2.3). The contents has many relations to other chapters of the handbook. In particular, the adjustment technology is a foundation of many data capture methods and geostatistical methods are applied in marine GIS and geology. Section 2.1 starts with an introduction to the Gauss–Markov model, discusses error propagation, and explains the role of covariance. The positional accuracy improvement, a key method for the reduction of geometrical errors present in old paper maps, is the main topic of the remainder of Sects. 2.1, 2.2. Many related topics of positional accuracy improvement are addressed such as datum and conformal transformation as well as the consideration of geometric constraints. Section 2.3 starts with an example and then describes the most common methods for processing and analysis of huge amounts of geodata, random fields and variograms. Important terms such as stationarity, intrinsic model, and ergodicity are explained. The chapter concludes with a discourse about kriging (Sect. 2.3.6), which is a common method for the interpolation of geodata in order to estimate the contents of unknown and inaccessible mineral deposits.

The geometrical properties of objects in Geographic Information Systems (GIS) are almost exclusively described by point coordinates referring to a global reference frame. However, it is impossible to measure these point coordinates directly; they are the result of a calculation process. The input parameters of these calculations are measured values. Even global navigation satellite system (GNSS) receivers do not measure coordinates directly but calculate them from distance measurements to satellites.

## 2.1 Data Integration with Adjustment Techniques

There exist several types of measured values, for instance, distances, directions, local coordinates from maps or orthophotos, etc. Mostly, single measured values are grouped into sets of local coordinates. So, the measured values of a total station – horizontal direction, vertical direction, and distance – can be seen as a set of spherical coordinates, rows and columns of digitized pixels as Cartesian coordinates in a local reference frame, etc. The final aim of most evaluation processes in surveying is the determination of point coordinates in a global reference frame.

However, measured values have two essential properties. Firstly, they are random variables. It is impossible to measure a value with arbitrary accuracy, which leads to the fact that any measured value contains some uncertainty. Secondly, they are redundant. Commonly, there exist more measured values then necessary to be able to calculate unique point coordinates.

A function of random variables results again in a random variable. Because point coordinates are functions of measurement values, they are random variables. The uncertainties contained in measured values lead necessarily to uncertainties in point coordinates.

For the unique determination of a number of coordinates, the exact same number of measured values is necessary (Fig. 2.1).

The positions of the control points A and B should be known. The distances $d_1$ and $d_2$ from the control points to the new point N were measured. The position of the new point can be calculated by intersection of arcs. The two unknown coordinate values of the new point, $x_N$ and $y_N$, can be determined uniquely from the two measured values $d_1$ and $d_2$.

However, what happens if we measure a third distance $d_3$ from a control point C to N (Fig. 2.2)?

A geometrical construction with a pair of compasses would yield a small triangle. The size of this triangle depends on the magnitude of the uncertainties in the measured values. This means for the calculation that its result depends on which measured values are used. A calculation with the distances $d_1$ and $d_2$ yields a different result than a calculation with the distances $d_2$ and $d_3$.

This example raises several questions. How can one get a unique result from redundant measured values containing uncertainties? How can one quantify the accuracy of the measured values on one hand, and of the result on the other? The reply to these questions is the objective of adjustment theory.

The objectives of adjustment theory are the determination of optimal and unique output parameters

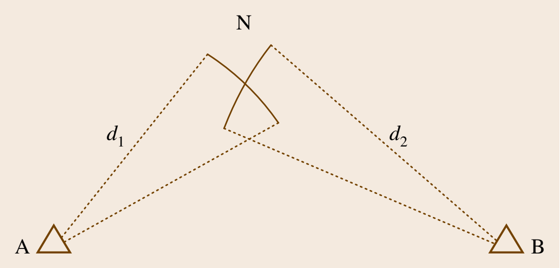

Fig. 2.1 Unique arc section

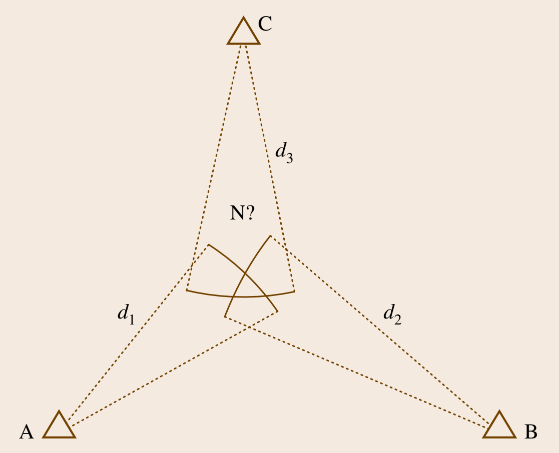

Fig. 2.2 Ambiguous arc section

(coordinates) from input parameters (measured values) which are redundant random variables, considering their accuracy, estimation of accuracy values for the output parameters, and detection and localization of blunders.

However, what is the relevance of adjustment techniques for positional accuracy improvement in GIS? Coordinates in GIS result from the evaluation of measured values. In a first step, these measured values often were local coordinates of digitized analog maps which were transformed into a global reference frame. The global coordinates so determined describe the geometry of the GIS objects uniquely, whereby the coordinates have to be addressed as random variables. During the process of positional accuracy improvement (PAI), new measured values are introduced with higher accuracy (even global coordinates can be seen as special measured values). The new measured values are redundant to the already existing coordinates. Therefore, the determination of new global coordinates with improved positional accuracy is a typical adjustment problem.

However, before the application of adjustment techniques for PAI is presented in detail, it is necessary to consider the basics of adjustment theory.

## 2.1.1 Estimation of Parameters

Section 2.1.1 describes the process of parameter estimation on the basis of the least-squares method. Figure 2.3 shows a symbolic representation of an adjustment process.

The input parameters are measured values called observations. The output parameters are the so-called unknown parameters (mostly coordinate values) and the residual errors of the observations. The number of observations is $n$ and the number of unknown parameters is $u$. Typical for an adjustment problem is the fact that the number of observations is larger than the number of unknown parameters. The difference between these two values is conterminous to the number of supernumerary observations and is called the redundancy $r$

$$r = n - u \,. \tag{2.1}$$

Observations are denoted by the symbol $l$, unknown parameters by the symbol $x$, and residual errors by the symbol $v$. To be able to formulate adjustment problems clearly it is necessary to use vector/matrix notation. The observations, unknown parameters, and residual errors can then be grouped in vectors

$$\boldsymbol{l} = \begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ l_n \end{pmatrix} , \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_u \end{pmatrix} , \quad \boldsymbol{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} . \tag{2.2}$$

We only want to consider adjustment of observation equations. The basis of this model is the presentation of the observations as explicit functions of the unknown parameters. This type of approach is called a Gauss–Markov model. The observation equations in matrix notation are

$$\boldsymbol{l} + \boldsymbol{v} = \mathbf{A}\boldsymbol{x} \tag{2.3}$$

or transformed

$$\boldsymbol{v} = \mathbf{A}\boldsymbol{x} - \boldsymbol{l} \,. \tag{2.4}$$

The matrix $\mathbf{A}$ in this observation equation system is called the configuration matrix or design matrix. It describes the linear dependency between observations and unknown parameters. The observation equation system has an infinite number of solutions. To obtain the optimal solution, it is necessary to formulate an additional constraint. This additional constraint is the request that the square sum of the residual errors should be minimal. In matrix notation it is

$$\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} \overset{!}{=} \min \,. \tag{2.5}$$

With this constraint, it is possible to formulate an extremum problem and to derive an optimal and consequently unique solution for the unknown parameters

$$\boldsymbol{x} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\boldsymbol{l} \,. \tag{2.6}$$

The equation system (2.6) is called the normal equation system. After solving the normal equations, the residual errors can be calculated by inserting $\boldsymbol{x}$ into the observation equations (2.4).

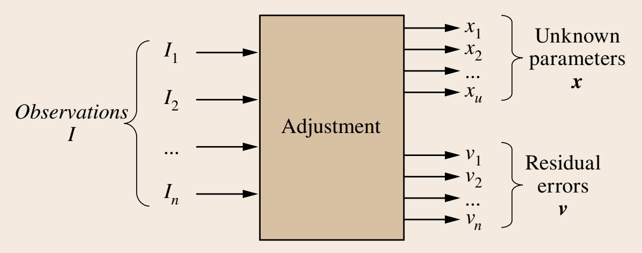

**Fig. 2.3** Symbolic representation of an adjustment

### 2.1.2 Arithmetic Mean

Adjustment theory is very complex, and it is impossible to impart its whole content herein. For further reading please refer to [2.1, 2]. Therefore, we will begin with very simple special examples and try to generalize them step by step.

The simplest case of an adjustment is the calculation of an arithmetic mean value.

We consider two points A and B. The distance between these two points was measured 10 times. The results of the measurement are 10 observation values, which we call $l_1 \ldots l_{10}$. The distance between A and B, which we call $x$, is the unknown parameter. Now, it would be possible to solve the problem by formulation of the equation

$$x = l_3 \,,$$

in which case all the other observations would be redundant. For each observation we could calculate a residual error $v_i$,

$$\begin{aligned} v_1 &= x - l_1 \,, \\ v_2 &= x - l_2 \,, \\ &\vdots \\ v_{10} &= x - l_{10} \,. \end{aligned} \tag{2.7}$$

The residual error $v_3$ would have the value 0 while the other residuals would have values different from 0. The value of $x$ depends in that case on the observation we used for its determination. However, this result is not optimal. The request for an optimal result can be formulated as follows: Find exactly that solution of $x$ for which the sum of the squares of the residual errors $v_i$ becomes minimal.

The first step in calculating an optimal $\boldsymbol{x}$ is formulation of the observation equations

$$\boldsymbol{v} = \mathbf{A}\boldsymbol{x} - \boldsymbol{l} \,.$$

In our special case the configuration matrix $\mathbf{A}$ has a very simple structure with just one column filled with ones, and the vector $\boldsymbol{x}$ has just one row

$$\mathbf{A} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} , \quad \boldsymbol{x} = (x) \,.$$

With $\mathbf{A}$ and $\boldsymbol{l}$ we are able to calculate $\boldsymbol{x}$ as (for the derivation see [2.1, 2])

$$\boldsymbol{x} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\boldsymbol{l} \,.$$

We now want to apply this formula to our special problem of arithmetic mean. The matrix product $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is 10

$$\begin{array}{cc} & \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} . \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} & (10) \end{array}$$

The product $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is called the normal equation matrix and is denoted by the symbol $\mathbf{N}$. We have the special case of an N-matrix with one row and one column, which is conterminous to a scalar. The inverse $(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}$ of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is equal to the reciprocal of the scalar 10

$$(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1} = \tfrac{1}{10} \,.$$

The expression $(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}$ (or $\mathbf{N}^{-1}$) is the cofactor matrix of the unknown parameters and is denoted by the symbol $\mathbf{Q}_{xx}$.

Cofactor matrix: $\mathbf{Q}_{xx} = \mathbf{N}^{-1}$.

The expression $\mathbf{A}^{\mathrm{T}}\boldsymbol{l}$ results in the sum of the observations

$$\begin{array}{ccc} & \begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ l_{10} \end{pmatrix} & \\ \mathbf{A}^{\mathrm{T}}\boldsymbol{l} = & & = \sum l_i \,, \\ \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix} & (l_1 + l_2 + \cdots + l_{10}) & \end{array}$$

so that $\boldsymbol{x}$ results in

$$\boldsymbol{x} = \tfrac{1}{10} \cdot (l_1 + l_2 + \cdots + l_{10}) \,.$$

As we can see, the solution for the least-squares approach is the same as for the arithmetic mean.

### 2.1.3 Weighted Arithmetic Mean

In the arithmetic mean example we assumed that all observations have the same accuracy. However, in reality, mostly this assumption is not applicable. In the general case, the observations have different accuracies, e.g., because of different measuring devices. For this reason it is necessary to modify the adjustment model of the preceding subsection. We now have to introduce a weight for each observation. The observation weight steers the influence of each observation on the resulting unknown parameters. The greater the weight, the stronger the influence of the corresponding observation.

The weight of an observation is a function of its standard deviation. The standard deviation quantifies the accuracy of a value. The standard deviations of the observations are mostly known before the adjustment. Usually, a standard deviation is denoted by the symbol $\sigma$. The square of the standard deviation $\sigma^2$ is called the variance. The meaning of a standard deviation will be explained in the following sections. The formula for a weight $p_i$ of an observation is

$$p_i = \frac{\sigma_0^2}{\sigma_i^2} \,,$$

where $p_i$ is the weight of observation $i$, $\sigma_i^2$ is the variance of observation $i$, and $\sigma_0^2$ is the variance of unit weight.

The variance of unit weight is a constant to be set. It results in weight value 1 for this variance. Often $\sigma_0$ is set to 1. If one introduces weighted observations in an adjustment, the least-squares constraint changes from

$$\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} \overset{!}{=} \min$$

to

$$\boldsymbol{v}^{\mathrm{T}}\mathbf{P}\boldsymbol{v} \overset{!}{=} \min \,. \tag{2.8}$$

In expression (2.8), $\mathbf{P}$ is the weight matrix, which is a diagonal matrix of dimension $n \times n$ with the observation weights on the principle diagonal, i.e.,

$$\mathbf{P} = \begin{pmatrix} p_1 & & 0 \\ & p_2 & \\ 0 & & \ddots \\ & & & p_n \end{pmatrix} .$$

The formula for the calculation of the unknown parameters $\boldsymbol{x}$ changes to

$$\boldsymbol{x} = (\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{P}\boldsymbol{l} \,,$$

and the formula for the cofactor matrix of the unknown parameters $\mathbf{Q}_{xx}$ becomes

$$\mathbf{Q}_{xx} = (\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A})^{-1} \,.$$

## 2.1.4 Adjustment with Several Unknown Parameters

The arithmetic mean represents a special case of adjustment with just one unknown parameter. In the general case, the number of unknown parameters is greater then one. This case will be explained with a simple example again.

Figure 2.4 shows points in a one-dimensional coordinate system. The coordinates $x_{\mathrm{A}}$ and $x_{\mathrm{B}}$ of the control points A and B are known. The distances between the points $d_1 \ldots d_{10}$ were measured. In demand are the coordinates of the new points $x_1 \ldots x_9$. The standard deviations are proportional to the corresponding distance

$$\sigma_i = 0.01 \cdot d_i \,. \tag{2.9}$$

In a first step we have to formulate the observation equations as

$$\begin{aligned}
d_1 + v_1 &= +1\cdot x_1 && -x_{\mathrm{A}} \\
d_2 + v_2 &= -1\cdot x_1 + 1\cdot x_2 && \\
d_3 + v_3 &= -1\cdot x_2 + 1\cdot x_3 && \\
d_4 + v_4 &= -1\cdot x_3 + 1\cdot x_4 && \\
d_5 + v_5 &= -1\cdot x_4 + 1\cdot x_5 && \\
d_6 + v_6 &= -1\cdot x_5 + 1\cdot x_6 && \quad . \\
d_7 + v_7 &= -1\cdot x_6 + 1\cdot x_7 && \\
d_8 + v_8 &= -1\cdot x_7 + 1\cdot x_8 && \\
d_9 + v_9 &= -1\cdot x_8 + 1\cdot x_9 && \\
d_{10} + v_{10} &= -1\cdot x_9 && +x_{\mathrm{B}}
\end{aligned}$$

If we shift the observations to the right-hand side of the equation system we get

$$\begin{aligned}
v_1 &= +1\cdot x_1 && -x_{\mathrm{A}} - d_1 \\
v_2 &= -1\cdot x_1 + 1\cdot x_2 && -d_2 \\
v_3 &= -1\cdot x_2 + 1\cdot x_3 && -d_3 \\
v_4 &= -1\cdot x_3 + 1\cdot x_4 && -d_4 \\
v_5 &= -1\cdot x_4 + 1\cdot x_5 && -d_5 \\
v_6 &= -1\cdot x_5 + 1\cdot x_6 && -d_6 \quad . \\
v_7 &= -1\cdot x_6 + 1\cdot x_7 && -d_7 \\
v_8 &= -1\cdot x_7 + 1\cdot x_8 && -d_8 \\
v_9 &= -1\cdot x_8 + 1\cdot x_9 && -d_9 \\
v_{10} &= -1\cdot x_9 && +x_{\mathrm{B}} - d_{10}
\end{aligned}$$

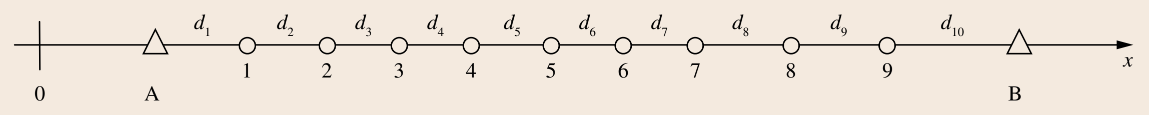

**Fig. 2.4** Points in a one-dimensional coordinate reference system

From this equation system we can directly derive the structure of our matrices

$$\mathbf{A}=\begin{pmatrix} 1 & & & & & & & & \\ -1 & 1 & & & & & & & \\ & -1 & 1 & & & & & & \\ & & -1 & 1 & & & & & \\ & & & -1 & 1 & & & & \\ & & & & -1 & 1 & & & \\ & & & & & -1 & 1 & & \\ & & & & & & -1 & 1 & \\ & & & & & & & -1 & 1 \\ & & & & & & & & -1 \end{pmatrix},$$

$$\boldsymbol{x}=\begin{pmatrix} x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\\ x_7\\ x_8\\ x_9 \end{pmatrix},\quad \boldsymbol{l}=\begin{pmatrix} d_1+x_\mathrm{A}\\ d_2\\ d_3\\ d_4\\ d_5\\ d_6\\ d_7\\ d_8\\ d_9\\ d_{10}-x_\mathrm{B} \end{pmatrix}.$$

For the calculation of the parameter vector $\boldsymbol{x}$, the weight matrix $\mathbf{P}$ is still needed. The weights can be calculated by (2.9)

$$\mathbf{P}=\begin{pmatrix} p_1 & & & & & & & & & \\ & p_2 & & & & & & & & \\ & & p_3 & & & & & & & \\ & & & p_4 & & & & & & \\ & & & & p_5 & & & & & \\ & & & & & p_6 & & & & \\ & & & & & & p_7 & & & \\ & & & & & & & p_8 & & \\ & & & & & & & & p_9 & \\ & & & & & & & & & p_{10} \end{pmatrix}.$$

Now, the parameter vector and the residual vector can be calculated as

$$\boldsymbol{x}=(\mathbf{A}^\mathrm{T}\mathbf{P}\mathbf{A})^{-1}\mathbf{A}^\mathrm{T}\mathbf{P}\boldsymbol{l}$$
$$\boldsymbol{v}=\mathbf{A}\boldsymbol{x}-\boldsymbol{l}\,.$$

## 2.1.5 The Law of Error Propagation

As mentioned above, adjustment techniques have two main objectives. One task is estimation of optimal parameter values, but the second task is estimation of parameter accuracy. The basis for this second task is the law of error propagation.

In mathematical statistics as well as in adjustment theory, the measure for the scatter of a random value is its standard deviation. The meaning of that value shall be explained with an example.

If we measure a distance $d$ with observation value of 123.45 m, then this value has to be seen as a random sample of an infinite population of observations. The average value of this infinite quantity of observations is the so-called expectation value or true value. A standard deviation of $\sigma_d = \pm 2\,\mathrm{cm}$ means that the true but unknown value of the distance $d$ falls, with probability 67%, in the interval from $123.45\,\mathrm{m} - 2\,\mathrm{cm}$ to $123.45\,\mathrm{m} + 2\,\mathrm{cm}$, if normally distributed observations can be assumed. The standard deviation is a measure of the average deviation of the observed value from the theoretical true value. Usually, the standard deviation is denoted by $\sigma$. The square of the standard deviation is called the variance and is denoted by $\sigma^2$.

The standard deviation is not identical to the maximal deviation from the theoretical true value. However, there exists a rule of thumb for the ratio of both values

$$\text{maximal deviation} \approx 3 \times \text{standard deviation}\,.$$

The input parameters of an adjustment are observations which are, in the sense of mathematical statistics, random samples of a population. The true value $\lambda_i$ of an observation $l_i$ is unknown and exists only in theory, but its standard deviation $\sigma_i$ is mostly known. The output parameters of an adjustment are the parameters $x_j$ and the residual errors $v_i$. Parameters as well as residual errors are functions of the observations and therefore also random variables.

The law of error propagation describes the propagation of accuracies for linear functions of random variables. Applying this law to an adjustment, it is possible to calculate the standard deviations of the unknown parameters and those of the residual errors.

## 2.1.6 Error Propagation for Linear Functions

If we have a linear function of random variables with the structure

$$x = f_1 \cdot l_1 + f_2 \cdot l_2 + \cdots + f_n \cdot l_n\,,$$

and the standard deviations $\sigma_1 \dots \sigma_n$ of the random variables $l_1 \dots l_n$ are known, then the standard deviation of the parameter $x$ can be calculated by the formula

$$\sigma_x^2 = f_1^2 \cdot \sigma_1^2 + f_2^2 \cdot \sigma_2^2 + \dots + f_n^2 \cdot \sigma_n^2 \,. \tag{2.10}$$

The simplest case of a linear function is a sum. For a sum of random variables

$$x = \sum_{i=1}^{n} l_i = l_1 + l_2 + \dots + l_n \,,$$

the formula for the calculation of its standard deviation is

$$\sigma_x^2 = \sum_{i=1}^{n} \sigma_i^2 = \sigma_1^2 + \sigma_2^2 + \cdots \sigma_n^2$$
$$\Rightarrow \sigma_x = \sqrt{\sigma_1^2 + \sigma_2^2 + \dots + \sigma_n^2} \,.$$

In the case of a difference, application of (2.10) yields

$$x = l_1 - l_2 \Rightarrow \sigma_x^2 = \sigma_1^2 + \sigma_2^2 \,. \tag{2.11}$$

Consider Fig. 2.5 as an example. We see our one-dimensional coordinate system of the previous section. The control point A is fixed, and we want to calculate the coordinates of the new points by adding the distances

$$x_1 = x_\mathrm{A} + d_1 \,,$$
$$x_2 = x_\mathrm{A} + d_1 + d_2 \,,$$
$$\vdots$$
$$x_9 = x_\mathrm{A} + d_1 + d_2 + \dots + d_9 \,.$$

We now apply the law of error propagation to these functions and get

$$\sigma_{x1}^2 = \sigma_{d1}^2 \,,$$
$$\sigma_{x2}^2 = \sigma_{d1}^2 + \sigma_{d2}^2 \,,$$
$$\vdots$$
$$\sigma_{x9}^2 = \sigma_{d1}^2 + \sigma_{d2}^2 + \dots + \sigma_{d9}^2 \,.$$

## 2.1.7 The Importance of Covariances

If we applied the law of error propagation (2.11) to calculate the standard deviation of the difference $x_8 - x_7$ and compared the result with the standard deviation of the distance $d_8$ then we could see that the values are not identical

$$\sigma_{x7}^2 = \sigma_{d1}^2 + \sigma_{d2}^2 + \sigma_{d3}^2 + \sigma_{d4}^2 + \sigma_{d5}^2 + \sigma_{d6}^2 + \sigma_{d7}^2 \,,$$
$$\sigma_{x8}^2 = \sigma_{d1}^2 + \sigma_{d2}^2 + \sigma_{d3}^2 + \sigma_{d4}^2 + \sigma_{d5}^2 + \sigma_{d6}^2 + \sigma_{d7}^2 + \sigma_{d8}^2 \,.$$

Why is this? Does this imply $\sigma_{x8-x7}^2 = \sigma_{x8}^2 + \sigma_{x7}^2$?

The answer is that we neglected the covariance between the random values $x_7$ and $x_8$. However, before we explain calculations with covariances, we want to interpret the standard deviations of the distances $d_i$ and those of the coordinates $x_i$. The standard deviations of the coordinates $\sigma_{xi}$ represent the absolute accuracy of the coordinates relative to the reference frame. On the other hand, the standard deviations of the distances $\sigma_{di}$ represent the accuracy of the coordinates relative to each other.

However, what does covariance mean? If two calculated random values are functions of partially the same random variable arguments, they are stochastically dependent. The degree of their stochastic dependency is quantified by their covariance.

The parameters $x_7$ and $x_8$ are stochastically dependent because they are functions of partials of the same random variables. The distances $d_1$ to $d_7$ are arguments of both functions; just $d_8$ is an argument of only one of the two functions.

However, how can we incorporate these dependencies into the error propagation? This problem can be solved by a generalization of the law of error propagation. The general form of the law of error propagation can be represented in matrix notation. If there is a system of linear equations describing the functional dependency of parameters $x_j$ on the arguments $l_i$

$$\boldsymbol{x} = \mathbf{F} \cdot \boldsymbol{l} \,, \tag{2.12}$$

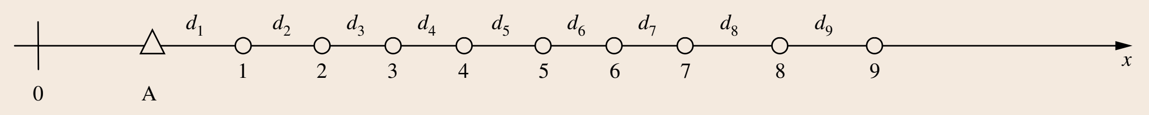

**Fig. 2.5** Addition of distances

and the standard deviations of the arguments $l_i$ are known, then the variances and covariances of the parameters $x_j$ can be calculated by the formula

$$\mathbf{C}_{xx} = \mathbf{F} \cdot \mathbf{C}_{ll} \cdot \mathbf{F}^{\mathrm{T}} \,. \tag{2.13}$$

In (2.13), the functional matrix $\mathbf{F}$ contains the coefficients of the linear functions. The matrix $\mathbf{C}_{ll}$ is called the covariance matrix of observations and contains the variances of observations on its principal diagonal and their covariances on its secondary diagonals which run from the lower left to the upper right corner. In the most common case of stochastically independent observations, $\mathbf{C}_{ll}$ is a diagonal matrix. $\mathbf{C}_{xx}$ is the covariance matrix of the unknown parameters and contains their variances and covariances

$$\mathbf{C}_{ll} = \begin{pmatrix} \sigma^2_{l1} & & & \\ & \sigma^2_{l2} & & \\ & & \ddots & \\ & & & \sigma^2_{\mathrm{ln}} \end{pmatrix} ,$$

$$\mathbf{C}_{xx} = \begin{pmatrix} \sigma^2_{x1} & \operatorname{cov}(x_1, x_2) & \cdots & \operatorname{cov}(x_1, x_u) \\ \operatorname{cov}(x_1, x_2) & \sigma^2_{x2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \operatorname{cov}(x_1, x_u) & \cdots & \cdots & \sigma^2_{xu} \end{pmatrix} .$$

Covariance matrices are always quadratic and symmetric.

Now, we want to apply the general law of error propagation to our example. First, we have to build the functional matrix $\mathbf{F}$. Because the functions are simple sums, $\mathbf{F}$ contains just the values one and zero

$$\mathbf{F} = \begin{pmatrix} 1 & & & & & & & & \\ 1 & 1 & & & & & & & \\ 1 & 1 & 1 & & & & & & \\ 1 & 1 & 1 & 1 & & & & & \\ 1 & 1 & 1 & 1 & 1 & & & & \\ 1 & 1 & 1 & 1 & 1 & 1 & & & \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & & \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} ,$$

$$\mathbf{C}_{ll} = \begin{pmatrix} \sigma^2_{d1} & & & & & & & & \\ & \sigma^2_{d2} & & & & & & & \\ & & \sigma^2_{d3} & & & & & & \\ & & & \sigma^2_{d4} & & & & & \\ & & & & \sigma^2_{d5} & & & & \\ & & & & & \sigma^2_{d6} & & & \\ & & & & & & \sigma^2_{d7} & & \\ & & & & & & & \sigma^2_{d8} & \\ & & & & & & & & \sigma^2_{d9} \end{pmatrix} .$$

In the result of the calculation, we get the covariance matrix $\mathbf{C}_{xx}$, which is fully allocated (fully equipped).

Now, we can use elements of $\mathbf{C}_{xx}$ for a further calculation to get the standard deviation of the coordinate difference $x_8 - x_7$. The functional matrix $\mathbf{F}$ for that case is simple again

$$\mathbf{F} = \begin{pmatrix} -1 & 1 \end{pmatrix} .$$

The $\mathbf{C}_{ll}$ matrix contains the variances of $x_7$ and $x_8$ as well as their covariance from the previous calculation

$$\mathbf{C}_{ll} = \begin{pmatrix} \sigma^2_{x7} & \operatorname{cov}(x_7, x_8) \\ \operatorname{cov}(x_7, x_8) & \sigma^2_{x8} \end{pmatrix} .$$

If we solve the matrix equation for the general law of error propagation in a symbolic way then we get the expression

$$\sigma^2_{x8-x7} = \sigma^2_{x7} + \sigma^2_{x8} - 2 \cdot \operatorname{cov}(x_7, x_8) \,.$$

As we can see, this formula which does not neglect the covariance between dependent random variables yields the correct result.

### 2.1.8 Adjustment and Error Propagation

If we consider the calculation formula for the unknown parameters in an adjustment then we can see that the parameters are linear functions of the observations, i. e.,

$$\boldsymbol{x} = \underbrace{(\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{P}}_{\mathbf{F}} \cdot \boldsymbol{l} \,.$$

The expression $(\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{P}$ in this formula is equivalent to the functional matrix $\mathbf{F}$ in (2.12). This approach allows for the application of the law of error propagation to calculate the covariance matrix of the unknown parameters.

The basis of further calculations is the empirical standard deviation of unit weight. The formula for its calculation is

$$s_0 = \sqrt{\frac{\boldsymbol{v}^{\mathrm{T}}\mathbf{P}\boldsymbol{v}}{r}}\,, \tag{2.14}$$

where $\boldsymbol{v}$ is the residual vector and $r$ is the redundancy. The redundancy is the number of surplus observations, equal to the difference between the number of observations $n$ and the number of unknown parameters $u$, i.e.,

$$r = n - u\,. \tag{2.15}$$

The value $s_0$ can be interpreted as the empirical standard deviation of an observation with weight $p = 1$.

A standard deviation $\sigma$ should not be mistaken for an empirical standard deviation $s$. A standard deviation is a constant value and an input parameter of an adjustment calculation, whereas an empirical standard deviation is a random variable which is estimated as an output parameter of an adjustment. After an adjustment calculation, the empirical standard deviation $s_0$ should be approximately equal to the standard deviation $\sigma_0$, which means that the quotient $s_0/\sigma_0$ should be in the interval 0.7–1.3. If $s_0$ is too small, the a priori assumption of observational accuracy ($\sigma_i$) was too pessimistic. Too large a value of $s_0$ often indicates that there is a blunder among the observations.

The complete derivation of the formula for the covariance matrix of unknown parameters is too complex to be presented here. Therefore, the formula is just given

$$\mathbf{C}_{xx} = s_0^2 \cdot \mathbf{Q}_{xx} = s_0^2 \cdot (\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A})^{-1}\,. \tag{2.16}$$

### 2.1.9 Positional Accuracy Improvement as an Adjustment Problem

In this section we want to give a typical example for a PAI adjustment problem. First, we consider the workflow from map digitalization to improved global coordinates from the point of view of adjustment. Then, we simplify the example to a one-dimensional problem and reproduce the process with a concrete calculation. Figure 2.6 depicts the workflow.

The result of scanning a map is a raster image with a row–column coordinate system. The coordinates of specific points (building corners, boundary points, etc.) are determined in the raster system. On the basis of the scan resolution, the raster coordinates can be converted into metrical map coordinates. If the scan resolution is measured in dots per inch (dpi) then the converting

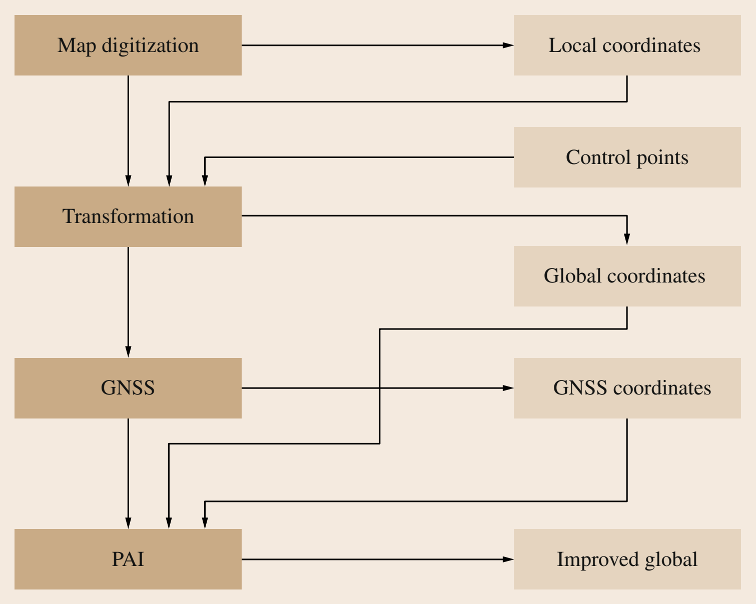

**Fig. 2.6** PAI workflow for map digitization

form is

$$x_{\text{map}} = x_{\text{raster}} \cdot \frac{0.0254\,\text{m}}{\text{resolution [dpi]}} \,.$$

The standard deviations and thereby the variances of the digitized map coordinates are dependent on the scale factor and the quality of the underlying map. As a rule of thumb, the standard deviation of a map coordinate value $\sigma_{\text{map}}$ is about 0.5 mm.

The measured coordinates are random variables, and they are stochastically dependent because the point positions on a map result from real-world measurements and plots, both of which were made following the principle of adjacent points. Therefore, the coordinate values are stochastically dependent, analogous to the example given in the section above for adjustment with several unknown parameters.

However, in general, it is impossible to reconstruct a map history in detail, and therefore the covariances between digitized coordinates are not known. The outcome of a digitalization process is an observation vector $\boldsymbol{l}_r$ with map coordinates and a covariance matrix of these observations $\mathbf{C}_{ll}$ with known principal diagonal but unknown covariances.

## 2.1.10 Transformation

Map coordinates are transformed into a higher-level global reference frame. Control points with known coordinates in the local map coordinate system and the global reference frame are used to determine the necessary transformation parameters. The general transformation approach is

$$\boldsymbol{x}_{\text{global}} = \boldsymbol{t} + \mathbf{R} \cdot \boldsymbol{x}_{\text{map}} \,, \qquad (2.17)$$

where $\boldsymbol{x}_{\text{global}}$ is the vector of global coordinates, $\boldsymbol{t}$ is the translation vector of transformation, $\mathbf{R}$ is the rotation matrix of transformation, and $\boldsymbol{x}_{\text{map}}$ is the vector of map coordinates.

As we can see, the global coordinates are linear functions of map coordinates. Applying the law of error propagation to (2.17), it is possible to calculate the standard deviations of global coordinates while their covariances remain unknown.

## 2.1.11 GNSS Measurement

For a number of points, global coordinates with higher accuracy are determined by GNSS measurements. Each GNSS point yields two redundant coordinates.

### Positional Accuracy Improvement

If the global coordinates of a GIS were stochastically independent, then we could just exchange the less accurate coordinates for more accurate ones. However, as they are stochastically dependent, such a procedure would lead to a violation of geometrical neighborhood relationships.

Figure 2.7 illustrates the necessity to consider stochastic dependencies between coordinates of adjacent points. In the presented example, coordinates with higher accuracy for the building corners were determined, but not for the adjacent tree. If the correlations remain unconsidered and the coordinates of the building are exchanged, then the tree seems to stand inside the building after PAI. However, if the stochastic dependencies are considered the tree is shifted with the building during PAI.

## 2.1.12 Improving Absolute Geometry

However, how can correlations be quantified and taken into account? One option is the introduction of artificial

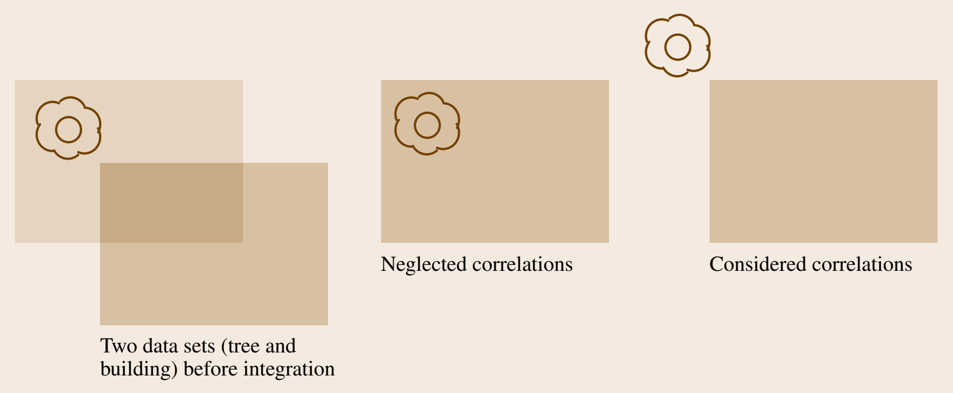

**Fig. 2.7** PAI neglecting and considering correlations

covariances. These can always be calculated as functions of the distances between two points. The smaller the distance between two points, the larger their covariance becomes. However, in practice, this approach is difficult to handle. Often there are more then 100 000 points to be processed, which would lead to extremely large covariance matrices.

A more practical solution is the introduction of pseudo-observations. In this approach stochastic dependencies between GIS coordinates are modeled by coordinate differences. The basis of the determination of these pseudo-observations is Delaunay triangulation (Fig. 2.8).

Before triangulation, the point positions are expressed uniquely by global coordinates. This parameterization is exchanged with pseudo observations generated from coordinate differences of points which are adjacent in the triangle network. This new parameterization is redundant but still consistent (Fig. 2.9).

Now the observation vector $\boldsymbol{l}_{\text{GNSS}}$ is extended with GNSS coordinates of higher accuracy (Fig. 2.10).

This leads to an adjustment problem with the improved global coordinates as unknown parameters and the coordinate distances and GNSS coordinates as observations. The observation equations have the structure

$$\begin{aligned} &\vdots \\ \Delta x_{ij} + v_{\Delta x} &= x_j - x_i \;, \\ \Delta y_{ij} + v_{\Delta y} &= y_j - y_i \;, \\ &\vdots \\ x_{\text{GNSS\_}i} + v_x &= x_i \;, \\ y_{\text{GNSS\_}i} + v_y &= y_i \;. \\ &\vdots \end{aligned}$$

These equations represent the functional model of the adjustment. The stochastic model is determined by the weights of the observations, which are functions of their standard deviations. The standard deviations of the GNSS coordinates depend on the measurement procedure applied and on the reproduction accuracy of the measured points. A practical value is $\sigma_{xy} = \pm 2\,\text{cm}$. The standard deviations of the coordinate differences are functions of those of the underlying map coordinates and of the corresponding point distances.

We want to explain this with an example. If map coordinates with standard deviation of $\sigma_{xy} = \pm 1\,\text{m}$ are given, then the formula for the calculation of the standard deviation of a coordinate difference could be

$$\sigma_\Delta = \sigma_{xy} \frac{d}{d_0} \quad \text{with} \quad d_0 = 100\,\text{m} \,.$$

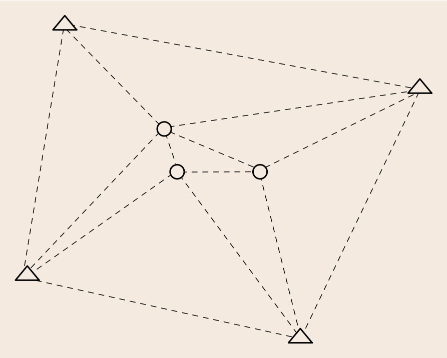

**Fig. 2.8** Delaunay triangulation

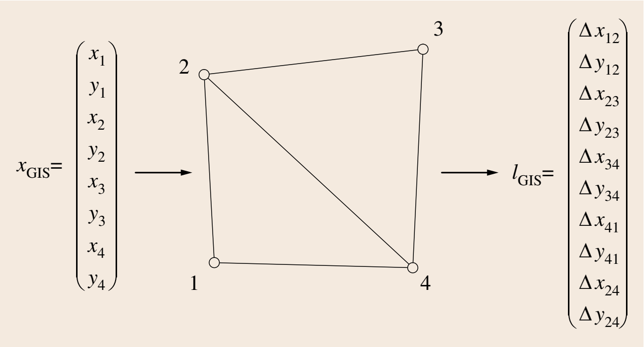
$x_{\text{GIS}}=$
$\begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{pmatrix}$
2
3
1
4
$l_{\text{GIS}}=$
$\begin{pmatrix} \Delta x_{12} \\ \Delta y_{12} \\ \Delta x_{23} \\ \Delta y_{23} \\ \Delta x_{34} \\ \Delta y_{34} \\ \Delta x_{41} \\ \Delta y_{41} \\ \Delta x_{24} \\ \Delta y_{24} \end{pmatrix}$

**Fig. 2.9** Replacement of point-coordinates by pseudo-observations (distances)

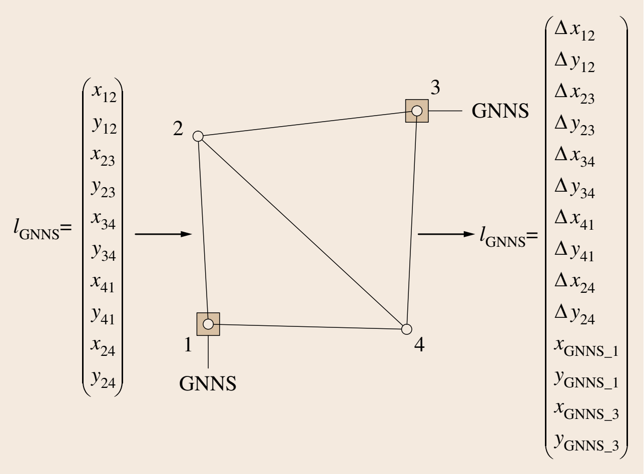
$l_{\text{GNNS}}=$
$\begin{pmatrix} x_{12} \\ y_{12} \\ x_{23} \\ y_{23} \\ x_{34} \\ y_{34} \\ x_{41} \\ y_{41} \\ x_{24} \\ y_{24} \end{pmatrix}$
2
3
GNNS
1
GNNS
4
$l_{\text{GNNS}}=$
$\begin{pmatrix} \Delta x_{12} \\ \Delta y_{12} \\ \Delta x_{23} \\ \Delta y_{23} \\ \Delta x_{34} \\ \Delta y_{34} \\ \Delta x_{41} \\ \Delta y_{41} \\ \Delta x_{24} \\ \Delta y_{24} \\ x_{\text{GNNS\_1}} \\ y_{\text{GNNS\_1}} \\ x_{\text{GNNS\_3}} \\ y_{\text{GNNS\_3}} \end{pmatrix}$

**Fig. 2.10** Extension of the observation vector by GNSS coordinates

In this case, an observational weight is inversely proportional to the square of the point distance.

The results of adjustment are improved coordinates for all points and their covariance matrix. The point accuracy depends then on the distance to the newly introduced GNSS points.

### 2.1.13 Improving Relative Geometry

Accuracy improvement of GIS coordinates is not exclusively effected by the introduction of precise global coordinates. There are also observations describing the relative geometry between adjacent points; for instance, it is known that, mostly, walls of buildings are rectangular and parcel limits are straight. Such geometrical constraints can be modeled by the introduction of corresponding observations. Rectangularity, for instance, can be expressed by a scalar product. The standard deviation of this scalar product observation can be calculated by error propagation of the corresponding point coordinates. Because construction workers can build a house with accuracy of approximately 2 cm, a standard deviation of coordinates of about the same size would be adequate.

## 2.2 2-D Datum Transformations

The position of points can be defined in a variety of different local coordinate systems, e.g., in the instrument system of a total station, the local system of a digitized map, or as pixel coordinates of an orthophoto. In such a case, usually it is the objective of processing to determine the coordinates of these points in a global reference frame. On the other hand, datasets often use a different datum for point coordinates. In all these cases it is necessary to transform point coordinates from one datum into another.

Coordinate transformation can generally be described as a mapping of a metric space $R_\mathrm{s}$ into another metric space $R_\mathrm{t}$. The mapping is modeled by a transformation function $f(\boldsymbol{p})$, where $\boldsymbol{p}$ is the vector of the transformation parameters, i.e.,

$$f(p): R_\mathrm{s} \rightarrow R_\mathrm{t} .$$

In the following, we call the space $R_\mathrm{s}$ the start system s, and $R_\mathrm{t}$ the target system t.

Frequently, the transformation parameters are not known a priori. Then, the parameters have to be determined by a calculation using control points. The coordinates of the control points are known in both the start and the target system, whereas the coordinates of the new points are known only in the start system.

In the majority of cases, the number of control point coordinates exceeds the number of transformation parameters to be determined. In such cases, the calculation of transformation parameters is an adjustment problem.

In the general approach, the control point coordinates in both the start and the target system are considered as observations. This leads to an adjustment by a (mostly) nonlinear Gauss–Helmert model, in which the connection between observations and unknowns is expressed in an implicit way as

$$f[(\boldsymbol{l}+\boldsymbol{v}),\, p] = 0 \quad \text{with} \quad \boldsymbol{l} = \begin{pmatrix} \boldsymbol{x}_\mathrm{s} \\ \boldsymbol{x}_\mathrm{t} \end{pmatrix} . \tag{2.18}$$

For simplicity, we will use a Gauss–Markov model with the coordinates in the target system $x_\mathrm{t}$ as observations and the coordinates in the start system as constants, i.e.,

$$\boldsymbol{x}_\mathrm{t} + \boldsymbol{v} = f(p, \boldsymbol{x}_\mathrm{s}) . \tag{2.19}$$

### 2.2.1 Centroid Reduction

For several reasons it is appropriate to perform centroid reduction before the actual transformation calculation takes place. For the four- and six-parameter approaches, centroid reduction leads to a diagonal structure of the normal equation matrix and simple closed forms for the calculation of the transformation parameters. For nonlinear approaches, this significantly improves the condition of the normal equation system. If the distance between the coordinate origin and coordinate centroid in the start system is much larger then the size of the transformed point set, a calculation may become impossible without centroid reduction.

First, the centroid coordinates of the control points of the start system are calculated as

$$x_\mathrm{c} = \frac{1}{n}\sum_{i=1}^{n} x_{\mathrm{s}i} , \qquad y_\mathrm{c} = \frac{1}{n}\sum_{i=1}^{n} y_{\mathrm{s}i} . \tag{2.20}$$

Then, the centroid coordinates are subtracted from all (control points and new points) coordinates of the start system, thus

$$\begin{aligned} x'_{\mathrm{s}i} &= x_{\mathrm{s}i} - x_\mathrm{c} , \\ y'_{\mathrm{s}i} &= y_{\mathrm{s}i} - y_\mathrm{c} . \end{aligned} \tag{2.21}$$

The transformation parameters are calculated using the reduced coordinates of the control points as

$$p = f(\boldsymbol{x}'_{\mathrm{s}}, \boldsymbol{x}_{\mathrm{t}}) \,. \tag{2.22}$$

With the now known transformation parameters, the coordinates of the new points in the target system can be calculated as

$$\boldsymbol{x}_{\mathrm{t}} = f(p, \boldsymbol{x}'_{\mathrm{s}}) \,. \tag{2.23}$$

## 2.2.2 The Four-Parameter (Helmert) Transformation

The four-parameter transformation is the most common approach in GIS. It provides four degrees of freedom: two translations, one rotation, and one scale factor. The residual equations have the structure

$$\begin{aligned} x_{\mathrm{t}i} + v_{xi} &= t_x + \cos\varphi \cdot s \cdot x'_{\mathrm{s}i} - \sin\varphi \cdot s \cdot y'_{\mathrm{s}i} \,, \\ y_{\mathrm{t}i} + v_{yi} &= t_y + \sin\varphi \cdot s \cdot x'_{\mathrm{s}i} + \cos\varphi \cdot s \cdot y'_{\mathrm{s}i} \,, \end{aligned} \tag{2.24}$$

where $t_x$ is the translation in $x$, $t_y$ is the translation in $y$, $\varphi$ is the rotation angle, and $s$ is the scale factor.

Usually, the expressions $\cos\varphi \cdot s$ and $\sin\varphi \cdot s$ are substituted by the variables $a$ and $b$, which act as unknowns in the adjustment calculation. We then get

$$\begin{aligned} x_{\mathrm{t}i} + v_{xi} &= t_x + a \cdot x'_{\mathrm{s}i} - b \cdot y'_{\mathrm{s}i} \,, \\ y_{\mathrm{t}i} + v_{yi} &= t_y + b \cdot x'_{\mathrm{s}i} + a \cdot y'_{\mathrm{s}i} \,. \end{aligned} \tag{2.25}$$

The original coordinates of the start system are substituted by their centroid-reduced coordinates, thus

$$\begin{aligned} x_{\mathrm{t}i} + v_{xi} &= t'_x + a \cdot x'_{\mathrm{s}i} - b \cdot y'_{\mathrm{s}i} \,, \\ y_{\mathrm{t}i} + v_{yi} &= t'_y + b \cdot x'_{\mathrm{s}i} + a \cdot y'_{\mathrm{s}i} \,. \end{aligned}$$

This approach can also be written in matrix notation as

$$\boldsymbol{x}_{\mathrm{t}} + \boldsymbol{v} = \boldsymbol{t} + \mathbf{R}s \cdot \boldsymbol{x}'_{\mathrm{s}} = \boldsymbol{t} + \mathbf{D} \cdot \boldsymbol{x}'_{\mathrm{s}} \quad \text{with} \quad \boldsymbol{t} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} ,$$

$$\mathbf{D} = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} = \begin{pmatrix} \cos\varphi \cdot s & -\sin\varphi \cdot s \\ \sin\varphi \cdot s & \cos\varphi \cdot s \end{pmatrix} .$$

For the calculation of the translation parameters, it is necessary to determine also the control point centroid coordinates in the target system. The parameters $t_x$ and $t_y$ can then be calculated directly as differences of the control point centroid coordinates between the start and the target system, i.e.,

$$\begin{aligned} t_x &= x_{\mathrm{tc}} - x_{\mathrm{sc}} \,, \\ t_y &= y_{\mathrm{tc}} - y_{\mathrm{sc}} \,. \end{aligned} \tag{2.26}$$

The substituted parameters $a$ and $b$ are calculated from the reduced control point coordinates in the start system and the control point coordinates in the target system as

$$\begin{aligned} a &= \frac{\sum \left( x'_{\mathrm{s}i} x_{\mathrm{t}i} + y'_{\mathrm{s}i} y_{\mathrm{t}i} \right)}{\sum \left( {x'_{\mathrm{s}i}}^2 + {y'_{\mathrm{s}i}}^2 \right)} \,, \\ b &= \frac{\sum \left( x'_{\mathrm{s}i} y_{\mathrm{t}i} + y'_{\mathrm{s}i} x_{\mathrm{t}i} \right)}{\sum \left( {x'_{\mathrm{s}i}}^2 + {y'_{\mathrm{s}i}}^2 \right)} \,. \end{aligned} \tag{2.27}$$

The target coordinates of the new points can now be calculated by inserting the reduced start coordinates into (2.25).

By converting (2.25) and inserting the start and target coordinates of the control points, we are able to calculate the residuals as

$$\begin{aligned} v_{xi} &= t_x + a \cdot x'_{\mathrm{s}i} - b \cdot y'_{\mathrm{s}i} - x_{\mathrm{t}i} \,, \\ v_{yi} &= t_y + b \cdot x'_{\mathrm{s}i} + a \cdot y'_{\mathrm{s}i} - y_{\mathrm{t}i} \,. \end{aligned} \tag{2.28}$$

A general value for the evaluation of the achieved accuracy is the empirical standard deviation $\hat{\sigma}_0$ of the observed target coordinates. This value can be calculated from the residuals as

$$\begin{aligned} \hat{\sigma}_0^2 &= \frac{\boldsymbol{v}^{\mathrm{T}} \boldsymbol{v}}{2n_{\mathrm{p}} - 4} = \frac{\sum \left( v_{xi}^2 + v_{yi}^2 \right)}{2n_{\mathrm{p}} - 4} \,, \\ \hat{\sigma}_0 &= \sqrt{\hat{\sigma}_0^2} \,, \end{aligned} \tag{2.29}$$

where $n_{\mathrm{p}}$ is the number of control points.

By variance propagation, the empirical variances of the transformation parameters can now be calculated as

$$\hat{\sigma}_{\mathrm{t}x}^2 = \hat{\sigma}_{\mathrm{t}y}^2 = \frac{\hat{\sigma}_0^2}{n_{\mathrm{p}}} \,, \tag{2.30}$$

$$\hat{\sigma}_{\mathrm{a}}^2 = \hat{\sigma}_{\mathrm{o}}^2 = \frac{\hat{\sigma}_0^2}{\sum \left( {x'_{\mathrm{s}}}^2 + {y'_{\mathrm{s}}}^2 \right)} \,. \tag{2.31}$$

With these values, the empirical variances (and thereby standard deviations) of the transformed new point coordinates in the target system can be calculated as

$$\begin{aligned} \hat{\sigma}_{x\mathrm{t}}^2 &= \hat{\sigma}_{y\mathrm{t}}^2 = \hat{\sigma}_{\mathrm{t}x}^2 + \hat{\sigma}_a^2 \cdot \left( {x'_{\mathrm{s}}}^2 + {y'_{\mathrm{s}}}^2 \right) \,, \\ \hat{\sigma}_{x\mathrm{t}} &= \hat{\sigma}_{y\mathrm{t}} = \sqrt{\hat{\sigma}_{x\mathrm{t}}^2} \,. \end{aligned} \tag{2.32}$$

The standard deviations of the transformed coordinates are dependent on the distance to the centroid in the start system. This is due to the influence of the uncertainty of the rotation parameter, which grows with this distance.

Sometimes, the rotation angle $\varphi$ and the scale factor $s$ are required, not just the substituted unknowns $a$ and $b$. These values can be derived from $a$ and $b$ as

$$\varphi = \arctan\frac{b}{a}\,,$$
$$s = \sqrt{a^2+b^2}\,.$$

### 2.2.3 Six-Parameter (Affine) Transformation

The six-parameter transformation, like the four-parameter transformation, is a linear adjustment problem. Also here, the transformation parameters can be calculated by simple sum formulas. This approach provides six degrees of freedom: two translations, two rotations, and two scale factors. The residual equations have the following structure

$$\begin{aligned} x_{\mathrm{t}i} + v_{xi} &= t_x + \cos\varphi_x \cdot s_x \cdot x'_{\mathrm{s}i} - \sin\varphi_x \cdot s_x \cdot y'_{\mathrm{s}i}\,,\\ y_{\mathrm{t}i} + v_{yi} &= t_y + \sin\varphi_y \cdot s_y \cdot x'_{\mathrm{s}i} + \cos\varphi_y \cdot s_y \cdot y'_{\mathrm{s}i}\,, \end{aligned} \tag{2.33}$$

where $t_x$ is the translation in $x$, $t_y$ is the translation in $y$, $\varphi_x$ is the rotation angle of the $x$-axis, $\varphi_y$ is the rotation angle of the $y$-axis, $s_x$ is the scale factor in $x$, and $s_y$ is the scale factor in $y$.

The expressions $\cos\varphi_x \cdot s_x$, $\sin\varphi_x \cdot s_x$, $\cos\varphi_y \cdot s_y$, and $\sin\varphi_y \cdot s_y$ can be substituted by the variables $a$–$d$, which act as unknowns in the adjustment calculation. We then get

$$\begin{aligned} x_{\mathrm{t}i} + v_{xi} &= t_x + a \cdot x'_{\mathrm{s}i} + b \cdot y'_{\mathrm{s}i}\,,\\ y_{\mathrm{t}i} + v_{yi} &= t_y + c \cdot x'_{\mathrm{s}i} + d \cdot y'_{\mathrm{s}i}\,. \end{aligned} \tag{2.34}$$

This approach can also be written in matrix notation as

$$\boldsymbol{x}_{\mathrm{t}} + \boldsymbol{v} = \boldsymbol{t} + \mathbf{R}s \cdot \boldsymbol{x}'_{\mathrm{s}} = \boldsymbol{t} + \mathbf{D} \cdot \boldsymbol{x}'_{\mathrm{s}} \quad \text{with} \quad \boldsymbol{t} = \begin{pmatrix} t_x \\ t_y \end{pmatrix}\,,$$
$$\mathbf{D} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \cos\varphi_x \cdot s_x & -\sin\varphi_x \cdot s_x \\ \sin\varphi_y \cdot s_y & \cos\varphi_y \cdot s_y \end{pmatrix}\,. \tag{2.35}$$

As in the four-parameter approach, the parameters $t_x$ and $t_y$ are directly calculated as differences of the control point centroid coordinates between the start and the target system as

$$t_x = x_{\mathrm{tc}} - x_{\mathrm{sc}}\,,$$
$$t_y = y_{\mathrm{tc}} - y_{\mathrm{sc}}\,.$$

The substituted parameters $a$–$d$ are calculated from the reduced control point coordinates in the start system and the control point coordinates in the target system using

$$\begin{aligned} a &= \frac{\sum x'_{\mathrm{s}i} x_{\mathrm{t}i}}{\sum {x'_{\mathrm{s}i}}^2}\,, & b &= \frac{\sum y'_{\mathrm{s}i} x_{\mathrm{t}i}}{\sum {y'_{\mathrm{s}i}}^2}\,,\\ c &= \frac{\sum x'_{\mathrm{s}i} y_{\mathrm{t}i}}{\sum {x'_{\mathrm{s}i}}^2}\,, & d &= \frac{\sum y'_{\mathrm{s}i} y_{\mathrm{t}i}}{\sum {y'_{\mathrm{s}i}}^2}\,. \end{aligned} \tag{2.36}$$

The target coordinates of the new points can now be calculated by inserting the reduced start coordinates into (2.36).

By converting (2.17) and inserting the start and target coordinates of the control points, we are able to calculate the residuals as

$$\begin{aligned} v_{xi} &= t_x + a \cdot x'_{\mathrm{s}i} + b \cdot y'_{\mathrm{s}i} - x_{\mathrm{t}i}\,,\\ v_{yi} &= t_y + c \cdot x'_{\mathrm{s}i} + d \cdot y'_{\mathrm{s}i} - y_{\mathrm{t}i}\,. \end{aligned} \tag{2.37}$$

The empirical standard deviation $\hat{\sigma}_0$ of the observed target coordinates can be calculated by (2.12), analogous to the four-parameter transformation.

Variance propagation provides the empirical variances of the transformation parameters as

$$\hat{\sigma}^2_{\mathrm{t}x} = \hat{\sigma}^2_{\mathrm{t}y} = \frac{\hat{\sigma}^2_0}{n_{\mathrm{p}}}\,, \tag{2.38}$$

$$\hat{\sigma}^2_a = \hat{\sigma}^2_c = \frac{\hat{\sigma}^2_0}{\sum {x'_{\mathrm{s}}}^2}\,, \tag{2.39}$$

$$\hat{\sigma}^2_b = \hat{\sigma}^2_d = \frac{\hat{\sigma}^2_0}{\sum {y'_{\mathrm{s}}}^2}\,. \tag{2.40}$$

With these values, the empirical variances (and thereby the standard deviations) of the transformed new point coordinates in the target system can be calculated as

$$\hat{\sigma}^2_{x\mathrm{t}} = \hat{\sigma}^2_{\mathrm{t}x} + \hat{\sigma}^2_a \cdot {x'_{\mathrm{s}}}^2 + \hat{\sigma}^2_b \cdot {y'_{\mathrm{s}}}^2\,, \quad \hat{\sigma}_{x\mathrm{t}} = \sqrt{\hat{\sigma}^2_{x\mathrm{t}}}\,, \tag{2.41}$$

$$\hat{\sigma}^2_{y\mathrm{t}} = \hat{\sigma}^2_{\mathrm{t}y} + \hat{\sigma}^2_c \cdot {x'_{\mathrm{s}}}^2 + \hat{\sigma}^2_d \cdot {y'_{\mathrm{s}}}^2\,, \quad \hat{\sigma}_{y\mathrm{t}} = \sqrt{\hat{\sigma}^2_{y\mathrm{t}}}\,. \tag{2.42}$$

Also here, the standard deviations of the transformed coordinates are dependent on the distance to the centroid in the start system.

The rotation angles $\varphi_x$ and $\varphi_y$ as well as the scale factors $s_x$ and $s_y$ can be derived from $a$–$d$ as follows

$$\varphi_x = \arctan\frac{b}{a}\,, \qquad \varphi_y = \arctan\frac{d}{c}\,, \tag{2.43}$$

$$s_x = \sqrt{a^2+b^2}\,, \qquad s_y = \sqrt{c^2+d^2}\,. \tag{2.44}$$

### 2.2.4 Three-Parameter Transformation

Unlike the four- and six-parameter approaches, the three-parameter transformation is a nonlinear adjustment problem. The solution is found through an iterative process by applying Newton's method. At the start of the iteration process, approximate values for the transformation parameters are needed. These approximate values can easily be obtained by performing a linear four-parameter transformation.

The three degrees of freedom here are two translations and one rotation. A scale factor is not modeled. The residual equations have the structure

$$\begin{aligned} x_{\text{t}i} + v_{xi} &= t_x + \cos\varphi \cdot x'_{\text{s}i} - \sin\varphi \cdot y'_{\text{s}i} \ , \\ y_{\text{t}i} + v_{yi} &= t_y + \sin\varphi \cdot x'_{\text{s}i} + \cos\varphi \cdot y'_{\text{s}i} \ . \end{aligned} \tag{2.45}$$

In general, we can say that the vector of the adjusted control point coordinates is a function of the parameter vector (the coordinates in the start systems are constants), i.e.,

$$\boldsymbol{x}_{\text{t}} + \boldsymbol{v} = f(p) \ . \tag{2.46}$$

In the considered case, $f(\boldsymbol{p})$ is a nonlinear function, therefore $f(\boldsymbol{p})$ has to be linearized. The linearization is done by developing $f(\boldsymbol{p})$ in a Taylor series of degree one as

$$\boldsymbol{x}_{\text{t}} + \boldsymbol{v} = f(p_0) + \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} \cdot \underbrace{(p - p_0)}_{\Delta p} \ . \tag{2.47}$$

In this expression, $\boldsymbol{p}_0$ is the vector of the proximity parameters. The vector $\Delta\boldsymbol{p}$ contains the substitute unknowns. To get the substitute observations, we rearrange expression (2.47) to

$$\underbrace{\boldsymbol{x}_{\text{t}} - f(p_0)}_{\tilde{\boldsymbol{x}}} + \boldsymbol{v} = \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} \cdot \Delta p \ . \tag{2.48}$$

In this form, $\tilde{\boldsymbol{x}}$ is the vector of the substitute observations. Its elements are

$$\begin{aligned} \tilde{x}_{\text{t}i} &= x_{\text{t}i} - t_x^0 - \cos\varphi^0 \cdot x'_{\text{s}i} + \sin\varphi^0 \cdot y'_{\text{s}i} \ , \\ \tilde{y}_{\text{t}i} &= y_{\text{t}i} - t_y^0 - \sin\varphi^0 \cdot x'_{\text{s}i} - \cos\varphi^0 \cdot y'_{\text{s}i} \ . \end{aligned} \tag{2.49}$$

The linearized residual equations are then

$$\begin{aligned} \tilde{x}_{\text{t}i} + v_{xi} &= \Delta t_x - \left(\sin\varphi^0 \cdot x'_{\text{s}i} + \cos\varphi^0 \cdot y'_{\text{s}i}\right) \cdot \Delta\varphi \ , \\ \tilde{y}_{\text{t}i} + v_{yi} &= \Delta t_y + \left(\cos\varphi^0 \cdot x'_{\text{s}i} - \sin\varphi^0 \cdot y'_{\text{s}i}\right) \cdot \Delta\varphi \ . \end{aligned} \tag{2.50}$$

The overdetermined equation system (2.50) can now be solved by using the rules of nonlinear Gauss–Markov model adjustment.

### 2.2.5 Five-Parameter Transformation

Like the three-parameter transformation, the five-parameter transformation is a nonlinear adjustment problem. The degrees of freedom are two translations, one rotation, and two scale factors. The nonlinear residual equations are

$$\begin{aligned} x_{\text{t}i} + v_{xi} &= t_x + \left(\cos\varphi \cdot x'_{\text{s}i} - \sin\varphi \cdot y'_{\text{s}i}\right) \cdot s_x \ , \\ y_{\text{t}i} + v_{yi} &= t_y + \left(\sin\varphi \cdot x'_{\text{s}i} + \cos\varphi \cdot y'_{\text{s}i}\right) \cdot s_y \ . \end{aligned} \tag{2.51}$$

The equations for the substitute observations are

$$\begin{aligned} \tilde{x}_{\text{t}i} &= x_{\text{t}i} - t_x^0 - \left(\cos\varphi^0 \cdot x'_{\text{s}i} + \sin\varphi^0 \cdot y'_{\text{s}i}\right) \cdot s_x^0 \ , \\ \tilde{y}_{\text{t}i} &= y_{\text{t}i} - t_y^0 - \left(\sin\varphi^0 \cdot x'_{\text{s}i} - \cos\varphi^0 \cdot y'_{\text{s}i}\right) \cdot s_y^0 \ . \end{aligned} \tag{2.52}$$

The linearized residual equations are

$$\begin{aligned} \tilde{x}_{\text{t}i} + v_{xi} &= \Delta t_x - \left(\sin\varphi^0 \cdot x'_{\text{s}i} + \cos\varphi^0 \cdot y'_{\text{s}i}\right) \cdot s_x^0 \cdot \Delta\varphi \\ &\quad + \left(\cos\varphi^0 \cdot x'_{\text{s}i} - \sin\varphi^0 \cdot y'_{\text{s}i}\right) \cdot \Delta s_x \ , \\ \tilde{y}_{\text{t}i} + v_{yi} &= \Delta t_y + \left(\cos\varphi^0 \cdot x'_{\text{s}i} - \sin\varphi^0 \cdot y'_{\text{s}i}\right) \cdot s_y^0 \cdot \Delta\varphi \\ &\quad + \left(\sin\varphi^0 \cdot x'_{\text{s}i} + \cos\varphi^0 \cdot y'_{\text{s}i}\right) \cdot \Delta s_y \ . \end{aligned} \tag{2.53}$$

Like the three-parameter transformation, the overdetermined system of the linearized residual equations (2.53) can be solved with by Gauss–Markov model adjustment.

### 2.2.6 Conformal Transformation with Complex Polynomials

A map is called conformal (or angle preserving) if it preserves oriented angles between curves with respect to their orientation (i. e., not just the acute angle). Conformal maps preserve both angles and the shapes of infinitesimally small figures, but not necessarily their size [2.3] (Chap. 8).

Many projections used in GIS are conformal mappings of a double-curved reference surface (ellipsoid or sphere) into a plane. Examples are the Universal Transverse Mercator (UTM), Gauss–Krüger, stereographic or Lambert projection. Frequently, the task is a datum transformation of point sets whose coordinates are given as a conformal projection. The classical approach for the solution of this task is a three-dimensional (3-D) seven-parameter transformation (Fig. 2.11). For this purpose, the projected coordinates have to be converted into 3-D geocentric Cartesian coordinates. A prerequisite for that conversion is the knowledge of the ellipsoidal heights of the control points in both the

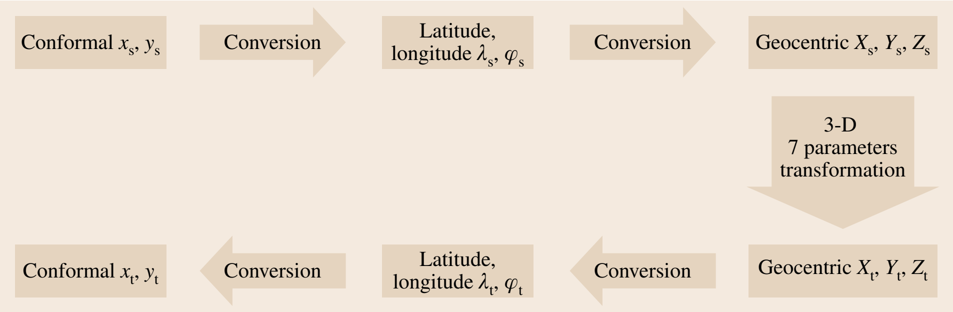
Conformal $x_s$, $y_s$
Conversion
Latitude, longitude $\lambda_s$, $\varphi_s$
Conversion
Geocentric $X_s$, $Y_s$, $Z_s$
3-D 7 parameters transformation
Geocentric $X_t$, $Y_t$, $Z_t$
Conversion
Latitude, longitude $\lambda_t$, $\varphi_t$
Conversion
Conformal $x_t$, $y_t$

**Fig. 2.11** Workflow of a 3-D seven-parameter transformation

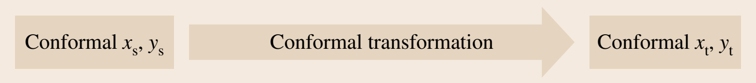

**Fig. 2.12** Workflow of a 2-D conformal transformation

start and the target system. Figure 2.11 shows the workflow of this transformation.

An alternative to this lengthy approach can be a 2-D conformal transformation (Fig. 2.12). This approach directly maps the conformal coordinates $x_s$, $y_s$ of the start system into the coordinates $x_t$, $y_t$ in the target system.

The underlying idea of this approach is the mathematical rule that the analytical function of a complex variable is a conformal mapping. One class of analytical functions is the polynomials. This means that a Taylor series developing a complex variable always results in a conformal mapping

$$z_t = \alpha_0 + \alpha_1 \cdot z_s + \alpha_2 \cdot z_s^2 + \alpha_3 \cdot z_s^3 + \cdots + \alpha_n \cdot z_s^n \,. \tag{2.54}$$

We express the coordinates of our control points $x_s$, $y_s$, $x_t$, and $y_t$ as complex numbers

$$\begin{aligned} z'_s &= x'_s + y'_s i \,, \\ z_t &= x_t + y_t i \,. \end{aligned} \tag{2.55}$$

Then, we develop $z_t$ as a polynomial function of $z'_s$ of degree one, i.e.,

$$z_t = \alpha_0 + \alpha_1 \cdot z_s \,. \tag{2.56}$$

If we expand expression (2.54) then we get

$$\begin{aligned} x_t + y_t i = {} & \mathrm{Re}(\alpha_0) + \mathrm{Im}(\alpha_0) i \\ & + [\mathrm{Re}(\alpha_1) + \mathrm{Im}(\alpha_1) i] \cdot \left(x'_s + y'_s i\right) \\ = {} & \mathrm{Re}(\alpha_0) + \mathrm{Im}(\alpha_0) i \\ & + \mathrm{Re}(\alpha_1) \cdot x'_s - \mathrm{Im}(\alpha_1) \cdot y'_s \\ & + \left[\mathrm{Im}(\alpha_1) \cdot x'_s + \mathrm{Re}(\alpha_1) \cdot y'_s\right] i \,. \end{aligned} \tag{2.57}$$

After separation of the real and imaginary parts we get the equations for the coordinates $x_t$ and $y_t$ as

$$\begin{aligned} x_t &= \mathrm{Re}(\alpha_0) + \mathrm{Re}(\alpha_1) \cdot x'_s - \mathrm{Im}(\alpha_1) \cdot y' \,, \\ y_t &= \mathrm{Im}(\alpha_0) + \mathrm{Im}(\alpha_1) \cdot x'_s + \mathrm{Re}(\alpha_1) \cdot y' \,. \end{aligned} \tag{2.58}$$

If we compare (2.58) with (2.8), we see that the four-parameter transformation is just a special case of complex transformation polynomials, namely the case of degree one. The components of $\alpha_0$ are identical to the components of the translation vector $\boldsymbol{t}$, whereas the components of $\alpha_1$ are identical to $a$ and $b$.

In the four-parameter approach, the rotation and scale factor are constant over the whole area under consideration. However, if we transform large areas, e.g., the size of a country, this assumption is not valid anymore. In that case, the rotation angle and scale factor are functions of the point coordinates. To be able to model this property, it is necessary to extend the complex transformation polynomial to elements of higher order.

The necessary degree of development depends on several factors, especially on the area size. The calculated elements of the polynomial can be tested for significance by using Student's $t$-test [2.4].

The adjustment approach is presented here for the example of a polynomial of degree two. The extension to higher degrees works analogously. The residual equations have the structure

$$\begin{aligned} x_{ti} + v_{xi} = {} & \mathrm{Re}(\alpha_0) + \mathrm{Re}(\alpha_1) \cdot x'_s \\ & - \mathrm{Im}(\alpha_1) \cdot y' + \mathrm{Re}(\alpha_2) \cdot \mathrm{Re}\left({z'_s}^2\right) \\ & - \mathrm{Im}(\alpha_2) \cdot \mathrm{Im}\left({z'_s}^2\right) + \cdots \,, \\ y_t + v_{yi} = {} & \mathrm{Im}(\alpha_0) + \mathrm{Im}(\alpha_1) \cdot x'_s \\ & + \mathrm{Re}(\alpha_1) \cdot y' + \mathrm{Im}(\alpha_2) \cdot \mathrm{Re}\left({z'_s}^2\right) \\ & + \mathrm{Re}\big(\alpha_2\big) \cdot \mathrm{Im}({z'_s}^2) + \cdots \end{aligned} \tag{2.59}$$

As we can see, the equations are linear. The coefficients are the real and imaginary parts of the powers of $z'_s$. The unknowns are the real and imaginary parts of the values $\alpha_i$. The unknowns can be calculated by a linear Gauss–Markov adjustment calculation.

### 2.2.7 Modeling of Correlations

A special problem in the integration of spatial data is distance-dependent correlations among coordinates within one dataset. This subsection discusses the reason for these correlations and shows how they can be modeled in an adjustment approach. It is shown that techniques such as proximity fitting, rubber sheeting, and constraint management can be traced back to the same adjustment problem.

### 2.2.8 Reasons for Correlations

In the pre-GNSS era, measurements were done following the principle of neighborhoods. Trigonometric networks were calculated from large to small. Manual mapping was done in the same manner. These circumstances led to the fact that coordinates originating from digitized paper maps or classical terrestrial surveys show distance-dependent correlations. In other words, one can say that the relative accuracy of two neighboring points is higher then their absolute accuracy relative to the reference frame. This fact can be expressed in the following equation by the covariance of two coordinate values

$$\begin{aligned} &\Delta x = x_\mathrm{B} - x_\mathrm{A}\,, \\ &\sigma_{\Delta x} = \sqrt{\sigma^2_{\Delta x_\mathrm{A}} + \sigma^2_{\Delta x_\mathrm{B}} - 2 \cdot \operatorname{cov}(x_\mathrm{A}, x_\mathrm{B})} \\ &\quad \text{with} \quad \operatorname{cov}(x_\mathrm{A}, x_\mathrm{B}) \neq 0\,. \end{aligned} \tag{2.60}$$

The quotient of the covariance $\operatorname{cov}(x_\mathrm{A}, x_\mathrm{E})$ and the standard deviations of the coordinates $\sigma_\mathrm{A}$ and $\sigma_\mathrm{E}$ represents the correlation coefficient of the coordinate values $x_\mathrm{A}$ and $x_\mathrm{E}$, i.e.,

$$\rho_{x_\mathrm{A}, x_\mathrm{B}} = \frac{\operatorname{cov}(x_\mathrm{A}, x_\mathrm{B})}{\sigma_\mathrm{A} \cdot \sigma_\mathrm{B}}\,. \tag{2.61}$$

The correlation coefficient always has a value between $-1$ and 1. In the case of the considered coordinates, $\rho_{x_\mathrm{A}, x_\mathrm{B}}$ is a function of the distance between point A and B with a codomain between 1 and 0 (Fig. 2.13).

As we can see, the correlation grows as the distance gets smaller, and it converges towards zero as the distance gets longer.

Many of the problems regarding integration of different GIS datasets are caused by these distance-dependent correlations. If they did not exist, all discrepancies accruing in the control points after a datum transformation could be considered as random errors. It would be enough to average them, whereas all new points could keep their transformed coordinates.

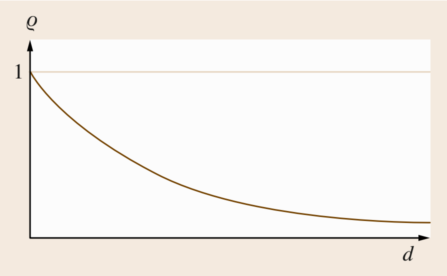

**Fig. 2.13** Correlation coefficient as a function of distance

However, because of the existence of distance-dependent correlations, it is necessary to account for them during the integration process. A simple transformation approach is not adequate in many cases. The mapping model has to be extended, otherwise local geometrical relationships of neighboring points would be violated.

Several approaches exist to model these correlations. Some of them are presented and discussed in the following.

Because the genesis of coordinates can almost never be completely reconstructed, all modeling approaches are based on hypotheses. For this reason, there is never only one possible approach. It has to be decided for the individual case which model is appropriate.

### 2.2.9 Rubber Sheeting

Several GIS provide rubber-sheeting tools for conflation of different vector datasets. The basic idea of most of these tools is a six-parameter affine transformation applied on single triangles of a triangulated irregular network (TIN) (Chap. 9). The triangles are the result of Delaunay triangulation over the control points in the start system. The six-parameter approach provides, for three control points, a unique solution. For this reason, no discrepancies occur in the control points. The result is an individual set of transformation parameters for each triangle in the network. All new points lying in a particular triangle are subsequently transformed using the parameters belonging to that triangle (Fig. 2.14).

In many cases this model is adequate, but it also has disadvantages. The coordinates of the target system are considered to be correct. So, it is not possible to con-

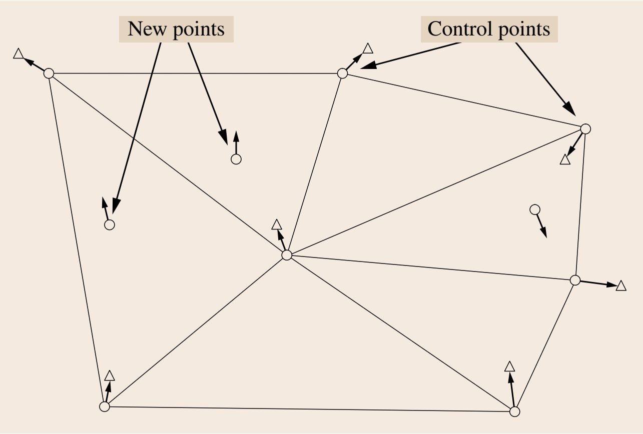

**Fig. 2.14** Principle of rubber sheeting

flate two datasets where the coordinates in the start and the target system are of limited accuracy. Because of the unambiguousness of the solution, it is not possible to detect blunders. Blunders are directly propagated to the new point coordinates. Also, geometrical constraints such as collinearities, rectangularities or parallelisms are not retained.

### 2.2.10 Stochastic Modeling

A further option to model distance-dependent correlations is their direct calculation with a hypothetic function $\rho(d)$. Examples for such a function are

$$\rho(d)=\frac{1}{1+d/d_0} \quad \text{or} \quad \rho(d)=\frac{1}{1+(d/d_0)^2} \,. \tag{2.62}$$

In these functions, $d$ is the distance and $d_0$ is a constant value. Clearly, $\rho(0)=1$ and $\rho(\infty)=0$ apply. The hypothetic function as well as the correlation and the standard deviations of the point coordinates lead to the covariances of the coordinates

$$\operatorname{cov}(x_\mathrm{A},x_\mathrm{B})=\sigma_{x_\mathrm{A}}\cdot\sigma_{x_\mathrm{B}}\cdot\rho_{\mathrm{A,B}} \,. \tag{2.63}$$

The covariances are collated in the covariance matrix $\mathbf{C}_{xx}$. This covariance matrix is introduced in an adjustment process for the calculation of the transformation parameters and is also used for the propagation of the residuals to the coordinates of the new points. During an adjustment calculation, $\mathbf{C}_{xx}$ has to be inverted. However, in practice the matrix $\mathbf{C}_{xx}$ can get very large, which leads to an enormous calculation effort. For this reason the stochastic modeling approach is more of a theoretical proposal than a method applicable in practice.

### 2.2.11 Functional Modeling

Distance-dependent correlations originate from observations between neighboring points. In most cases, the original observations are not known, or their acquisition would require an indefensible effort. These problems can be solved by the introduction of artificial observations.

The introduction of these artificial observations is done after the datum transformation. The results of the datum transformation are the coordinates of all points in the target system as well as the coordinate residuals for the control points. Those residuals are caused by random errors and distance-dependent correlations. To be able to model the correlations, the transformation approach is now extended by a further step: proximity fitting. For this purpose, artificial observations are introduced between points of the start system.

An easy approach is distance-dependent distribution of coordinate residuals. For each new point, a weighted average of coordinate residuals is calculated and applied to its coordinates. The particular weights for the average calculation are functions of the distance between the new point and the neighboring control points, i.e.,

$$p_i=\frac{d_0}{d_i} \quad \text{or} \quad p_i=\left(\frac{d_0}{d_i}\right)^2 \quad \text{or} \quad p_i=\left(\frac{d_0}{d_i}\right)^{3/2} \,. \tag{2.64}$$

The result is identical to that of an adjustment calculation with coordinate difference observations between the new and the control points. It is advantageous that the adjustment for each new point can be calculated

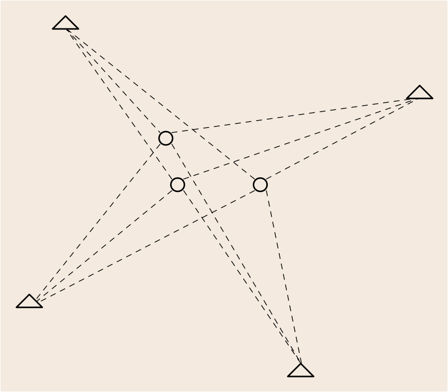

**Fig. 2.15** Observation topology of distance-dependent averaging

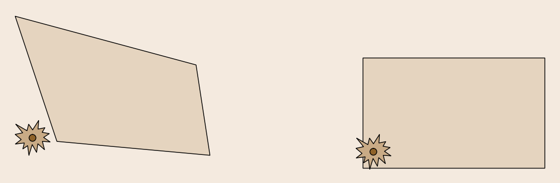

**Fig. 2.16** Before and after the application of rectangularity constraints

separately, which leads to very small calculation effort (Fig. 2.15).

A disadvantage of this approach is the absence of a direct link between the topology of observations and the real neighborhood relationship of points. The method fails if additional observations are introduced into the adjustment model. Such configurations may lead to significant violations of local geometry relationships.

Figure 2.16 shows the situation before and after the introduction of rectangularity observations in the adjustment. The rectangularity observations result in a change of the building geometry, while the tree in the lower-left corner keeps its position. The reason for this behavior is the nonmodeled neighborhood between the building corner and the tree. As a result the tree is situated inside the house.

A better way of creating an observation topology is Delaunay triangulation of all points (control and new points) of the considered dataset. The triangle edges are the carriers of neighborhood information. Along the triangle edges, artificial coordinate distance observations are generated (Fig. 2.17).

In this approach the neighborhood relationships are modeled directly. Further observations such as geometrical constraints can be introduced into the adjustment without problems. It is disadvantageous that all point coordinates have to be introduced as unknowns in the same adjustment, which can lead to a very large normal equation matrix. Therefore the solution of sophisticated geometrical data integration problems requires special software.

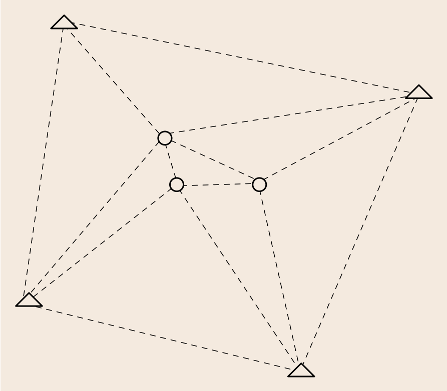

**Fig. 2.17** Topology of observations (edges) of a Delaunay triangulation

## 2.2.12 Modeling of Point Identities

There are two methods to model point identities, in which the identity information is expressed either topologically or geometrically. The most common method is topological modeling of identities by introducing the same point identifier to corresponding points in different datasets. However, this method can lead to problems because of inevitable point confusions. Confusions only indirectly affect the residuals of corresponding observations. To eliminate such confusion, it is necessary to *disfuse* points by the generation of new point objects, whereby referential integrity requirements have to be observed.

Geometrical modeling of point identities is an alternative approach. An identity observation is introduced. This means that a coordinate difference with value zero is observed between two potentially identical points instead of assigning a common identifier. The square sum of the residuals of such an identity observation is $\chi^2$ distributed and can be tested for significance. Misidentifications can easily be detected and then eliminated.

Figure 2.18 shows the principle of a point identity observation. It is a relative measurement between two points with different point numbers. The point identity is weighted with a standard deviation derived from connected observations (e.g., map accuracy). It reacts like a *rubber band* (see the left part of Fig. 2.18) with elasticity corresponding to its weight and can be analyzed like all other observations. Unreliable measurements can easily be removed without violating the topology.

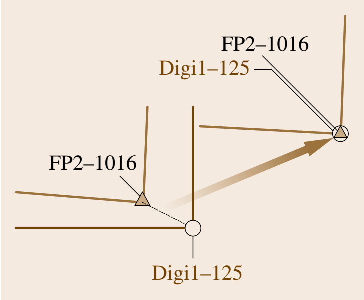

**Fig. 2.18** Handling of point identities

If all remaining point identities are reliable, their standard deviation is fixed (set to zero) and the connected points get the same coordinates (see the right part of the figure). Finally, the points can be fused by a GIS to obtain a topology without redundancies.

### 2.2.13 Geometrical Constraints, Known Relative Geometry

Sometimes, conflation of the coordinates of different datasets is not the only task. Additionally, geometrical constraints may have to be retained or known values such as distances, aligning bases, and so on be considered.

Typical geometrical constraints are the rectangularity of buildings, the collinearity of boundary points, or the parallelism of waysides.

All these constraints can be expressed by either a scalar product or a cross-product of two vectors. The vector components in this case are the coordinate differences of the points involved in the target system, i.e.,

$$\boldsymbol{v}_{\mathrm{AB}} = \begin{pmatrix} \Delta x_{\mathrm{AB}} \\ \Delta y_{\mathrm{AB}} \end{pmatrix} = \begin{pmatrix} x_{\mathrm{B}} - x_{\mathrm{A}} \\ y_{\mathrm{B}} - y_{\mathrm{A}} \end{pmatrix} , \tag{2.65}$$

where rectangularity is expressed as

$$\boldsymbol{v}_{\mathrm{AB}} \cdot \boldsymbol{v}_{\mathrm{BC}} \overset{!}{=} 0 , \tag{2.66}$$

collinearity as

$$|\boldsymbol{v}_{\mathrm{AB}} \times \boldsymbol{v}_{\mathrm{BC}}| \overset{!}{=} 0 , \tag{2.67}$$

and parallelism as

$$|\boldsymbol{v}_{\mathrm{AB}} \times \boldsymbol{v}_{\mathrm{CD}}| \overset{!}{=} 0 . \tag{2.68}$$

In theory, there are two options to model these constraints in the adjustment approach. The first option is the formulation of restriction equations of unknowns. These equations force strict compliance with the constraints. However, this method has two disadvantages. Firstly, the normal equation matrix grows with each constraint by one row or column, which leads to enormous inflation of the calculation. Secondly, it is not possible to check particular constraints for plausibility. Constraints are often detected by automated snooping algorithms that are not completely error free. Therefore, it is important to detect erroneously identified constraints.

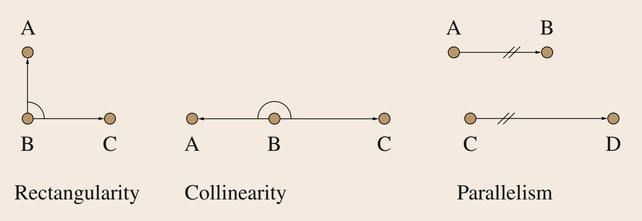

**Fig. 2.19** Geometrical constraints

A better approach is the formulation of constraints as observations. The observation is in all cases zero. As an example, the residual equation for a rectangularity constraint is given by

$$0 + v = (x_{\mathrm{A}} - x_{\mathrm{B}})(x_{\mathrm{C}} - x_{\mathrm{B}}) + (y_{\mathrm{A}} - y_{\mathrm{B}})(y_{\mathrm{C}} - y_{\mathrm{B}}) . \tag{2.69}$$

This approach does not influence the dimension of the normal equation matrix. On the other hand, the normalized residual of each observation can be used as a test value for an outlier test.

The weight for the constraint observation is calculated by variance propagation from the coordinates to the observation value.

Analogously to the constraint observations, known measurements such as lengths and widths of buildings, widths of streets, distances between objects, and so on can be introduced into the adjustment calculation.

### 2.2.14 Matching and Constraint Snooping

A prerequisite for application of adjustment techniques for spatial data integration is knowledge about identical objects in the relevant datasets. This subsection describes how identical objects in different datasets can automatically be detected. It will be shown how identity information can be modeled by identity observations. The basics of statistical test theory will be given. Furthermore, it will be shown that matching and adjustment interact in an alternating iterative process.

### 2.2.15 Topology and Extraction of Subgraphs

Each vector dataset can be considered as a graph. According to the general definition, a graph is a special case of a topology, containing a set of two-valued subsets called edges defined on a basic set of vertices [2.5] (Chap. 10). The graph itself contains no geometric information at all. Even if the particular vertices have geometrical properties (coordinates), the topological information of the graph is invariant to transformations

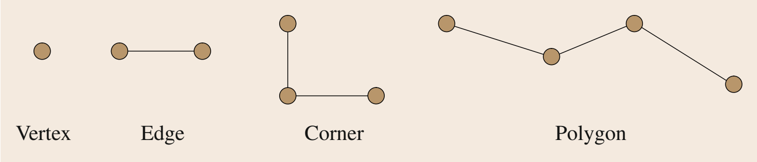

**Fig. 2.20** Example of subgraphs

of this geometry. Prerequisites for this behavior are a unique object identifier for each vertex and nonredundant storage of the geometry in the data model.

However, in most GIS, a vertex is identified by its coordinates and stored redundantly in the geometry of several lines or shapes. To be able to use graph operators it is necessary to reconstruct the topological information from such geometry.

Graph operators are helpful tools to find identical objects in different vector datasets. Each object in a vector dataset can be seen as a subgraph. A special case of a subgraph is a single vertex. Simple matching operators work on the basis of vertex identities. However, for complex vector datasets, the results of such operators are often ambiguous. To get unique results, it is necessary to use subgraphs of higher complexity. The higher the complexity of the subgraphs used, the lower the probability of ambiguities in the matching results. Figure 2.20 shows some examples of subgraphs.

With graph operators, subgraphs of the same type can be extracted from the two datasets to be matched. In a second step, their geometrical properties are compared and tested for identity. A very efficient approach for extraction of subgraphs with defined structure is the use of adjacency tensors. These tensors result from a sequence of tensor products of the graph's adjacency matrix with itself.

### 2.2.16 Geometrical Parameterization of Subgraphs

Identified subgraphs contain no geometrical information. However, to be able to test pairs of them for identity, it is necessary to quantify their geometrical properties. In general, one can distinguish between datum-dependent and datum-independent parameterization of their geometry.

First we consider the datum-dependent approach. A prerequisite for datum-dependent matching is that the vector datasets to be compared have to be in the same datum. Datum-dependent parameterization uses the coordinates of the vertices involved, or functions of them. The simplest approach is direct use of coordinates. Using a vertex subgraph, one pair of coordinate tuples has to be compared; at an edge, two tuples have to be compared; at a corner, three tuples; and so on.

However, often a coordinate comparison is not appropriate, as illustrated in Fig. 2.21. If we compare the corner of the building with the boundary corner we find that not all coordinate tuples of the involved vertices of the corners are identical. The same geometric fault is valid for the comparison of the building edge with the boundary edge.

In geometry, a linear point set can be given as a line segment, ray or line. The geometrical parameterization of these features is different. A line segment is simply parameterized by the coordinates of its end points

$$\boldsymbol{g}_{\text{line segment}} = \begin{pmatrix} x_\text{A} & y_\text{A} & x_\text{B} & y_\text{B} \end{pmatrix}^\text{T} , \tag{2.70}$$

where $\boldsymbol{g}_{\text{line segment}}$ is the vector of the geometrical parameters. A ray can be parameterized by the coordinates of its start point and a normalized direction vector as

$$\boldsymbol{g}_{\text{ray}} = \begin{pmatrix} x & y & d_x & d_y \end{pmatrix}^\text{T} . \tag{2.71}$$

Note that the covariance matrix of $\boldsymbol{g}_{\text{ray}}$ is singular because of the overparameterization of the direction. However, this overparameterization provides the advantage that the direction can be expressed without discontinuities. A line can be parameterized by its normal form $\boldsymbol{nx} - d = 0$, where $\boldsymbol{n}$ is the normal vector of the line, $\boldsymbol{x}$ is the position vector of a point of the line, and $d$ is the orthogonal distance between the line and the coordinate origin. Then, the parameter vector $\boldsymbol{g}_{\text{line}}$ contains the components of the normal vector and the

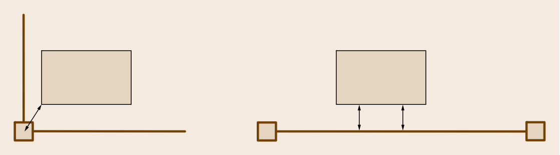

**Fig. 2.21** Matching of corners and edges

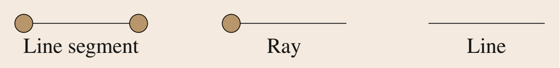

**Fig. 2.22** A linear point set

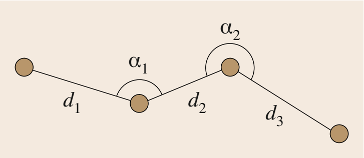

**Fig. 2.23** Datum-independent parameterization of a polygon

translation parameter $d$, i.e.,

$$g_{\text{line}} = \begin{pmatrix} n_x & n_y & d \end{pmatrix}^{\mathrm{T}} . \tag{2.72}$$

Analogously to the ray exposition, $g_{\text{line}}$ is overparameterized and its covariance matrix is singular.

Subgraphs of different types can be parameterized by a combination of line segment, ray, and line parameterization. Each corner in Fig. 2.23 would be parameterized by one point and two normalized direction vectors as

$$g_{\text{corner}} = \begin{pmatrix} x & y & d_{1x} & d_{1y} & d_{2x} & d_{2y} \end{pmatrix}^{\mathrm{T}} . \tag{2.73}$$

The search for control points that are necessary for a transformation leads to the identification of identical objects in different vector datasets which are not referenced to the same coordinate reference system. In such cases, the parameterization of the subgraphs must be done independent from the coordinate reference system. Let us take a closer look at the example of the polygon in Fig. 2.23. The inner geometry of this subgraph can be described by two angles and three distances.

Then, the parameter vector has the form

$$g_{\text{polygon}} = \begin{pmatrix} \alpha_1 & \alpha_2 & d_1 & d_2 & d_3 \end{pmatrix}^{\mathrm{T}} . \tag{2.74}$$

### 2.2.17 Search for Candidates

The aim of matching is to find subgraphs with identical geometrical parameter vectors in different vector datasets. However, if we think of corners, for instance, the number of extracted subgraphs can get very large.

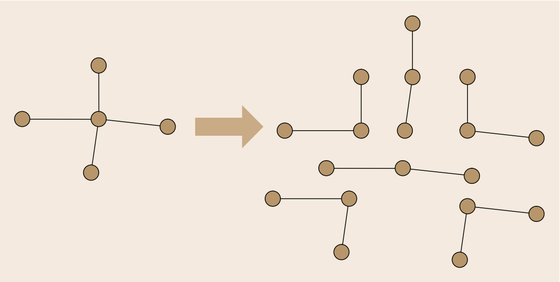

**Fig. 2.24** Extracted corners

Figure 2.24 demonstrates that a simple junction of four edges leads to six corners. The total number of corners in a dataset can easily reach several hundred thousand.

If each parameter vector of one dataset is compared with each parameter vector of the other dataset, the number of comparisons increases with the square of the number of parameter vectors, i.e.,

$$\text{comparisons} = n\frac{n-1}{2} .$$

So, 100 000 corners require about $5 \times 10^9$ comparisons. With 500 000 corners, the number would reach about $10^{11}$. Such a procedure results in unacceptably long computing time.

An appropriate approach is the application of $k$d-trees. In the case of a corner with six geometrical parameters, a 6d search tree is created. In this tree a search is performed in a $k$-dimensional window. In a well-balanced tree, the number of compare operations is the logarithmic function

$$\text{comparisons} \approx \log_d n .$$

Using a $k$d-tree, 100 000 corners require only seven comparisons, while the number reaches eight with about 500 000 corners (Chap. 3).

In any case a $k$d window search results in a set of matching candidates. Whether one or more of them are identical in the start parameter vector can only be decided by the application of a statistical test.

### 2.2.18 Statistical Tests

To be able to test the parameter vectors $g_1$ and $g_2$ for identity, it is necessary to know their stochastic properties. The elements of $g_1$ and $g_2$ are random values. The elements inside one of these vectors are algebraically correlated because they are functions of the same set of random values – the coordinates of the involved vertices. The stochastic properties of the parameter vector elements are quantified by their covariance matrices $\mathbf{C}_{gg1}$ and $\mathbf{C}_{gg2}$.

These covariance matrices can be calculated by variance propagation from the coordinates to the parameters. First, the parameters have to be expressed as linear functions of the coordinates, i.e.,

$$g = \mathbf{F}x . \tag{2.75}$$

Then, the covariance matrix of the parameters results as

$$\mathbf{C}_{gg} = \mathbf{F} \cdot \mathbf{C}_{xx} \cdot \mathbf{F}^{\mathrm{T}} . \tag{2.76}$$

The distance between the vertices involved in a subgraph (e.g., a corner) is mostly quite small. Therefore, it seems reasonable to consider the distance-dependent correlations between these vertices. In that case, the secondary diagonal elements of $\mathbf{C}_{xx}$ are not zero. The hypothetic covariances can be calculated from (2.62) and (2.63).

In general, a statistical test consists of the following steps.

1. Stating a null hypothesis.
2. Stating an alternative hypothesis.
3. Calculating a test value.
4. Calculating a critical value.
5. Making the test decision.

In our case, the null hypothesis is the identity in the comparison

$$H_0: \quad \boldsymbol{g}_1 = \boldsymbol{g}_2 . \tag{2.77}$$

The alternative hypothesis is the disparity of the parameter vectors, i.e.,

$$H_{\text{alt}}: \quad \boldsymbol{g}_1 \neq \boldsymbol{g}_2 . \tag{2.78}$$

The test value is always a random value with a known distribution function. From the null hypothesis, we can conclude that the difference vector of both parameter vectors is zero, i.e.,

$$\boldsymbol{d} = \boldsymbol{g}_1 - \boldsymbol{g}_2 = 0 . \tag{2.79}$$

The elements of the difference vector $\boldsymbol{d}$ are correlated random values. Its covariance matrix $\mathbf{C}_{dd}$ is calculated by variance propagation as

$$\mathbf{C}_{dd} = \mathbf{C}_{gg1} + \mathbf{C}_{gg2} . \tag{2.80}$$

With this covariance matrix, we are able to calculate a quadratic form with known $\chi^2$ distribution as

$$\chi^2 = \boldsymbol{d}^{\mathrm{T}} \mathbf{C}_{dd}^{-1} \boldsymbol{d} . \tag{2.81}$$

If the subgraph is overparameterized – for instance, the case with directions expressed as normalized vectors – then the covariance matrix of the parameters is singular. In that case, an infinite number of inverse matrices exist. This problem can be solved if the pseudoinverse $\mathbf{C}_{dd}^{+}$ of $\mathbf{C}_{dd}$ is used, which is exactly the inverse with minimal trace. For this purpose, $\mathbf{C}_{dd}$ might be rendered with its eigenvectors of eigenvalue zero. In the case of a normalized direction vector, the eigenvector is identical to the direction vector itself.

The value $\chi^2$ is $\chi^2$ distributed, whereby the number of degrees of freedom is identical to the rank of $\mathbf{C}_{dd}$. In the case of a corner parameterized with a coordinate tuple and two direction vectors, the number of parameters is six but the rank of $\mathbf{C}_{dd}$ is just four. If the same corner is parameterized by three coordinate tuples then $\mathbf{C}_{dd}$ has the full rank of six.

The critical value $\chi^2_{\mathrm{c}}$ is a function of the number of degrees of freedom and the significance level $\alpha$. Common $\alpha$ values are 1% and 5%. The comparison of $\chi^2$ with $\chi^2_{\mathrm{c}}$ leads to the test decision.

If $\chi^2 < \chi^2_{\mathrm{c}}$, then we accept the null hypothesis, therefore $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are identical.

If $\chi^2 > \chi^2_{\mathrm{c}}$, then we reject the null hypothesis in favor of the alternative hypothesis, therefore $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are not identical.

After testing all matching candidates, in principle, three results are possible.

1. No candidate was accepted as identical.
2. Exactly one candidate was accepted as identical.
3. More than one candidate was accepted as identical.

Solution 3 is ambiguous. In that case, all candidates should be rejected.

### 2.2.19 Search for Geometrical Constraints

Frequently, it is advisable to keep geometrical constraints during the adjustment calculation. It was shown in Sect. 2.2 how these constraints can be modeled by observations. However, before constraints can be introduced in an adjustment calculation, one has to know where they occur. In the same way that subgraphs of different datasets can be used to find related subgraphs, they can also be used within one dataset to find geometrical constraints. This is shown here with the example of rectangularity and collinearity constraints.

If a corner subgraph is parameterized by three coordinate tuples, then it can easily be tested whether the related vectors of coordinate differences are either rectangular or collinear. In the case of rectangularity, its scalar product has to be zero, i.e.,

$$\boldsymbol{v}_{\mathrm{AB}} \cdot \boldsymbol{v}_{\mathrm{BC}} \mathrel{\hat{=}} \mathrm{sp} \overset{!}{=} 0 . \tag{2.82}$$

The scalar product sp is a function of the related coordinates as

$$\begin{aligned} \mathrm{sp} &= f(\boldsymbol{g}) \\ &= (x_{\mathrm{A}} - x_{\mathrm{B}})(x_{\mathrm{C}} - x_{\mathrm{B}}) + (y_{\mathrm{A}} - y_{\mathrm{B}})(y_{\mathrm{C}} - y_{\mathrm{B}}) . \end{aligned} \tag{2.83}$$

It is a normally distributed random value whose standard deviation can be calculated by variance propa-

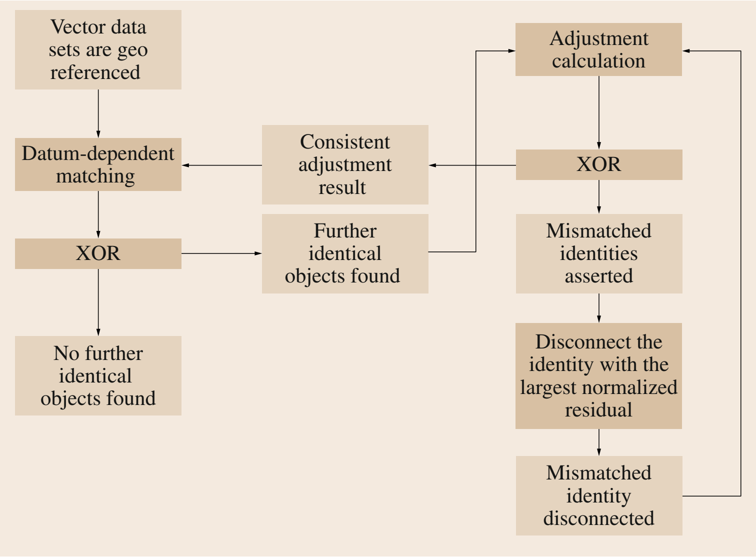
Vector data sets are geo referenced
Datum-dependent matching
XOR
No further identical objects found
Consistent adjustment result
Further identical objects found
Adjustment calculation
XOR
Mismatched identities asserted
Disconnect the identity with the largest normalized residual
Mismatched identity disconnected

**Fig. 2.25** Matching and adjustment in an event-driven process chain

gation as

$$\sigma_{\text{sp}}^2 = \mathbf{F} \cdot \mathbf{C}_{gg} \cdot \mathbf{F}^{\text{T}} \quad \text{with} \quad \mathbf{F} = \frac{\text{dsp}}{\text{d}\boldsymbol{g}} \,. \tag{2.84}$$

Note that the covariance matrix of the related coordinates $\mathbf{C}_{gg}$ has nonzero secondary diagonal elements if distance-dependent correlations were modeled. With sp and $\sigma_{\text{sp}}$, a test value $u$ can be calculated as

$$u = \frac{\text{sp}}{\sigma_{\text{sp}}} \,, \tag{2.85}$$

where $u$ is normalized and normally distributed ($u \sim$ N(0, 1)). The null hypothesis is that $u$ is equal to zero, i.e.,

$$H_0: \quad u = 0 \,. \tag{2.86}$$

The alternative hypothesis is that $u$ is not equal to zero, which means that the test is a two-sided problem, i.e.,

$$H_{\text{alt}}: \quad u \neq 0 \,. \tag{2.87}$$

The critical value $u_{\text{c}}$ is a function of the significance level $\alpha$. It is calculated as the inverse of the distribution function of the normalized normal distribution as

$$u_{\text{c}} = \pm \Phi^{-1}\left(\frac{\alpha}{2}\right) \,. \tag{2.88}$$

Common $\alpha$ values are 1% or 5%. Comparison of $u$ with $u_{\text{c}}$ leads to the test decision.

If $u < u_{\text{c}}$, then we accept the null hypothesis, therefore $u$ is zero, and the vectors are rectangular.

If $u > u_{\text{c}}$, then we reject the null hypothesis in favor of the alternative hypothesis, therefore $u$ is not zero, and the vectors are not rectangular.

In the case of collinearity, the length of the cross-product of the vectors has to be zero, i.e.,

$$|\boldsymbol{v}_{\text{AB}} \times \boldsymbol{v}_{\text{BC}}| \mathrel{\hat{=}} \text{cp} \overset{!}{=} 0 \,. \tag{2.89}$$

The length of the cross-product cp is a function of the related coordinates through

$$\begin{aligned} \text{cp} &= f(\boldsymbol{g}) \\ &= (x_{\text{A}} - x_{\text{B}})(y_{\text{C}} - y_{\text{B}}) - (y_{\text{A}} - y_{\text{B}})(x_{\text{C}} - x_{\text{B}}) \,. \end{aligned} \tag{2.90}$$

Like sp, also cp is normal distributed and can be tested for significance in the same way.

## 2.2.20 Interaction of Matching and Adjustment

Frequently, matching and adjustment are seen as completely independent processes. However, this view does not reflect reality. In fact, both processes interact. Datum-independent matching provides control points for the transformation. The transformation itself is an adjustment problem. After georeferencing, datum-dependent matching and a search for geometrical constraints can be accomplished. Results are additional identity constraint observations. With these observations, the observation vector of the adjustment can be extended, and a further adjustment calculation can be performed. The adjustment result gives information about mismatches. In an iterative process, false identity observations are removed. With the improved coordinates a new matching step is possible, and so on. Figure 2.25 shows the interaction of matching and adjustment as an event-driven process chain [2.6]. As one can see, the interaction of matching and adjustment is an alternating iterative process whereby the adjustment itself constitutes an inner iteration loop.

## 2.3 Geostatistics

In contrast to descriptive and inductive statistics where random variables are independent, geostatistics is based on location-dependent random variables. Typical examples can be found in geosciences – hence the name geostatistics – (the thickness of a coal seam, the salt concentration in a salt mine, the surface magnetization, etc.), the environmental sciences (the spread of viruses in groundwater, air pollution, forest degradation, epidemiological problems), but also in the technical sciences (the pressure distribution for sophisticated structures). This dependence on location and directional random fields is described by a variogram. Such empirical models are a basis for the estimation of any values. The best known of these is kriging estimation. Different kriging estimators can be chosen depending on the different bases of the data (dots, gradients). Geostatistics is used whenever the phenomena to be studied are so complex that they cannot be grasped by means of classical statistics, which would not lead to useful results. Geostatistics is an application method for the scientist or engineer. It is a method for knowlegeable geoscientific modelers rather than a pure mathematical approach. In addition, cross-validation, an iterative method for estimating the variogram, and simulations are treated in this section.

This section provides an introduction to geostatistics; for further information refer to the literature [2.7–10].

### 2.3.1 Example

The purpose and the general steps of geostatistical prediction will be explained using an example prior to presentation of the theory. Say that the unknown value $z$ at location $\boldsymbol{x}$ is needed. This can be a single point, but of course a run of a profile, a surface, or a body as well. This means that discrete points should be estimated at any position. The statistics provides simple means for making such estimates, for example, linear interpolation, moving average, and inverse distance weighting. However, the estimated variances of these methods are generally higher than those of the geostatistical method – kriging. Kriging computes the best linear unbiased estimator values $\hat{Z}(\boldsymbol{x}_0)$ of $Z(\boldsymbol{x}_0)$ based on a stochastic model of the spatial dependence.

The covariance function is needed to set up the kriging equation. This can be estimated based on the variogram (2.111) and (2.115–2.119). The kriging estimator is not only unbiased, but also interpolated exactly. The estimated point or area matches with the actual observed data. This behavior is not self-evident for interpolation techniques; for example, it is not valid for methods of least-squares adjustment. The kriging equation and kriging variance depend on the structure of the covariance matrix or the variogram only, and of course on the relative positioning of the various locations and their Euclidean distances $\boldsymbol{h}$, but not on the specific values of the data $z(\boldsymbol{x})$. The estimation variance can be affected by the location of the exploration points. Thus, the location and number of sampling points can be further optimized, thereby minimizing exploration costs.

The practice of geostatistical prediction will be described using an example of a salt deposit. The deposit is located in northern Germany and has an area of about $14\times18\,\mathrm{km}^2$. The deposit has been sampled by geological, geomechanical, and geophysical explorations. The data are available as 991 boreholes, 1010 horizontal gradients, 999 surface curvatures, 25 sampling stations for absolute gravity, and 13 sampling stations of Bouguer anomalies. Over the history of its geological exploration, this sample was assembled during various epochs and thus has an uneven distribution, including clusters of anomalies in some areas. Only the 991 boreholes are considered here, for reasons of simplicity and clarity.

If the second-moment stationarity or intrinsic hypothesis (see later) is assumed, the covariance function $\mathrm{cov}(\boldsymbol{h})$ or variogram $\gamma(\boldsymbol{h})$ can be estimated. These functions lead to the characteristic elements of sill, nugget effect, and range, and form the basic information that a geoscientist takes into account for the definition of a good model that tells him what is invisible below the Earth's surface. Mostly, more than one model assumption is plausible. However, which is the best model? Using cross-validation of kriging, the best model can be determined. This model function is the basis for the kriging equation system and may be computed for any location. The result is the kriging estimation variance $D_{\mathrm{K}}^2$ and the estimated value at the location $\hat{z}(\boldsymbol{x}_0)$.

This example will be continued later after the theory of spatial dependence (Sect. 2.3.5) and the estimation of values by kriging (Sect. 2.3.6).

## 2.3.2 Random Fields in Geostatistics

The estimation of stochastic moments and/or their distribution function describe the random field.

Stochastic moments or simply moments are values used to characterize the probability distribution of random variables. The $k$-th moment about the origin for a variable $x$ is defined as

$$m_k = \frac{1}{n}\sum_{i=1}^{n}\{Z(x) - E[Z(x)]\}^k .$$

A random field is defined as a family of random variables $Z(\boldsymbol{x})$, $x \in D$. The quantity $D$ is called the parameter quantity of $Z$. $D$ defines all time or spatial points. For the case $D = \mathbb{N}$ the field $Z$ is called discrete, and for $D = \mathbb{R}_+$ the field $Z$ is called continuous; for example, predictions of thicknesses and ore layers using a few specific samples or a few gradients are discrete fields, while predictions of physically describable surfaces (geoid, dust particles in the atmosphere, etc.) are continuous fields. The values $z_i = Z(\boldsymbol{x}_i)$ are realizations of the random field $Z$ at the positions $\boldsymbol{x}_i$. The estimation of stochastic moments and/or their distribution function describe the random field.

Moments are characteristics of random variables. They are parameters of descriptive statistics and play a theoretical role in stochastics. The expected value, variance, skewness, and curvature describe a random variable. A distribution function is determined by giving all its moments. The first moment is the expected value $E(Z(\boldsymbol{x}))$, and the second moment is the variance $\mathrm{var}(Z(\boldsymbol{x}))$.

In general, in nature there is only one realization of random variables. The description of random variables is limited by the number of samples. In linear geostatistics, the determination of the first two moments of a random field is sufficient, i.e.,

- First moment
  a) Expected value

$$E[Z(\boldsymbol{x})] = m(\boldsymbol{x}) . \tag{2.91}$$

- Second moment
  a) Variance function

$$\mathrm{var}[Z(\boldsymbol{x})] = E\{Z(\boldsymbol{x} - E[Z(\boldsymbol{x})])\}^2 . \tag{2.92}$$

  b) Autocovariance function

$$\begin{aligned}&\mathrm{cov}[Z(\boldsymbol{x}),\, Z(\boldsymbol{x}+\boldsymbol{h})]\\ &= E(\{Z(\boldsymbol{x}) - E[Z(\boldsymbol{x})]\}\\ &\quad \times\{Z(\boldsymbol{x}+\boldsymbol{h}) - E[Z(\boldsymbol{x}+\boldsymbol{h})]\}) .\end{aligned} \tag{2.93}$$

  c) Variogram

$$2\gamma[Z(\boldsymbol{x}),\, Z(\boldsymbol{x}+\boldsymbol{h})] = E[Z(\boldsymbol{x}) - Z(\boldsymbol{x}+\boldsymbol{h})]^2 . \tag{2.94}$$

The normalized autocovariance is called the autocorrelation function or correlogram

$$\rho(h) = \frac{\mathrm{cov}[Z(\boldsymbol{x}),\, Z(\boldsymbol{x}+\boldsymbol{h})]}{\sqrt{\mathrm{var}Z(\boldsymbol{x})\mathrm{var}Z(\boldsymbol{x}+\boldsymbol{h})}} . \tag{2.95}$$

The variable $\boldsymbol{h}$ is defined as the distance vector between the values $Z(\boldsymbol{x})$ and $Z(\boldsymbol{x}+\boldsymbol{h})$ observed in space or in time. A random field with normalized Gaussian distribution is characterized by $EZ(\boldsymbol{x}) = 0$ and $\mathrm{var}Z(\boldsymbol{x}) = 1$.

## 2.3.3 Terms of Stationarity

Using the first of the two moments described in Sect. 3.2.2, it is possible to formulate different hypotheses based on the precision of the stationarity.

### Stationarity One

A random field $Z$ is strictly stationary if all the $n$-dimensional distribution functions are invariant to translation

$$\begin{aligned}&P[Z(\boldsymbol{x}_1) < a_1,\, Z(\boldsymbol{x}_2) < a_2,\, \dots,\, Z(\boldsymbol{x}_n) < a_n]\\ &= P[Z(\boldsymbol{x}_1+\boldsymbol{h}) < a_1,\, Z(\boldsymbol{x}_2+\boldsymbol{h}) < a_2,\\ &\quad \dots,\, Z(\boldsymbol{x}_n+\boldsymbol{h}) < a_n] .\end{aligned} \tag{2.96}$$

The vectors then have the same distribution law for any translation vector $\boldsymbol{h}$. The expected value and the second moments are independent of $\boldsymbol{x}$. Linear geostatistics requires only the first two statistical moments, so it is sufficient to accept their existence.

### Second-Moment Stationarity

A random function is second-moment stationary (wide-sense stationary), if the expected value $E[Z(\boldsymbol{x})]$ is independent of the location $\boldsymbol{x}$

$$E[Z(\boldsymbol{x})] = m \tag{2.97}$$

and if there is a covariance for each pair of random variables

$$\operatorname{cov}[Z(\boldsymbol{x}),\, Z(\boldsymbol{x}+\boldsymbol{h})] = E[Z(\boldsymbol{x})Z(\boldsymbol{x}+\boldsymbol{h})] - m^2\,, \tag{2.98}$$

$$\operatorname{cov}[Z(\boldsymbol{x}),\, Z(\boldsymbol{x}+\boldsymbol{h})] = \operatorname{cov}[Z(\boldsymbol{h})] = \operatorname{cov}(\boldsymbol{h}) = \sigma(\boldsymbol{h})\,. \tag{2.99}$$

The translation invariance of the covariance means that the variance

$$\operatorname{var}[Z(\boldsymbol{x})] = D^2 Z = E\{Z(\boldsymbol{x}) - E[Z(\boldsymbol{x})]^2\} = \operatorname{cov}(0) \tag{2.100}$$

and the semivariogram

$$\begin{aligned}\gamma[Z(\boldsymbol{x}),\, Z(\boldsymbol{x}+\boldsymbol{h})] &= \frac{1}{2}E\{Z(\boldsymbol{x}) - E[Z(\boldsymbol{x}+\boldsymbol{h})]^2\}\\ &= \gamma(h)\,,\end{aligned} \tag{2.101}$$

$$\gamma(h) = \operatorname{cov}(0) - \operatorname{cov}(\boldsymbol{h})\,. \tag{2.102}$$

depend on $\boldsymbol{h}$ only. The covariance $\gamma$ of the correlogram is derived from (2.101) and (2.102) as

$$\rho(\boldsymbol{h}) = \frac{K(\boldsymbol{h})}{K(0)} = 1 - \frac{\gamma(\boldsymbol{h})}{K(0)}\,. \tag{2.103}$$

The covariance is computed as

$$\operatorname{var}[Z(\boldsymbol{h})] = K(0)\,. \tag{2.104}$$

In contrast to the variogram, this equation exists in most cases.

### Intrinsic Model

A random function is called intrinsic if the expected value is between $Z(\boldsymbol{x})$ and $Z(\boldsymbol{x}+\boldsymbol{h})$, where $\boldsymbol{h}$ is the Euclidian distance, and thus the trend is zero, so

$$E[Z(\boldsymbol{x}) - Z(\boldsymbol{x}+\boldsymbol{h})] = m(\boldsymbol{x}) = 0\,, \tag{2.105}$$

and holds for all distance vectors have a finite increment of variance independent of $\boldsymbol{x}$

$$\begin{aligned}\operatorname{var}[Z(\boldsymbol{x}) - Z(\boldsymbol{x}+\boldsymbol{h})] &= E[Z(\boldsymbol{x}) - Z(\boldsymbol{x}+\boldsymbol{h})]^2\\ &= 2\gamma(\boldsymbol{h})\,.\end{aligned} \tag{2.106}$$

Second-moment stationarity includes the intrinsic model. The reverse conclusion does not apply.

In linear geostatistics, only second-moment stationarity or the intrinsic model is assumed, which means that the covariance or variogram function must be known. The term “isotropy” means that, in addition to the required translational invariance (stationary one), the covariances of the random field $Z$ are independent of a rotation of the coordinate system, i.e.,

$$\operatorname{cov}[Z(\boldsymbol{x}),\, Z(\boldsymbol{x}+\boldsymbol{h})] = \operatorname{cov}|\boldsymbol{h}|\,. \tag{2.107}$$

If a directional dependence of the field exists, this is called anisotropy.

### Ergodicity

Stochastic moments describe the behavior of a random variable. If some quite general conditions exist such as continuity of the mean, the variance and the covariance are consistent estimates of $m(x)$, $\sigma^2$ and $\operatorname{cov}(x, y)$.

Ergodicity is a characteristic of dynamical systems which refers to the average behavior of the system. This system is described as a function which determines the temporal evolution of the system depending on its current situation. There are two interpretations. First the development can be determined over a long period of time and transmitted over time (time mean), or it can be considered as including all possible states andconverted to a mean (ensemble mean). The system has ergodicity when both means lead to the same result. Intuitively, this means that all possible states will be achieved during the development of the system. The state space is filled completely at the end of the process. This means in particular that the expected value depends on the initial state of such systems. When a single realization of the random field contains all information about the overall process, describing its stochastic properties in full, the process is called ergodic. Regarding the model of stationarity, it should be noted that an ergodic process is also a strictly stationary process. Ergodic is a synonym for strictly stationary [2.7]. When statements about the process characteristics based on only one realization can be made, ergodicity can be assumed.

## 2.3.4 Structure of Random Fields

The description of the structure of random fields was introduced by *Matheron*, who developed the concept of regionalized variables $Z(\boldsymbol{x})$. A regionalized variable is a synonym for a random field. The designed geostatistical methods are based on the assumption of regularities between the random variables $Z(\boldsymbol{x})$.

An approach to the description of a regionalized variable is given by the superimposition of a homogeneous random field $Z_S(\boldsymbol{x})$ with a trend $m(\boldsymbol{x})$ and a disturbance variable $\xi$ of *Menz* [2.11],

$$Z(\boldsymbol{x}) = Z_S(\boldsymbol{x}) + m(\boldsymbol{x}) + \xi\,. \tag{2.108}$$

A trend $m(\boldsymbol{x})$ is a regular part of the systematic variables that can be represented by mathematical functions. The disturbance variable $\xi$ is a random variable that reflects the microvariability of the variable and the measurement error that always exists. The idea of (2.108) is based on the assumption of the existence of several realizations of a random process, because all realizations at a specific location $\boldsymbol{x}_i$ of the random value $Z(\boldsymbol{x}_I)$ can be split into a deterministic and a stochastic part. The deterministic part describes, under the assumption

$$E[Z_s(\boldsymbol{x})+\xi]=0\,, \tag{2.109}$$

the expected value of the random process at point $\boldsymbol{x}_i$. The stochastic part from a global perspective can be decomposed into correlated and uncorrelated components.

## 2.3.5 Spatial Dependence

Assuming second-moment stationarity, a semivariogram, a covariance function, and a correlogram are equivalent descriptions of the spatial dependence of the random variables $Z(\boldsymbol{x})$. In geostatistics, it is common practice to estimate the characterization of regionalized variables using a semivariogram (Sect. 2.3.4). The semivariogram $\gamma(h)$ describes the average variance of a pair of points $(Z(\boldsymbol{x}), Z(\boldsymbol{x}+\boldsymbol{h}))$ in space. A prerequisite for estimation of the semivariogram

$$\gamma(h)=\frac{1}{2}E[Z(\boldsymbol{x})-Z(\boldsymbol{x}+\boldsymbol{h})]^2 \tag{2.110}$$

is the assumption that all values $z_i = Z(\boldsymbol{x}_i)$ are realizations of a random field at the location $\boldsymbol{x}_i$. For the random variable $Z$, the intrinsic model is required.

The experimental semivariogram

$$\begin{aligned}\gamma(h)=\frac{1}{2(N-1)}\Big\{&\sum[Z(\boldsymbol{x})-Z(\boldsymbol{x}+\boldsymbol{h})]^2\\ &-\frac{1}{N}\sum[Z(\boldsymbol{x})-Z(\boldsymbol{x}+\boldsymbol{h})]^2\Big\}\end{aligned} \tag{2.111}$$

results for all combinations of pairs of values $(Z(\boldsymbol{x}), Z(\boldsymbol{x}+\boldsymbol{h}))$, depending on the distance vector $\boldsymbol{h}$. $N$ is the number of pairs found for $\boldsymbol{h}=$ constant. In irregular sampling nets, the number of pairs is low or one. For reasons of statistical certainty, the number of pairs should be large enough. Therefore, distance and direction classes are made. The distance vector $\boldsymbol{h}$ is determined by its absolute value $|h|$ and its direction angle $t$. An equivalent representation is the decomposition of $\boldsymbol{h}$ into its components $\boldsymbol{h}_x$ and $\boldsymbol{h}_y$ in relation to a rectangular Cartesian coordinate system.

The summary of pair values into classes leads to smoothing of the semivariogram. Depending on the number and the spatial distribution of the sample values, the optimum of the ratio between the size of the class and the number of value pairs is sought. An empirical solution is possible using the variogram cloud. Through the display of all numbers of pairs, the influences of single and extreme values on the semivariogram become apparent.

Let the field $Z$ with elements $h$ with small absolute value be a dataset of observations. If the field $Z$ contains a relatively high number of extreme values, an indicator function of the semivariogram (the indicator semivariogram) can provide an estimation that delivers a robust statement about the variability [2.12].

If a location-dependent variable $z(\boldsymbol{x})$ will be transformed such that

$$I(\boldsymbol{x}, Z_c)=\begin{cases}1: & Z(\boldsymbol{x})\geq Z_c\\ 0: & Z(\boldsymbol{x})<Z_c\end{cases} \tag{2.112}$$

then $I(\boldsymbol{x}, Z_c)$ is the indicator variable, the $\boldsymbol{x}$ and the limit depends on $Z_c$. The limit value $Z_c$ should start from the median of $Z(x)$ and should change over the period of the calculation.

The cross-variogram is an expansion of the semivariogram for multivariate statistics. Several statistical variables or random variables are investigated simultaneously in multivariate statistics. The consideration of two dependent variables in the geostatistical estimation method can be described using co-regionalized variables. The cross-variogram $\gamma_{kl}(h)$ for the pair of points $(z_k(\boldsymbol{x}_i), z_l(\boldsymbol{x}_i))$ is defined analogously to the semivariogram as

$$\begin{aligned}\gamma_{kl}(h)=\tfrac{1}{2}E\{&[Z_k(\boldsymbol{x})-Z_k(\boldsymbol{x}+\boldsymbol{h})]\\ &\times[Z_l(\boldsymbol{x})-Z_l(\boldsymbol{x}+\boldsymbol{h})]\}\,.\end{aligned} \tag{2.113}$$

Another definition of the cross-variogram provides the pseudo-cross-variogram of *Myers* [2.13]

$$\gamma_{kl}(h)^*=\tfrac{1}{2}E[Z_l(\boldsymbol{x})-Z_k(\boldsymbol{x}+\boldsymbol{h})]^2\,. \tag{2.114}$$

### Characteristics of the Experimental Variogram

The experimental variogram is approximated by a parametric model function, which must be positive definite [2.14]. This condition of positive definiteness of the functional model is required, because positive definiteness of the variance–covariance matrix ensures the applicability of geostatistical estimation methods. Characteristic functions are positive definite if they are concave, continuous, and symmetrical about the $y$-axis.

Characteristic features in a semivariogram are the behavior at the origin, the sill $C$, and the range $a$. It is of great interest to know the continuity and differentiability behavior at the origin. Discontinuities indicate uncorrelated random shares, and differentiability is a prerequisite for the application of some geostatistical estimation techniques. The sill $C$ is the variance of uncorrelated pairs $[Z(\boldsymbol{x}), Z(\boldsymbol{x}+\boldsymbol{h})]$ of values corresponding to half of the variance of uncorrelated values $Z(\boldsymbol{x})$ of the random field. The range of a semivariogram indicates up to what impact the distance has between two points as neighbors.

The most common models (Figs. 2.26, 2.27) are presented in the following.

Exponential model

$$\gamma(h)=\begin{cases} C_0 & \text{for} \quad h=0 \\ C_0+C\left(1-\mathrm{e}^{-\frac{1}{a}|h|}\right) & \text{for} \quad h\neq 0 \end{cases} . \tag{2.115}$$

Gaussian model

$$\gamma(h)=\begin{cases} C_0 & \text{for} \quad h=0 \\ C_0+C\left(1-\mathrm{e}^{-\left(\frac{1}{a}|h|\right)^2}\right) & \text{for} \quad h\neq 0 \end{cases} . \tag{2.116}$$

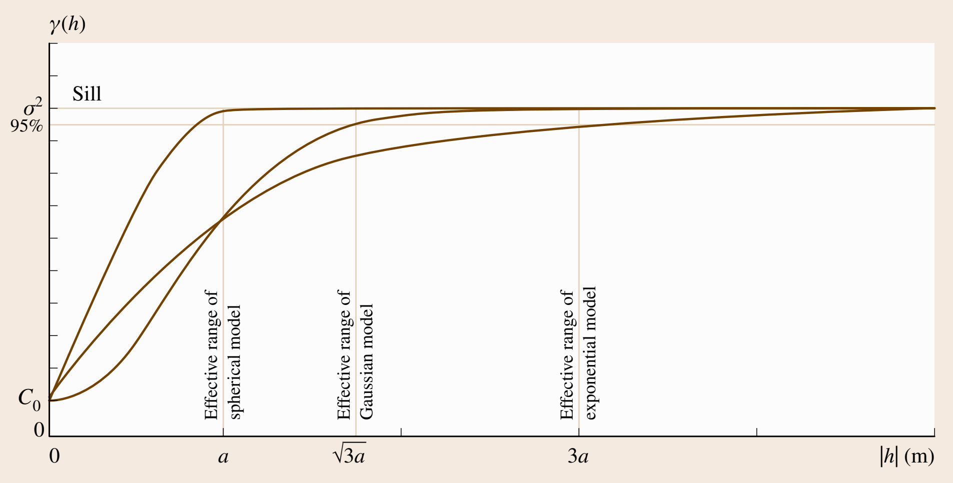

**Fig. 2.26** Representation of the spherical, Gaussian, and exponential variogram models

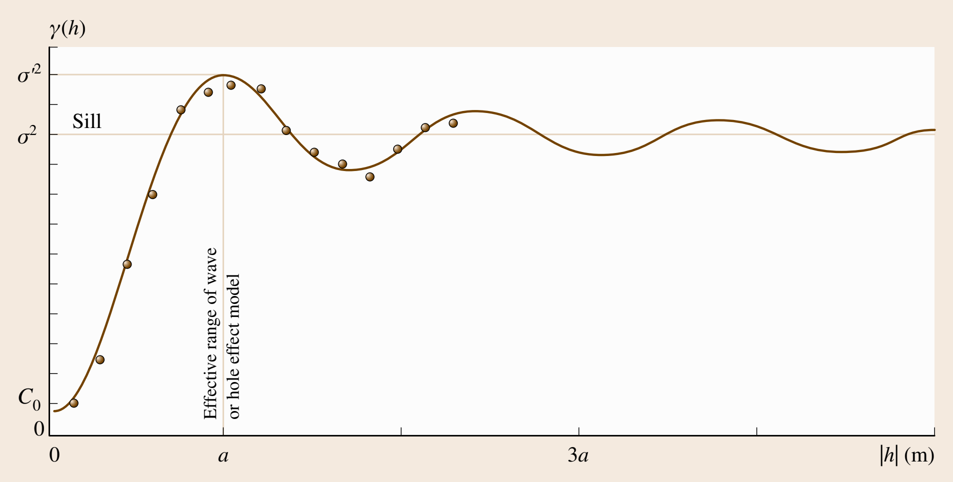

**Fig. 2.27** Experimental variogram with hole-effect and a periodic course

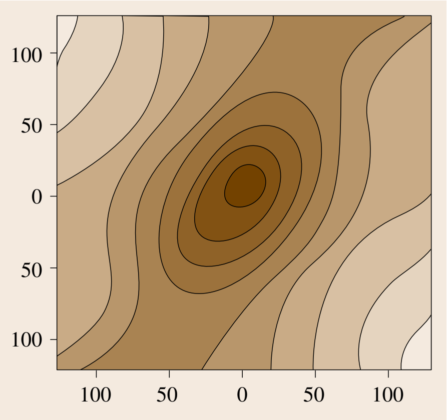

**Fig. 2.28** Two-dimensional correlation function of $Z_s(\boldsymbol{x})$

Spherical model

$$\gamma(h) = \begin{cases} C_0 & \text{for} \quad h = 0 \\ C_0 + C\left[\frac{3}{2a}(|h|) \right. \\ \quad \left. -\frac{1}{2}\left(\frac{|h|}{a}\right)^3\right] & \text{for} \quad 0 < |h| \leq a \\ C_0 + C & \text{for} \quad |h| = 0 \geq a \end{cases} . \tag{2.117}$$

Wave or hole-effect model

$$\gamma(h) = \begin{cases} C_0 & \text{for} \quad h = 0 \\ C_0 + C\left(1 - \frac{\sin\left(\frac{1}{a}|h|\right)}{a|h|}\right) & \text{for} \quad h \neq 0 \end{cases} . \tag{2.118}$$

If vibrations exist in the experimental semivariogram, such a model can be adapted with cosine vibrations

$$\gamma(h) = (2.115), (2.116), (2.117) \cos(\beta|h|)$$
$$\text{with} \quad \beta = \frac{2\pi}{\lambda} , \tag{2.119}$$

where $\lambda$ is the wavelength of the oscillation.

The values of sill and range are shown in Figs. 2.26 and 2.27. The graph of the spherical semivariogram approaches sill $C$ within a finite distance of points. The range of the experimental semivariogram is given for a distance of points, where the graph reaches the threshold or reaches 95% of the threshold.

The superposition of several independent functions for the formation of a model function is called a nested structure. The nugget effect $C_0$ is expressed in a semivariogram by a jump at the point $|h| = 0$. This is caused by the superposition of a random field and a location-independent disturbance $\xi$ (microvariability, measurement error of the analyzed parameter). The total variance $\sigma^2$ is the sum of a stationary portion $s_s^2$ and the nugget variance $s_\xi^2$, i.e.,

$$\sigma^2 = s_s^2 + s_\xi^2 . \tag{2.120}$$

If the random field is isotropic, the semivariogram is the same in all directions; otherwise, the random field is anisotropic.

If the threshold $C$ is constant for all directional semivariograms and the range is different (Fig. 2.28) then the random field is simply geometrical anisotropic. For further calculation the values can be transformed into polar coordinates by an ellipse equation

$$\gamma(h, \varphi) = \frac{a_{\min}^2}{1 - \left(\frac{\sqrt{a_{\max}^2 + a_{\min}^2}}{a_{\max}}\right)^2 \cos^2(\psi)} . \tag{2.121}$$

The random field is simply geometrically anisotropic. If the condition of constant threshold value $C$ fails, the semivariogram has zonal anisotropy. In (2.121), the variables $a_{\max}$ and $a_{\min}$ are defined as the semiaxes, and $\psi$ as the realignment angle of the ellipse (Fig. 2.29).

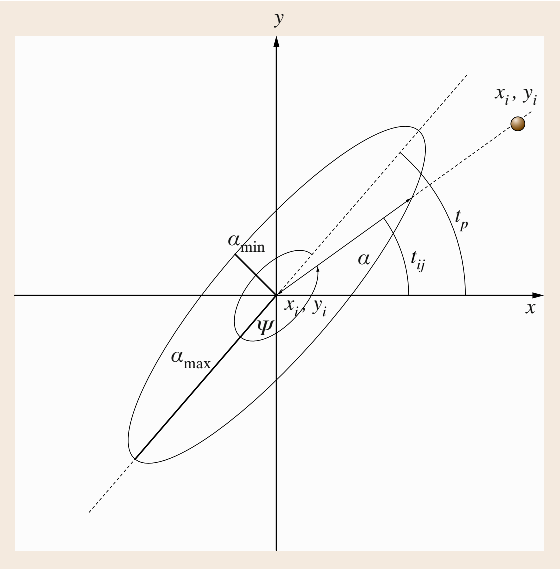

**Fig. 2.29** Elements to calculate the geometric anisotropy

### Back-Fitting Model

The idea of this alternative method of estimating the variogram or the covariance function was published by *Menz* [2.11]. In a realization $z(\boldsymbol{x})$ of a random field $Z(\boldsymbol{x})$, $N$ values are available, which have been measured at the points $\boldsymbol{x}_i$. The value $z_i$ is estimated by the values taken at the $N$ sampling locations in the environment, but without using the value $z_i$ itself. This approach is analogous to cross-validation of kriging. The prediction of a measurement location provides the estimated value $\hat{z}(x_i)$ and the associated kriging variance $D_\mathrm{K}^2(x_i)$. The difference between the estimates $\hat{z}(x_i)$ and the measured values $z(x_i)$ is called the residuum (2.126).

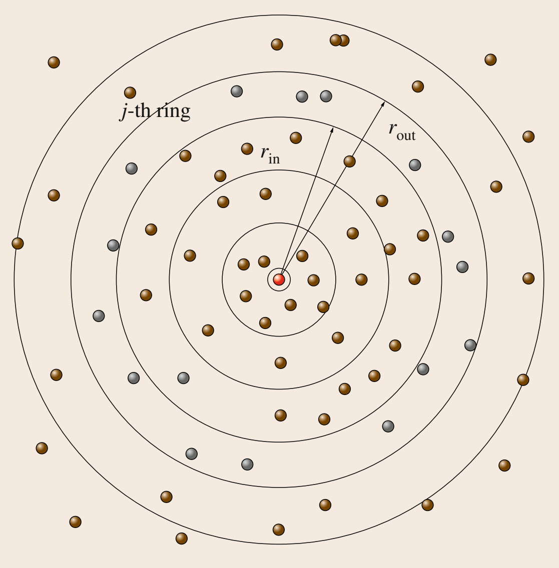

**Fig. 2.30** Back-fitting model, in which only the *gray dots* are used for the prediction

This allows the mean empirical error to be calculated as

$$m_\mathrm{empir} = \pm\sqrt{\frac{1}{N-1}\sum_{i=1}^{N} f^2(x_i)}\,, \tag{2.122}$$

which is to be compared with the mean theoretical error

$$m_\mathrm{theor} = \pm\sqrt{\frac{1}{N-1}\sum_{i=1}^{N} D_\mathrm{K}^2(x_i)}\,. \tag{2.123}$$

Both errors are influenced by the parameters of the covariance function. The idea of this method is convergence of these two errors by changing those parameters. If the two errors are almost identical, the influence of covariance is assumed.

The dependence of the errors on the distance should be considered by comparing (2.122) with (2.123). An exposure ring $s_j$ (Fig. 2.30) is used to determine the average errors for different distances $|h|$ in this method. Only the points within the exposure ring are used for the prediction. The mean empirical and theoretical errors refer to a specific width of the action of the ring $(r_\mathrm{in}, r_\mathrm{out})$. Both errors are plotted as a function of the average search radius

$$r = \tfrac{1}{2}(r_\mathrm{in} + r_\mathrm{out}) \tag{2.124}$$

in the so-called accuracy chart (Fig. 2.31).

The $L$ mean radii of the rings $s_j$ is determined for different distances to calculate both mean errors. A reference value for the dimension of the first medium-sized

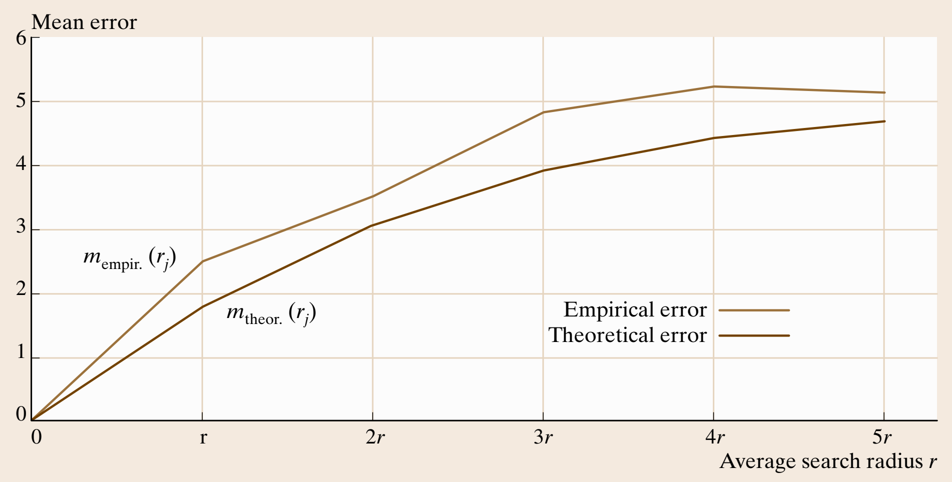

**Fig. 2.31** Accuracy graph representation of the two errors above the mean search radius

search radius $r_{j=1}$ is the mean point distance. The other mean radii are integer multiples of the first radius. Similar to the variogram ranges, a maximum search radius is used up to half of the maximum expansion of the field. The repetition of the forecast returns in a manner analogous to the mean error for each search radius $r_j$ with $j = 1, \dots, L$. The empirical and theoretical errors are plotted in every exposure ring above the mean search radius in the accuracy chart. By changing the parameters $\sigma^2_{xi}$, $\sigma^2$, and $a$, the best fit between the two curves is found.

Based on the calculation of a spatial correlation function, the parameter $c$ can be read off directly from the correlogram at the ordinate. The parameter $c$ is of the dimension of the stationary portion of total scattering. Prior knowledge about $c$ simplifies the iteration, so that the iteration is limited to $\sigma^2$ and $a$. The iteration ends with

$$m_{\text{empir}} - m_{\text{theor}} = \min \ . \tag{2.125}$$

The sensitivity property of a geostatistical estimation is helpful in this iterative process. This means that changes in the model parameters have a stronger influence on the kriging variance $D^2_{\text{K}}$ than the estimates $\hat{z}(x_i)$. The kriging variance also depends on the distance between the measurement positions and the predicted positions. Therefore, the mean theoretical error curve has been adapted to the mean empirical error curve.

The main differences between the back-fitting model and cross-validation of kriging are

- The back-fitting model is a real parameter estimation and not a model verification like cross-validation of kriging,
- The back-fitting model takes into account the $z$-values through the empirical error,
- The back-fitting model estimates $N \times L$ values while the cross-validation estimates only $N$ values.

However, graphs of the variogram and the accuracy graph are similar.

### Model Fit

As described in the previous subsection, the experimental semivariogram is approximated by a positive-definite model function. This model function can be visually adjusted, by the method of least squares, maximum-likelihood estimation, or cross-validation. Visual adaptation does not meet optimality criteria, although it is sufficient in many cases. In the method of least-squares models, the number of pairs per distance class can be considered as a weight function.

### Cross-Validation of Kriging

The method of cross-validation of kriging was published in [2.15, 16]. The method involves comparison of estimation methods, an estimation process, search strategies, a covariance function, and its parameters by reviewing the results of the prediction of well-known measurement points. The algorithm works as follows: Choose a model and its descriptive parameters for the underlying variogram. One measurement point has been ignored from the existing record. With the remaining $(N-1)$ points, the ignored point will be estimated again by kriging with the selected model. For each measuring point, the difference between the measured and the estimated values is calculated as

$$f(\boldsymbol{x}_i) = z(x_i) - \hat{z}(x_i) \quad i = 1, \dots, N \ . \tag{2.126}$$

For all $N$ measurement sites, $m_{\text{cv}}$ is determined by

$$m_{\text{cv}} = \frac{f(\boldsymbol{x}_i)}{\sqrt{D^2_{\text{K}}}} \ . \tag{2.127}$$

The arithmetic average and standard deviation for all values $m_{\text{cv}}$ are calculated. The calculation is then repeated for each model, or for each potential parameter combination.

The model that is chosen depends on several considerations: What degree of error is appropriate for the data? In which cases should a general approach be applied, and which subareas eventually need a specific model? Which methods are available to connect the measurement of the error with the estimator? There are only empirical proposals to answer these important questions. To establish the best model, the measure of discrepancy is calculated as

$$d_i = \frac{1}{D^2_{K_i}} \left[\hat{z}(x_i) - z(x_i)\right]^2 \ . \tag{2.128}$$

The degree of discrepancy should meet the conditions

$$\bar{d} = \frac{1}{N} \sum d_i \to 0 \quad \text{and} \quad \sigma^2 = \frac{1}{N} \sum d_i^2 \to 1 \ . \tag{2.129}$$

Cross-validation of kriging is a good method for objective assessment of estimation results, but problems can arise when

- Differentiating the points distribution in the following groups,
  - regular (grid),
  - uniformly distributed (same number of points in limited areas),

  – randomly (irregular pattern of location),
  – clustered (local accumulation),

  due to clustering issues. The ideal case of a well-proportioned representation of the study area by the measured data rarely exists. Clusters are overrepresented due to exploration of areas with high density of abnormalities (tectonics, areas of geologic faults). Therefore, cluster formations are the norm. Cluster effects in cross-validation of kriging can be avoided by decluster calculations or by allocating the same weight as a single sample instead of a cluster.

- A disadvantage of this procedure is the evaluation of the prediction of the estimation error $f_i$ only at the sampling location. Estimations at intermediate points cannot be classified. Thus, the results are not representative.
- The variance in cross-validation of kriging is higher than the prediction. This is explained by the fact that, in cross-validation, the distances between neighboring points used are always greater than for a prediction on a grid. If measurement errors for linearly arranged data (profile, borehole) are analyzed, differences from estimation arise. Thus, the results may claim a higher accuracy.

### Continuation of the Example – Spatial Dependence

Adequate description of location-dependent properties of the variables of a deposit using a variogram is certainly the most complex, but also the most important part, of geostatistical processing. The quality of an experimental semivariogram (2.111) is affected by many sources of errors that are often found in the data and in their structure, such as failure analysis, incorrect treatment of analytical values at the detection threshold, discordant values, incorrect choice of step size and tolerance in the variogram calculation, etc. To determine the experimental variogram, first, all single variogram values are plotted as a variogram cloud as a function of sample distances on the graph.

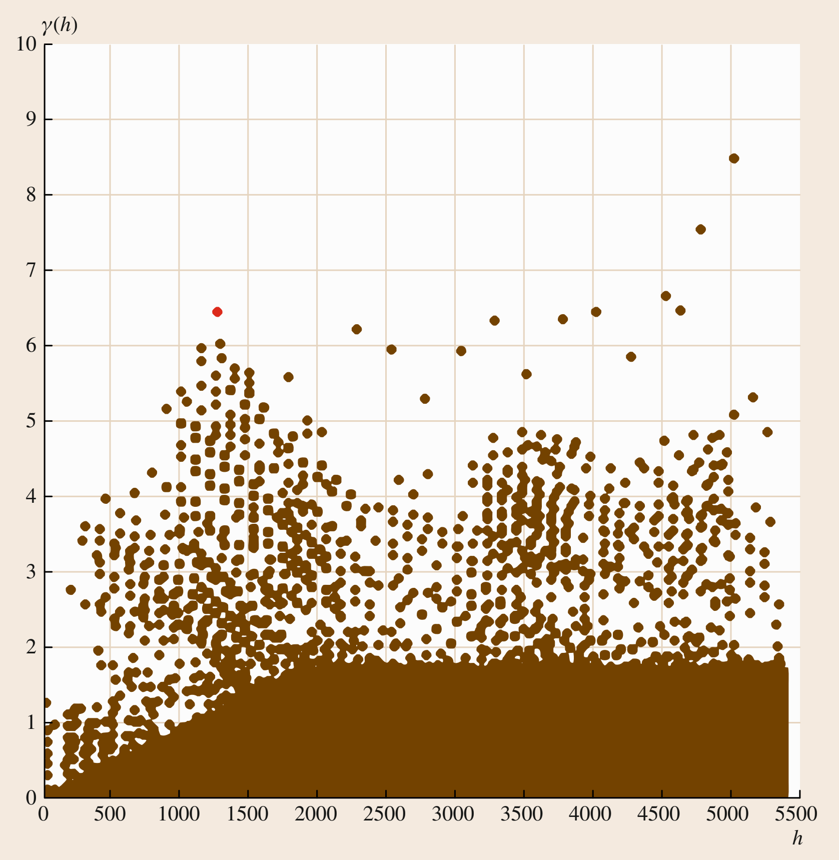

**Fig. 2.32** Variogram cloud for $h$

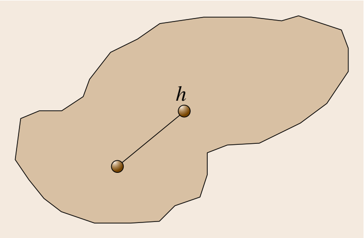

**Fig. 2.33** Sampling area ($18\,\mathrm{km}^2$), Distance ***h*** from *red point* of the variogram cloud in Fig. 2.32

In a computer-based version, the underlying pair of points is displayed on the map by clicking at a point in the variogram cloud. The points of this variogram cloud are classified to estimate the influence of individual sample values on experimental variogram values more precisely. Using the variogram cloud, it can easily be seen whether the distance classes with their step size and tolerance have been chosen correctly. Here, further analysis is needed to examine the data regarding compatibility with the hypothesis of stationarity, in order to exclude possibly nonstationary areas or to delineate several areas for each of which the hypothesis of stationarity can be assumed locally. Distance and direction classes are now selected for the experimental variogram. The first class corresponds to the distance from the midpoint spacing. The remaining distance classes are integral multiples of this.

To evaluate the isotropy or anisotropy of the variogram cloud, three profiles that intersect at the same point are calculated, oriented in the directions 0°, 45°, and 90°, respectively. Each profile represents all points within a sector of ±22.5° around each of those directions, forming a so-called direction class.

In the case of all profiles being approximately the same, the variogram cloud is called isotropic. In case of anisotropy, the ellipse of anisotropy of different ranges with their principal axes and the orientation can be computed from these three directions (Figs. 2.28, 2.29). An experimental variogram is now adapted by the variogram model (2.115–2.118). The weight is received by the individual points of the experimental variogram in the model fit. It is given by the number of pairs of values. The actual adjustment may use the method of least squares with a weighting given by the number of pairs of values. However, it is also possible to do this with an experienced eye. Experience in the application of geostatistical techniques shows that an acceptable variogram can be derived from location-dependent variables, especially if the sample values have relatively small variation. This variogram can be interpreted as geological–structural or generally genetic. Problems arise especially when the initial data vary strongly and contain discordant values. For detection and removal of such discordant values, statistical methods are available. First indications of discordant values can be found in the variogram cloud. Discordant values have a strong disruptive effect on the experimental variogram, because large squared differences correspond to small values in the variogram calculation. There are a number of publications dealing with robust variogram estimators. However, they also destroy the variability and sensitivity of the variogram model and the subsequent estimated values. A robust procedure is the indicator variogram (2.112), for example. Robust procedures are not used in the example.

The deposit with the 991 boreholes was analyzed for geometric and zonal anisotropy or isotropy. This was done as described above, with the variogram calculated in three different directions. The analysis shows the same estimated values for range, nugget effect, and variance in all directions. This means that the examination field has no geometric or zonal anisotropy. The knowledge of a neighboring deposit with the same geological formation confirms this assumption.

The experimental variogram is uniquely reflected by the isotropic spherical model (2.117) with parameter values $a = 2020.0\,\mathrm{m}$, $C_0 = 0.0$, and $C = 1.2$

$$\gamma(h) = \begin{cases} 0 & \text{for } h = 0 \\ 1.2\left[\frac{3}{4040}(|h|) \right. & \\ \left. \quad -\frac{1}{2}\left(\frac{|h|}{2020}\right)^3\right] & \text{for } 0 < |h| \leq 2020\,\mathrm{m} \\ 1.2 & \text{for } |h| = 0 \geq 2020\,\mathrm{m} \end{cases} .$$

This result is unique because no other type of model can be adjusted with the same precision to the experimental variogram. Therefore, investigation of alternative models to determine the best model function through cross-validation of kriging is not necessary.

The alternative analysis using the back-fitting model results in the the same estimated values for $a$, $C_0$, and $C$.

## 2.3.6 Kriging

Kriging is a summary of geostatistical methods to estimate from observations the value of a random field at an unobserved location. The mathematican Matheron developed the theory of interpolation and extrapolation by kriging with regionalized variables. The regionalized variable is the theoretical basis for the interpolation method by Krige.

The main advantage over simpler methods, e.g. inverse distance weighting, is the consideration of spatial variance. This can be determined with a semivariogram.

The weights of the used values are determined such that the estimation error variance is minimized. The error depends on the quality of the semivariogram.

With simpler interpolation techniques, problems can occur during the accumulation of data points. Kriging avoids this by considering the distances to neighboring points. The weighted averages are optimized so that the estimator determines the real value. Kriging is the best linear unbiased estimator.

Different kriging methods are applied depending on the stochastic properties of the random fields. Kriging is the estimation procedure with the smallest estimated variances. The weights of the kriging system are determined depending on the database. Known types of kriging are

- *Simple kriging*, which assumes a known constant trend,
- *Ordinary kriging*, which assumes an unknown constant trend,
- *Universal kriging*, which assumes a general linear trend model,
- *IRFk kriging*, which takes into account complex trends (intrinsic random function of order k),
- *Indicator kriging*, which uses indicator functions instead of the process itself, in order to estimate transition probabilities,
- *Cokriging*, which takes into account the relations with another additional parameter,
- *Gradients kriging*, which uses data points and gradients for the prediction in addition,
- *Bayesian kriging*, which combines actual data with a priori data (objective and subjective data).

Different kriging methods are described in the following paragraphs.

### Ordinary Kriging

The value $z(\boldsymbol{x}_0)$ is estimated at any point $\boldsymbol{x}_0$ in the field $D$ of observations $z(\boldsymbol{x}_i)$. In addition to the condition that the variable $z(\boldsymbol{x})$ is a realization of a stationary random field $Z(\boldsymbol{x})$, second-moment stationarity is required. The estimate is intended to provide the best linear unbiased values.

At the measurement point $\boldsymbol{x}_0$ there exists the unknown random variable $Z(\boldsymbol{x}_0)$. The random variable $\hat{Z}(x_0)$ is estimated by a linear combination of the random variables $Z(\boldsymbol{x}_i)$ as an approximation of the observation points $\hat{Z}(x_0)$ (linear estimation) by

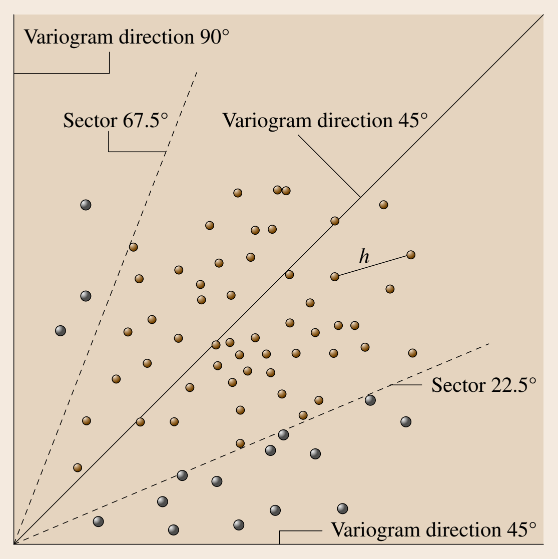

**Fig. 2.34** Direction classes 0°, 45°, and 90° and opening angle of ±22.5°

$$\hat{Z}(\boldsymbol{x}_0) = \hat{Z}_0 = \boldsymbol{w}^{\mathrm{T}}\mathbf{Z}\,. \tag{2.130}$$

$\mathbf{Z}$ is the vector of random variables $Z(\boldsymbol{x}_i)$, and $\boldsymbol{w}$ is the vector of weights. The weights $w$ are determined under the conditions of unbiasedness

$$E\big(Z_0 - \hat{Z}_0\big) = 0 \tag{2.131}$$

and under the condition that the expected squared error is minimized

$$E\big(Z_0 - \hat{Z}_0\big)^2 = \min\ . \tag{2.132}$$

In (2.131), $\hat{Z}_0$ will be replaced by (2.130)

$$\boldsymbol{w}^{\mathrm{T}} E\mathbf{Z} - EZ_0 = 0 \tag{2.133}$$

and based on the assumption of second-moment stationarity follows

$$w^{\mathrm{T}} 1m - m = 0\,. \tag{2.134}$$

The following condition for $\boldsymbol{w}$ can be derived from the condition for unbiased values (2.131) of $\hat{Z}_0$

$$a = w^{\mathrm{T}}\mathbf{1} - 1 = 0\,. \tag{2.135}$$

For the expected square error, (2.132) applies, and

$$\begin{aligned} E\big(\hat{Z}_0 - Z_0\big)^2 &= D_0^2 \\ &= \operatorname{var}\big(Z_0 - \hat{Z}_0\big) + \big[E\big(Z_0 - \hat{Z}_0\big)\big]^2\,. \end{aligned} \tag{2.136}$$

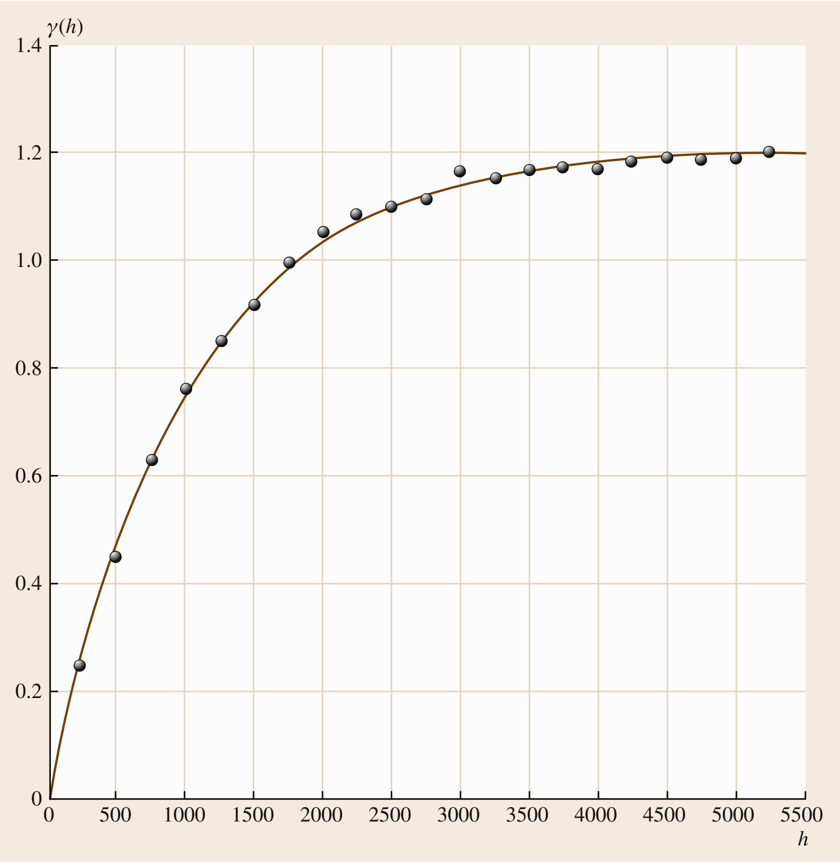

**Fig. 2.35** Variogram

The bias portion $\left[E(\hat{Z}_0 - Z_0)\right]^2$ for unbiased estimator disappears and becomes

$$\begin{aligned} E\left(\hat{Z}_0 - Z_0\right)^2 = D_0^2 &= \operatorname{var}\left(Z_0 - \hat{Z}_0\right) \\ &= \operatorname{var}(Z_0) + \operatorname{var}\left(\hat{Z}_0\right) - 2\operatorname{cov}\left(Z_0, \hat{Z}_0\right) \\ &= \sigma^2 + \boldsymbol{w}^{\mathrm{T}}\mathbf{K}\boldsymbol{w} - 2\boldsymbol{w}^{\mathrm{T}}\boldsymbol{c}_0 \end{aligned} \tag{2.137}$$

is applicable. The matrix $\mathbf{K}$ is called the covariance matrix. It contains all variances and covariances between the observation points $\boldsymbol{x}_i$. The variances and covariances are determined by the adapted covariance function, which only depends on the position of points with respect to each other. The vector $\boldsymbol{c}_0$ contains all covariances between pairs of observation $\boldsymbol{x}_0$ and the prediction location $\boldsymbol{x}_0$. The function of Lagrange [2.17] is formed and minimized from the variance (2.137) and the condition (2.135)

$$\begin{aligned} H &= D^{2,0} + 2\lambda a \\ &= \sigma^2 - 2\boldsymbol{w}^{\mathrm{T}}\boldsymbol{c}_0 + \boldsymbol{w}^{\mathrm{T}}\mathbf{K}\boldsymbol{w} + 2\lambda(\boldsymbol{w}^{\mathrm{T}}\mathbf{1} - 1)\,. \end{aligned} \tag{2.138}$$

This function $H$ has the value of the variance $D_0^2$ based on $a = 0$. The first derivatives of $H$ with respect to their $n+1$ unknowns ($\boldsymbol{w}$ and $\lambda$) are formed as

$$\frac{\delta H}{\delta w} = 0 \quad \text{and} \quad \frac{\delta H}{\delta \lambda} = 0\,. \tag{2.139}$$

Thus, minimization of (2.137) leads to the solution of the equation system

$$\mathbf{K} \cdot w + l = c_0\,, \tag{2.140}$$

$$1^{\mathrm{T}}\boldsymbol{w} = 1\,, \tag{2.141}$$

or, in matrix notation,

$$\begin{pmatrix} \mathbf{K} & \mathbf{1} \\ \mathbf{1}^{\mathrm{T}} & 0 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{w} \\ \lambda \end{pmatrix} = \begin{pmatrix} \boldsymbol{c}_0 \\ 1 \end{pmatrix}\,. \tag{2.142}$$

The variogram

$$\begin{pmatrix} \boldsymbol{\gamma} & \mathbf{1} \\ \mathbf{1}^{\mathrm{T}} & 0 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{w} \\ \lambda_\gamma \end{pmatrix} = \begin{pmatrix} g_0 \\ 1 \end{pmatrix} \quad \text{with} \quad \lambda_\gamma = -\lambda \tag{2.143}$$

or the correlation function

$$\begin{pmatrix} \boldsymbol{\rho} & \mathbf{1} \\ \mathbf{1}^{\mathrm{T}} & 0 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{w} \\ \lambda_\rho \end{pmatrix} = \begin{pmatrix} r_0 \\ 1 \end{pmatrix} \quad \text{with} \quad \lambda_\rho = \frac{\lambda}{\sigma^2} \tag{2.144}$$

can be used for determination of the kriging coefficients. One of the equation systems (2.142–2.144) is solved to calculate the estimated value $\hat{z}_0$ at the point $x_0$ and its kriging variance $D_{\mathrm{K}}^2$. Equation (2.130) provides the approach for determining the estimated value as

$$\hat{\tilde{z}}_0 = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{z} \,. \tag{2.145}$$

The variable $\boldsymbol{z}$ is the vector of observations with the measured values $z(x_i)$ for $i = 1, \ldots, n$.

In accordance with the determined weights $\boldsymbol{w}_i$ for $i = 1, \ldots, n$, $\mathbf{K}\boldsymbol{w}$ in (2.137) is replaced by (2.140) and we obtain the kriging variance as

$$D_{\mathrm{K}}^2 = \sigma^2 - \boldsymbol{w}^{\mathrm{T}} \boldsymbol{c}_0 - \lambda \,. \tag{2.146}$$

The derivation of other methods of kriging involves basically the same steps as ordinary kriging. For this reason, in the following we only describe the structure for the initial model for minimizing the function, the system of equations, the estimator, and the kriging variance.

### Universal Kriging

If a nonstationary random field with a position-dependent systematic size exists, the random field $Z(\boldsymbol{x})$ is described by a modified model approach. The random field (2.108) is composed of a stationary portion $Z_{\mathrm{s}}(\boldsymbol{x})$ and a trend $m(\boldsymbol{x})$, i.e.,

$$Z(\boldsymbol{x}) = Z_{\mathrm{s}}(\boldsymbol{x}) + m(\boldsymbol{x}) \,. \tag{2.147}$$

The expected value of the random field is dependent on the location

$$E[Z(\boldsymbol{x})] = m(\boldsymbol{x}) \,. \tag{2.148}$$

The goal of prediction, i. e., the estimation of the realization $z(x)$ of the random field at the location $x_0$, was explained in the previous section. The method of universal kriging can be derived for best linear unbiased prediction. The trend $m(\boldsymbol{x})$ is detected by a linear function with parameters $\vartheta_{\mathrm{K}}$ with $k = 1(1)p$ as

$$m(\boldsymbol{x}_i) = \boldsymbol{f}^{\mathrm{T}}{}_i \boldsymbol{\vartheta} = f_i^{(1)} \vartheta_1 + f_i^{(2)} \vartheta_2 + \ldots + f_i^{(p)} \vartheta_p \,. \tag{2.149}$$

The degree of a polynomial should be not greater than two when considering regionalized variables $Z(\boldsymbol{x})$ in the space $\mathbb{R}^2$. Two-dimensional random fields and a second-order polynomial ($p = 6$) require the following components

$$f_i^{(1)} = 1 \,, \quad f_i^{(2)} = x_i \,, \quad f_i^{(3)} = y_i \,, \quad f_i^{(4)} = x_i^2 \,, \\ f_i^{(5)} = y_i^2 \,, \quad f_i^{(6)} = x_i y_i \,. \tag{2.150}$$

The model approach (2.147) is reformulated as

$$Z(\boldsymbol{x}) = \mathbf{F} \cdot \boldsymbol{\vartheta} + Z_{\mathrm{s}}(\boldsymbol{x}) \,. \tag{2.151}$$

The $n \times p$-matrix ($n$ observations, $p$ components) is the so-called design or regressor matrix

$$\mathbf{F} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n)^{\mathrm{T}} \,. \tag{2.152}$$

The random variable $Z_0$ at location $x_0$ is given by

$$Z_0 = \boldsymbol{f}_0 \cdot \boldsymbol{\vartheta} + Z_{s_0} \,. \tag{2.153}$$

Under the unbiased condition, it follows from (2.130), (2.151), and (2.153) that

$$\boldsymbol{f}_0{}^{\mathrm{T}} \cdot \boldsymbol{\vartheta} - \boldsymbol{w}^{\mathrm{T}} \cdot \mathbf{F} \cdot \boldsymbol{\vartheta} = \boldsymbol{b}^{\mathrm{T}} \cdot \boldsymbol{\vartheta} = 0 \,, \tag{2.154}$$

$$\boldsymbol{b} = \boldsymbol{f}_0 - \mathbf{F}^{\mathrm{T}} \boldsymbol{w} = 0 \,. \tag{2.155}$$

Equation (2.155) is also known as the condition of universality, because the estimation is unbiased independent of $\boldsymbol{\vartheta}$. From the differentiated Lagrange function

$$H = \sigma^2 - 2\boldsymbol{w}^{\mathrm{T}} \boldsymbol{c}_0 + \boldsymbol{w}^{\mathrm{T}} \mathbf{K} \boldsymbol{w} + 2\lambda^{\mathrm{T}} \left( \boldsymbol{f}_0 - \mathbf{F}^{\mathrm{T}} \boldsymbol{w} \right) \tag{2.156}$$

follows the system of equations

$$\mathbf{K}\boldsymbol{w} + \mathbf{F}\boldsymbol{\lambda} = \boldsymbol{c}_0 \,, \tag{2.157}$$

$$\mathbf{F}^{\mathrm{T}} \boldsymbol{w} = \boldsymbol{f}_0 \,, \tag{2.158}$$

or

$$\begin{pmatrix} \mathbf{K} & F \\ \mathbf{F}^{\mathrm{T}} & 0 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{w} \\ \lambda \end{pmatrix} = \begin{pmatrix} \boldsymbol{c}_0 \\ \boldsymbol{f}_0 \end{pmatrix} \,. \tag{2.159}$$

The calculation of an estimated value is similar to in ordinary kriging (2.145). The kriging variance is given by

$$D_{\mathrm{K}}^2 = \sigma^2 - \boldsymbol{w}^{\mathrm{T}} \boldsymbol{c}_0 - \lambda^{\mathrm{T}} \boldsymbol{f}_0 \,. \tag{2.160}$$

In addition to the polynomial trend function, other approaches exist for trend adjustment. The adaptive trend function is selected, as it presents the theoretical considerations best. This is called a genetic–mathematical dependence. The systematic part of the realization can be estimated from additional knowledge about the genesis and geological processes.

### Universal Kriging with Filtering of Disturbances

The most general case (2.108) will be considered according to the representations of ordinary kriging and universal kriging. The random field is defined as the superposition of the stochastic random process $Z_\mathrm{s}(\boldsymbol{x})$ with a trend $m(\boldsymbol{x})$ and a disturbance $\xi$, i.e.,

$$Z(\boldsymbol{x}) = Z_\mathrm{s}(\boldsymbol{x}) + m(\boldsymbol{x}) + \xi \,. \tag{2.161}$$

Disturbances (so-called random noises) are measurement errors or random microvariability from a general course. The disturbance $\xi$ is a location-independent random variable with variance $\sigma_\xi^2$ and expectation value

$$E(\xi) = 0 \,. \tag{2.162}$$

The system of equations to be solved does not change, due to the location independence of $\xi$ and (2.162). The difference is in the construction of the covariance matrix $\mathbf{K}$ on its principal diagonal and the variance $\sigma^2$. Since the stationary part $Z_\mathrm{s}(\boldsymbol{x})$ and the disturbance $\xi$ are not correlated, (2.120) is therefore valid for the variance. The covariance matrix $\mathbf{K}$ is given by the sum

$$\mathbf{K} = \mathbf{K}_{Z_\mathrm{s}} + \mathbf{K}_\xi \,. \tag{2.163}$$

where $\mathbf{K}_{Z_\mathrm{s}}$ and $\mathbf{K}_\xi$ are the so-called covariance matrix of a random field $Z_\mathrm{s}(\boldsymbol{x})$ and the disturbance $\xi$. The observations have the same accuracy and the measurement errors are uncorrelated, which implies that

$$\mathbf{K}_\xi = \sigma^2 \mathbf{I} \,, \tag{2.164}$$

where $\mathbf{I}$ is the $n$-dimensional unit matrix. Weights are introduced for observations, which differ significantly with regard to their accuracy. Weights are used for dependent data class disturbances $\xi_i$, which are added to appropriate elements of the principal diagonal of the covariance matrix $\mathbf{K}$. The calculation of the estimated value is similar to in previous methods (2.145). The kriging variance differs from (2.160), being

$$D_\mathrm{K}^2 = \left(\sigma^2 - \sigma_\xi^2\right) - \boldsymbol{w}^\mathrm{T}\boldsymbol{c}_0 - \lambda^\mathrm{T}\boldsymbol{f}_0 \,. \tag{2.165}$$

*Menz* and *Pilz* [2.18] compared universal kriging with usual collocation in geodesy. The concept of collocation is the summary of equalization, prediction, and filtering. Universal kriging with filtering of disturbances and collocation leads to the same point estimates and the same estimation variances.

### Gradients Kriging

In nature, there are a variety of phenomena that affect each other; for example, the content of chemical elements of mineral deposits depends on the geological situation at the early stage of their evolution. The chemical content and the geometric parameters are independent random variables. The different genetic dependent random fields can be evaluated by cokriging. Relative changes of regionalized variables are available in addition to the observations. These gradients are also realizations of random variables of a stochastic field that was created by partial derivatives of the main field.

Gradients kriging was developed by *Menz* [2.11]. The assumptions on the co-regionalized variable are equal to a single regionalized variable.

Under the hypothesis of second-moment stationarity, the following is considered for each random field

- The population mean is constant

$$E[Z_k(\boldsymbol{x})] = m_k \,, \tag{2.166}$$

- For each pair of random fields $Z_k(x)$, $Z_l(x)$, the cross-variance function solely depends on the distance $\boldsymbol{h}$ of the random variables

$$E[Z_k(\boldsymbol{x}+\boldsymbol{h})Z_l(\boldsymbol{x})] - m_k m_l = \mathrm{cov}_{kl}(\boldsymbol{h}) = \sigma_{kl}(\boldsymbol{h}) \,, \tag{2.167}$$

- The cross-variogram is defined as

$$E[Z_k(\boldsymbol{x}+\boldsymbol{h}) - Z_k(\boldsymbol{x})][Z_l(x+h)Z_l(h)]) = 2\gamma_{kl}(\boldsymbol{h}) \,. \tag{2.168}$$

The spatial dependence of a random variable between the random fields is described by the cross-covariance function. The empirical cross-covariance function is not necessarily symmetric, in contrast to the autocovariance function (Fig. 2.36). The cross-covariance function contains more information than the autocovariance function.

The stochastic field $Z(\boldsymbol{x})$ is second-moment stationary and differentiable. The derivation of $Z(\boldsymbol{x})$ is defined as the derivative in the quadratic mean as

$$\lim E\left\{\left[\frac{Z(\boldsymbol{x}+\boldsymbol{a}) - Z(\boldsymbol{x})}{a} - Z'(\boldsymbol{x})\right]^2\right\} = 0 \,. \tag{2.169}$$

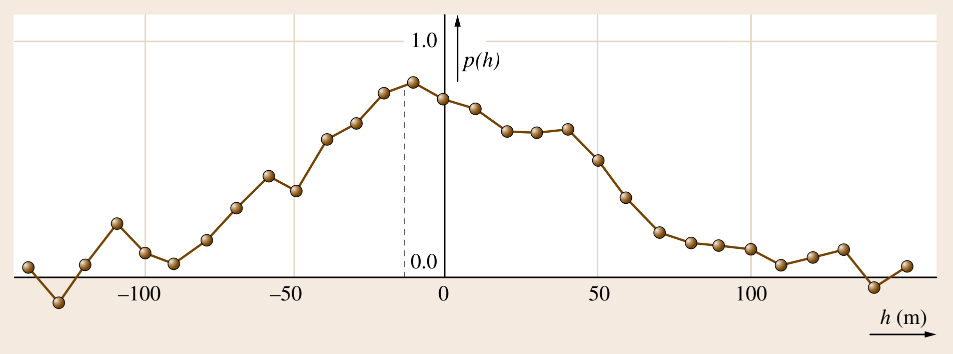

**Fig. 2.36** Example of an empirical cross-covariance function

The partial derivatives exists according to (2.169).

$$Z_x(x,y) = \frac{\partial Z(x,y)}{\partial x} \quad \text{and} \quad Z_y(x,y) = \frac{\partial Z(x,y)}{\partial y}\,, \tag{2.170}$$

$Z_x(\boldsymbol{x})$ and $Z_y(\boldsymbol{x})$ are also second-moment-stationary random fields whose autocovariance functions $\sigma_{Z_xZ_x}(\boldsymbol{x})$ and $\sigma_{Z_yZ_y}(\boldsymbol{x})$ can be derived from the covariance $\sigma_z(\boldsymbol{x})$ of the source field. The model function adopted for the empirical cross- and autocovariance functions must be at least twice continuously differentiable. For this reason, the exponential and spherical models are eliminated. Both of these models are not differentiable at the point $|\boldsymbol{h}| = 0$. The Gaussian model remains the most common model.

Two stationary random fields $Z(\boldsymbol{x})$ and $Y(\boldsymbol{x})$ and a stationary random field of gradients $G(\boldsymbol{x})$ are used for the estimation of $Z(\boldsymbol{x}_0)$. The random field of gradients can originate from any stochastic field that can be described with a co-regionalized variable. This means that $G(\boldsymbol{x})$ can be the derivative of $Z(\boldsymbol{x})$ or $Y(\boldsymbol{x})$, but does not need to be. Knowledge of the auto- and cross-covariance function of the source fields is a condition for inclusion of the gradient.

The estimate of the random variable $Z(\boldsymbol{x}_0)$ continues with the linear combination of random variables of all co-regionalized variables as

$$\hat{Z}_0 = \boldsymbol{w}^{\mathrm{T}}\mathbf{X} = \boldsymbol{a}^{\mathrm{T}}\mathbf{Z} = \boldsymbol{b}^{\mathrm{T}}\mathbf{Y} + \boldsymbol{c}^{\mathrm{T}}\mathbf{G}_x + \boldsymbol{d}^{\mathrm{T}}\mathbf{G}_y \tag{2.171}$$

with

$$\boldsymbol{w}^{\mathrm{T}} = (\boldsymbol{a}^{\mathrm{T}}, \boldsymbol{b}^{\mathrm{T}}, \boldsymbol{c}^{\mathrm{T}}, \boldsymbol{d}^{\mathrm{T}})\,, \tag{2.172}$$

$$\mathbf{X}^{\mathrm{T}} = \left(\mathbf{Z}^{\mathrm{T}}, \mathbf{Y}^{\mathrm{T}}, \mathbf{G}_x{}^{\mathrm{T}}, \boldsymbol{d}_y{}^{\mathrm{T}}\right). \tag{2.173}$$

The linear combination (2.130) is extended by only weighted random variables from other fields with spatial dependence on the estimate location. Trends and disturbances should be considered as an alternative and (2.130) is valid, therefore (2.161) is used. The expected values are constant if second-moment stationarity (2.166) applies, and location-dependent otherwise. The population means are

$$\begin{aligned} E[Z(x)] = m_Z(x),\ E[Y(x)] = m_Y(x),\ E[G(x)] \\ = m_G(x)\,, \end{aligned} \tag{2.174}$$

if it is assumed that every random field could have a trend. The trend is described by polynomials up to second degree (2.149).

The kriging system uses derivatives of the trend function instead of the trend of the random field of gradients.

The design matrix $\mathbf{F}$ is generally given by

$$F^{\mathrm{T}} = \begin{pmatrix} F_z{}^{\mathrm{T}} & 0^{\mathrm{T}} & 0^{\mathrm{T}} & 0^{\mathrm{T}} \\ 0^{\mathrm{T}} & F_y{}^{\mathrm{T}} & 0^{\mathrm{T}} & 0^{\mathrm{T}} \\ 0^{\mathrm{T}} & 0^{\mathrm{T}} & F_{G_x}{}^{\mathrm{T}} & F_{G_y}{}^{\mathrm{T}} \end{pmatrix}. \tag{2.175}$$

Due to the universality condition, the function to be minimized and the system of equations to be solved are identical to (2.155–2.159) for universal kriging.

The covariance matrix $\mathbf{K}$ is built up in the general case as

$$K^{\mathrm{T}} = \begin{pmatrix} K_{ZZ} & K_{ZY} & K_{ZG_x} & K_{ZG_y} \\ K_{YZ} & K_{YY} & K_{YG_x} & K_{YG_y} \\ K_{G_xZ} & K_{G_xY} & K_{G_xG_x} & K_{G_xG_y} \\ K_{G_yZ} & K_{G_yY} & K_{G_yG_x} & K_{G_yG_y} \end{pmatrix}. \tag{2.176}$$

The elements of the submatrices are computed from the auto- or cross-covariance functions.

The estimated value and the kriging variance result from (2.145) and (2.165). When disturbances occur, they are added to the principal diagonal of the autocovariance of the respective random field.

### Continuation of the Example – Kriging Part

At any location $\boldsymbol{x}$, the value $\hat{z}$ can be estimated after modeling the spatial dependence. Points, profiles, and surfaces are predicted by the kriging equation system. The kriging equation system is set up and solved on the basis of the estimated variogram (section on *Ordinary Kriging*). Ordinary kriging is used because there is no trend in the 991 boreholes. The result is the estimated value $\hat{z}_0$ and the kriging variance $D_{\mathrm{K}}^2$.

For simplicity, three neighboring locations will be used to estimate $\hat{z}_0(\boldsymbol{x})$. The kriging system of equations for $n = 3$ reads as follows (2.142) or (2.143)

$$\begin{pmatrix} \gamma(x_1x_1) & \gamma(x_1x_2) & \gamma(x_1x_3) & 1 \\ \gamma(x_2x_1) & \gamma(x_2x_2) & \gamma(x_2x_3) & 1 \\ \gamma(x_3x_1) & \gamma(x_3x_2) & \gamma(x_3x_3) & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \boldsymbol{w}_3 \\ \lambda_\gamma \end{pmatrix}$$

$$= \begin{pmatrix} g_0(x_1x_0) \\ g_0(x_2x_0) \\ g_0(x_3x_0) \\ 1 \end{pmatrix} .$$

For the matrix, the Euclidean distances between the three locations and between the estimated location $x_0$ and the three location are needed (Fig. 2.37).

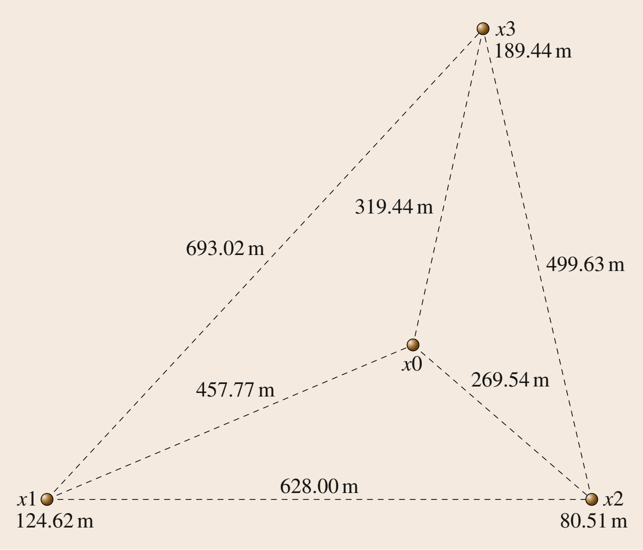

**Fig. 2.37** Estimation at the location $x_0$ with Euclidean distances from $x_1$, $x_2$, and $x_3$

The individual elements of the kriging equation system are obtained as

$$\begin{aligned} \gamma(x_1x_1) &= \gamma(x_2x_2) = \gamma(x_3x_3) = 0 , \\ \gamma(x_1x_2) &= \gamma(x_2x_1) = C_0 + C_\gamma(x_2 - x_1) \\ &= 1.2\gamma\left(\frac{628}{2020}\right) = \gamma(0.311) = 0.5446 , \\ \gamma(x_1x_3) &= \gamma(x_3x_1) = C_0 + C_\gamma(x_3 - x_1) \\ &= 1.2\gamma\left(\frac{693}{2020}\right) = \gamma(0.343) = 0.5973 , \\ \gamma(x_2x_3) &= \gamma(x_3x_2) = C_0 + C_\gamma(x_3 - x_2) \\ &= 1.2\gamma\left(\frac{500}{2020}\right) = \gamma(0.247) = 0.4380 , \\ \gamma(x_1x_0) &= C_0 + C_\gamma(x_1 - x_0) = 1.2\gamma\left(\frac{458}{2020}\right) \\ &= \gamma(0.227) = 0.4022 , \\ \gamma(x_2x_0) &= C_0 + C_\gamma(x_2 - x_0) = 1.2\gamma\left(\frac{270}{2020}\right) \\ &= \gamma(0.134) = 0.2394 , \\ \gamma(x_3x_0) &= C_0 + C_\gamma(x_3 - x_0) = 1.2\gamma\left(\frac{319}{2020}\right) \\ &= \gamma(0.158) = 0.2823 . \end{aligned}$$

After inserting the values into the kriging equation system, the resulting equation is

$$\begin{pmatrix} 0 & 0.5446 & 0.5973 & 1 \\ 0.5446 & 0 & 0.4380 & 1 \\ 0.5973 & 0.4380 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \boldsymbol{w}_3 \\ \lambda_\gamma \end{pmatrix} = \begin{pmatrix} 0.4022 \\ 0.2394 \\ 0.2823 \\ 1 \end{pmatrix} . \tag{2.177}$$

The result of the matrix is four parameters

$$\boldsymbol{w}_1 = 0.2262 , \quad \boldsymbol{w}_2 = 0.4223 , \quad \boldsymbol{w}_3 = 0.3515 ,$$
$$\boldsymbol{\lambda}_\gamma = -0.0378 .$$

Including $z(\boldsymbol{x}_1)$, $z(\boldsymbol{x}_2)$, $z(\boldsymbol{x}_3)$, and $w_i$, the estimated value $\hat{z}_0$ can be computed by using (2.145) as

$$\begin{aligned} \hat{z}_0 &= 0.2262 \cdot 124.62\,\mathrm{m} + 0.4223 \cdot 80.51\,\mathrm{m} \\ &\quad + 0.3515 \cdot 189.44\,\mathrm{m} = 120.65\,\mathrm{m} . \end{aligned}$$

The estimated value is a $z$-value for the representation of a surface elevation.

The kriging variance results from (2.146) as

$$\begin{aligned} D_{\mathrm{K}}^2 &= 1.2 - 0.2262 \cdot 0.4022 + 0.4223 \cdot 0.2394 \\ &\quad + 0.3515 \cdot 0.2823 + 0.0378 = 0.946 . \end{aligned}$$

The location $x_0$ has the best unbiased estimate with the smallest estimated variance. All other estimates have larger estimated variance.

### 2.3.7 Geostatistical Simulation

The generation of random variables for a specific location, which describes a random process $Z$ as a whole, is referred to as geostatistical simulation. The random variables must fulfill a condition regarding the terms of their $n$-dimensional distribution function and its statistical moments; for example, random variables with constant distribution function and covariance function characterize a stochastic process with second-moment stationarity. If random values are normally distributed, then the random field is completely described by its covariance function. If agreement between simulated and observed values is requested for specific locations $\boldsymbol{x}_i$, this is called conditional simulation. Figure 2.38 compares simulation, conditional simulation, and kriging.

The simulated values show the same structural characteristics as those of the investigated parameter. This applies in particular for its variability; $n$ realizations of this process can be simulated with the same parameter of the natural random process on demand. Each of these realizations is a theoretical image by reality. The geostatistical simulation is applied for investigations where the spatial variability of the parameter has an important influence (for example, risk analysis or uniformity considerations).

Sequential Gaussian simulation (SGS) uses standards for the simulation of a random field or a field value. SGS considers the neighborhood relation to previously defined and simulated pressure points. The conditional simulation is then automatically included in the simulation process, which must be transformed into a $N(0, 1)$ distribution of points for consideration (as real data). The process of simulation can be represented by decomposition of the simulated random values into their stochastic and deterministic components. The deterministic part describes the influence from considered sampling values to the simulated value. The stochastic part includes the variability by considering a weighted, normally distributed random variable $\psi$. The weighting coefficient is the square root of the kriging variance $D_{\mathrm{K}}^2$ of the estimated value $\hat{z}_0$. It follows that the simulated value $\hat{z}_0^s$ is the sum of the value estimated by simple kriging $\hat{z}_0$ and a $N(0, D_{\mathrm{K}}^2)$-distributed random variable, thus

$$\hat{z}_0^s = \hat{z}_0 + D_{\mathrm{K}}\psi\ , \qquad \psi \in N(0, 1)\ . \tag{2.178}$$

The simulated value is considered for the simulation of other random values in the simulation algorithm. Assuming that $E[Z(\boldsymbol{x})] = \mu$ and $E[Z(\boldsymbol{x}_0)] = \mu(\boldsymbol{x})_0$ are known, simple kriging provides the best linear unbiased estimates for $\hat{z}_0$ as

$$\hat{z}_0 = \boldsymbol{c}_0{}^{\mathrm{T}}\mathbf{K}(\boldsymbol{z}^{\mathrm{T}} - \boldsymbol{\mu}) + \mu(\boldsymbol{x}_0)\ , \tag{2.179}$$

$$D_{\mathrm{K}}^2 = \operatorname{cov}(\boldsymbol{x}_0, \boldsymbol{x}_0) - \boldsymbol{c}_0{}^{\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{c}_0\ . \tag{2.180}$$

The order of the simulation for the values $\hat{z}_0^s$ is determined by a random-number generator. Deviations from this procedure result in anisotropies of the covariance.

From this general scheme [2.19] further kriging methods and other variants can be derived. Simple krig-

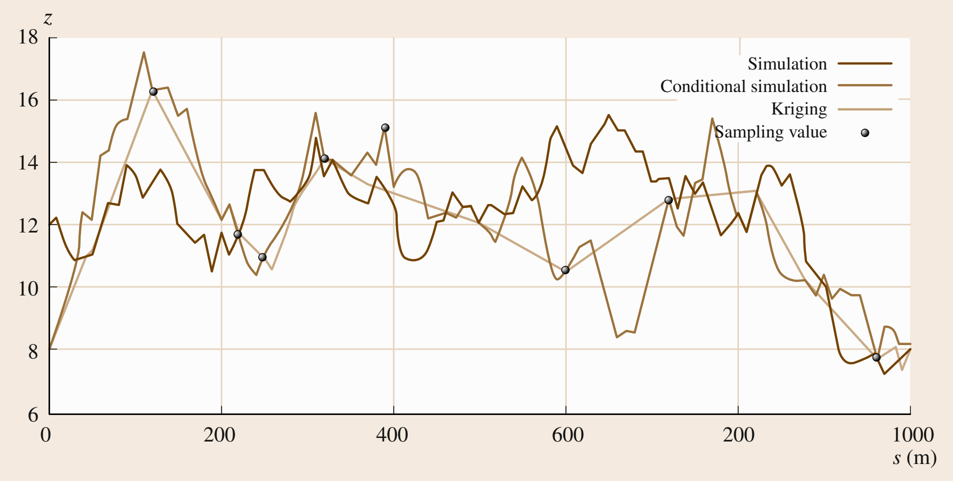

**Fig. 2.38** Representation of simulation, conditional simulation, and kriging

ing is replaced by cokriging. Additional parameters are taken into account in the simulation. Another approach is to estimate the deterministic component of the simulated value at random by using indicator kriging.

## References

2.1 E. Mikhail, F. Ackermann: *Observations and Least Squares* (Univ. Press America, New York 1976)

2.2 W. Niemeier: *Ausgleichungsrechnung* (De Gruyter, Berlin 2002)

2.3 Conformal map, http://en.wikipedia.org/wiki/Conformal_map (Wikipedia 2011)

2.4 Student's t-test, http://en.wikipedia.org/wiki/T-test (Wikipedia 2011)

2.5 http://en.wikipedia.org/wiki/Graph_(mathematics) (Wikipedia 2011)

2.6 Event-driven process chain, http://en.wikipedia.org/wiki/Event-driven_process_chain (Wikipedia 2011)

2.7 N.A.C. Cressie: *Statistics for Spatial Data* (Wiley, New York 1991)

2.8 G. Matheron: *Estimating an Choosing* (Springer, Berlin, Heidelberg 1989)

2.9 S. Meier, W. Keller: *Geostatistik: Einführung in die Theorie der Zufallsprozesse* (Akademie, Berlin 1990)

2.10 H. Wackernagel: *Multivariate geostatistics* (Springer, Berlin, Heidelberg 2003)

2.11 J. Menz: Geostatistische Vorhersage des Schichtenverlaufes im Gebirge auf der Grundlage von Bohrungen, Stoßbemusterungen und geophysikalischen Messungen, Markscheidewesen **98**(2), 70–73 (1991)

2.12 A.G. Journel: Nonparametric estimation of spatial distributions, Math. Geol. **15**, 445–468 (1991)

2.13 D.E. Myers: Pseudo-cross variograms, positive-definitness, and cokriging, Math. Geol. **23**, 805–816 (1991)

2.14 M. Armstrong, P. Diamond: Testing variograms for positive-definitness, Math. Geol. **16**(4), 407–421 (1992)

2.15 M. Stone: Cross-validadory choice and assesment of statistical predictions, J. R. Stat. Soc. **36**, 111–133 (1974)

2.16 S. Geisser: The predictive sample reuse method with apllications, J. Am. Stat. Assoc. **70**, 320–328 (1975)

2.17 E.H. Isaaks, R.M. Srivastava: *An Introduction to Applied Geostatistics* (Oxford Univ. Press, New York 1989) p. 561

2.18 J. Menz, J. Pilz: Kollokation, Universelles Kriging und Bayes'scher Zugang, Markscheidewesen **101**(2), 62–66 (1994)

2.19 P.A. Dowd: A review of recent developments in geostatistics, Comput. Geosci. **17**(10), 1481–1500 (1991)

# 3. Databases

Thomas Brinkhoff, Wolfgang Kresse

As geographic information storage and applications matured, their use as databases followed. A typical geoconfiguration consists of a map combined with an object-relational database, similar to the 300 year-old example shown in Fig. 3.1. Other geographic databases such as the well-known Earth browsers Bing or Google Maps contain a simple, but large, collection of raster orthophoto maps. Vector maps require a far more sophisticated data model and are usually rendered while being read from the database and presented on a display device.

Sections 3.1–3.4 provide basic knowledge about database theory. The two most common models, namely the relational and the object-oriented model, are explained. The second part of this chapter (Sects. 3.5–3.11) explains the geospecific aspects of database technology. It starts with Sect. 3.5 about spatial databases with vector and raster models, referencing the relevant standards. Section 3.6 covers spatial queries and filtering. Section 3.7 explains indexing, which supports acceleration of queries. Section 3.8 provides an overview of network databases and some prominent network search algorithms. Section 3.9 is dedicated to raster databases and Sect. 3.10 introduces time in the context of spatiotemporal databases. Section 3.11 summarizes the most widespread database software solutions.

## 3.1 Historical Background

The term *database* was coined with the advent of the computer. However, databases are not a 20th century concept. Though known under different names, databases have been applied for centuries. The early taxation systems are one of the oldest examples. Figure 3.1 shows a late 17th century tax-cadastral map that is linked to a table of tax-relevant parameters necessary to determine the correct farmer's tax.

Today, databases are present in every corner of our life. Modern database technology was developed for administration of bank accounts, followed by warehouse management. Databases are scalable from small desktop solutions to huge server clusters, e.g., a private photograph collection versus administration of an entire cellphone system. Some databases may typically have only a small number of modifications per time period, while others are very dynamic. A telephone directory or a railroad schedule has a static nature, while an Internet shop or a monitoring system for vehicle movements is very dynamic.

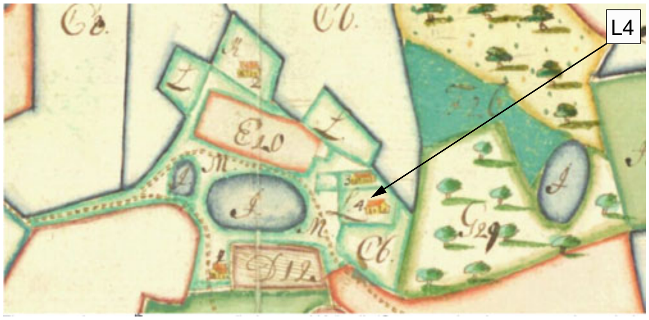

**Fig. 3.1** Late 17th century *Matrikel* map of Kröpelin/Germany, showing property boundaries, land use, and pointers to the tax data tables (after [3.1])

### 3.1.1 Features of a Database

A *database* is an organized computer-based collection of data that allows the management of those data including insertion, modification, retrieval, and deletion. The term database comprises the involved software and the hardware including the physical data storage. The operation of a database is controlled by a database management system (DBMS) that provides the interfaces for the communication from the outside and conceals the physical data storage from the applications.

A database has four typical properties [3.2].

#### Reliability

It shall deliver an uninterrupted service and be able to cope with unforeseen situations like the interruption of

**Table 3.1** Example of a tax data table

| **Attribute** |
|---|
| Pointer: L4 |
| Municipality: Kröpelin |
| Location of the village |
| Proprietor |
| Municipality |
| Name of owner |
| Societal position of owner (full farmer, half farmer, *kossat*) or profession (tailor, blacksmith) |
| Size and quality of fields |
| Description of meadows, pastures, woods, waters, roads |
| Soil quality, quantity of sowing, quantity of yield, livestock |
| Tax obligation |
| Labor service obligation |

a data link. For example during a purchase via the Internet, we expect that the database will be reset to its original state if the Internet-connection breaks down after we had paid with our credit card but before the purchase had been confirmed.

### Correctness and Consistency

The internal logics of the data of our database shall be correct. For example, if we have created a database of our photographs we expect that the photos shown in the table of contents do exist in the database and the other way around the photos that we have loaded appear in the table of contents, too.

### Technology Proof

The DBMS shall be independent of the details of the hardware and software. It is commonplace that computer hardware and software keep on developing fast. However, if we visit an Internet shop we do not want to know, which software and hardware it is built of and if this has been updated recently.

### Security

The data of a database shall be protected against loss and unauthorized read and write access. The typical example regards a bank account. We do not want to let anybody else read how much money we have or our depts (read access). At the same time the bank does not allow us to change the amount of money on our account be simply typing in new figures (write access). Typically data access is partitioned in two or more levels of rights such as user and administrator rights.

## 3.1.2 Database Architecture

A database is structured in three levels, the external level, the conceptual level, and the internal level.

A user has usually only a limited access to the database. We again may think of our bank account. Such an access is called a view of the database. A view is defined by a subset of the database content and a level of authorization. The totality of all views forms the external level.

Within the database the data are organized and stored in a way that had been designed before the database was set up. The database design is explained further below. For example, consider again the database of our photographs. It may have a simple hierarchy with folders named according to events such as holiday_2011 on the top level and the photographies on the bottom level. Such a structure is called a user-oriented conceptual schema. It resides on the conceptual level of the database.

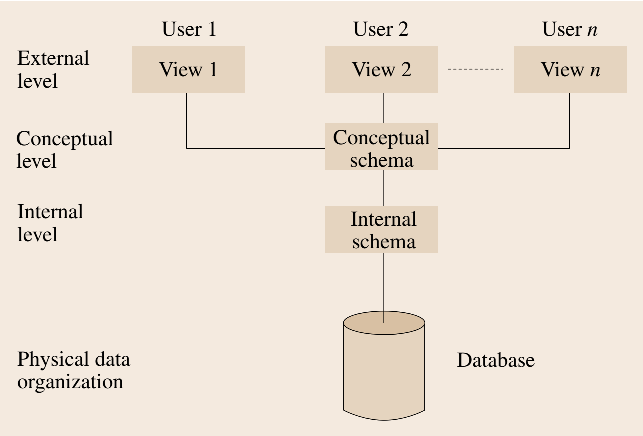

**Fig. 3.2** ANSI/SPARC three-level architecture (after [3.2, 3])

The internal schema comprises all aspects of the physical data storage. The related level is called the internal level.

The three-level architecture has been developed in the 1970s and standardized by the American National Standards Institute (ANSI)/Standards Planning and Requirements Committee (SPARC) (Fig. 3.2, [3.3]).

### Database Management System (DBMS)

A Database Management System (DBMS) is the totality of software components that define the data model, realize all database properties, and provide the interface between the application programming interface (API) and the physical storage of the data.

Figure 3.3 explains the structure of and the workflow within a DBMS. A query is submitted to the DBMS through the user interface using a data interaction language, usually the Structured Query Language (SQL). The user interface is typically an API or the console for the administrator's access. The query is forwarded to the query compiler, which either sends it directly to the runtime database processor for forming the final retrieval command or to the query optimizer. The query optimizer is invoked to improve the database performance. For instance, if a query requires a Cartesian product of three tables that are significantly different in size, say 100, 10 000 and 1 000 000 rows, then it is faster to first compute the product of the two smaller tables followed by the large one compared to computing the product of the two larger tables first.

The database may hold, for example, land parcels whose areas are constrained to values larger than zero

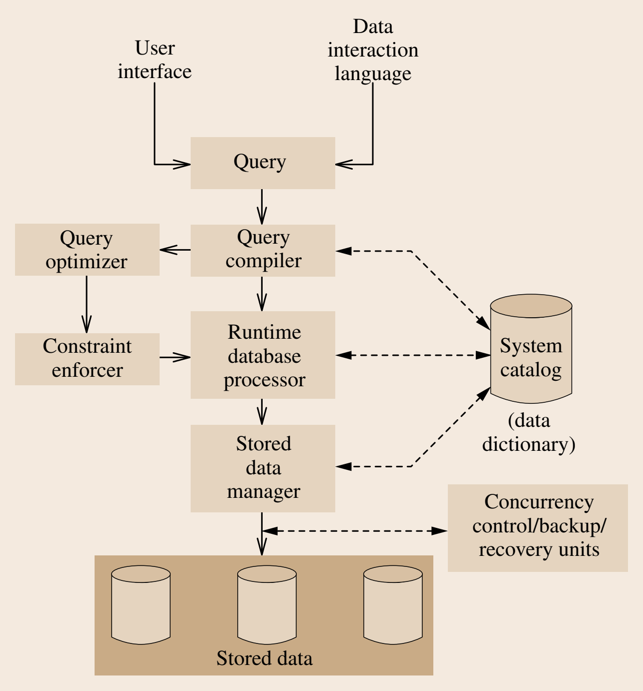

**Fig. 3.3** DBMS components used to process user queries (after [3.2])

square meters. Every time the database is modified, the constraint enforcer checks the permissibility under the defined constraints and refuses a change request when necessary. The runtime database processor handles the access to the physical data storage, technically spoken to the files in which the data actually reside. At this level critical situations regarding the overlapping access to the data and data preservation in case of unforeseen disruptions are cared for. These situations are named concurrency control, backups and recovery which are explained below. Sometimes, the part of the DBMS which controls the critical situations is called the transaction engine. The query compiler, the runtime data processor, and the stored data manager read their control data from the system catalog which is a data dictionary that holds the three schemas explained in the upper part of this Sect. 3.1.2.

### 3.1.3 Operational Requirements

The basic demands on a database are quite self-evident. A database query shall be answered after a reasonable time span, and the works of simultaneous operations shall not interfere. In other words: The availability and the performance shall be good. The accesses of many users shall be independent. The latter can be established by locking that part of a database to which one user has granted access to.

#### Transactions

Above, we have seen the example of a disruption of the network connection while paying for an Internet-purchase. Such an irregularity must not lead to a loss of our money and, seen from the logics of the database, it must not lead to an inconsistent dataset. The payment is an example of a database transaction. Transactions are triggered with the commands *Insert*, *Modify*, *Delete*, and *Retrieve*. A transaction follows four rules.

- The transaction is *atomic* as it cannot be split into smaller transactions. That is, the user can rely on a transaction performing all its actions in one step or performing none of the actions at all.
- The transaction is *consistent* as it begins and ends with a consistent database.
- The transaction is *isolated* as the involved data cannot be accessed from other users during the transaction. While interacting with the database a user should feel that he or she is the only user at that time.
- The transaction is *durable* as it ends with a permanent modification of the database.

Because of the first letters of the four keywords the rules are often called ACID-rules.

After a successful transaction the database is set to the new state. The related command is named *Commit*. If a problem occurred the database is set back to its original state. This is called a recovery of the database. The related command is named *Rollback*.

The simultaneous access of many users to the same database requires an exclusion of more than one write access to the same data. This is called concurrency control. Reference [3.2] gave a good example of the consequences of disregarding concurrent accesses to the same data which lead to an inconsistent database.

For example, suppose that my bank balance is $ 1000. Two transactions are in progress: $T_1$ to credit my account with $ 300, and $T_2$ to debit $ 400 from my account. Table 3.2 shows a particular sequence of the constituent operations of each transaction, termed interleaving. Transaction $T_1$ begins by reading my balance $B$ from the database into a program variable $X$ and increasing $X$ to $ 1300. Transaction $T_2$ then starts by reading the same balance from the database into $Y$ and decreasing $Y$ to $ 600. $T_1$ then concludes by writ-

**Table 3.2** Lost update for nonatomic interleaved transactions, $T_1$ and $T_2$, with variables $X$ and $Y$ and bank balance $B$ (after [3.2])

| $T_1$ | $T_2$ | $B$ (US $) | $X$ (US $) | $Y$ (US $) |
|---|---|---|---|---|
| | | 1000 | | |
| $X \leftarrow B$ | | 1000 | 1000 | |
| $X \leftarrow X + \text{US\,\$\,300}$ | | 1000 | 1300 | |
| | $Y \leftarrow B$ | 1000 | 1300 | 1000 |
| | $Y \leftarrow Y - \text{US\,\$\,400}$ | 1000 | 1300 | 600 |
| $B \leftarrow X$ | | 1300 | 1300 | 600 |
| | $B \leftarrow Y$ | 600 | | 600 |

ing $X$ to the database as the new balance of $ 1300, and $T_2$ writes $Y$ to the database as the new balance of $ 600. It is as if transaction $T_1$ never occurred, a problem known as lost update. Interleaving can improve database performance, because shorter operations may be executed while more lengthy operations are still in progress. However, interleaving must be controlled to avoid problems such as lost update [3.2].

### 3.1.4 Data Models

In Sect. 3.1.2 we mentioned the example of a database of photographs with folders named according to events and the photographs of those events underneath those folders. This structure can be imagined as a simple tree with the folders as the branches and the photographs as the leaves. The corresponding data model is a hierarchical model.

A data model is an explicit definition of the structure of data. In modern database technology only two types of data models (*database models*) are common: The relational model and the object-oriented model. The relational model subdivides into the relational model by *Codd* [3.4] and the entity-relationship model by *Chen* [3.5, 6].

The object-relational database management system (ORDBMS) combines elements of the relational and the object-oriented model. The hierarchical model is applied in the simple example above. It offers an efficient data storage, but its application is slightly decreasing because of the limitations. For instance, it cannot model the links between many house-owners and many insurance companies, which is a many-to-many relationship shown in Sect. 3.2.4.

The relational model is by far the most common model of today and discussed in Sect. 3.2.

The applications of the entity-relationship model are rarer. But after almost four decades it is still acknowledged because of its semantic expressiveness. It is common ground that its concepts have influenced the development of the object-oriented model and probably will continue playing their role in future developments. The entity-relationship model is discussed in Sect. 3.4.

The object-oriented model overcomes many limitations of the relational model. For instance, the storage of a polygon in a relational model according to Codd is laborious because it cannot be extended to inhabit new data types. Such limitations have been overcome by the object-oriented model which is discussed in Sect. 3.3.

## 3.2 Relational Model

This section regards the relational database in the way it has been defined by *Codd* in 1970 [3.4]. The data model is based on set theory and predicate logic. The set theory is explained below in Sect. 3.2.2. The predicate logic is a branch of the mathematical logic that considers the subject–predicate relation [3.7].

A relational database consists of a group of unordered tables. Each table has a number of rows and columns. Each row represents an item that is described by the attributes which are placed in the columns. *Codd* called such a table a relation because it inhabits a related set of information [3.2]. Said in other words, each row provides information that is inter-related by residing in the same table. This is in contrast to the popular but wrong thinking that the name relational database refers to the relation between the tables.

Often, the term tuple is used instead of row. Originating in the set theory a tuple is a sequence of $n$ elements where $n$ is a positive integer.

The relational database model gained wide acceptance because of its simplicity. The mayor benefits are quoted in [3.8].

- Data entry, updates and deletions will be efficient.
- Data retrieval, summarization and reporting can be efficiently computed by introducing techniques like indexes (Sect. 3.4).
- Since the database follows a well-formulated model, it behaves predictably.
- Since much of the information is stored in the database rather than in the application, the database is somewhat self-documenting.
- Changes to the database schema are easy to make.

### 3.2.1 Design

To state the obvious: A database is always an abstracted representation of the real world and thus can never contain any element that one could think of. The definition of the conceptual schema, the real world's subset and its structure, and the mapping to the tables of a database is called the design of the database. The design is based on the three-level-architecture explained in Sect. 3.1.2.

Unfortunately the design of a database cannot be automated. It rather requires decisions on the shape of the future database. For example, a database shall manage the plants of a horticulture company. We assume that apple-trees, oak-trees, black currant bushes and lilac bushes exist. If we want to create two types of tables we may put the trees in one table (apple and oak) and the bushes in the other (currant and lilac). Alternatively we could put the fruit plants in one table (apples and currant) and nonfruit plants in the other (oaks and lilacs), which is probably the more adequate solution for a horticulture company.

### 3.2.2 Tables

In a relational model a table is the central element. A table represents the real world as far as it shall be modeled for a given task. A relational database consists of one or many tables that are related to each other. Obviously, one table or even a few would be an exceptional case. As shown in the following examples, a table may represent the gardens/parcels of a city, the owners of those garden/parcels, or further information about the owners such as their telephone numbers. A table consists of rows and columns. Each row represents an instance of the real-world item the table represents. Each column represents an attribute of the item [3.2, 4, 9].

There are a number of rules regarding the creation of the rows of a table in the relational model. These rules guarantee unambiguous addressing of every item of information in the relational database by programming. The uniqueness of a row within a table can be guaranteed by designating a primary key. This is a column or a combination of columns that contains unique values throughout the table. It is used to address every individual row and its information by giving the table name and the primary key. By definition, each table can have only one primary key, even though several columns or combinations of columns (called candidate keys) may have unique values. The primary key must be selected from one of the candidate keys.

It is up the designer of the relational database to decide which of the candidate keys shall become the primary key. The decision should be based upon the principle of [3.10]

- minimality (choose the fewest columns necessary),
- stability (choose a key that seldom changes), and
- simplicity/familiarity (choose a key that is both simple and familiar to users).

The main properties regarding tables – relationships and normalization – are elaborated using an example featuring owners, gardens, and plants. Let us assume that a city has a table of garden owners called *owner*, which looks like the table shown in Table 3.3.

The candidate keys for *owner* include

- OwnerID,
- Telephone.

The telephone is not favorable because they might change frequently over time.

How should we decide regarding OwnerID and LastName + FirstName? Both choices are reasonable, though names change sometimes, for instance with marriage. However, the answer is easy because numeric columns can be searched and sorted more efficiently than character columns.

The previous paragraphs outlined the theory. In many practical applications the primary key is an additional static integer number that does not change over time because it is not selected from the information given in a table. A typical primary key is an order identifier (ID) or simply a row counter.

**Table 3.3** Table *owner*: The best choice for primary key for *owner* would be OwnerID

| Table: owner | | | | | | |
|---|---|---|---|---|---|---|
| **OwnerID** | **LastName** | **FirstName** | **Address** | **ZipCode** | **City** | **Telephone** |
| 2 | Johs | Paul | Goethestraße 17 | 10625 | Berlin | 030/8886902 |
| 4 | Nebel | Gregor | Kantstraße 102 | 22089 | Hamburg | 040/9809099 |
| 15 | Magnus | Karl | Brechtstraße 25 | 17034 | Neubrandenburg | 0395/7837890 |
| 23 | Johs | Georg | Goethestraße 17 | 10625 | Berlin | 030/8886902 |

**Table 3.4** Table *garden0*: OwnerID is a foreign key in *garden0* which can be used to reference an owner stored in the *owner* table

| Table: garden0 | | | | |
|---|---|---|---|---|
| **GardenID** | **OwnerID** | **Area ($m^2$)** | **Price (€)** | **Soiltype** |
| 1 | 4 | 1303 | 1303 | Mould clay |
| 2 | 23 | 4075 | 4075 | Sandy clay |
| 3 | 15 | 911 | 911 | Clay |
| 4 | 2 | 2423 | 2423 | Loam |
| 5 | 23 | 892 | 892 | Silt–loam |
| 6 | 2 | 1550 | 1550 | Loam |

#### Foreign Keys

A relational database is built of tables. Primary keys are used to create the relationships between the tables. A foreign key in one table points to a primary key in another table, or expressed the other way around, a foreign key is a column in a table used to reference a primary key in another table.

Continuing the example given in the last section, let us assume that we choose OwnerID as the primary key for *owner*. Now we define a second table, *garden0*, as shown in Table 3.4.

OwnerID is considered a foreign key in *garden0*, since it can be used to refer to a given person, i. e., a row in the *owner* table [3.8].

**Table 3.5** The tables *garden01* and *confidential* are related in a one-to-one relationship. The primary key of both tables is GardenID

| Table: garden01 |
|---|
| GardenID |
| Area |
| Soiltype |

One-to-one (1 : 1)

| Table: confidential |
|---|
| GardenID |
| OwnerID |
| Price |

### 3.2.3 Relationships

Relationships in the real world might be quite complex. Think of a resident in a city. He or she has numerous private and official relationships within the neighborhood, with the city council, and beyond. However, the model of a relational database only allows relationships between pairs of tables. These tables can be related in one of three different ways [3.8]

1. one to one (1 : 1),
2. one to many (1 : $n$),
3. many to many ($m$ : $n$).

#### One-to-One Relationships

In this simple case, two tables are related such that each row of the first table has at most one partner row in the second table. This type of relation rarely exists in the real world. However, it is often applied in order to hide information such as personal data in a second table.

In the example shown in Table 3.5, the prices of the gardens are separated from the technical data, e.g., soil type.

Another example may be a large table of which only a part is needed in another application. Then it may be advisable to split the large table into two which are related with a one-to-one relation, in order to use only the smaller part-table in the other application.

For the sake of clarity, tables that are related in a one-to-one relationship should always have the same primary key [3.8].

### One-to-Many Relationships

A one-to-many relationship of two tables creates a tree-like structure. This means that, for each row in the first table, there can be zero, one, or many rows in the second table, but for every row in the second table there is exactly one row in the first table.

For example, each garden can have many trees. Therefore, *garden0* is related to *gardenDetails* in a one-to-many relationship (Table 3.6). It is obvious that the one-to-many relationships are the standard case in relational databases.

### Many-to-Many Relationships

A many-to-many $(m : n)$ relationship of two tables means that a row of the first table may have many related rows in the second table and that at the same time a row in the second table may have many related rows in the first table. An example is the relation between house owners and insurance companies/their insurance programs. A house owner usually has contracts with several insurance companies, while an insurance company has many house owners as customers. Thus, the *houseOwner* table in a real-estate database would be related to the *insurer* table in a many-to-many relationship.

Many-to-many relationships cannot be modeled in relational databases and therefore have to be split into two steps: a one-to-many $(1 : n)$ relation and a many-to-one $(n : 1)$ relation. This requires an intermediate table that links to both original tables, in the example above the house owners and the insurance companies. In the example shown in Table 3.7, this linking table is called *houseOwnerLinksInsurer* and would contain one row for each insurance program of each house owner.

**Table 3.6** There can be many trees, bushes, and vegetables in a garden, so *garden0* and *gardenDetails* are related in a one-to-many relationship

| **Table: garden0** |
|---|
| GardenID |
| Soiltype |

One-to-many $(1 : n)$

| **Table: gardenDetails** |
|---|
| GardenDetailsID |
| Trees |
| Bushes |
| Vegetables |

## 3.2.4 Normalization

The design of a database schema allows for a number of choices. Those choices include the definition of tables which may result in a few large tables or many smaller tables. A large table with many columns may have all information in one row. A small table with a few columns may allow for a better overview and more flexible creation of relationships between the tables. The primary and foreign keys have been discussed above.

Normalization may be considered as the process of simplifying the database structure to minimize the dependency between tables and to allow for the greatest diversity of database queries. The normal forms are a linear progression of rules that you apply to your database, with each higher normal form achieving a better, more efficient design [3.8].

The basic normal forms are

- first normal form (1NF)
- second normal form (2NF)
- third normal form (3NF).

**Table 3.7** A linking table, *houseOwnerLinksInsurer*, is used to model the many-to-many relationship between *houseOwner* and *insurer* (after [3.8])

| **Table: houseOwner** |
|---|
| IDNumber |
| LastName |
| FirstName |
| Address |
| ZipCode |
| City |

One-to-many $(1 : n)$

| **Table: houseOwnerLinksInsurer** |
|---|
| IDNumber |
| InsurerID |

Many-to-one $(n : 1)$

| **Table: insurer** |
|---|
| InsurerID |
| CompanyName |
| Address |
| ZipCode |
| City |

The higher levels of normal forms, such as the Boyce–Codd normal form and the fourth normal form are less important and are therefore not discussed here.

### First Normal Form

The first normal form (1NF) says that all column values are atomic. This literally means that they are indivisible.

This means that each column contains only one value such as *apple tree* or *plum tree*. Arrays of values are not allowed. This structure is advantageous for easy update or retrieval of data.

The table *garden1* (Table 3.8) violates the 1NF.

It is quite obvious that it would take some programming effort to query the data such as: give me all trees of all species in all gardens. Overall, such programs tend to contain more programming errors than simpler programs.

The same table could be improved by replacing the single Plants column with six columns: Quant1, Plant1, Quant2, Plant2, Quant3, and Plant3 (Table 3.9), but it still violates the 1NF.

The design shown in Table 3.9 is still problematic.

1. A search for specific species would need to read through all columns, e.g., to retrieve the apple trees.
2. The number of different species in a garden is limited to three.

Obviously the table could be expanded to more than three species. However, where is the limit? Any limit might be too narrow for some exceptional cases. Otherwise, if the table is defined large, in most cases it would cause a waste of empty space.

A table in first normal form (1NF) does not have those problems. The table *garden3* in Table 3.10 is in 1NF, as each column only contains one value and there are no groups of repeating columns.

A column headed SpeciesNumber has been added. This column allows the definition of a primary key which could be a composite key of GardenID and SpeciesNumber.

The rules for 1NF can be summarized [3.11] as follows

- Eliminate repeating groups in individual tables.
- Create a separate table for each set of related data.
- Identify each set of related data with a primary key.

### Second Normal Form

A table is considered to be in second normal form (2NF) if it is in 1NF and every nonkey column is fully dependent on the entire primary key. In other words, tables should only contain data that are related to one entity (thing) in the real world.

The table *garden4* in Table 3.11 is a slightly extended version of *garden3* by the SpeciesID. It is in first

**Table 3.8** *garden1* violates first normal form because the data stored in the plants column are not atomic

| Table: garden1 | | |
|---|---|---|
| **GardenID** | **OwnerID** | **Plants** |
| 1 | 4 | 5 Apple trees, 3 Plum trees, 6 Peach trees |
| 2 | 23 | 1 Apple tree |
| 3 | 15 | 2 Rhododendrons, 2 Lilacs |
| 4 | 2 | 15 Wild roses |
| 5 | 23 | 1 Plum tree |
| 6 | 2 | 5 Willows |

**Table 3.9** *garden2*: A better, but still erroneous, version of the gardens table. The repeating groups of information violate first normal form

| Table: garden2 | | | | | | | |
|---|---|---|---|---|---|---|---|
| **GardenID** | **OwnerID** | **Quant1** | **Plant1** | **Quant2** | **Plant2** | **Quant3** | **Plant3** |
| 1 | 4 | 5 | Apple tree | 3 | Plum tree | 6 | Peach tree |
| 2 | 23 | 1 | Apple tree | | | | |
| 3 | 15 | 2 | Rhododendron | 2 | Lilac | | |
| 4 | 2 | 15 | Wild rose | | | | |
| 5 | 23 | 1 | Plum tree | | | | |
| 6 | 2 | 5 | Willow | | | | |

**Table 3.10** The *garden3* table is in first normal form

| Table: garden3<br>GardenID | OwnerID | SpeciesNumber | Quantity | Name |
|---|---|---|---|---|
| 1 | 4 | 1 | 5 | Apple tree |
| 1 | 4 | 2 | 3 | Plum tree |
| 1 | 4 | 3 | 6 | Peach tree |
| 2 | 23 | 1 | 1 | Apple tree |
| 3 | 15 | 1 | 2 | Rhododendron |
| 3 | 15 | 2 | 2 | Lilac |
| 4 | 2 | 1 | 15 | Wild rose |
| 5 | 23 | 1 | 1 | Plum tree |
| 6 | 2 | 1 | 5 | Willow |

**Table 3.11** The table *garden4* is in first normal form. Its primary key is a composite of GardenID and SpeciesNumber

| Table: garden4<br>GardenID | OwnerID | Area (m$^2$) | SpeciesNumber | Quantity | SpeciesID | Name |
|---|---|---|---|---|---|---|
| 1 | 4 | 1303 | 1 | 5 | 32 | Apple tree |
| 1 | 4 | 1303 | 2 | 3 | 2 | Plum tree |
| 1 | 4 | 1303 | 3 | 6 | 41 | Peach tree |
| 2 | 23 | 4075 | 1 | 1 | 32 | Apple tree |
| 3 | 15 | 911 | 1 | 2 | 113 | Rhododendron |
| 3 | 15 | 911 | 2 | 2 | 121 | Lilac |
| 4 | 2 | 2423 | 1 | 15 | 124 | Wild rose |
| 5 | 23 | 892 | 1 | 1 | 2 | Plum tree |
| 6 | 2 | 1550 | 1 | 5 | 152 | Willow |

normal form. Each column is atomic, and there are no repeating groups.

To find out whether table *garden4* meets second normal form (2NF), one has to investigate whether all columns fully depend on the primary key (in the given case, a composite of the columns GardenID and SpeciesNumber).

However, not all columns depend on all parts of the primary key; for instance, the garden owner (column OwnerID) and the garden area (column Area) do not depend on SpeciesNumber. This problem is also indicated by the fact that both values are repeated for GardenID = 1 and GardenID = 3. For this reason *garden4* is not 2NF.

The second normal form can be achieved by splitting *garden4* into two tables. The process of splitting a nonnormalized table into its normalized parts is called decomposition. Since *garden4* has a composite primary key, the decomposition process is easy: one simply puts everything that applies to each *garden* in one table, and everything that applies to each *species* in the second table. The two decomposed tables, *garden0* and *gardenSpecies*, are shown in Table 3.12.

Two thoughts should be considered.

During the normalization process, no information is deleted. This form of decomposition is termed nonloss decomposition, because no information is sacrificed to the normalization process.

The decomposed table can be recombined to one table. This is guaranteed by the common foreign key, in the example GardenID.

The rules for 2NF can be summarized [3.11] as follows.

- Create separate tables for sets of values that apply to multiple records.
- Relate these tables with a foreign key.

### Third Normal Form

A table is said to be in third normal form (3NF) if it is in 2NF and if all nonkey columns are mutually independent.

The dependency that violates the 3NF may be of various kinds, for instance:

A new column is derived from two others. An example could be the price for a garden which is computed

**Table 3.12** The *garden0* and *gardenDetail* tables satisfy second normal form (2NF). GardenID is a foreign key in *gardenDetail* that can be used to rejoin the tables

| Table: garden0 | | | | |
|---|---|---|---|---|
| **GardenID** | **OwnerID** | **Area ($m^2$)** | **Price (€)** | **Soiltype** |
| 1 | 4 | 1303 | 1303 | Mould clay |
| 2 | 23 | 4075 | 4075 | Sandy clay |
| 3 | 15 | 911 | 911 | Clay |
| 4 | 2 | 2423 | 2423 | Loam |
| 5 | 23 | 892 | 892 | Silt-loam |
| 6 | 2 | 1550 | 1550 | Loam |

| Table: gardenSpecies | | | | |
|---|---|---|---|---|
| **GardenID** | **SpeciesNumber** | **Quantity** | **SpeciesID** | **Name** |
| 1 | 1 | 5 | 32 | Apple tree |
| 1 | 2 | 3 | 2 | Plum tree |
| 1 | 3 | 6 | 41 | Peach tree |
| 2 | 1 | 1 | 32 | Apple tree |
| 3 | 1 | 2 | 113 | Rhododendron |
| 3 | 2 | 2 | 121 | Lilac |
| 4 | 1 | 15 | 124 | Wild rose |
| 5 | 1 | 1 | 2 | Plum tree |
| 6 | 1 | 5 | 152 | Willow |

by the product of the area (in one column) and the price per $m^2$ (in a second column). If this price is stored in a third column, then that column depends on the other two and must be updated if the value in one of the others is changed.

Two columns have the same content but are simply coded in a different way. In Table 3.12, the value 32 in the column SpeciesID = 32 is exchangeable with the value Apple tree in the column Name. So this is redundant information in the table.

Redundancy may cause anomalies with insertion, update, and deletion of records. Such types of problems can be avoided if tables are developed towards the third normal form (3NF).

In our example, the table *gardenSpecies* can be further decomposed to achieve 3NF by breaking out the SpeciesID–Name dependency into a *lookup table*, as shown in Table 3.13. As a result we have the new table *gardenDetail1* and the lookup table *species*. The column SpeciesID is duplicated to appear in both tables, as a foreign key in *gardenDetail1* and as a primary key in *species*. This allows an easy join of the two tables using a query.

The rule for 3NF can be summarized as

- eliminate fields that do not depend on the key [3.11].

**Table 3.13** The *gardenDetail1* and *species* tables are in third normal form (3NF). The SpeciesID column in *gardenDetail1* is a foreign key referencing *species*

| Table: gardenDetail1 | | | |
|---|---|---|---|
| **GardenID** | **SpeciesNumber** | **Quantity** | **SpeciesID** |
| 1 | 1 | 5 | 32 |
| 1 | 2 | 3 | 2 |
| 1 | 3 | 6 | 41 |
| 2 | 1 | 1 | 32 |
| 3 | 1 | 2 | 113 |
| 3 | 2 | 2 | 121 |
| 4 | 1 | 15 | 124 |
| 5 | 1 | 1 | 2 |
| 6 | 1 | 5 | 152 |

| Table: species | |
|---|---|
| **SpeciesID** | **Name** |
| 2 | Plum tree |
| 32 | Apple tree |
| 41 | Peach tree |
| 113 | Rhododendron |
| 121 | Lilac |
| 124 | Wild rose |
| 152 | Willow |