UNIVERSITÄT OSNABRÜCK

Building change detection using high resolution remotely sensed data and GIS

Dipl. Eng. Natalia Sofina

Dissertation

A thesis submitted to the Faculty Mathematics-Informatics

of the University of Osnabrueck

In partial fulfillment of the requirements for the degree of Dr. rer. nat.

June 2014

Supervisor: Prof. Dr. Eng. Manfred Ehlers
Institute of Geoinformatic and Remote Sensing
University of Osnabrueck
Osnabrueck, Germany

Co-Supervisor: Prof. Dr. Eng. Peter Reinartz
German Aerospace Center (DLR)
Remote Sensing Technology Institute
Photogrammetry and Image Analysis
Oberpfaffenhofen
Wessling, Germany

Abstract

In recent years, natural disasters have had an increasing impact, involving immense economical and human losses. Remote sensing technologies are being more frequently used for the rapid registration and visualization of changes in the affected areas, providing essential information for damage elimination, as well as the planning and coordination of recovery activities. Numerous methods of image processing have been proposed to automate a detection of changes on the Earth's surface, most of which focus on the comparison of remotely sensed images of the same area acquired at different dates. However, atmospheric influences (e.g. clouds covering the objects of interest) often render the observations ineffective in the optical domain. In addition, the accuracy of the change detection analysis decreases if the images are acquired with different acquisition angles. These situations can be common in the case of sudden catastrophes (e.g. earthquakes, landslides or military actions), when there is no time to wait for the perfect conditions to acquire the data. This study presents a GIS-based approach for the detection of destroyed buildings. The methodology is based on the integrated analysis of vector data containing information about the original urban layout and remotely sensed image obtained after a catastrophic event.

The integrated data processing enables minimizing the influence of the atmosphere and illumination effects, thus improving damage detection. Moreover, the object-oriented GIS technology makes it possible to concentrate on the investigation of specified objects, thereby reducing false alarms due to natural changes that occur around the investigated objects (e.g. seasonal changes of vegetation). Additionally, GIS based change detection analysis produces a tangible end product, namely damage maps.

A new feature ‘Detected Part of Contour’ (DPC) was developed to identify a building’s condition. The basic idea behind the proposed feature is the assessment of building contour integrity. The feature defines a part of the building contour that can be detected in the remotely sensed image, reaching a maximum value (100%) if the investigated building is intact. Furthermore, several features based upon the analysis of textural information are analyzed. Finally, a binary classification of building state concludes the change detection analysis.

The experiments performed during this research indicate that employing a GIS-based analysis for change detection can essentially improve the potential for remotely sensed data interpretation and can be considered a powerful tool for the detection of destroyed building.

The proposed methodology has been developed solely within the Open Source software environment (GRASS GIS, Python, Orange), whose use implies an innovative, flexible and cost-effective solution for change detection analyses.

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Dr. Eng. Manfred Ehlers for his unwavering support and valuable guidance during the research period. I am also very grateful to him for providing a workplace fully equipped with hard- and software, which enabled me to develop and implement all my ideas presented in this thesis. During this research, he supported me in various ways, including a financial base that allowed me to participate in international conferences to exchange experience with other researchers. Special thanks go to Prof. Dr. Eng. Peter Reinartz, the second supervisor, for sincere and objective support and criticism.

I cannot forget to thank all my colleagues for their close cooperation and the wonderful intelligent atmosphere from which I have been able to benefit during the work at the Institute of Geoinformatic and Remote Sensing of University of Osnabrueck. In particular, I would like to thank Thomas Jarmer, Sascha Klonus and Daniel Tomowski for their helpful advices and comments, as well as their help at the beginning of the research by introducing me to the world of remote sensing and digital image processing.

I would like to express my gratitude to the Pool Frauenfoerderung of the University of Osnabrueck for their financial contribution by way of a six-month scholarship.

This work uses Open Source Software projects such as GRASS GIS, Orange data mining software, and Python programming language. I thank the development teams for providing and supporting the software. I am also grateful to Digital Globe company, Google Earth and OpenStreetMap services for providing the data that made this investigation possible.

My great thanks go to my family, my husband Andrey and daughters Sofia and Marina, who entered this world during my PhD study, for their continuing support and persevering belief in my success and endless motivation. I offer them my thanks for understanding, patience and donating the time that I could have spent with them.

I am deeply indebted to my mother Liudmila for standing by and providing encouragement. I am very grateful for her help at the end phase of my PhD, for her long stay in Germany to take care of my children, thereby granting me time to finish this thesis.

Finally, I would like to thank all those people who have influenced my ideas and whose comments helped to improve this work.

Content

List of Figures

List of Tables

List of Appendices

Glossary

AI	Artificial Intelligence
ANN	Artificial Neural Networks
ARI	Area Ration Index
AUC	Area Under the ROC Curve
CERL	Construction Engineering Research Laboratory
CLI	Canada Land Inventory
DEM	Digital Elevation Model
DPC	Detected Part of Contour
ERDAS	Earth Resource Data Analysis System
ESRI	Environmental System Research Institute
FFT	Fast Fourier Transformation
FSF	Free Software Foundation
GIS	Geographic Information System
GLCM	Grey Level Co-occurrence Matrix
GPL	General Public License
GPS	Global Positioning System
GIS GRASS	Geographic Resources Analysis Support System
GS	Gramm-Schmidt
GUI	Graphical User Interface
IDE	Integrated Development Environment
k-NN	k-Nearest Neighbor algorithm
OpenCV	Open Source Computer Vision Library
ODE	Ordinary Differential Equations
OSI	Open Source Initiative
OSS	Open Source Software
PCA	Principal Component Analysis
PCT	Pseudo-colour Transformation
PIL	Python Imaging Library
PSF	Python Software Foundation
PSFL	Python Software Foundation License
SSI	Shape Similarity Index
SVM	Support Vector Machines
TC	Tasselled Cap

1. Introduction

Humans have always been closely connected with the surrounding world. At the turn of the 21st century, humanity is increasingly feeling the problems resulting from life in highly industrialized societies, whereby a dangerous human-induced interference with nature has become distinctly noticeable. Almost daily, mass media reports about different emergencies occur here and there in the world, including industrial catastrophes, natural disasters, military conflicts and acts of terrorism. A catastrophic impact is not restricted by any nationals borders; moreover, it can lead to huge loss of lives and socio-political crises (for example, Chernobyl disaster (1986), the Gulf of Mexico oil spill (2010), the Great East Japan Earthquake (2011)) (International Atomic Energy Agency, 2006; Republic of the Marshall Islands Maritime Administrator, 2011; Johnson, 2011).

The number of disasters and the severity of their consequences have been constantly growing. Large-scale natural disasters (earthquakes, tsunamis, typhoons and floods) cause mass deaths of people accompanied by the destruction of transport and telecommunication infrastructure, as well as paralysing the management and economy of the crisis regions. This complicates the adequate assessment of the situation and making timely decisions. For the organization of an efficient emergency management, it is very important to have a system of a rapid evaluation of the current situation.

Remote sensing technologies enabling the observation of large areas of the Earth without physical contact offer an undeniable advantage over other ways of acquiring geospatial information. They allow for a quick acquisition of actual information in the form of images of the area affected by the catastrophe. Remote sensing has found a wide application for information and documentary support in the emergency response for almost all major disasters that have occurred in the current decade, including the 2010 Haiti earthquake, the 2010 Yushu earthquake and the 2011 Tohoku earthquake (Dong & Shan, 2013). This technology has become an essential tool in post-event damage mapping. Interactive maps of disaster-affected countries created based upon aerial photography and satellite images have become a crucial element for the support of disaster management. More importantly, numerous agencies and organizations all over the world participate in the effective international coordination of efforts intended for the collection and processing of satellite information.

The assessment of destructions can be performed based upon the comparison of pre- and post- event images by means of change detection analysis. However, a reliable automatic change detection analysis based solely on remote sensed images is not always possible, with the accuracy of the analysis strongly depending on the acquisition conditions. The significant improvement of this situation can be reached by additional utilization of vector

information, which is usually available in the form of cadastral data and contains precise description of object geometry and location, as well as attributive information related to it. The combined treatment of vector and raster information can be performed by means of geographic informational systems (GIS), which provide a broad range of instruments for storage, visualization and processing of spatial information of different types. The tandem of GIS – Remote Sensing is a powerful platform for geographical analysis. In this respect, research works directed to the techniques of the analysis bridging remote sensing and GIS have constantly attracted growing interest (Ehlers, 1989; Blaschke, 2010; Chen et al., 2012; Hussain et al., 2013).

1.1 Research Objective

The main goal of this study is to develop a methodology for the detection of changes in urban areas affected by a catastrophic event through the integrated analysis of post-event remotely sensed data and pre-event vector data. The study was performed in the following successive steps:

1. Overview of approaches to change detection analysis related to separate and integrated processing of raster and vector data.
2. Choice of strategy for the implementation and assessment of developed technique within a GIS environment.
3. Based on step 1, the identification of raster and vector information that is relevant for the detection of building states. Choice/development of methods for the extraction of this information for further change detection analysis.
4. Development of change detection technique based on the results of step 3.
5. Implementation of developed technique based upon the results of steps 2, 3 and 4.
6. Assessment of techniques and software developed in steps 4 and 5 by means of application to a number of study cases.
7. Discussion related to the advantages and disadvantages of developed techniques, as well as their further development.

Based on the aforementioned objectives, the structure of this thesis is provided in the section below.

1.2 Organization of the Thesis

The content of the thesis is presented in six chapters, as follows:
Chapter 2 provides an introduction to change detection, remote sensing and GIS technologies. It describes the properties and handing of the raster and vector data types. In

particular, attention is focused on integrated data processing and urban change detection analysis.

Chapter 3 discusses the motivation for an application of Open Source Software and describes the construction of a testing and implementation environment for the developed change detection method.

Chapter 4 deals with detailed description of the developed methodology, including the processing of vector data, image enhancement, extraction of valuable information and classification.

In Chapter 5, the efficiency of the developed change detection method is experimentally demonstrated based upon data sets from several study areas. Finally, the discussion of advantages and disadvantages of the method developed during the research work is provided.

Chapter 6 briefly summaries the findings and conclusions of this research work. In closing, recommendations for further improvement of the efficiency and reliability of the developed change detection technique are given.

2. Remote Sensing and GIS in Change Detection

The recent progress in computer hardware, software development and space technology has enabled the creation of new highly effective methods for the gathering, analysis, manipulation and visualization of information on the Earth surface, with these new approaches drastically changing the work with cartographic data.

2.1 Geographic Information Systems

Methods of computer data processing have been markedly improving during the last decade, and are now becoming routinely applied in many areas of science and engineering. It has become customary to see documents, tables, diagrams and schemes on a computer's display. Spatial data representation on digital maps has justifiably received widespread attention in various fields of engineering, knowledge management, business operations, commerce and everyday life. This became possible due to the development of geographic information systems (GIS) providing a principally new approach of geodata processing. GIS present a digital model of the geospatial data and enable storing, managing, updating, analysing and graphically presenting the spatial information. This technology combines traditional data base operations, such as client requests and statistical analysis, with a visualisation system and conventional map based spatial analysis. GIS are widely used in a range of applications including the analysis and prediction of natural processes, understanding and interpreting the major factors influencing the processes and their possible consequences.

GIS present the graphic information as a collection of thematic layers with the data separated into different layers depending on their actual thematic content. This concept derives from the principles of classical cartography, where the basic map was overlaid by transparent films with various thematic patterns (hydrology, transportation, soils, geology, land cover etc.). Moreover, GIS contain attribute information concerning each object, which is stored in a database or table and can be returned to the user when provided with a suitably formulated request. Modern GIS include numerous mathematical tools and integrated programming languages. This renders GIS not only a powerful tool for storing and presenting large amounts of multi-source information, but also for spatial analysis, monitoring and prognostics.

Given that a computer system only deals with digital information, paper-based maps were transformed into digital format at the beginning of GIS development, which was a very time consuming process. Digitizing was performed manually by the input of object description (coordinates and attributes) or semi-automatically by applying tracing software to scanned images or using digitizer devices to capture coordinates on the paper-based maps.

Nowadays, the data is inputted directly in digital form as images from remote sensing devices, with the objects information obtained by an automatic analysis (image segmentation techniques, object recognition) or manual input (coordinates, attribute and topological information).
Information transformed into the digital format is represented by two general data models: vector data model and raster data model.

2.2 Remote Sensing

Remote sensing is a combination of methods for collecting information about objects, areas of the Earth's surface or phenomena without physical contact with the objects. Remote sensing data can be used for the investigation of the object's geometrical, physical and chemical properties. The natural forms of remote sensing are human sense organs – eyes, ears or nose. For the analysis of constructions, vegetation and areas that cover the Earth's surface, remote sensing methods based on recording and interpreting measurements of the electromagnetic radiation reflected or emitted from objects in the Earth's surface are applied. Photography is one way to capture the current pattern of electromagnetic radiation.

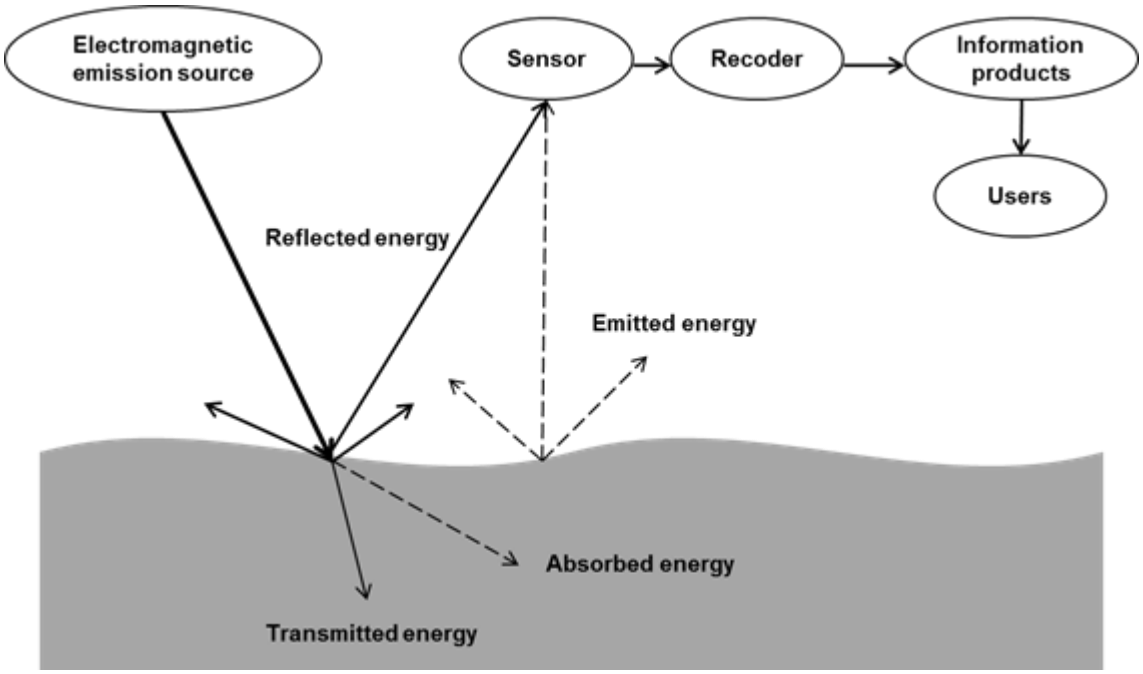


Figure 2.1: General principle of remote sensing (adapted from Chandra & Ghosh, 2006).

Current methods of the remote sensing are generally based on the usage of sensor systems located on spacecraft or aircraft. Sensors can be divided into two groups: active and passive devices. Active sensors (e.g. laser scanners) emit the electromagnetic radiation and detect the reflected part. By contrast, passive sensors (e.g. cameras) register a radiation reflected

from the objects or an object's natural radiation (in the thermal domain), and are the most frequently used. Figure 2.1 shows a general scheme of remote sensing.

The detected energy is converted into digital image data consisting of a two-dimensional rectangular grid of numerical values representing different brightness levels. Each value corresponds to a fixed square area on the Earth and is equal to the average brightness level of detected radiation from this area. The grid is commonly called 'raster' and the square units or 'pixels' can be subdivided into pure and mixed pixels. The square area of a pure pixel includes only one object type, whereas the mixed pixel corresponds to an area with objects of different types. The value of the pure pixel facilitates conclusion about the object type, while the interpretation of mixed pixel values can be quite complex.

A remotely sensed image is characterised by its spatial, spectral, radiometric and temporal resolution. The spatial resolution is a measure of the spatial detail in the image, and represents the area size corresponding to a single pixel.

Spectral resolution is the ability of a remote sensing system to distinguish different ranges of wavelengths. The image produced by a sensor system can consist of many bands corresponding to different wavelength ranges. The finer the spectral resolution the narrower the wavelength range represented by the particular band. Many remote sensing systems use multi-spectral sensors to record radiation energy over several separate wavelength ranges. Advanced multi-spectral sensors, called hyperspectral sensors, can detect hundreds of very narrow spectral bands throughout the visible, near-infrared, and mid-infrared portions of the electromagnetic spectrum. Their very high spectral resolution facilitates a fine discrimination between different targets.

In order to obtain digital brightness value, the continuous signal is subdivided into several levels, with the number of the detected brightness level saved as the pixel value. Radiometric resolution defines a number of used brightness levels, and depends on the number of information bits assigned to a single pixel.

Temporal resolution represents the ability of a remote sensing system to collect imagery of the same area of the Earth's surface at different periods of time. Radiation spectrum may change over time, and such changes can be detected by collecting and comparing multi-temporal imagery. The analysis of multi-temporal images enables monitoring naturally occurring changes on the Earth's surface (such as changes in natural vegetation cover or flooding) or those induced by humans (such as urban development or deforestation).

Since the data obtained from space-based systems cover large areas of the Earth's surface, they are effectively used for case studies of both a global and regional scope. Regularly obtained satellite images enable the monitoring of water recourses, agricultural productivity

and urban developments, as well as the investigation of objects and phenomena undergoing changes due to natural and anthropogenic factors.

Remote sensing techniques are especially effective for data gathering from areas that are difficult to access, where the situation analysis is particularly important for prevention and recovery activities involving natural and human-caused disasters.

Remote sensing is used in many fields, such as agriculture and forestry, geography, geology, archaeology, meteorology and climatology, monitoring of marine environment, water resources management, and change detection. Table 2.1 summarizes selected possible application areas of remote sensing in detail.

Agriculture and forestry	Land-use	Geology	Water-resources	Oceanography	Environmental researches
Identification of vegetation types	Land-use classification	Identification of rocks and minerals	Water boundary mapping	Investigation of living organisms	Supplementary exploration and development of new fields
Crop control and monitoring the crop state	Mapping and map update	Mapping of geological structures	Flood mapping	Stream mapping and estimating the water turbidity	Mapping of atmospheric pollution
Forest inventory	Land-use suitability assessment	Updating of geological maps	Mapping and estimation of snowline	Coastal Mapping	Monitoring the atmospheric pollution
Estimation of plant biomass	Urban and land territory planning	Mapping of friable deposits	Glaciology	Mapping of shallow water	Damage assessment caused by natural disasters
Monitoring vegetation cover	Regional planning	Analysis of volcanic intrusions	Water quality monitoring, water bloom prediction	Analysis of ice conditions	Anthropogenic influence on environments
Monitoring changes in soil covering	Transport network analysis	Relief mapping	Delineation of irrigation fields	Wind speed or wave heights	

Table 2.1: Application areas of remote sensing.

Remotely sensed data must be georeferenced for comparison with each other or for interpretation in relation to other geodata. Moreover, remotely sensed images are usually processed by mathematical algorithms, which help to reduce noise influence or to enhance or suppress certain image features. GIS provide numerous possibilities for such processing

and are widely used as integrated systems for management, processing, and visualization of remote sensing data.

2.3 Historical Overview of Remote Sensing and GIS Integration

Rather than being a recent phenomenon, the spatial location of physical objects has always been interesting for people. Primitive people already took an interest in geography via the expansion of their living space in search of the most suitable areas for hunting and gathering. Nowadays, dependence on information about what and where things are located is rapidly increasing. For instance, Google Maps service is intensively used in order to find restaurants, hotels, and other points of interest. Moreover, modern in-car navigation systems provide the most efficient route and increasingly replace traditional paper-based maps.

Information about the spatial description of objects on the Earth's surface is termed 'geoinformation'. Throughout recent decades, large amounts of spatial information have been stored in digital form and processed on computers. Subsequently, geo-data gathering, storing and processing have been consolidated in a new discipline – 'geoinformatics', which covers a wide range of different derivative scientific directions and enables integration of conventional and digital cartographic techniques.

The first significant step toward the integration of geo-sciences can be traced back to the 1960s and 70s starting with the computer implementation of a powerful arsenal of mathematical methods and statistical techniques in cartography. It became evident, that such new data analyses should be included in the range of cartography's interests as a new discipline of 'solving cartographic problems' entitled 'Analytical Cartography' (Tobler, 1959).The focus on a new technique of map production by means of mathematical analysis marked analytical cartography as very different from the conventional definition of traditional hard copy cartography. New feasibilities including neighborhood analysis, map generalization, shape analysis, map overlay, expanded dramatically the scientific capacity of classic cartography.

Another aspect of the integration is associated with the extremely rapid development of computer and information technologies, which has considerably promoted the extension of cartography's conception. New products such as cartographic databases, digital images and digital terrain models were generated and by the end of the 1960s, studies addressing mapping and modeling of spatial digital data had become widespread. The first true operational 'Geographic Information System' (GIS) was developed in the 1960s in a project headed by Roger Tomlinson, known as "father of GIS". This system improved the 'computer mapping' applications and was used to store, manipulate and analyze data collected for the Canada Land Inventory (CLI). However, due to the high cost of the hardware, the new

approaches for map production did not become advantageous in comparison to the manual systems until the 1980s (Tomlinson, 1985). With the continued development of smaller, faster and cheaper computers, various GIS software had emerged by the end of the 20th century: ERDAS (Earth Resource Data Analysis System), ESRI (Environmental System Research Institute), Intergraph etc.

The next step towards the integration of geosciences was associated with the intensive development of satellite technologies with remote sensing methods of data collection from the Earth's surface being commonly used in recent decades. This data gathering is performed by devices (or cameras) located away from the area of interest by using aircraft or satellites, with the information provided by sensors yielded in the form of digital images. Such remotely sensed images became considered as maps, which led to the development of new sophisticated methods of image interpretation including combination of photogrammetric and image processing.

The application of remotely sensed imagery in geology (identification of rock type, mapping faults and geological structure), meteorology (estimation of atmospheric temperature, pressure and wind velocity), agriculture (monitoring of biomass of land vegetation), disaster warning and assessment (monitoring of floods and landslides, assessment of damaged zones from natural disasters), ecology (planning of ecological zones, monitoring of deforestation and urban land use) etc., bear testimony toits high economic efficiency.

A rather recent step towards the integration of remote sensing and GIS technologies is due to the rapid development of space-based satellite positioning systems (e.g. GPS) providing accurate location and time information about objects on or near the Earth's surface. The first satellite navigation system was developed to coordinate underwater positions for U.S. Navy submarines in the 1960s. Thanks to a precise and accurate detection of object location, the technology brought an end to major errors in navigation of sea vessels. Nowadays, GPS devices can be found everywhere – they are used in cars, boats and even in cellular phones.

The integration of GIS and remote sensing was preceded by a long time period, in which many innovative products emerged, with the development of GIS software, cost-effective computers and scanner technologies, and powerful methods of image processing playing a decisive role in this eventual integration. The following main technological steps of the integration can be emphasized:

1. Mathematization and computization of cartography.
2. Development of remote sensing technologies.
3. Development of geoinformatics technologies.
4. Development of GPS.

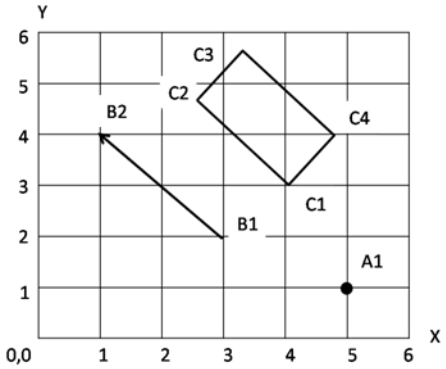
The integration of cartography, geoinformatics and remote sensing has simultaneously affected all branches of Earth sciences, economic and socio-economic studies.

2.4 Spatial Data Models

There are two fundamental approaches to the digital representation of geographic information: the vector data model and raster data model. Accordingly, definitions, properties, advantages and disadvantages of raster and vector data models with regard to change detection analysis are discussed in this chapter.

2.4.1 Vector Data Model

The vector data model, or object-based model, describes geographic objects by the specification of their boundaries, coordinates and attribute information. It should be noted that the vector model represents the real world in a similar way as maps. There are three main elements depicting objects in the vector data model: points, lines and polygons. Points represent geographic objects that are very small with respect to the considered map scale, which can include a meeting point, traffic light, bus station, etc. The points have no extent in any directions and consequently refer to zero-dimensional objects. Lines are one-dimensional objects having length, but no width; and can represent, for example, rivers and roads. Polygons form bounded areas referring to two-dimensional objects having length and width, and can represent objects such as lakes, buildings, agricultural fields.



Point	X	Y
A1	5,0	1,0
B1	3,0	2,0
B2	1,0	4,0
C1	4,0	3,0
C2	2,6	4,7
C3	3,3	5,6
C4	4,8	4,0

B = {B1, B2}

C = {C1, C2, C3, C4}

Figure 2.2: Presentation and description of objects in the vector data model.

All vector objects are built based on a sequence of points, or vertexes, each of which has a set of exact coordinates (Figure 2.2). Each point is represented by a position vector connecting the reference origin with the point. Based on the position vectors and using the operations of vector algebra, different 1D and 2D object such as lines or polygons can be built. This data representation is called the 'vector data model'.

Evidently, the objects cannot be fully described only by a set of the position vector coordinates. The whole mathematical description of the object geometry implies location, coordinate system, object topology, Euclidean metric (definition of distance between two points) and neighbourhood relations (topology). Topological properties describe relations between the distinct vector objects, such as connections and neighborhood. For example, in order to find the fastest route from A to B, the topological information about streets and their intersections is exploited.

2.4.2 Raster Data Model

Remote sensing systems record measurements of electromagnetic radiation in the form of raster image data. The raster data model is more akin to a photo than a map. If a raster image is viewed through a magnifier, a rectangular grid of cells (or pixels) with individual colour values can be seen (Figure 2.3).

125	116	121	124	121
146	145	134	140	153
167	183	177	164	159
148	151	167	174	167
101	92	117	153	168

Figure 2.3: Raster image structure.

A pixel is the simplest element of the raster image, and corresponds uniquely to an area on the Earth's surface. In order to define the pixel's position, a coordinate system in the raster space is used. The position of each pixel is determined by a unique address consisting of row number and column number in the rectangular raster grid (Figure 2.4).

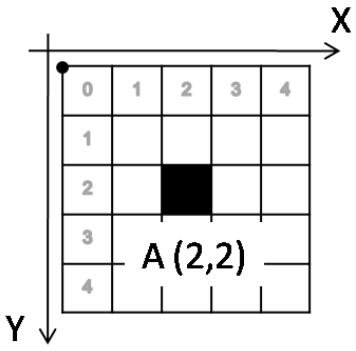


Figure 2.4: Pixel location in the rectangular raster matrix.

Using the upper-left or lower-left corner of the raster grid as a reference point, the raster image can be georeferenced to the Earth surface by specifying the corresponding coordinate system. In the georeferenced image, the geographic location of the image pixel areas and distance between them can be determined.

Each pixel can be assigned several values indicating a brightness level of corresponding remote sensed data, type of pixel area, relief elevation, result of mathematical image processing, etc. If the pixel has several values, the image consists of several rectangular raster grids– bands. Each band contains only one value per pixel, which is represented in the image by so-called grey values.

The maximum number of colours, or grey tones, in the case of a panchromatic image, that can be simultaneously represented is defined by radiometric resolution. This parameter is one of the most important characteristics of the raster image and is equal to a number of bits, which defines how much computer memory is required to store one pixel of the image.

The computer memory required to store an image depends on the number of bands in the image. For example, an RGB-image has three bands corresponding to the radiation brightness in the red, green and blue colour spectrum. If each band contains 8-bit values, the pixel storage requires 8 bits × 3 = 24 bits.

Generally, the sensors installed on the remote sensing satellites have the radiometric resolution of at least 8 bits per pixel (Table 2.2). There are sensors with higher radiometric resolution, for example, 11 bits for WorldView, IKONOS and QuickBird, and 16 bits for the EO-1.

Bit Depth	Number of grey levels
6 bits	64
8 bits	256
11 bits	2 048
14 bits	16 384
16 bits	65 536

Table 2.2: The most commonly used pixel depths and associated number of grey levels.

2.4.3 Integrated Data Analysis

Vector data is often considered as a passive aid to image analysis, typically for geo-referencing, image segmentation or data conversion. Handling of vector and raster data is still predominantly separate with the complexities of integrated data analysis arising from differences in data structure and the conceptual philosophy of data processing. Vector data

provides an accurate geometric representation of objects (buildings, roads or agricultural field boundaries). In many cases, such as cadastral data processing or infrastructure planning, it is particularly important to precisely represent the object's boundaries. For the monitoring of building states, the ability to extract and analyse a single object from vector data represents an essential feature. Vector graphics are easily scalable and can be transformed to any other cartographic projections without loss of quality due to their mathematically defined geometry.

The vector data model provides the means to manipulate descriptive attribute information concerning each object on the map. However, it is associated with an initial complexity of the data base creation, as both the data base filling and building of a vector object and its topology can be very time-consuming. However, this information makes it possible to operate thematically selected objects (object-based analysis) and their relations with neighbouring objects. Additionally, vector data can be graphically symbolized according their attribute characteristics; for example, line objects, such as rivers or roads, can have different width and colours.

However, vector data representation becomes disadvantageous for the modelling of diffusion processes, such as temperature distribution along the surface or urban air pollution. Furthermore, there are spatial objects with a rather high level of detail that cannot be represented using the vector format. Therefore, for objects with numerous essential small details and data that has a continuous distribution, the raster data model is considered more suitable. For instance, information about atmospheric pressure and cloud cover, or digital elevation models are difficult to represent with vector graphics. However, in contrast to the vector graphics, the analysis of the raster data requires a large memory capacity for data storage.

Any object in the raster data model consists of a set of pixels. Therefore, line objects are built from a series of pixels and at a certain magnification appear as a sequence of squares, which is not always convenient. The pixel size does not usually conform to the size of the real object, and thus the representation of continuous objects by a sequence of discrete pixels is only possible with a certain approximation. The accuracy of real object representation depends on the relation between pixel and object sizes.

Preference was given to the modelling of the raster data at the beginning of GIS development since the vector graphic tools and algorithms for data modelling were not yet sufficient to cover user's needs. With the further developments of both the hardware and GIS-software, the vector data presentation of geographic objects has been generally prevailed over the raster data model due to the reduced requirements of computer capacities and compatibility with principles of engineering graphics. However, the raster data format is

still used in remote sensing as an essential part of this technology. The raster data representation and modelling are not only widely used, but have also recently achieved considerable progress in GIS applications. While the purported role of raster images was only a passive layer for decoration of an output map in the past, they are now used as an important source of spatial information. Modern methods of digital image collection, such as aerial photos or high-resolution satellite images, allow for regular data obtaining from large areas at frequent intervals, thereby providing actual geodata for detection of changes in the Earth's surface. Consequently, this data is usually used as real-time information. The vector data is usually obtained by digitizing existing maps when an operator traces points, lines, and polygon boundaries from a map. The vector information contains detailed data related to the objects on the map, yet the update of this content can be tedious and time-consuming. The combined analysis of raster and vector data enables analysing the state of each single object separately by using real-time or the most actual information in raster format and comprehensive information in vector format for the past state.

The comparison of vector and raster data is provided in Table 2.3.

	Advantages	Disadvantages
Vector model	• High geometric accuracy • Detailed object description • Small volumes of data • Similar data representation with traditional paper maps • Selective data analysis • Flexible management of attribute information	• Complex data structures • Time-consuming process of geometry and topology compilation • Complex and compute-intensive mathematic and logic operations (overlay operations) • Parallel description of object geometry and topology
Raster model	• Easy data structure • Simple geometry and topology compilation • Compatible with remotely sensed and scan data • Easy updatable • Simple implementation of overlay and intersection operations • Simple implementation of logical operations	• Inaccurate recording objects form and position • High compute and storage investment • Accuracy of data representation depends on the pixel size • Complicated coordinate transformation

Table 2.3: Comparison of vector and raster data models (de Lange, 2006; Ehlers & Schiewe, 2012).

2.5 GIS-Based Change Detection

Remote sensing is an excellent instrument for monitoring a current situation. However, a large amount of incoming data can only be efficiently used through the application of automatic or semi-automatic techniques for data processing. Data derived from remote sensing are increasingly being used as a data source in GIS. For this purpose, numerous methods and approaches have been adopted for the analysis of multi-temporal vector and raster data in recent decades.

2.5.1 Definition and Application of Change Detection

The term ‘change’ is usually associated with a transformation or modification of something over an extended period of time. As such, change detection aims at an identification of changes occurred in objects or phenomena by observing them at different times (Singh, 1989). This analysis yields a quantitative assessment of the differences in the state of the investigated objects by interpretation of multi-temporal data. Furthermore, it is not only important to determine the fact that the change exist, but also the location, dynamics and kind of change.

Increasing changes of the environment through long-term natural and seasonal climate change, ecological and geomorphologic processes, as well as human-induced changes of the Earth’s surface have led to a wide range of problems. The ability to detect changes is very important for human life, in order to correctly understand relationships between man-made and natural events on the Earth to assist further decision-making. Change detection is one of the major applications of remote sensing, given its feasibility to obtain actual information from large areas of the Earth’s surface in relatively short periods of time and with an appropriate image quality (Ingram et al., 1981; Singh, 1984). Therefore, remotely sensed data deliver valuable information for solving social, economic and ecological problems. The increasing spatial dynamics of urban development have resulted in environmental disturbances. Mas (1999) considers the growing need for remote sensing employment in land use analysis and evaluation. Application of remote sensing techniques for observation, analysis and assessment of phenomena on the Earth’s surface presents a very wide spectrum of data and methods. The ability of remote sensing to capture large land areas with a high temporal and spatial resolution represents its unique feature, finally enabling an automatic classification of land cover or the detection of changes: Albertz, 2001; Briger, 2002; Gutman, 2004; Jensen, 2007; Lu et al., 2004. Coupled with remote sensing technology, change detection is widely used in a large range of applications such as land use and land cover analysis, monitoring of urban land use, estimation of deforestation or forest

fires, monitoring of biomass of land vegetation, monitoring of floods and landslides, and the assessment of damage zones from natural or man-made disasters (Table 2.4).

Thematic application	Review
1. Land-use and land-cover change	Adams et al., 1995; Alves and Skole 1996; Gautam and Chennaiah 1985; Green, Kempka and Lackey 1994; Dewidar 2004; Kaufmann and Seto 2001; Petit and Lambin 2001; D. A. Roberts, G. T. Batista and J. G. Pereira, et al. 1998; and Sommer, Hill and Megier 1998.
1.1 Forest or vegetation change	Coppin, Nackaerts, Quenn, & Brewer, 2001; Graetz, Pech, & Davis, 1988; Köhl, Magnussen, & Marchetti, 2006; Sohn, Moran, & Gurri, 1999; Vogelmann, 1988; and Cihlar, Pultz, & Gray, 1992.
1.2 Forest defoliation and damage assessment	Nelson, 1983; Vogelmann, 1989; Price, Pyke, & Mendes, 1992; Gopal & Woodcock, 1996; Radeloff, Mladenoff, & Boyce, 1999; and Rigina, Baklanov, Hagner, & Olsson, 1999.
1.3 Forest fire	Jakubauskas, Kamlesh, & Mausel, 1990; Elvidge, Pack, Prins, Kihn, Kendall, & Baugh, 1998; Fuller, 2000; Cuomo, Lasaponara, & Tramutoli, 2001; Garcia-Haro, Gilabert, & Melia, 2001; and Bourgeau-Chavez, Kasischke, Brunzell, Mudd, & Tukman, 2002.
1.4 Wetland change	Jensen, Cowen, Althausen, Narumalani, & Weatherbee, 1993; MacLeod & Congalton, 1998; and Michener & Houhoulis, 1997.
1.5 Landscape change	Zhen, Wallin, & Hao, 1997; Franklin, Dickson, Farr, Hansen, & Moskal, 2000,and Taylor, Brewer, & Bird, 2000.
2. Urban change	Li & Yeh, 1998; Ward, Phinn & Murray, 2000; Liu & Lathrop, 2002; Agouris, Stefanidis & Gyftakis, 2003; Gautama, Haeyer & Philips, 2006; Howarth & Wickware, 1981; Armour, Tanaka, Ohkura & Saito, 1998; Peters, Walter-Shea, Lei, Vina, Hayes & Svoboda, 2002.

Thematic application	Review
2.1 Catastrophic damage	Kumar, Chingkhei & Dolendro, 2007; Lupo, Reginster & Lambin, 2001; Stramodo, Bignami, Chini, Pierdicca & Tertulliani, 2006; ; Wikantika, Sinaga, Hadi & Darmawan, 2007; and Tomowski, 2011.
3. Other applications	
3.1 Crop monitoring	Manavalan, Kesavasamy, & Adiga, 1995.
3.2 Road change	Agouris, Stefanidis, & Gyftakis, 2003; and Gautama, Haeyer, & Philips, 2006.

Table 2.4: Change detection applications using remote sensing technologies (Lu et al., 2004).

The main reason for the employment of remote sensing technologies for change detection can be briefly described as follows. Changes in the state of the observed object provoke differences in its natural radiation, which can be registered by remote sensing as changes of radiance values and recorded in form of satellite imagery (Ingram et al., 1981). The differences in natural radiance should be markedly larger than radiance changes caused by seasonal, weather or acquisition factors. The effectiveness of the change detection analysis depends on the accuracy of the geometric registration and atmospheric correction or normalization between the multi-temporal images being processed. To reduce such effects, the images for comparison should be acquired by the same sensor, at the same time of a season and the same time of a day. The implementation of an appropriate method is also one of the crucial factors in producing a good change detection result (Lu et al., 2004). However, it is difficult to select the most effective algorithm of change detection for all specific problems. The next chapter provides an overview of the frequently used change detection techniques and their relevant applications.

2.5.2 Review of Change Detection Methods

Following the emergence of remote sensing, many methods of change detection have been developed and examined for numerous purposes. Comparative reviews of change detection methods based on the analysis of multi-temporal remotely sensed data are provided in Briger, 2002; Coppin et al., 2004; Singh, 1989; or Lu et al., 2004.

The selection of change detection method must be performed depending on the purposes of the investigation. Thus, different change detection approaches are required for detection of destroyed buildings and for land-use analysis.

Niemeyer and Nussbaum (2006) categorized the existing methods according to their flexibility, robustness and applicability, furthermore incorporating technical aspects within the categorization: automatic or visual, pixel-based or object-oriented, bi- or multi-temporal, and by the capability to define the art of the change.

Mas (1999) subdivided existing methods into the following subject groups: image differencing, vegetation index differencing, selective principal component analysis, multi-date classification, post-classification analysis and image enhancement.

In this work, the classification based on (Lu et al.,, 2004) is described, whereby the author groups a wide range of change detection methods into seven groups: (1) algebra, (2) transformation, (3) classification, (4) advanced, (5) geographic information system (GIS) approaches, (6) visual analysis, and (7) other change detection techniques.

1) Algebra

This group includes image differencing, image ratio, image regression, vegetation index differencing and change vector analysis.

Image differencing

Images taken on two different dates are subtracted pixel-by-pixel from another in this technique.

$$Dx_{ij}^{k} = x_{ij}^{k}(t_2) - x_{ij}^{k}(t_1) + c \qquad (2.1)$$

Where x_{ij}^{k} is the pixel value of band k for pixel at row i and at column j; t_1 and t_2 denote the acquisition time; c is a constant to provide positive result values.

If the pixel values in two images are close, the pixel values of the resulting image are near zero. By setting a value larger zero as a threshold the changed areas are selected.

Image differencing is a very simple technique, offering a straightforward implementation and interpretation of the results. However, the resulted values Dx_{ij}^{k} are absolute and the same value can be interpreted variously. Moreover, the efficiency of method can be strongly affected by atmospheric effects.

Price et al. (1992) make use of image differencing for detection of shrub dieback. However, they found rather poor agreement with field verification. Green et al. (1994) report better results of image differencing of a single band in contrast to differences in a vegetation index. Saksa et al. (2003) apply image differencing for detecting clear cut areas in boreal forest. Three methods were tested using Landsat satellite imagery and aerial photographs. In the first method, they calculated difference image using pixel-by-pixel differencing. Then, they

segmented the image to delineate the clear cut areas. In the second method, pixel block-level differencing, they included neighbouring pixels into the calculation of the difference image, thereby reducing negative effects of misregistration in the resulting image. In the third method, they first segmented the images. After that, they calculated a segment-level image difference and labeled clear cut areas by using an unsupervised classification. They found that the image differencing in pixel blocks or in the segments provides better results with a smaller number of misinterpreted areas.

Image regression

This algorithm is based on the assumption that the pixel values of the first image $x_{ij}^k(t_1)$ are linearly related to the pixel values of the second $x_{ij}^k(t_2)$ (Singh, 1984). The regression function can be expressed as:

$$\hat{x}_{ij}^k(t_2) = ax_{ij}^k(t_1) + b \tag{2.2}$$

Here, $x_{ij}^k(t_2)$- the pixel values of the second image, $\hat{x}_{ij}^k(t_1)$- the expected pixel values of the first image in the absence of changes.

The values a and b can be found by using the least square method. Thereby, the residual is defined based on difference of the first image pixel values $x_{ij}^k(t_1)$ and the regression function $\hat{x}_{ij}^k(t_2)$.

$$\min_{a\,b} \sum_{ij} (x_{ij}^k(t_1) - ax_{ij}^k(t_2) - b)^2 \tag{2.3}$$

A change image can be obtained through the subtraction of the two images after applying the linear regression function to the second image.

$$Dx_{ij}^k = \hat{x}_{ij}^k(t_2) - x_{ij}^k(t_1) + c \tag{2.4}$$

The changed area can be marked out using suitable threshold value. This technique enables reducing the influence of atmospheric conditions on the results of change detection analysis. However, the selection of an appropriate regression function and thresholds between change and no change areas can be rather challenging and clearly influences the final result (Theau, 2012).

Image rationing

In this technique, images from different times are divided on a pixel-by-pixel basis.

$$Rx_{ij}^{k} = \frac{x_{ij}^{k}(t_1)}{x_{ij}^{k}(t_2)} \tag{2.5}$$

If the pixel values of two images are close, the pixel values of the resulting image reach a value close to 1. The pixels corresponding to the areas of changes have ratio values significantly greater or less than 1. Therefore, the technique requires definition of two threshold values for separation of values greater and less than 1.
The method enables reducing the impact of Sun angle, shadow and topography (Lu et al., 2004).

Vegetation Index Differencing

The vegetation index is calculated for each pixel based on the pixel values from different bands, highlighting a particular property of vegetation. The change image is a subtraction result of the index values of considered images. Detection of changes is performed based on the suitable threshold and the obtained change image. There are many approaches for calculations of vegetation indexes. One of the most frequently used is NDVI (normalized difference vegetation index).

$$x_{ij}^{NDVI} = \frac{x_{ij}^{NIR} - x_{ij}^{RED}}{x_{ij}^{NIR} + x_{ij}^{RED}} \tag{2.6}$$

Here, x_{ij}^{NIR}- pixel value from a band corresponding to infrared spectrum, x_{ij}^{RED}- pixel value from a band corresponding to red spectrum.
The application of the vegetation index difference method enables reducing the influence of topography and illumination however it leads to an increase random or coherence noise (Lu et al., 2004).

Change vector analysis

The method based on the comparison of vectors built of pixel values from different bands. The vector corresponding to the pixel can be expressed as:

$$V_{ij} = \left(x_{ij}^{1}, x_{ij}^{2}, \ldots, x_{ij}^{n}\right) \tag{2.7}$$

Performing subtraction of the vector of considered images, the spectral change vector is obtained as:

$$\Delta V_{ij} = V_{ij}(t_2) - V_{ij}(t_1) = \left(x_{ij}^1(t_2) - x_{ij}^1(t_1), x_{ij}^2(t_2) - x_{ij}^2(t_1), \ldots, x_{ij}^n(t_2) - x_{ij}^n(t_1)\right) \quad (2.8)$$

The change magnitude is calculated as the Euclidian norm of the vector ΔV_{ij}:

$$||\Delta V_{ij}|| = \sqrt{\left(x_{ij}^1(t_2) - x_{ij}^1(t_1)\right)^2 + \left(x_{ij}^2(t_2) - x_{ij}^2(t_1)\right)^2 + \ldots + \left(x_{ij}^n(t_2) - x_{ij}^n(t_1)\right)^2} \quad (2.9)$$

Naturally, the unchanged pixels have a change magnitude close to zero. In general cases, the threshold for the detection of change areas can be defined as the boundary of a region in n-dimensional vector space of the change vector.

Based on the magnitude and direction of change vector, the approach not only enables detecting changes, but also to define the type of changes (Tomowski, 2011). This technique is commonly used for land-use or land-cover analysis.

The algebra methods are rather simple to implement. However, it is somewhat challenging to select a suitable threshold to detect the change areas. The result of change detection particularly depends on the accurate geometric registration and radiometric normalization of the input data. As all of these methods are based on a pixel-by-pixel comparison, a misregistration leads to false results due to a mismatch between the studied pixels.

2) Transformation

This category includes principal component analysis (PCA), Tasseled cap (TC), Gramm-Schmidt (GS) and Chi-square transformations. The most popular methods, namely PCA and TC, described here.

Principal Component Analysis

This technique uses a linear transformation of the band components into the principal components. The new components better enable highlighting uncorrelated data corresponding to changes, and thereby the information concerning changes is almost entirely presented by only a few new components. Thus, the set of considered components can be reduced for further change detection analysis.

Change detection using PCA can be performed in the two following ways.

1. The considered images are merged into one image with doubled number of bands, with the obtained component set transformed into principal components and the few minor components used for change detections. An example of such an approach is shown in Figure 2.5. The one band from each compared image is merged into one image with two bands. Thus, two components T1 and T2 from each band correspond

to each pixel. Applying PCA, the components T1 and T2 can be replaced by the principal components PC1 and PC2. Separation of objects can be performed using only the value of component PC2.

2. PCA is performed separately for each considered image. The obtained images are subtracted from each other, and the resulted image is analysed.

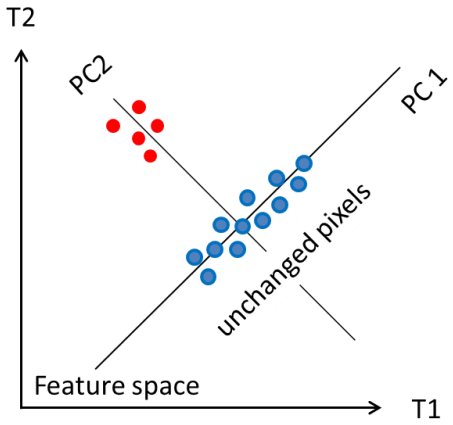


Figure 2.5: PCA for change detection (Tomowski, 2011).

Mas highlighted that PCA allows for the reduction of sensor and atmospheric condition (Mas, 1999). However, the major weakness of this technique lies in the difficulty of result interpretation.

Tasseled Cap (TC)

The 'Kauth-Thomas –Tasseled Cap Transformation' (Kauth & Thomas, 1976) was developed particularly for agricultural applications. The method is similar to the PCA and is widely used in studies of forest, ecology and landscape. The basic idea behind the Tasseled Cap transformation is a compression of multispectral Landsat data into three components: brightness, greenness and wetness. The new components are used for further change detection analysis.

Nordberg & Evertson (2005) apply image differencing to create a change image from each transformed image pair. They detected changes in greenness due to vegetation removal and in wetness because of increased soil exposure.

The main advantages of transformation methods include reducing data redundancy between bands and highlighting the different information in the derived components. However, it is rather difficult to interpret the information in the transformed images (Lu et al., 2004).

3) Classification

The following methods belong to this group: Post-Classification comparison, spectral-temporal combined analysis, expectation-maximization (EM) detection, unsupervised change detection, hybrid change detection, and artificial neural networks (ANN).

Post-Classification Comparison

To perform this method, multi-temporal images are initially classified independently from each other, with new thematic maps are generated based on the classification, and the new maps finally compared on a pixel-by-pixel basis. The benefits of this approach include a reduction of atmospheric, environmental and sensor caused differences between the multi-temporal images. Moreover, this method enables detecting the changed area and identifying the type of change. However, accuracy of the change detection strongly depends on the accuracy of each individual classification of multi-temporal images (Stow et al., 1980).

Spectral-Temporal Combined Analysis

This technique is a classification of combined data sets from two or more differing points in time. For instance, the two images with four bands can be merged in one image with eight bands. The new image is then analysed by means of supervised or unsupervised classification techniques. This method is easier than Post-Classification comparison, however, the identification of change classes is rather complicated.

Expectation-Maximum Algorithm (EM Algorithm)

The method uses an iterative procedure to find parameters for statistical distribution model of cluster members in the feature space. The method yields a probability of each sample that falls in a certain class. For example, a difference image can be calculated from images of T1 and T2. Then, a probability that pixels are changed or unchanged can be estimated using EM algorithm (Dellaert, 2002).
The estimation of a priori class probabilities is rather challenging. Therefore, the EM method is not widely used in remote sensing practices (Lu et al., 2004).

Unsupervised change detection

The unsupervised change detection methods do not need a ground truth information (training data). The main idea is to select groups of objects with similar properties. There are numerous approaches of unsupervised change detection. One of the most popular method is based on the results of image differencing analysis. By this method, the “change” and “non-

change" groups of pixels are selected by means of a threshold value applied to the difference image (Bruzzone & Prieto, 2000; Melgani et al., 2002).

Hybrid change detection

The hybrid method is based on a combined use of two or more methods for change detection. This methods can be categorized as: procedure –based (using different change detection methods), result-based (using the results obtained by different methods). The approach enables combination of benefits from different methods. However, it can be difficult to define the influence of each involved method on the result of analysis (Hussain et al., 2013).

Artificial Neural Networks (ANN)

Neural networks attempt to simulate the basic mechanism of a human brain, providing human capacity for learning and further decision making. Thus, the neural network is an adapted tool that enables a computer system to find the solution to problems. After learning the procedure, the computer "remembers" the solution to problems from a learning set and thus can find an answer for similar problems.

There are two main steps in data processing by means of a neural network: the learning phase and the retrieval phase. During the learning phase, the weight parameters of a net structure are calculated based on the training data defining the neural model and are then used in the next phase. In the retrieval phase, the built model recognizes and classifies real test patterns.

The performance of the technique depends on the available training set and structure of the neural network. By applying the ANN approach, Liu and Lathrop (Liu & Lathrop, 2002) reported significant accuracy improvement of around 20-30% in comparison with the post classification comparison.

4) Advanced methods

This group consists of the Li-Strahler reflectance model, spectral mixture model and biophysical parameter method, with the former two models discussed in this subsection.

Li-Strahler reflectance model

This approach is based on the calculation of pixel value through the simulation of reflectance from the set of objects with simple geometric shapes. Some field measurements are required for definition of the model, representing a difficulty in application. This method can be applied

for estimation of size, shape and spacing of discrete objects that are imaged at resolutions sufficiently coarse that they cannot be resolved individually (Li & Strahler, 1985). This method is particularly efficient for quantitative estimation of vegetation changes. For the monitoring of conifer deadliness, Macomber & Woodcock (1994) use the Li-Strahler model for estimation of the size and density of trees from remotely sensed images. They demonstrated advantages of the method as giving both the statistical results of an inventory changes in forest canopy, with a high level of accuracy, and also maps showing the geographic distribution of mortality patterns.

Spectral mixture model

The pixels of a satellite image represent the sensor measurements of spectral radiance, with the values of the pixels resulting from an integrated sum of radiance of different neighbouring materials within the area covered on the ground by a single pixel. As a rule, the pixels do not represent a homogeneous area, but rather a mixture of several spectral classes. For example, the pixel can include information from vegetation, ground and water. The spectral unmixing procedure decomposes a reflectance source spectrum into a set of components or endmembers and corresponding factions (abundances). The abundances indicate the proportional composition of each endmember in the mixed pixel (Keshava & Mustard, 2002). A more detailed description of the spectral mixture model can be found in Adams et al.(1995), Quintano et al. (2012).
Roberts et al. (1998) describe a new method based on a multiple endmember spectral mixture analysis to map California chaparral. The technique enabled improved discrimination of vegetation classes through pixel-unique endmember selection. Lu & Weng (2004) examine characteristics of urban land-use and land-cover classes using spectral mixture analysis (SMA). The research indicated that SMA is useful to solve problem in the low resolution data, effectively characterizing the urban landscape patterns, and significantly improves classification accuracy in contrast to maximum-likelihood classifier.

5) GIS approaches

Significant advantages of change detection analysis are provided by GIS application due to its various features, such as the incorporation of different source data, support of multi-layered structure, numerous mathematical tools and integrated programming languages.
GIS are able to process a wide range of data with dissimilar nature, properties and origin. For handling different data types, GIS are equipped with interfaces for data import and export, as well as tools for data generation, analysis, actualization and visualisation. A GIS is well suited as a platform for the implementation of new tools for change detection analysis.

A straightforward application of GIS for change detection involves a visual analysis of multi-source and multi-scale maps, satellite images and field observation data, whereas complex applications include analysis by using self-developed modules and a combination of change detection techniques. A map overlay and binary masking are commonly employed for the quantitative assessment of change dynamics.

Many GIS applications are particularly focused on urban areas, where traditional methods often have poor change detection results due to complexity of landscapes (Lu et al., 2004). However, it should be noted that data fusion in a GIS environment can be rather complex due to different data standards, resolutions, data acquisition techniques, etc. (Koch & Ei-Baz, 1998).

6) Visual analysis

The remote sensed data of different dates are stored in different bands of the composed image. The image consists of 2 or 3 bands, which correspond to red, green and blue colors.The analysis is performed by visual analysis of a multi-data colour composed image.

The visual interpretation by human analyst is very flexible and can take use of the analysis of texture, shape, size, patterns and so on. However, the result quality depends strongly on the expert skills. Moreover, the analysis of large areas is very time-consuming.

7) Other change detection techniques

There are methods, which cannot be classified to the one of above described categories. These methods have not yet frequently been used and do not deserve separate own categories. This groups consists of the following methods: measures of spatial dependence (Henebry, 1993), knowledge-based vision system (Wang, 1993), area production method (Hussin, De Gier, & Hargyono, 1994), combination of three indicators: vegetation indices, land surface temperature, and spatial structure (Lambin & Strahler, 1994), change curves (Lawrence & Ripple, 1999), generalized linear models (Morisette et al., 1999), curve-theorem-based approach (Yue, et al., 2002), structure-based approach (Zhang et al, 2002), spatial statistics-based method (Read & Lam, 2002).

2.5.3 Change detection inside urban areas

Urban areas are characterized by the occurrence of numerous distinct objects-buildings, with such changes being consequences of new building, rebuilding, building destruction or vegetation growth. This section focuses on the selection of building destruction caused by natural or man-made catastrophic events, such as an earthquake or a civil war. Naturally, it is desirable to obtain states of separate buildings as a result of the change detection analysis.

Due to an increasing worldwide number of natural disasters, emergency situations and civil hazards, remote sensing methods have been widely adopted for obtaining reliable and continuous information on the scale and effects of a disaster and the current situation in the affected areas. Furthermore, post-disaster remotely sensed data is especially important for seriously damaged or difficult-to-access areas, with the generation of image-based damage maps becoming an indispensable component of information support for search and rescue services and humanitarian communities.

Many techniques for detection of damaged buildings have been developed based on remote sensed images. Such methods can be divided into two groups: 'image-to-image' and 'map-to-image' strategies.

The commonly used 'image-to-image' approach involves a comparison of pre- and post-event images on a pixel-by-pixel or object-by-object basis. Gupta et al. applied the image differencing change detection technique to pre- and post-earthquake remotely sensed images obtained by the LISS-II sensor of the Indian Remote Sensing Satellites (Gupta et al., 1994). The authors conclude that the applied method is a valuable tool for delineating areas of changes and damage assessment caused from the catastrophe.

Collapsed buildings can be also detected based on digital elevation models (DEMs) generated from pre- and post-event stereo aerial photographs. Within this approach, pre- and post-event DEMs are analysed to obtain the difference of detected building heights. The building is then recognized as damaged if the difference exceeds a specified threshold level (Turker & Cetinkaya, 2005).This proposed method has proved to be highly accurate.

Another frequently used application of aerial photography is the detection of buildings from their shadows. The central idea behind this approach is that destroyed buildings do not have any shadows or have very small shadows compared to their pre-event stage. Change detection is performed by comparison the buildings sets detected from pre- and post-event aerial photographs (Irvin & McKeown, 1989), (Turker & San, 2004), (Huertas & Nevatia, 1988).

Saraf et al. (2002) proposed a detection of damage using a pseudo-colour transformation (PCT) technique. The pre- and post-earthquake images from Indian Remote Sensing Satellites (IRS) were used to study the Bhuj earthquake-induced damage in the Bhachau area (India) in 2001.

Ehlers et al., 2012 applied texture-based analysis for destruction detection of buildings and infrastructure in areas affected by a rapid catastrophic event (Ehlers et al, 2010), with multi-temporal images filtered by a Fourier transform based band pass filter. The filtered images are then used for the calculation of textural features at each pixel, with change detection performed based on the obtained textural feature images. The authors analysed the usage of

different textural features (energy, correlation, contrast and inverse difference moment) for different change detection methods (difference, ratio, regression and PCA).

Methods based on the comparison of multi-temporal images provide reliable and effective change detection. However, pixel-based analyses require a precise geo-referencing of all processed data. Moreover, to achieve successful results it is desirable to take images acquired by the same sensor at the same time of a season, the same time of a day, and – for electro-optical sensors - in cloudless conditions (Hall et al., 1991). The accuracy of the change detection analysis suffers from variation of acquisition angles. The situation can be improved by taking additional information into account, such as vector-based maps, used as a primary information source through the 'map-to-image' approach. The vector map provides information about the exact position and size of each building, representing a major benefit of this approach. A comparative analysis of two different data types (vector map and remotely sensed image) is generally performed by extracting spectral, textural and structural parameters from the image for each single building, thereby excluding the image area outside of building footprints from consideration. A principal strength of the 'map-to-image' strategy is its capability to adapt image processing methods to the building-by-building analysis and to assess the state of each building separately.

Samadzadegan and Rastiveisi (2008) used vector map information for calculation of the textural features from the pre- and post-event satellite images over the building footprint areas. Furthermore, a classification of building conditions was performed by using a fuzzy logic system. The calculated overall classification accuracy indicates a high capability of the method in terms of the automated detection of collapsed buildings. However, the main difficulty of this approach is the selection of the optimum feature set. For redundancy minimization of the feature space, Goldberg (1989) proposed using an adapted genetic algorithm.

Chesnel et al. (2008) used very high-resolution images of the crisis region and roof footprints as vector data, with change detection performed based on the correlation value between the roof footprint images within the vector contour of a building object. In order to take the difference between acquisition angles into account, the contour positions on the images are chosen with a certain shift. For the detection of building state, rather than a fixed threshold correlation value, the authors used a state classification based on Support Vector Machines (SVM) technique (Vapnik, 1995; Burges, 1998). This approach enables adapting the change detection method to an investigated case (damage types, building types, atmospheric conditions, etc.). The authors showed the efficiency of the method on six different very high-resolution images acquired before and after the bombings of Beirut occurring between 12th July 2006 and 14th August 2006, with the vector map generated manually. The proposed

method demonstrates high efficiency in terms of the detection of damaged buildings, however, difficulties exist in the limitation of false alarms due to the difference between acquisition angles.

Sumer and Turker (2006) proposed a method for the detection of collapsed buildings based on differences in the grey-values and gradient orientations within a building footprint area. It was observed that the image brightness or image grey-values may vary depending of building state, with intact buildings having lower brightness and more regular distribution of gradient orientations then collapsed buildings. The proposed method was applied for the detection of damaged buildings caused by the Kocaeli earthquake in the northwest of Turkey, occurring on 17th August 1999.

Using vector information, DEM-based method enables performing change detection on a building-by-building basis and excluding pixels outside the building footprint from the consideration (Turker & Cetinkaya, 2005). This approach was tested for a study area of the city Golcuk, which was strongly affected by the Kocaeli earthquake. The experiments performed achieved a high level of accuracy, and demonstrating the major benefit of the integrated analysis of remotely sensed data and vector information.

Shi and Hao (2012) assumed that the footprint area of an intact building has a homogeneous structure, whereas the area corresponding to the damaged building is heterogeneous. Based on the vector data, the authors define a bounding rectangle around the expected building footprint on the post-event image. Within the rectangle area, a continuous homogeneous region is selected, which is compared in shape and area with the original building contour given by the pre-event vector data. The authors calculate Shape Similarity Index (SSI) (Ling & Jacobs, 2007) indicating the degree of similarity between the contour shape extracted from the image and the original building shape, and Area Ration Index (ARI) defining the ratio of pixel numbers from the extracted homogeneous area and the area corresponding to the original building footprint. In the SSI and ARI feature space, a k-means clustering is performed to select two groups of buildings (damaged and undamaged). The proposed method was applied to the high resolution GeoEye-1 and Quickbird test images obtained after the Yushu earthquake in China on 14th April 2010. The experiments performed indicated that the described approach is effective and particularly robust for the detection of entirely collapsed buildings. However, the experimental results pointed to limitations of building recognition when buildings are covered by trees or shadows.

Due to its object-oriented approach, the ‘map-to-image’ strategy enables the separate analysis of each building state, and this appears to be a better choice for change detection inside urban areas. Moreover, the vector data includes additional attributive information (building type, building age, utility, number of stories, etc.), which can also be very useful.

Change detection analysis after catastrophic events requires the utilization of all available information, and in such cases the application of GIS techniques are indispensable. GIS enables the manipulation and visualization of large amounts of multi-source data, and the integration of remote sensing technology and geoinformatics. Therefore, it has justifiably received widespread attention as a tool for change detection analysis in recent years.

In this research, the integrated approach was selected as a paradigm for change detection. For a general applicability of the method developed within this work, it was decided to use Open Source software as a basic development platform. The author believes that this approach yields more flexibility for innovative research studies and can be used as a basis for the further development of cost-effective software for practical application of the proposed change detection technique.

3. Open Source Software as Implementation Environment

Open Source Software (OSS) is a very interesting phenomenon. The software is developed by hundreds of thousands of worldwide programmers voluntarily producing, sharing and supporting their product with no monetary compensation. OSS is becoming more and more popular, with the most well-known OSS including Firefox web browser, Thunderbird e-mail client, Eclipse software development environment, Linux operating system and OpenOffice software suite.

Just one of the popular web-based source code repositories, SourceForge, supports over 324,000 open source projects which are developed by 3,4 million developers. SourceForge connects over 46 million consumers with open source projects and serves more than four million downloads a day (www.sourceforge.net/about).

3.1 Historical Overview and Definition

There are two similar movements, Free Software and Open Source Software. The definition of Free Software is published by the Free Software Foundation (FSF), the definition of Open Source Software by Open Source Initiative (OSI). Both software types basically refer to the same licenses with a few minor extensions; in other words, nearly all free software is open source and nearly all Open Source Software is free (www.gnu.org/philosophy/categories.html).

Both Free Software and Open Source Software are software products with available source code that is free to modify and to distribute copies of the original or modified program without paying any fees to the original creators of the software package. It enables the appropriation of the program code for own use. The license of the Open Source Software does not require providing the software for free. It means that a user is free to run the software, change the software and redistribute the software with or without charges. However, most of the Open Source Software are distributed free of charge.

Open Source Software started with one man: Richard Stallman. He used the term ‘free software’ for the first time and carefully explained its definition. Until the end of the 1970s, the major part of software was developed in academic institutions. The academic community does not encourage hiding information. Therefore, software developed in university laboratories was freely distributed internally, including the source code, with a release typically under very easy zero-cost licenses. Commercial companies often used the software developed under the academic licenses and integrated it into their own products. The new software became proprietary with a release under paid-for commercial licenses, under which the source code was not accessible. The tendency to hide the source code, namely a

restriction on viewing and amending by other programmers, seemed to Stallman to be especially negative. He saw the advantages of collaborative work in creating robust and high quality software due to the distribution of the program codes to different experts who understood the program functionality in depth and could contribute to the software effectively in each specific part. Stallman believed that it was unfair that the commercial companies made financial gains from the work done by the academic developers, which they obtained for free. To avoid this, a licensing mechanism was needed to guarantee that, once software was made in an open environment, free for modification and distribution, it cannot ever be proprietary.

On 27 September 1983, Stallman announced the GNU project (GNU is a recursive acronym that stands for “GNU’s Not Unix”). The goal of the project was the development of a free software environment that enables working without use of any non-free software. In a search for persons holding the same views, Stallman founded the non-profit organization “Free Software Foundation” (FSF) on 4 October 1985. After a short period of development, in 1989, FSF established a GNU General Public License (GPL) and started to release software under this license. The software under GPL may be used for any commercial and non-commercial purposes. The source code availability is not required for internal use, but in the case of sale or distribution the source code including all source changes must be made available to the end users. In order to emphasize the difference to the copyright licenses of proprietary software, the new form of licensing became the name “copyleft”.

By the early 1990s, Stallman and FSF developed such software as EMACS text editor, GCC compiler, GDB debugger, and Bash command processor as a part of the GNU project.

In 1991, Linus Torvalds, a student at the University of Helsinki (Finland), started work on a new operating system which was eventually named Linux. He used the FSF’s software as a basis for the system kernel. The development of Linux progressed rapidly due to the collaborative work model, which became possible with the steady progress in Internet technologies. Thus, thousands of programmers dispersed around the world can work on the different aspects in order to produce a stable, reliable, and fast running version of the new open source operating system. At the suggestion of Stallman, the Linux operating system was released under the GPL. To date, Linux has become one of the most popular operation systems. Numerous open source and proprietary software have been developed to run on Linux.

In 1997, Eric Raymond published a collection of essays in The Cathedral and the Bazaar (Raymond, 1999). He described the software engineering methods used for free software development. The publication received significant attention and motivated Netscape Communications Corporation to release the Netscape Communicator Internet suite as free

software. The Mozilla Application suite was later developed based on this released source code.

In February 1998, the Open Source Initiative (OSI) was founded by Eric Raymond and Bruce Perens and promoted by publishing the source code of the Netscape Communicator. The members of the OSI created a new term, “open source”. They emphasized the efficiency of the source code availability, distinguishing the term “open source” from the philosophically and politically focused label “free software”. Depending on the license, however, the main software is now free or open source but mostly the software is both free and open source, since many licenses are accepted by both OSI and FSF.

Nowadays, there are many licenses approved by OSI and FSF, however, GPL still remains the most popular of them.

Lately, the open source concept has increasingly attracted the attention of large commercial companies. The companies not only use the OSS, but also support and develop it. Thus, for example, Google develops Android (an open source operation system based on Linux) and Chromium (open source web browser), Nokia possesses the Qt project (Open Source Software development framework), and Autodesk releases MapGuide Open Source (web-based platform for developing web mapping applications and geospatial web services).

Recent years are characterized by the intensive development of geographical information systems with open source codes. The FreeGIS.org catalogue counts about 350 different GIS projects with open source codes.

In 2006, the Open Source Geospatial Foundation (http://www.osgeo.org) was established with the purpose to support and promote the development of open geospatial technologies, data and educational material. The following projects are supported by the foundation:

1. Web Mapping (deegree, geomajas, GeoMoose, GeoServer, Mapbender, MapBuilder, MapFish, MapGuide Open Source, MapServer, OpenLayers, ZOO-Project).
2. Desktop Applications (GRASS GIS, Quantum GIS, gvSIG, Opticks).
3. Geospatial Libraries (FDO, GDAL/OGR, GEOS, GeoTools, MetaCRS, OSSIM, PostGIS, rasdaman).
4. Metadata Catalog (GeoNetwork).
5. Public Geospatial Data.
6. Education and Curriculum.

In Germany, OSGeo is represented by FOSSGIS e.V. (abbreviation for Freie und Open Source Software für Geoinformationssysteme). Originally founded in 2000 as GRASS-Anwender-Vereinigung e.V. (GAV), it has been operating since September 2008 as FOSSGIS e.V. which organizes the FOSSGIS conferences, provides free geodata by the

Open Street Map project, presents free and open source GIS software at conferences and meetings, operates the FreeGIS.org portal and initiates round tables throughout Germany.

3.2 Used Software

The application of change detection techniques considered in this study requires GIS software for processing and visualization of raster and vector data, programming possibilities for implementation of algorithms as well as data mining software for classification of object states. In the performed study, GRASS GIS was used as the GIS software, Python as the programming language and Orange as the data mining software. Figure 3.1 presents the general scheme for the integration of GRASS, Python and Orange into a single change detection tool. The scripts written by using Python and Orange are included in GRASS as external modules. In this way, the user working with GRASS GIS can perform a change detection analysis by using new algorithms without leaving the GIS system; all processing scripts run in the background.

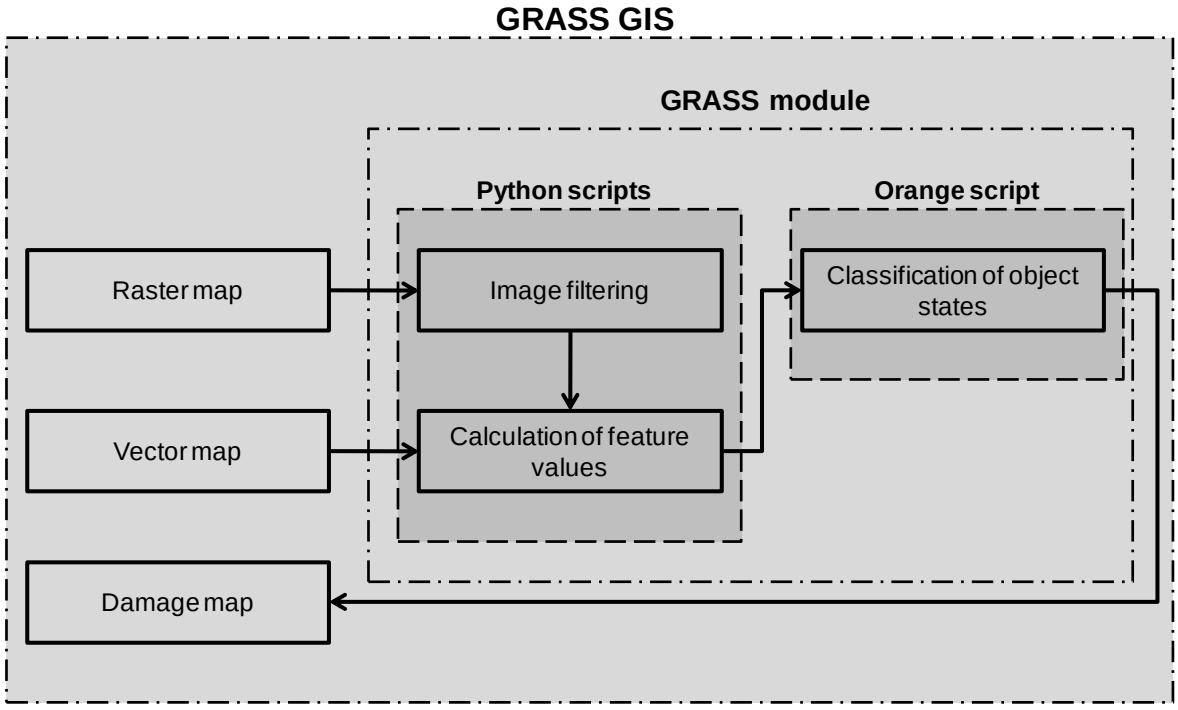


Figure 3.1: General structure of the change detection module in GRASS GIS.

3.2.1 GRASS GIS

GIS GRASS (Geographic Resources Analysis Support System) is currently the most popular system among open source GIS (Neteler et al., 2012). Its development was initiated by the U.S. Army Construction Engineering Research Laboratory (CERL) in 1982 in Champaign,

Illinois to support land management and environmental planning at military installations. From 1982 to 1995, CERL led the development of GRASS with the involvement of numerous universities and federal agencies. GRASS, with its complete source code, was published as public domain software, i.e. without ownership such as copyright or trademark. After finishing the GRASS 4.1 release in 1995, CERL officially stopped its active participation in the project. In 1997, the developer group at Baylor University took control of GRASS development from CERL and released GRASS 4.2. In 1998, the main development was concentrated at the University of Hannover. In October 1999, Version 5.0 was released under GNU General Public License (GPL) which is used for all later program versions. In 2001, the main GRASS development activity was moved to ITC-irst (Centro per la Ricerca Scientifica e Technologica), Trento, Italy. In 2006, the GRASS project became a founding member of the Open Source Geospatial Foundation (OSGeo) and, in the same year, the GRASS Project Steering Committee (PSC) was established, which is currently responsible for the management of the project. Since 2008, the GRASS website, mailing lists and bug tracker are hosted by OSGeo. To date, the development team has grown into a multi-national team of programmers at numerous locations. Presently, GRASS has a large user community that makes bug searches and software learning much easier.

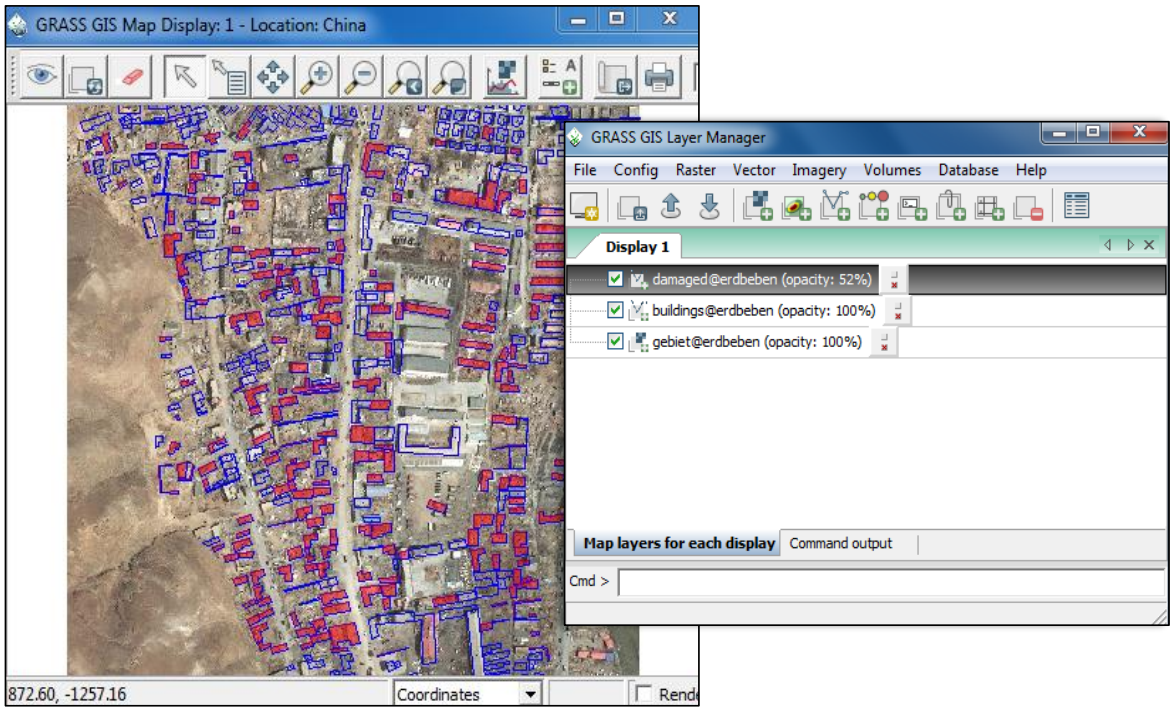


Figure 3.2: The graphical user interface of GRASS GIS with data representation.

GRASS GIS is a multi-purpose GIS running on common operating systems: Windows, Linux, Mac OSX, BSD etc (Neteler & Mitasova, 2005). The software has a modern Graphical User Interface (GUI) (Figure 3.2) which includes an integrated location wizard, vector digitizer,

SQL query builder, attribute editor, model builder, 3D view mode, and georeferencing tool. Besides the window-oriented user interface, GRASS provides a command-line oriented tool for manual command entering. Visual tools in GRASS enable animation of spatial data, visualization of different layers, and creation of 3D-views.The GUI text and messages are translated into over 20 languages.

GRASS provides a flexible tool set for work with map projections, partially due to the usage of PROJ.4 open source cartographic projections library (http://proj.osgeo.org). Through the application of the GDAL/OGR library (http://www.gdal.org), GRASS is compatible with the majority of common raster and vector data formats such as GeoTIFF, Erdas Imagine, SDTS, ECW, MrSID, JPEG2000, DTED, NITF, ESRI Shape file, ESRI ArcSDE, MapInfo, GML, KML, PostGIS and Oracle Spatial. GRASS supports SQL interface to database engines: PostgreSQL, MySQL, and SQLite.

GRASS is a modular system which provides access to over 400 modules. Each function of data processing is performed with a single module, thereby clearly dividing the GIS components and ensuring transparency. The modules contain tools for managing vector data, processing multispectral data, analyzing spatial data, as well as simulating different events and visualizing final results. Various surface processes, such as rainfall-runoff modeling, flow path calculation, ground water control, and slope stability analysis are only a few of the many applications. GRASS can be easily connected with a printer, plotter, or digitizing table.

Nowadays, GRASS GIS is widely utilized in various academic and commercial projects around the world. Many governmental agencies including NASA, NOAA, USDA and DLR use it. The functionality of the GRASS is comparable with the popular commercial GIS programs, for example ArcGIS (ESRI, Inc.). GRASS is distributed under GPL license and can be downloaded from its web page at no cost.

In contrast to proprietary software, the GRASS GIS enables both the utilization of existing source codes and the implementation of own modules. A new module can be built by combining other GRASS modules and by implementing scripts written in different programming languages including Python, Bash, C, JAVA, and Perl. The GRASS parser allows for easy generation of GUI and documentation for the user module.

Using the GRASS parser tool, GUI and a html documentation file with an optional description for a user module are generated automatically. The generated documentation can be extended by an additional detailed description of the module. Below, the structure of the GRASS module for the change detection procedure is presented.

```
# Description of GUI
#%Module
#% description: Detection of building state
#%End
#%Option
#% key: rastermap
#% type: string
#% gisprompt: old,cell,raster
#% description: Raster input map
#% required : yes
#%End
#%Option
#% key: vectormap
#% type: string
#% gisprompt: old,vector,vector
#% description: Vector input map
#% required : yes
#%End
#%Option
#% key: damagemap
#% type: string
#% gisprompt: new,vector,vector
#% description: Damage map
#% required : yes
#%End
# Import of module for image filtering
Import image_filtering
# Import of module for feature generation
Import features_generation
# Import of module for classification
import classification
# Import of GRASS modules
fromgrass.script import core as grass
...
# Import of standard python modules
import sys
...
def main():
    # Image filtering, using “image_filtering”
    # module
   ...
    # Extract raster area around vector objects
    ...
    # Calculation of features values, using
    # “features_generation” module
    ...
    # Classification, using “classification” module
    ...
```

```
    # Load data into damage map
    ...
return 0
if __name__ == "__main__":
    options, flags = grass.parser()
    sys.exit(main())
```

At the beginning of the module code, the description of GUI is provided, which results in the interface shown in Figure 3.3. Then, the module code defines the import of Python modules. The image filtering procedure is implemented in module “image_filtering”. The “feature_generation” module contains the Python functions for the feature generation. The “classification” module includes the functions developed for classification based on the generated feature set. The entire processing is coded in the “main” function. Firstly, the image filtering is performed. The next processing step cuts out and stores the areas of the raster map around the selected vector objects in separated files. The generated raster files are then used for calculations of a feature set for each vector object. On the following step the object state classification is performed. The obtained object states are loaded in the new vector damage map.

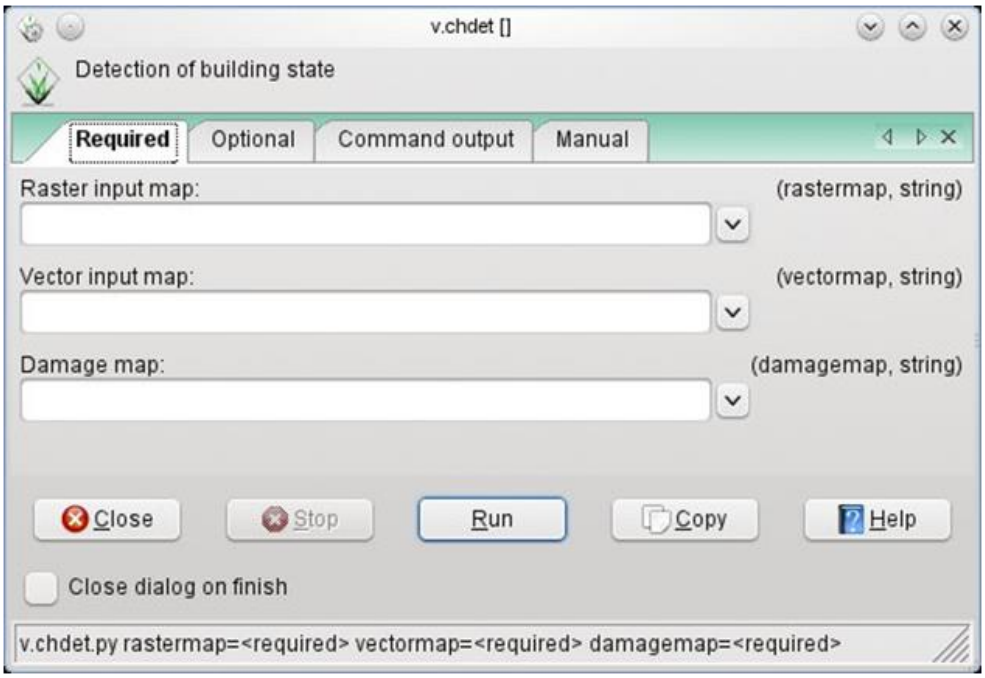


Figure 3.3: Dialog window of the change detection module.

3.2.2 Python

There is a striking trend of the emergence of programming languages with open source code like Perl, Python and Ruby in the last decade. For scripting interfaces, Python is the most used of them, not only in open source, but also in proprietary software. Thus, for example, ERSI uses Python as the scripting language for writing scripts in ArcGIS. The main advantage of Python for building the change detection tool is its usage as a scripting language for the implementation of the interface in GRASS and Orange.

The first implementation of Python was started by Guido van Rossum at CWI (National Research Institute for Mathematics and Computer Science) in the Netherlands in December 1989. Python 1.0 was released in January 1994, Python 2.0 in October 2000, Python 3.0 in December 2008. On 6 March 2001 the Python Software Foundation (PSF) was founded as a non-profit organization devoted to the Python programming language. In 2005, the PSF won the prestigious Computerworld Horizon Award for cutting-edge technology.

Interest in developments in Python is constantly increasing. Based on the measurement by the TIOBE index, Python was awarded a TIOBE Programming Language Award twice, for the years 2007 and 2010, as a language with the greatest growth in popularity over the course of a year.

Python is distributed under the Python Software Foundation License (PSFL) approved by both FSF and OSI. Unlike GPL, the PSFL enables distributing derivative software without making their code open source. The source code as well as its binary code compiled for different operation systems can be downloaded free of charge from the Internet. Python is developed by a worldwide distribution team supported by PSF. Python program can be written in object-oriented or functional programming styles. The following key distinguishing features are enumerated on the Python official website (www.python.org):

1. Very clear, readable syntax.
2. Strong introspection capabilities.
3. Intuitive object orientation.
4. Natural expression of procedural code.
5. Full modularity, supporting hierarchical packages.
6. Exception-based error handling.
7. Very high level dynamic data types.
8. Extensive standard libraries and third-party modules for virtually every task.
9. Extensions and modules easily written in C, C++ (or Java for Python, or .NET languages for IronPython).
10. Embeddable within applications as a scripting interface.

Python programs can be executed on all major operating systems such as Windows, Linux/Unix, OS/2, MacOS and Amiga. Thanks to the large user community and numerous documentations and tutorials, Python is easy to learn and provides a powerful standard library which allows for rapid application development. Moreover, there are many third-party libraries developed within distinct open source projects due to the popularity of Python. The most popular libraries which are very valuable for analysis of raster and vector information are:

1. Python Imaging Library (PIL). The library allows for opening, manipulating, and saving many different image file formats.
2. SciPy. It is a set of algorithms and mathematical tools for optimization, linear algebra, integration, interpolation, Fast Fourier Transformation (FFT), signal and image processing, solution of ordinary differential equations (ODE) etc.
3. NumPy. NumPy enables supporting large, multi-dimensional arrays and matrices providing a large number of high-level mathematical functions to operate on them.
4. OpenCV (Open Source Computer Vision Library). The library focuses on the real-time image processing.
5. matplotlib. A Python 2D plotting library. The library allows for generation of plots, histograms, power spectra, bar charts, error charts, scatterplots etc.

The following compact example consists of only a few lines of Python code which perform the algebraic operation on two images:

$$C_{ij} = (A_{ij} - B_{ij})/2 + 128 \tag{3.1}$$

where A_{ij}, B_{ij}, C_{ij} are the pixel values at row i and column j of images A, B, C.

```
from PIL import Image
from PIL import ImageChops
A = Image.open('A.tif')
B = Image.open('B.tif')
C = ImageChops.subtract(A,B,2,128)
C.save('C.tif')
```

The input data is read from the A.tif and B.tif files, with the resulting image saved in the C.tif file. The image can be treated as an image object (see the example above) or as a two-dimensional array with elements corresponding to separate pixels. Thus, for example, the following script set all pixel values of the image from the A.tif file to value of 255 if the pixel value is larger than the threshold value of 200, otherwise to 0. The result of the transformation is saved back in the A.tif file.

```
from PIL import Image
A = Image.open('A.tif')
A_Array = A.load()
for i in range(1, A.size[0],1):
   for j in range(1,A.size[1],1):
      if A_Array[i,j]>200:
         A_Array[i,j]=255
      else:
         A_Array[i,j]=0
A.save('A.tif')
```

Development of Python scripts can be significantly simplified by using an Integrated Development Environment (IDE). In Python distribution the IDLE integrated development environment is included (Figure3.4). This IDE is coded in Python using the tkinter graphical user interface tool.

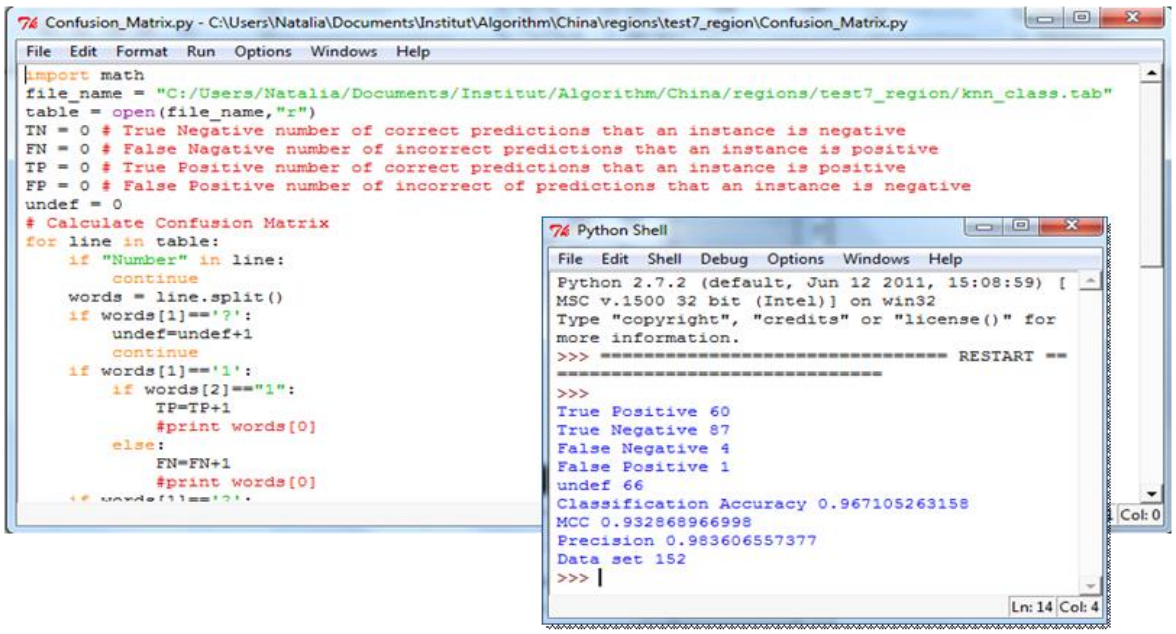


Figure 3.4: Creating and editing a Python script in an IDLE environment.

The more powerful IDE is PyDev. It is an open source plugin for the Eclipse multi-language software development environment. The PyDev enables editing, debugging, refactoring, quick navigation, code analysis, unit test integration etc. Another popular IDE is Spyder which is intended for development of scientific software. Spyder is an open-source scientific Python development environment providing functions for editing and debugging in the MATLAB-style.

Python properties make this scripting language a flexible and powerful tool for building a data interchange between Orange and GRASS. Moreover, it allows for fast implementation of new

algorithms for feature generation and image filtering. The created Python code can be easily integrated into the change detection workflow.

3.2.3 Orange Data Mining Software

The Orange development began in 1997 by Janez Demsar and Blaz Zupan, Orange is a general-purpose machine learning and data mining software. Further development was continued by members of the Artificial Intelligence Laboratory and then the Laboratory of Bioinformatics at the University of Ljubljana (http://ailab.ijs.si/dunja/TuringSLAIS-2012/), (Demsar & Zupan, 2012).

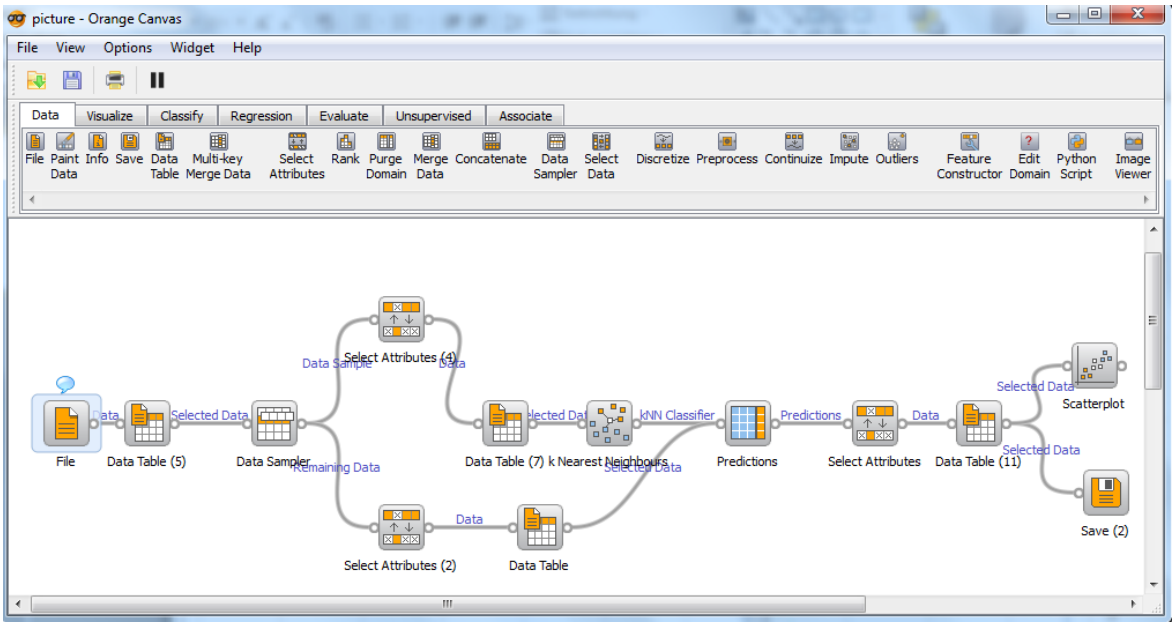


Figure 3.5: Snapshot of the Orange canvas.

At the beginning, Orange was a C++ library containing machine learning algorithms and related procedures, like preprocessing, sampling and other data manipulation. Mostly, Orange was used to assess a classification strategy for different data sets. The users utilized different programs developed based on the Orange library. The explicit usage of C++ components was seldom. In order to improve the usability of Orange, a Python scripting interface was developed. The functionality of the programs was transformed into Python objects. Thus, instead of the sequence of program calls, one Python script could be used. Since 1999, Orange has mostly been used as a Python module. Exporting a C++ code into Python enables the creation of fast and well readable program code, thereby significantly improving the usability of Orange and the collaboration within the developer teams. The majority of developers began to create new code in Python, avoiding C++ programming. One of the most advantageous applications of Python is the simplification of GUI creation.

The next significant development step was implementation of a GUI, which enables usage of Orange without any programming efforts. Nowadays, Orange is continuously being developed with distribution under the GPL license at no cost (http://orange.biolab.si).
Orange is a component-based data mining and machine learning system with powerful data visualization capabilities. The program supports numerous tools including data processing, modeling and evaluation: data sampling, filtering, discretization, classification and regression models, calibration of probability predictions of the classification, association rules and clustering, methods for assessment and scoring of prediction models etc. (Figure 3.5).
The graphical user interface (Orange Canvas) provides a flexible creation of own schemes for assessment of various machine learning or data mining algorithms on available data sets. In Orange, the concept of so-called visual programming is realized, with the visual program built of distinct objects-widgets connected by communication channels (Figure 3.6).

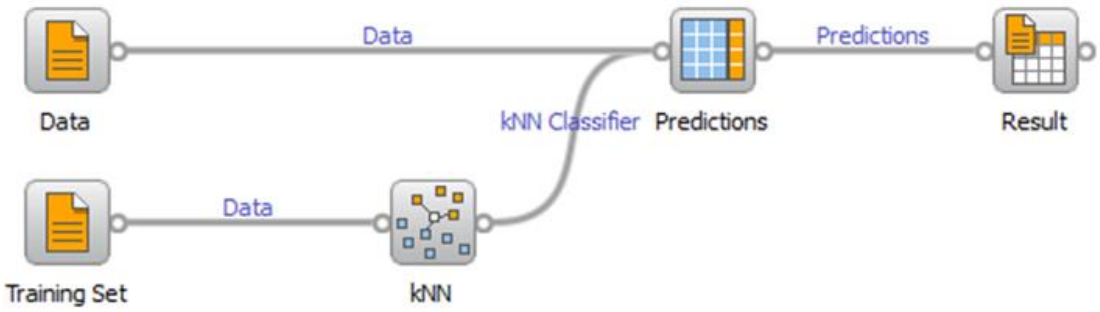


Figure 3.6: Visual program for k-NN classification.

The widgets provide some basic functionality, for example, reading and saving data, building learners, creating plots, etc. There are over 100 different widgets and their number is constantly growing. Each widget has a set of properties which define its functionality. If the user changes the configuration of any widgets, the whole generated analysis schema is updated automatically.
The analysis of classification techniques is enhanced by numerous widgets responsible for visualization, which enable the creation of scatter plots, bar charts, trees, dendrograms, networks, heat maps etc.
Orange is an Open Source Software with an active user community. The user can study existing code and develop own widgets, scripting interface or new add-ons. The software has good documentation containing the tutorial, description of all widgets and scripting manual. Orange can run on Windows, Mac OS and Linux.
Using a visual programming approach, the fast testing of ideas and optimization of

classification strategies can be performed. The obtained visual program can be transformed into a Python script.

In the example in Figure 3.6, the visual program for the assessment of k-NN classification is presented. The “Data” widget reads input data to be classified. The “Training Set” widget loads data to be used as a training set for the building of a k-NN classifier with the “kNN” widget. The classification is performed by the “Prediction” widget. The “Result” widget is used to visualize the classification results (Figure 3.7). The described schema can be also implemented in the form of the following Python script.

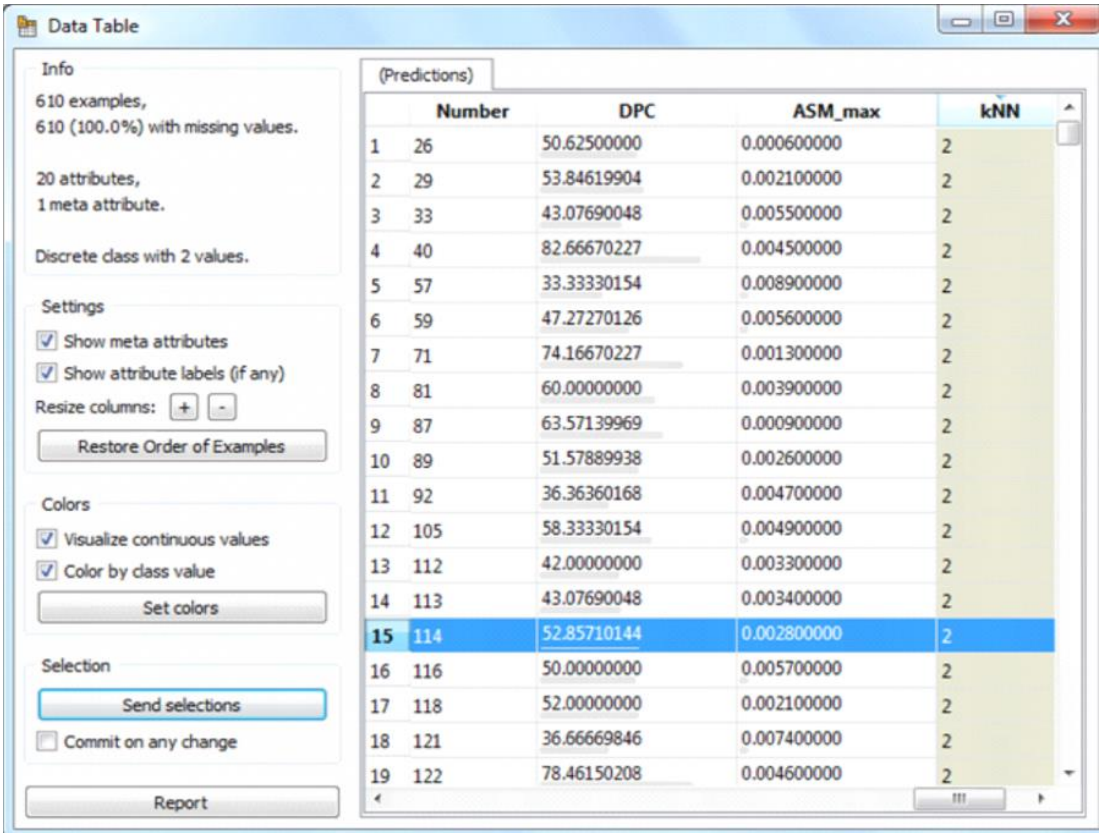


Figure 3.7: Visualization of classification result by using the "Data Table" widget. The “kNN” column contains the numbers of the obtained classes.

```
import Orange
data = Orange.data.Table("samples.tab")
training_set = Orange.data.Table("training_set.tab")
knn = Orange.classification.knn.kNNLearner(training_set,k=3)
result = open("result.dat","w")
for i in range(len(data)):
   result.write(str(i)+" "+str(knn(data[i]))+"\n")
result.close()
```

The script reads feature data from the sample.tab file and training set from the training_set.tab file. The results of the classification are stored in the result.dat file. The presented script can be easily transformed into a function and included in a Python module. Further, the obtained function can be called from the Python programs. In this way, the built classification scheme can be integrated in the GRASS module, as shown in section 3.2.1.

4. Integration of Remote Sensing and GIS for Building Change Detection

4.1 Basic requirements for the new change detection method

In section 2.5.2, various methods for change detection were introduced and discussed in detail. The effectiveness of the presented methods can be differentiated based upon the following criteria (Macleod & Congalton, 1998):

1. The capability to define the size of the changed area.
2. The capability to locate the changes.
3. The capability to identify the nature of the occurred changes.
4. The accuracy assessment.

The method developed by this research is supposed to be predominantly used for the purposes of emergency management, which advances the additional requirements such as robustness, a high degree of automation, the utilization of full available information and the efficient representation of results.

Comprehensive analysis capabilities of GIS, coupled with methods of data mining, techniques of spatial association, spatial clustering, spatial relation and distribution, provide a powerful intelligence apparatus for detection of changes. The visualisation capacities supported by GIS enable an effective representation of damage maps and a combination of results with various additional data stored in different thematic layers (road network, water resources, geology, relief, etc.). The use of GIS for processing remotely sensed data has eventually become an essential approach for change detection analysis.

The most valuable GIS information with respect to the change detection analysis is the detailed geometrical description of objects within the studied area. Unfortunately, the geometrical and context information available in GIS is seldom used as a data source for explicit comparative analysis with remotely sensed images. Combined analysis of vector and raster data supplements the conventional techniques based solely on raster data, enabling a concentration of analysis in areas related to the considered objects, the selection of objects depending on attributive information, usage of object geometry information, etc. For urban areas, the usage of vector information is especially advantageous, given that it allows performing change detection analysis individually for each building.

The development of methods based on the integrated analysis of vector and raster data sets is a point of general interest. However, most research addresses the estimation of land-use change dynamics (Centeno, 2000; Lo & Shipman, 1990; Li, 2010; Mattikalli, 1995; Weng, 2002), whereas only a few deal with damage assessment. Some techniques for deriving urban changes from remotely sensed data and GIS are presented in section 2.5.3.

This thesis presents a development of a new change detection algorithm for the identification of destroyed buildings caused by natural or man-made catastrophic events. The innovative aspect of this research lies in the employment of a single post-event remotely sensed image and pre-event vector data as an initial data set. Thus, rather than commonly used satellite images obtained at different points in time, a vector map presenting information concerning the original urban layout is involved in the change detection analysis.
Table 4.1 presents the basic requirements for the change detection algorithm, which this work aimed to fulfil. Furthermore, this chapter is concerned with achieving a detailed, stepwise description of the proposed method.

Objective	Development of change detection algorithm based on the integrated analysis of remotely sensed and vector data.
Source data	Post-event remotely sensed image with a spatial resolution equal to or finer than 1 meter and a vector map containing information concerning the pre-event urban layout.
Result	Vector damage map presenting destroyed and intact buildings.
Implementation	Open Source Software as implementation environment (GRASS GIS, Python, Orange data mining).

Table 4.1: Requirements for a new change detection algorithm developed in this study.

4.2 General concept

As mentioned in the previous chapters, most of the numerous change detection methods have been established based upon the analysis of remotely sensed images acquired at different times. These techniques are based on the processing of raster data and they cannot be applied to vector data. The raster and vector data models are completely different approaches concerning the representation of the real world. The raster data contains information about the area occupied by considered objects, whereby the data is stored in an array of pixels, each of which represents a small part of the area. The raster data does not explicitly contain any information about objects within the considered area. The vector data provides information describing the object, such as the accurate spatial location, boundaries, neighbourhood and attribute information. Thus, given that raster and vector data supplement each other, their combined analysis naturally allows the improvement of change detection results. The integrated analysis of these data types requires a new approach taking into account the specific structures of both data types.

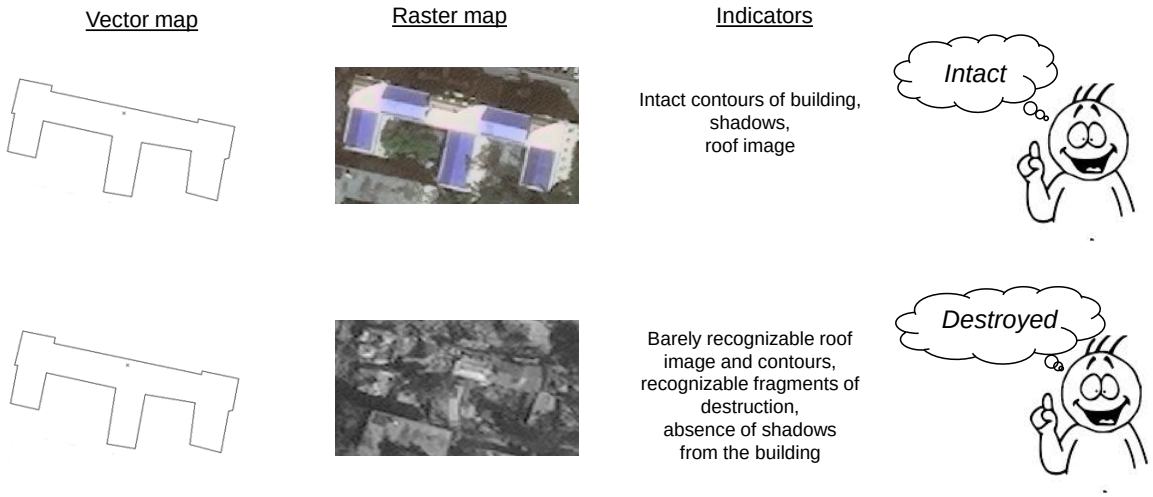


Figure 4.1: Combined analysis of vector and raster information by human expert.

In most cases, a human expert can easily define the state of a building through the combined visual interpretation of vector and raster maps. The expert makes a decision based on the visible features of destruction and his own experience (see Figure 4.1). In order to use a computer for such an analysis, the work of human experts must be modelled by a system of artificial intelligence (AI) (see Figure 4.2).

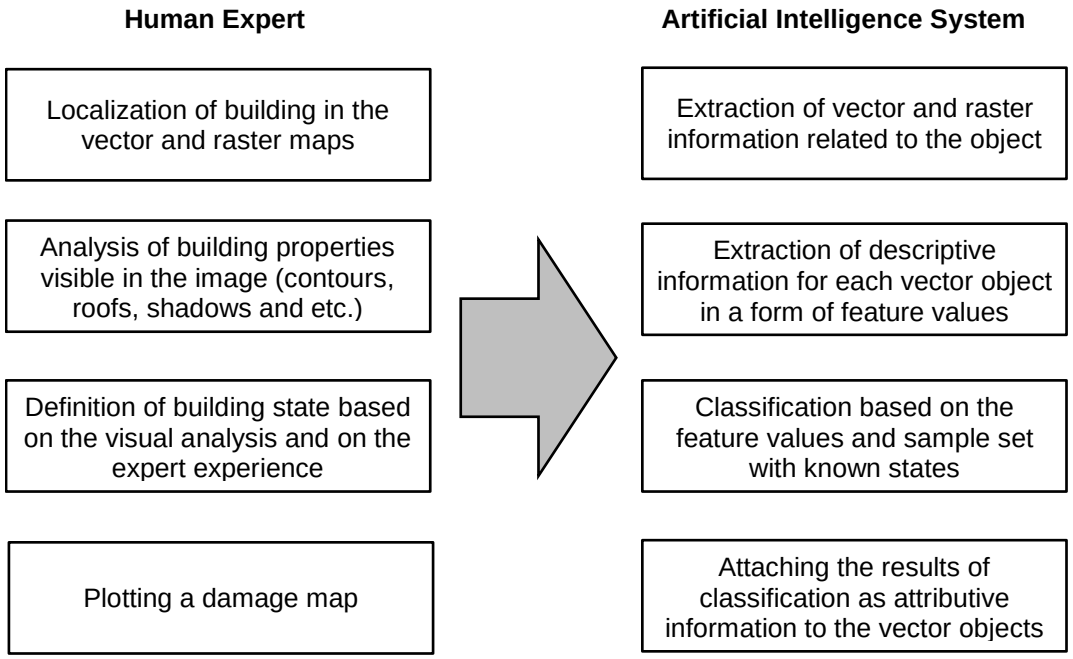


Figure 4.2: Building system of an artificial intelligence on the human expert workflow.

The object analysis by a human expert can be described by the following steps:

1. Selection and localization of the object in vector and raster maps.

2. Search for typical destruction features such as a lack of object contours, fragments of destruction, etc.
3. Making a decision about the building state based on the performed visual study and practical experience.
4. Building the damage map.

The AI system must model these steps by following the sequence of operations detailed below:

1. Selection of vector objects and extraction of related raster data.
2. Extraction of information related to the object state in the form of feature values, which can be characterized as quantitative measures of destruction signs.
3. Classification of building states based on the feature values and a set of examples with known building states; the experience acquisition is modelled by a learning procedure, with the resulting building of classifier structure allowing for an accurate classification of the given examples.
4. Attachment of the building states as attributive information to the building objects.

This sequence of operations defines the structure of the methodology developed in this study, which includes the pre-processing of remotely sensed data (image filtering), the selection of vector objects, generation of a feature set, classification and visualisation of the final results. The main steps of the proposed change detection methodology are summarized in the scheme provided in Figure 4.3.

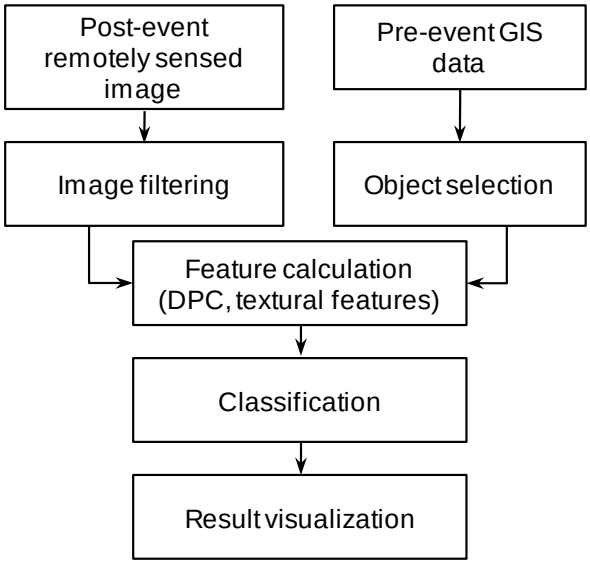


Figure 4.3: A general scheme of the proposed change detection method.

First, the enhancement of remotely sensed images is performed, before the vector object and part of the image containing its footprint are selected. Based on the selected vector and raster data, the information describing the building state is extracted in the form of features. The feature values are calculated by analysing the contour integrity and the raster image within the building footprint. For the description of the contour integrity, a new feature, the 'Detected Part of Contour' (DPC) is developed, which describes the integrity of the building contours. For the area within the contour, textural characteristics of the remotely sensed image are calculated by means of the well-known grey-level co-occurrence matrix (Haralick et al. 1973). Eventually, the classification of the building conditions is performed and the results are imported in GIS.

4.3 Pre-processing of raster data (image filtering)

The objective of the image filtering is to transform the raster data into a state that is more suitable for further processing. In most cases, filtering is used to remove unwanted information, i.e. noise, from the original image. Noise can occur during image capture (acquisition process, acquisition conditions such as illumination levels, calibration, positioning, etc.), transmission in a noisy channel or preprocessing. There are three main types of noise (Beaurepaire et al., 1997):

1. Additive noise, whereby the noise signal is added to the relevant signal during acquisition, transmission or processing.
2. Multiplicative noise, within which the noise signal is multiplied to the relevant signal during acquisition, transmission or processing.
3. Impulse noise, also called 'salt and pepper' noise, referring to single pixel noise, usually represented by randomly occurring white and black pixels.

Among the numerous approaches of image enhancement, two groups of the most commonly used algorithms of data filtering can be identified:

1. Filtering in the spatial domain with direct manipulation of image pixels.
2. Filtering in the frequency domain with manipulation of the frequency representation of the image.

Filtering in spatial domain is mainly performed to eliminate an additive noise, whereas filtering in frequency domain is a more universal approach. It is widely used to reduce different types of noise, as well as for other purposes of image processing. Moreover, filtering in frequency domain often serves as a basis for more complex algorithms.
It is evident that not all existing algorithms of image filtering can be covered in detail in the one chapter. This chapter is concerned with the basics of the spatial filtering and the

methods of the frequency domain filtering employed within this research. Accordingly, the interested reader is referred to (Gonzalez & Woods, 2002) for more information.

4.3.1 Filtering in spatial domain

The basic idea to modify the pixel value based on pixel values from its neighbourhood. Around the considered pixel, the neighbourhood area is defined by applying a filter mask of the size $n \times m$. A certain weight is prescribed to each of the selected pixels from the area. The filtered pixel value is found as the sum of pixels values multiplied with their weights. An example of the smooth filtering with neighbourhood area 3x3 is shown in Figure 4.4.

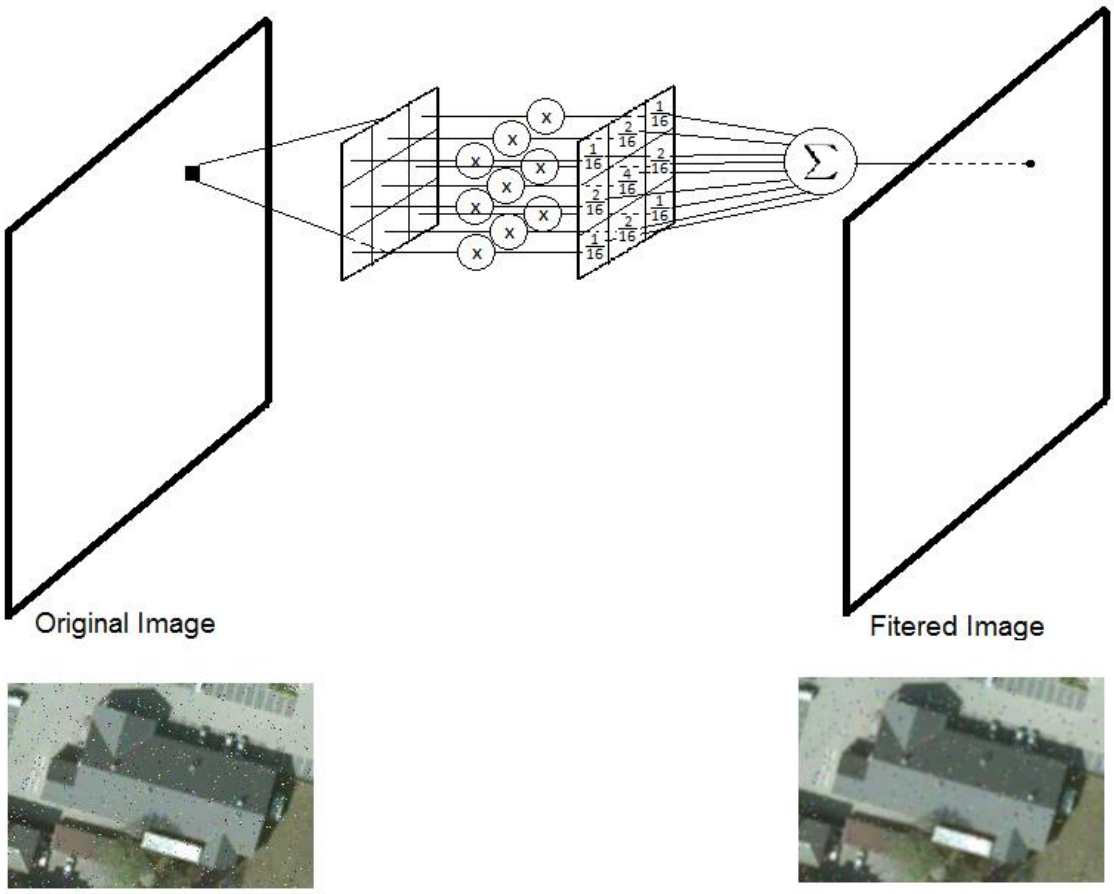


Figure 4.4: The procedure of spatial filtering. The picture (left) is affected by an additive noise.

The pixel values $\hat{x}_{i,j}$ of filtered image can be calculated by:

$$\hat{x}_{i,j} = \frac{1}{16}x_{i-1,j-1} + \frac{2}{16}x_{i,j-1} + \frac{1}{16}x_{i+1,j-1} + \frac{2}{16}x_{i-1,j} + \frac{4}{16}x_{i,j} + \frac{2}{16}x_{i+1,j} + \frac{1}{16}x_{i-1,j+1} + \frac{2}{16}x_{i,j+1} + \frac{1}{16}x_{i+1,j+1} \quad (4.1)$$

The matrix of weights is called the convolution matrix or convolution kernel. From the above example, it is equal to:

$$w = \begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix} \quad (4.2)$$

The above convolution matrix is normalized through dividing by the sum of all elements of the kernel. The normalization is used to obtain the resulting pixel values in the magnitude range of the input image. Within existing literature, the convolution matrixes are often presented in the unnormalized form; for example, the above matrix is often expressed as:

$$w = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (4.3)$$

In general form, the spatial filtering for the filter mask $n \times m$ can be expressed as:

$$\hat{x}_{i,j} = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w_{s,t} x_{i+s,j+t}\,, \quad (4.4)$$

Here, $a = \frac{n-1}{2}$ and $b = \frac{m-1}{2}$.

Evidently, special treatment is required for the pixels close to the image boundary, given that the filter mask for these pixels is not fully contained in the image. The easiest remedy is to perform filtering with the filter part included in the image area. Other approaches to the solution of this problem can be found in (Gonzalez & Woods, 2002).

Unnormalized convolution kernel	Description
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	Image remains unchanged
$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	Sharpening filter
$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	Smoothing filter
$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	Edge detection filter

Table 4.2: Examples of convolution kernels.

The properties of a spatial filter depend on the convolution kernel. Table 4.2 shows several spatial filter examples with a windows size of 3×3.
In this research, the spatial filtering is employed within the Canny edge detection technique described in section 4.5.1.1.

4.3.2 Filtering in frequency domain

A panchromatic image can be considered as a 2D discrete function of the image brightness $f(x,y)$. Representation of the image as function $f(x,y)$ is used here for the sake of convenience. It is related to the matrix notation used in the previous sections as $x_{i,j} = f(x=i, y=j) = f(i,j)$.The function $f(x,y)$ can be decomposed into a sum based upon 2D discrete functions $b_m(x,y)$ (Jayraman et al., 2011):

$$f(x,y) = \sum_{m=1}^{N} A(m) b_m(x,y) \tag{4.5}$$

Here, $A(m)$ are decomposition coefficients.
The appropriate choice of basic functions allows extracting the items containing the information about the image distortion from the whole sum of components:

$$f(x,y) = \sum_{m \in I} A(m) b_m(x,y) + \sum_{m \in D} A(m) b_m(x,y) \tag{4.6}$$

Here, I is a set of numbers of items containing a valuable information, D is a set of numbers of items responsible for distortions. Composing the image only from items defined by the set I, the image can be cleaned from undesired effects such as noise or brightness variation.

$$\tilde{f}(x,y) = \sum_{m \in I} A(m) b_m(x,y) \tag{4.7}$$

In general cases, it is mostly impossible to select items that are solely responsible for distortion effects. Thus, as a rule, removing a part of decomposition necessarily leads to a loss of the valuable information. A quality of the image enhancement strongly depends on the choice of the basic functions. In the image processing practice, the application of Fourier decomposition is one of the most popular approaches.

Using the Fourier approach, the decomposition can be expressed as:

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{i2\pi\left(\frac{ux}{N}+\frac{vy}{M}\right)} \tag{4.8}$$

The $F(u,v)$ is defined by the following expression:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-i2\pi\left(\frac{ux}{N}+\frac{vy}{M}\right)} \tag{4.9}$$

Expression 4.9 defines the Fourier transform of the function $f(x,y)$ and Expression 4.8 - the inverse Fourier transform.

Based on Euler's formula, the function $e^{i2\pi\left(\frac{ux}{N}+\frac{vy}{M}\right)}$ can be written as:

$$e^{i2\pi\left(\frac{ux}{N}+\frac{vy}{M}\right)} = cos\left(2\pi\left(\frac{ux}{N}+\frac{vy}{M}\right)\right) + i\, sin\left(2\pi\left(\frac{ux}{N}+\frac{vy}{M}\right)\right) \tag{4.10}$$

From (4.10), it follows that the 2D discrete brightness function is decomposed into a sum of the sine and cosine functions of various frequencies defined by values (u,v). The domain (u,v) is called the frequency domain. Each of the $M \times N$ terms of $F(u,v)$ is called a frequency component of transform.

The $F(u,v)$ is a complex function, i.e. $F(u,v) = F^{Re}(u,v) + iF^{Im}(u,v)$. The function $F(u,v)$ is often plotted as a 2D $M \times N$ image. For visualization of frequency components, the magnitude or spectrum of a Fourier transform is used, defined as $\sqrt{\left(F^{Re}(u,v)\right)^2 + \left(F^{Im}(u,v)\right)^2}$. The value $F(0,0)$ corresponds to the mean value of image brightness and is usually much higher than other frequency components. In order to reduce

the influence of the difference in the values concerning the visualization quality, a logarithm of the spectrum value is usually used for plotting the image spectrum.

For the sake of convenience in practice, the centered Fourier spectrum is used for processing in the frequency domain. In this representation, the low frequency components are located in the center of the spectrum image, with the high frequency components arranged on the edges (Figure 4.5). In order to obtain the centered Fourier spectrum, the function $f(x,y)$ is multiplied by $(-1)^{x+y}$. Thus, the centered Fourier transformation can be expressed by:

$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(-1)^{x+y}f(x,y)e^{-i2\pi\left(\frac{ui}{N}+\frac{vj}{M}\right)} \tag{4.11}$$

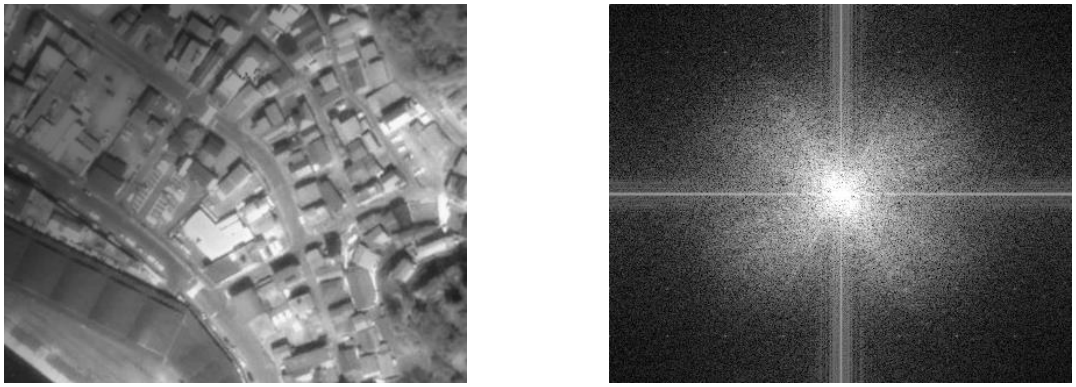


Figure 4.5: Centred Fourier spectrum.

When the centered Fourier spectrum is used, the function resulting from the inverse Fourier transform must be multiplied by $(-1)^{x+y}$:

$$f(x,y) = (-1)^{x+y}\sum_{u=0}^{N-1}\sum_{v=0}^{M-1}F(u,v)\frac{e^{i2\pi\left(\frac{ui}{N}+\frac{vj}{M}\right)}}{NM} \tag{4.12}$$

The filtering is performed by the elimination or reduction of frequency components of Fourier decomposition through a multiplication of $F(u,v)$ on the filter function $H(u,v)$. The filter function $H(u,v)$ is the real 2D discrete function, which takes values from 0 to 1. The value of 1 corresponds to intact components, while the value of 0 relates to the eliminated components, and values in-between correspond to the reduced components.

Thus, the filtered Fourier transform is calculated as:

$$G(u,v) = H(u,v)F(u,v) \tag{4.13}$$

The resulting brightness function can be obtained by using the inverse Fourier transformation; however, due to the application of the filter function, the obtained brightness function can be complex. The filtered image is obtained by extracting the real part of the resulted brightness function. From (4.12), it follows:

$$\tilde{f}(x,y) = (-1)^{x+y} Re\left(\sum_{u=0}^{N-1}\sum_{v=0}^{M-1} G(u,v)\frac{e^{i2\pi\left(\frac{ui}{N}+\frac{vj}{M}\right)}}{NM}\right) \tag{4.14}$$

Summarizing the above discussion, the Fourier filtering can be described by the following steps:

1. Multiplication of the input image by $(-1)^{x+y}$.
2. Calculation of the Fourier transform $F(u,v)$.
3. Calculation of the filtered Fourier transform $G(u,v) = H(u,v)F(u,v)$.
4. Calculation of the brightness function corresponding to $G(u,v)$ using the inverse Fourier transformation.
5. Extraction of the real part of the brightness function.
6. Multiplication of the obtained brightness function by $(-1)^{x+y}$.

As can be seen, the Fourier filter is defined by choice of the filter function $H(u,v)$. The simplest way to define $H(u,v)$ is to set a zero values for the Fourier decomposition components that must be removed and the value of 1 for other components. The filter eliminating the high frequency components can be expressed as:

$$H(u,v) = \begin{cases} 0 \; if \; D(u,v) \geq D_0 \\ 1 \; if \; D(u,v) < D_0 \end{cases} \tag{4.15}$$

Here, $D(u,v)$ is the distance from point (u,v) to the center of the frequency rectangle. For the spectrum image of size $M \times N$, the distance is defined as:

$$D(u,v) = \left[\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2\right]^{\frac{1}{2}} \tag{4.16}$$

The filter function defined by Equation 4.11 passes the low frequency components and is called the ideal lowpass filer. The lowpass filter is often applied to remove noise from images. However, the application of the ideal lowpass filter leads to the appearance of artifacts on the filtered image. A better quality of enhancement can be achieved by using smooth filter functions such as the Butterworth and Gaussian lowpass filter functions. The Butterworth lowpass function is defined as:

$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)}{D_0}\right]^{2n}} \tag{4.17}$$

Here, n is the order of the Butterworth filter function.
The Gaussian lowpass filter function is as follows:

$$H(u,v) = e^{-\frac{D^2(u,v)}{2{D_0}^2}} \tag{4.18}$$

For the elimination of low frequency components, the highpass filters are used. The highpass filter function can be obtained from the lowpass filter function as:

$$H_{hp}(u,v) = 1 - H_{lp}(u,v) \tag{4.19}$$

Thus, based on the lowpass filters discussed above, the following highpass filter functions can be obtained:

The ideal highpass filter:

$$H(u,v) = \begin{cases} 1 \; if \; D(u,v) \geq D_0 \\ 0 \; if \; D(u,v) < D_0 \end{cases} \tag{4.20}$$

The Butterworth highpass filter:

$$H(u,v) = 1 - \frac{1}{1 + \left[\frac{D(u,v)}{D_0}\right]^{2n}} \tag{4.21}$$

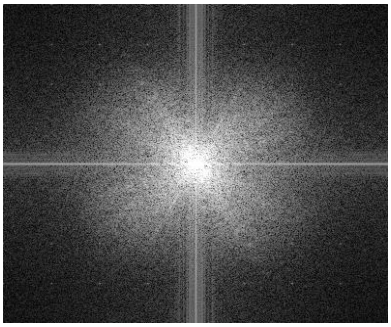
The Gaussian highpass filter:

$$H(u,v) = 1 - e^{-\frac{D^2(u,v)}{2{D_0}^2}} \tag{4.22}$$

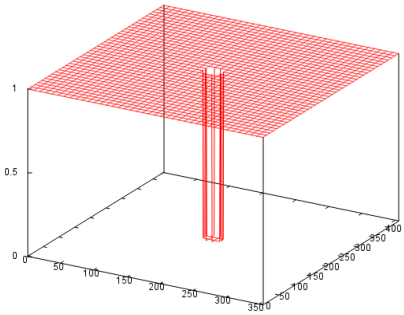
In practice, a combination of low and high filters is frequently used in the form of band filters that reduce the low and high frequency components, thereby passing the middle frequency components. A more detail comparative description of different filter function can be found in (Gonzales & Woods, 2002).
For the assessment of the building state, information about building edges and fragments of destruction is very important. This information is represented by the high frequency components and should be retained. Consequently, for the pre-processing of raster data, only filtering procedures based on high pass filtering were used in this study. Figure 4.6 illustrates examples of the image filtering with different highpass filters: the ideal filter, Butterworth filter and Gaussian filter. As can be observed, the ideal filter typically produces

the ringing artifacts in the filtered image. Visually, the artifacts are presented as a series of faint bands located parallel to the edges, which negatively affects the quality of the subsequent edge detection analysis. Therefore, the ideal filter is often inappropriate for real applications in the image processing. Filters with Gaussian and Butterworth profiles do not produce ringing artifacts and they are frequently chosen for image enhancement. The result of filtering strongly depends on the parameters used; thus, the value of the order n defines the appearance of ringing in the case of the Butterworth filter. The higher the n value, the sharper the filter function becomes, creating stronger ringing in the filtered image. In practice, the Butterworth function with n values of 1 or 2 is usually used.

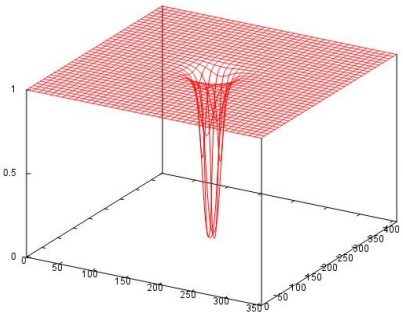
Original image
(@Digital Globe 2011)



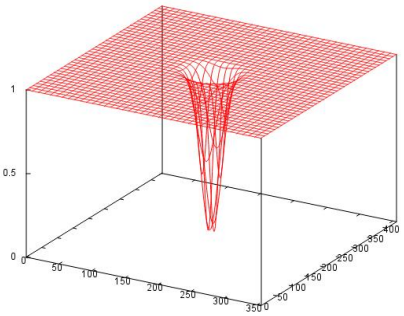
Frequency spectrum



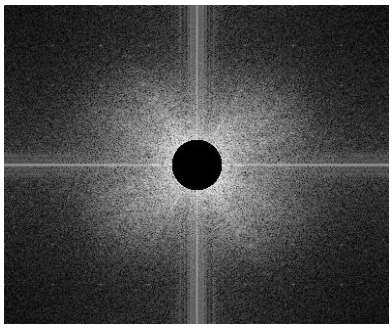
Ideal highpass
filter function
(a)

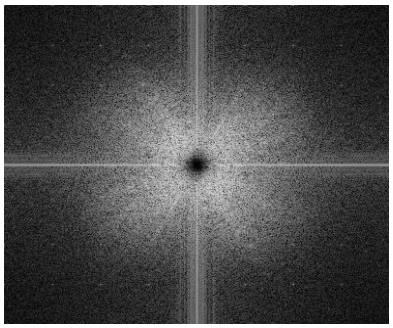


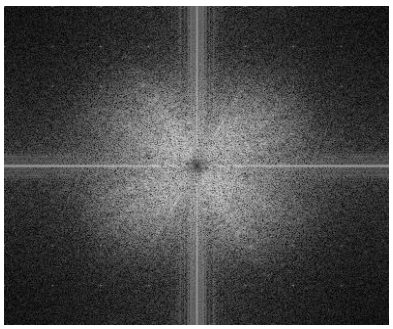
Butterworth highpass
filter function (order 2)
(b)

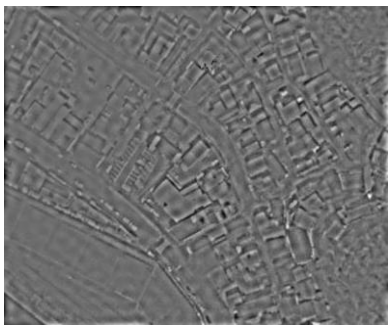


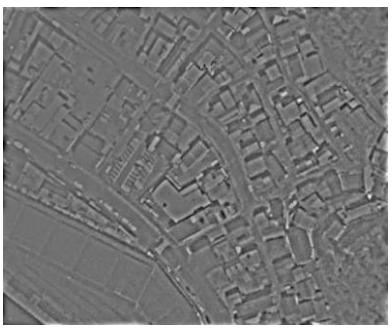
Gaussian highpass
filter function
(c)











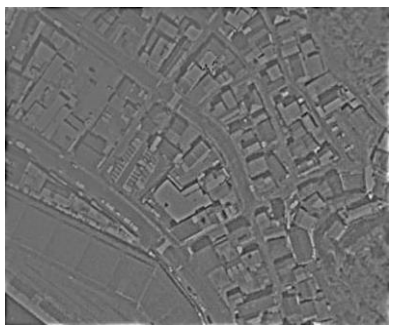


Figure 4.6: Examples of image filtering by means of different highpass filters: (a) ideal filter; (b) Butterworth filter; and (c) Gaussian filter.

4.3.3 Homomorphic filtering

Remotely sensed images can have areas with non-uniform illumination, low contrast and diminished colour, caused by lighting variations and complex environmental settings, which can negatively influence the result of the further image processing, including edge detection and texture analysis. For example, the selection of a threshold value for the edge detection process is chosen depending on a number of factors related to the mean values of the entire image, such as brightness, contrast and the level of noise. Consequently, illumination variation can lead to poor results for edge detection in image areas that are either too dark or too bright.

The most commonly used technique of reduction of non-uniform illumination is homomorphic filtering. This method is often preferred to other algorithms due to its ability to correct the image brightness and contrast by sharpening the image features at the same time (Gonzalez & Woods, 2002; Delac et al., 2006).

In general, an image $f(x,y)$ is formed as a product of two components: illumination $i(x,y)$ and reflectance $r(x,y)$:

$$f(x,y) = i(x,y)r(x,y) \tag{4.23}$$

The illumination describes the amount of source incident on the scene being viewed, usually with slow-moving variation across the image. Thus, in the frequency domain, the illumination function is represented by low frequency components. The reflectance is the amount of illumination reflected by the objects in the scene representing their details and edges, with the frequency components concentrated in the high frequency segment.

If the image is literally a sum of the illumination and reflectance, it can be decomposed by means of the Fourier transform. Bearing in mind that the image is a multiplicative function, the separation of illumination and reflectance is not possible in the frequency domain, given that the frequency transform of the product of two components is not separable. One way of overcoming this problem is to take a natural logarithm of both sides, thereby converting the multiplication into a sum:

$$ln(f(x,y)) = ln(i(x,y)) + ln(r(x,y)) \tag{4.24}$$

Applying Fourier transform to the above expression the following form can be obtained:

$$F(ln(f(x,y))) = F(ln(i(x,y))) + F(ln(r(x,y))) \tag{4.25}$$

Denoting the Fourier transforms of $ln(f(x,y))$, $ln(i(x,y))$ and $ln(r(x,y))$ as $Z(u,v)$, $F_I(u,v)$ and $F_R(u,v)$, respectively, Equation 4.25 can be written as:

$$Z(u,v) = F_I(u,v) + F_R(u,v) \tag{4.26}$$

As mentioned above, the illumination is represented by low frequency components and the reflectance by high frequency components. After applying a high pass filtering to the $Z(u,v)$ by means of a filter function $H(u,v)$ in the frequency domain, a filtered Fourier transformation is obtained as:

$$S(u,v) = H(u,v)Z(u,v) \tag{4.27}$$

The subsequent inverse Fourier transformation converts the modified signal back into the spatial domain:

$$s(x,y) = F^{-1}(S(u,v)) \tag{4.28}$$

Finally, the resulting enhanced image $\tilde{f}(x,y)$ is provided by the exponential operation:

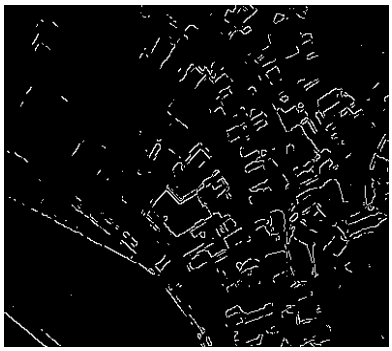
$$\tilde{f}(x,y) = e^{s(x,y)} \tag{4.29}$$

The homomorphic filtering yields both the normalization of the image brightness and the amplification of high-frequency components, such as building edges or fragments of destruction. Extending the algorithm of Fourier filtering from section 4.3.1, the entire process can be summarized in the following steps:

1. Taking the natural logarithm of $f(x,y)$.
2. Multiplication of the input image by $(-1)^{x+y}$.
3. Calculation of the Fourier transform $F(u,v)$.
4. Calculation of the filtered Fourier transform $G(u,v) = H(u,v)F(u,v)$.
5. Calculation of the brightness function corresponding to $G(u,v)$ using the inverse Fourier transformation.
6. Extraction of the real part of the brightness function.
7. Multiplication of the brightness function by $(-1)^{x+y}$.
8. Taking the exponential operation of the brightness function.

Homomorphic filtering is used for image enhancement in this research. The algorithm described above was implemented in Python, which enables its easy integration in the change detection workflow. Figure 4.7 illustrates experimental results of homomorphic filtering compared to pure Fourier filtering. The tests were performed by means of the ideal, Butterworth and Gaussian filter functions. The enhanced images were used for the subsequent edge detection using the Canny method (see section 4.5.1.1). The performed experiments demonstrate a poor result of the edge detection when directly applied to the original image. The difficulties of edge detection arise in the dark areas of the image.

Original image with
an irregular brightness



Canny edge detection
applied to the original image

(1) Image Fourier filtering

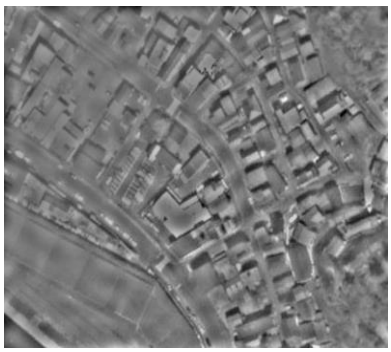


Image filtered using ideal
highpass filter function
(a)

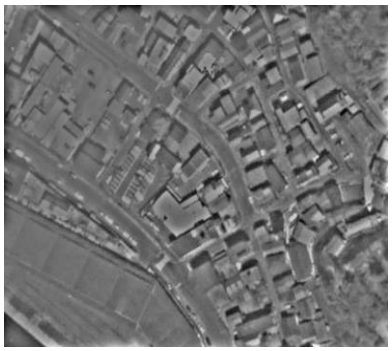


Image filtered using Butterworth
highpass filter function
(b)

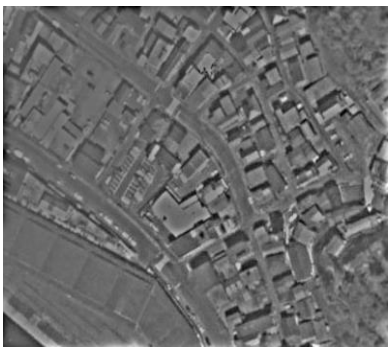
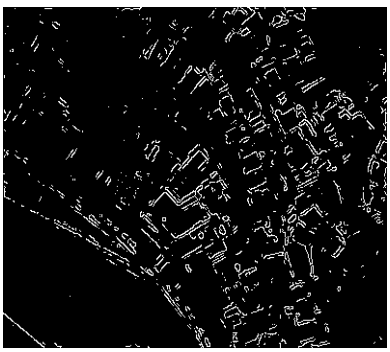


Image filtered using Gaussian
highpass filter function
(c)



Edge detection with the Canny operator

Figure 4.7*:* (1) Image filtering by means of Fourier transformation with different highpass filters and resultant edge detection with the Canny operator*.*

Fourier filtering above does not drastically improve the situation, given that numerous details are still missing (Figure 4.7 (1)). Homomorphic filtering yields the best results, with the detected contours evenly distributed across the resulting image (Figure 4.7 (2)).

(2) Image homomorphic filtering

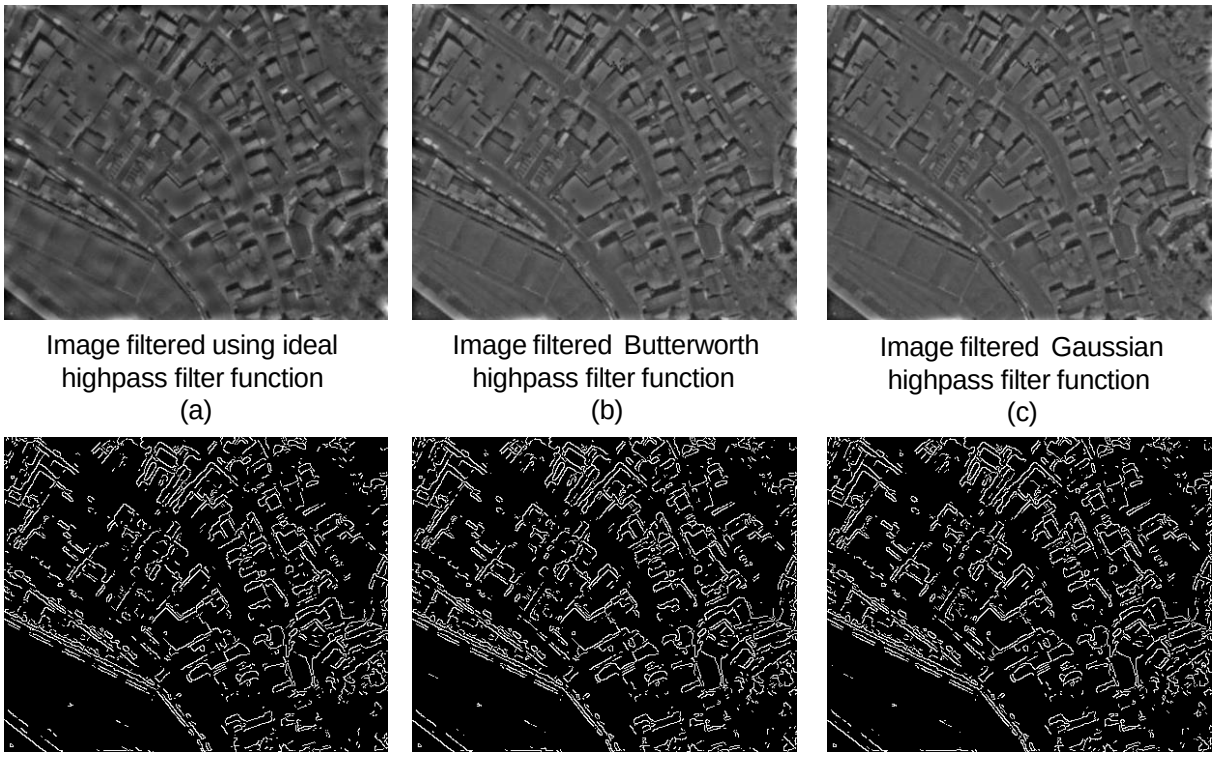


Figure 4.7: (2) Homomorpfic filtering with different highpass filters.

4.4 Selection of vector data

Destruction of urban areas can be caused by different catastrophic events, including natural or human-induced activities. Earthquakes are the most dramatic example of such events; they are characterized by the large loss of human lives, difficulties in eliminating the consequences of the disaster and large financial losses.

A crucial element of the rescue measures intended for the relief phase is an effective system of emergency management. During the first days after the event, numerous problems must be rapidly solved, with administration of medical aid, coordination of rescue activities, evacuation of injured persons, providing accommodation for the survivors, etc. Emergency management requires knowledge about the state of objects such as road networks, electricity network, bridges, buildings related to detrimental production, hospitals, police departments, fire stations, schools, nurseries, apartment houses, etc. Naturally, these building types have different significance for emergency management, which must be taken into account in succession and the detail of object investigations. For example, a complete destruction of storage facilities can be left without attention, but even a suspicion about a

minor destruction of hazardous substance storage must be checked on the site. Depending on the object, the results of an AI system can be accepted only with a sufficient reliability of its classification, otherwise the analysis must be repeated by a human expert. Not only does the type of building play an important role, but also which, where, when and under which circumstances the catastrophe happens. The priority of objects can be defined depending on the following criteria:

1. Potential influence on the environment.
2. Transport connections.
3. Extension of affected area.
4. Number of potentially injured persons.
5. Relation to the relief activity.
6. Areas with enhanced risk of destruction e.g. due to geological features.
7. Objects with enhanced risk of destruction e.g. due to some design features.
8. Living condition of survivors.
9. Time of day.
10. Time of year, etc.

The vector maps included in a GIS can provide comprehensive information concerning the affected area (geology, relief, water resources, etc.) and the objects (construction, age, location, etc.). Moreover, a GIS includes numerous tools for processing vector data and modelling processes that influence the rescue measures. The rational choice of vector objects involved in the subsequent analysis enables a significant improvement of the efficiency of emergency management, and thus GIS proves a very flexible tool set for various selection strategies.

4.5 Feature extraction

Independent of the classification approach, the results of the final classification strongly depend on the features used to depict the information about objects. The features are the quantitative measures of image/vector object properties. Ideally, they should refer to the differing information, identifying the most effective characteristics that are specific for all objects belonging to the same class (e.g. class of intact objects or destroyed objects).
A human expert can infer the state of a building based on the visual analysis of texture and contours in the roof image. Moreover, these indicators of building state can also be used in an AI system. The objective of the feature extraction in this study is to depict information about the integrity of building contour and roof image texture. In the further development of the proposed technique, the feature set can be extended to the other indicators used by human experts.

4.5.1 Detected Part of Contour

In most cases, visual interpretation of the remotely sensed images enables identification of building state based on the integrity of its contour. An intact building usually has markedly visible boundaries. In the case of damage or destruction, the contour of the building can be partially corrupted (or displaced) or completely lacking. Thus, a possibility to recognize the building's contour in the image is one of the most representative indicators of its state. In order to extract this information from the available data and transform it into a form of a certain numerical characteristic, a new feature, entitled 'Detected Part of Contour' (DPC), has been developed (Sofina et al., 2011). This feature reaches a maximum value of 100% if the contour of the investigated building can be entirely identified. The calculation of DPC can be described briefly as follows. An edge detection algorithm is applied to the filtered remotely sensed image to extract building edges. The developed algorithm yields a raster map with pixel values corresponding to the direction of the detected edges. The pixels that do not belong to any edge have a 'no data' value. On the vector map, test points are selected along the respective contours. A search area is defined on the raster map around these points, where pixels with appropriate contour directions are counted. The DPC value is calculated as the ratio of the number of detected pixels and the number of pixels expected for the intact building. A general scheme for calculation of DPC is presented in Figure 4.8. Each step of the calculation method is discussed in detail in the following sections.

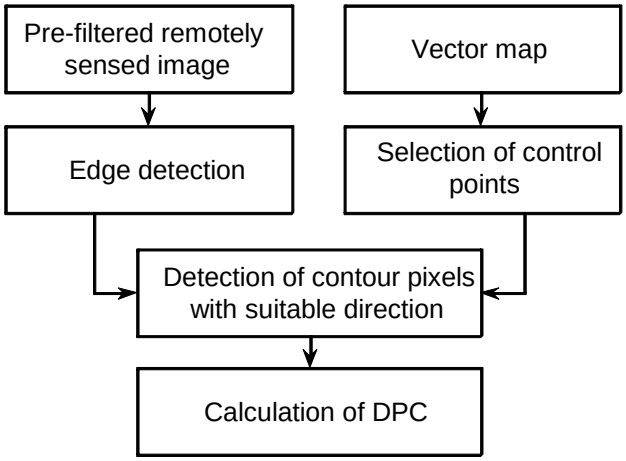


Figure 4.8: Main steps involved into the DPC calculation.

4.5.1.1 Edge detection

The edges on the pictures are represented as drastic changes in the brightness perpendicular to the edge line. Taking derivative (e.g. by means of Roberts, Prewitt or Sobel

operators) of the brightness function, the edges can be detected as peaks in derivative magnitude. The derivative is a vector pointing to the direction of greatest rate of brightness change, i.e. perpendicular to the contour line. Thus, the direction of derivative can be used for the definition of contour direction. For further simplification of the peak search, the second derivative can be used, whose magnitude takes a zero-value at the middle of the peaks. Most edge detection methods can be grouped into two categories: detecting the local maxima or minima of the first derivative (e.g. Roberts, Prewitt or Sobel operator); and detecting the zero-crossing of the second derivative, with the most popular methods including the Laplacian operator and Gaussian smoothing (Gonzalez & Woods, 2002).
In this study, the Canny edge detector (Canny, 1986) is employed. It is a widely used algorithm developed to suit the following requirements:

1. The detection of all and only real edges;
2. Minimal difference in the localization of detected and real edges; and
3. One real edge should not be detected as more than one edge.

The algorithm results in one pixel wide edge lines as an output, providing the perfect basis for a follow-up comparative analysis. The Canny method can be described by the following steps:

1. Noise reduction by filtering with Gaussian filter: the kernel of Gaussian filter with standard deviation value of 1.4 is expressed as:

$$\mathrm{B} = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

2. Derivative of the brightness function: the derivative can be found by a convolution operation with a Sobel kernel for horizontal $(\mathrm{G_x})$ and vertical $(\mathrm{G_y})$ components of derivative. The corresponding Sobel kernels are expressed as:

$$\mathrm{K_{G_x}} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \mathrm{K_{G_y}} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The magnitude of the derivative can be obtained as:

$$\mathrm{G} = \sqrt{\mathrm{G_x^2} + \mathrm{G_y^2}} \tag{4.30}$$

The direction of the derivative is found as:

$$\theta = \tan^{-1}\left(\frac{Gy}{Gx}\right) \quad (4.31)$$

Given that the edge is perpendicular to the direction of the derivative, the edge orientation can be found as:

$$\alpha = \tan^{-1}\left(\frac{Gy}{Gx}\right) + 90° \quad (4.32)$$

3. Non-maximum suppression: it is the edge thinning procedure. At each pixel, the magnitude of the derivative is compared with the values at the neighbor pixels along the direction of the derivative. If the comparison shows that the value of the magnitude is not greater than those of chosen neighbors, it is set to zero. As a result, lines of one-pixel width mark the top of the magnitude peaks are obtained.
4. Double thresholding: using two threshold values, the found contour pixels are split into three groups: non-contour pixels, weak contour pixels and strong contour pixels.
5. Edge tracking: the strong contour pixels are interpreted as certain edge pixels and included in the final edge image. The weak contour pixels are only included if they are directly connected with the strong pixels or through the sequence of other weak pixels.

The obtained angle values of α (see Equation 4.32) are within the range of $[0, 180°]$, which is classified into four intervals (see Figure 4.9).

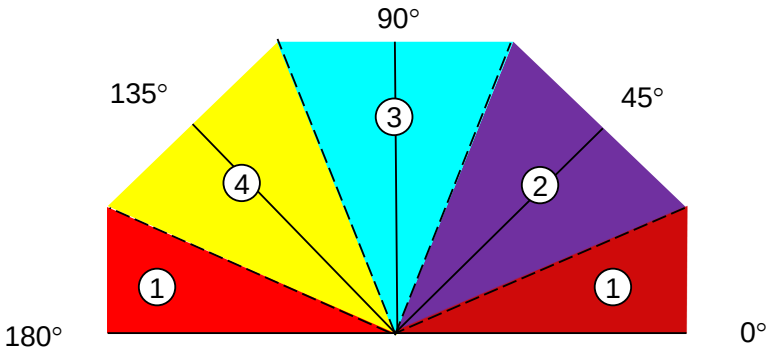
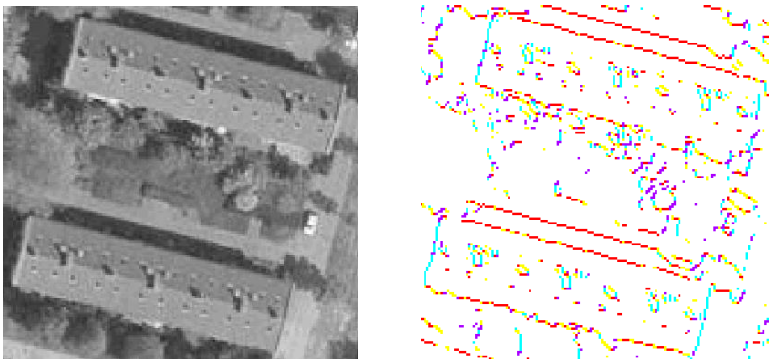


Figure 4.9: Four intervals of edge directions. The colors are used to present the orientation intervals on the edge image.

The Canny algorithm yields an image where pixels are marked as either an edge pixel or a non-edge pixel, while the pixels have values corresponding to the contour orientation interval. The edge image is shown in Figure 4.10, with all contours pixels colored according to their orientation intervals.



Original image Edge detection

Figure 4.10: Result of the edge detection.

4.5.1.2 Detection of matched parts of contours

The vector map illustrating the original urban state presents the buildings as polygons, whereby each side of the building is split into small segments to search for contours on the raster map created by the edge detection. At the center of each segment, a control point and an angle to the horizontal axis (the contour direction) are defined: $\alpha \in [0°, 180°]$. Schematic representation of the proposed methodology is shown in Figure 4.11.

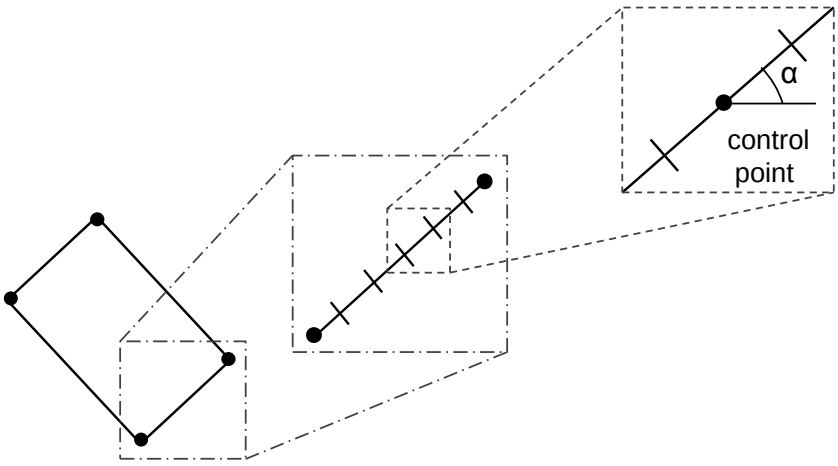


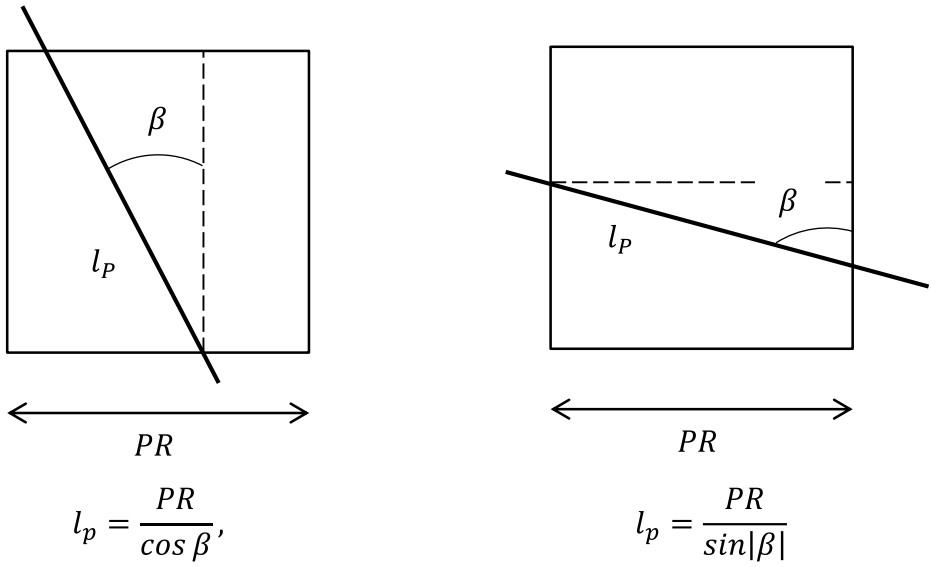
Figure 4.11: Splitting a building contour.

The control point is used as a center of the search area (see section 4.4.1.3) in the raster map, where the contour part is supposed to be detected. The search area in the form of a $P \times P$ pixel square (see Figure 4.12) is used, where the raster contour is supposed to cross the center of the search window. If this does not occur, a correction of the window position is performed. Based on the window size and contour declination angle, the contour length

inside the window can be calculated. Assuming that declination angle takes values in the range of $[-90°, 90°]$, the following expression for contour length can be derived:

$$l_P = \begin{cases} \dfrac{PR}{\cos\beta} & if\ |\beta| \leq 45 \\ \dfrac{PR}{sin|\beta|} & if\ |\beta| > 45 \end{cases} \tag{4.33}$$

Where l_P is the contour length inside the search area; P is the size of the search area in pixels; R is the image resolution (pixel size); and β is the angle of the contour to the vertical axis with $\beta\epsilon[-90°, 90°]$ (naturally, $\beta = \alpha - 90$).



$$l_p = \frac{PR}{cos\,\beta},$$

$$l_p = \frac{PR}{sin|\beta|}$$

given that $cos\,x\ = cos(-x\ )$

Figure 4.12: Part of the contour studied in the search window.

The above-expression can be written in a more compact form as follows:

$$l_P = \frac{PR}{max(sin|\beta|, cos\,\beta)} \tag{4.34}$$

Where l_P is the contour length inside the search area; P is the size of the search area in pixels; R is the image resolution; and β is the angle between the contour and the vertical axis (obviously, $\beta = \alpha - 90$).

Using the l_P value, the number of segments per building side is defined by:

$$N = int\left(\frac{L}{l_P}\right) = int\left(\frac{max(sin|\beta|,\ cos\,\beta)L}{PR}\right) \tag{4.35}$$

Here, $int(x)$ refers the floor of x, i.e. the largest integer less than or equal to x; and L *is the* side of the building contour (distance between two vertexes).
Consequently, the length of a segment is expressed as:

$$l_s = \frac{L}{N} = \frac{L}{int\left(\frac{max(sin|\beta|,\ cos\,\beta)L}{PR}\right)} \tag{4.36}$$

It is evident that $l_s \geq l_P$, and thus a small part of the polygon is dropped from the search procedure. This reflects that the arbitrary contour side cannot be covered without gaps with non-overlapping search windows of fixed size. If the side length is smaller than the l_P value calculated, the number of segments along the side is equal to zero (N=0) i.e. the side is not taken into account for the calculation of DPC. Figure 4.13 shows search areas along the building contour.

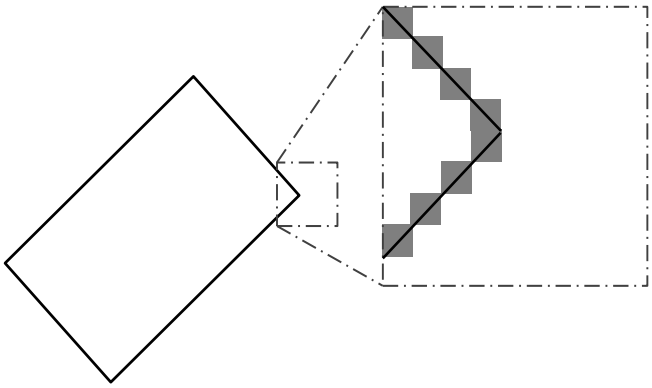


Figure 4.13: Placement of the search areas over the raster contour.

4.5.1.3 Size of search window

As indicated in the previous section, a search area, also called a search window, with a $P \times P$ pixel size defines an area in the edge image, in which pixels belonging to the analyzed building contour are searched. The center of the search window is located at the position of the related vector control point obtained by splitting the vector contour (Figure 4.14).
The size of the search window is an important parameter that defines the number of control points and affects the accuracy of the DPC calculation. The spatial size of the search window can be expressed as:

$$W = PR \tag{4.37}$$

where P is the size of the search widow in pixels and R is the resolution of the raster image.

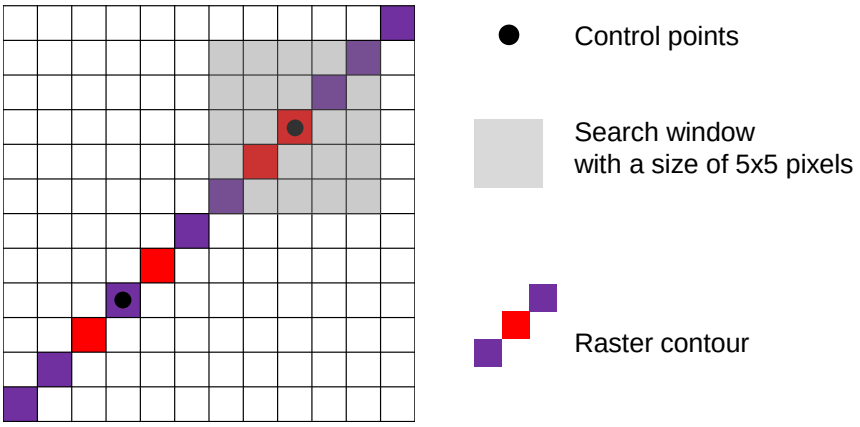


Figure 4.14: Placement of control points and search window over a raster contour.

The following criteria must be taken into account when choosing the window size:

1. Distance between neighboring buildings: the window must be small enough to avoid counting the pixels of neighboring buildings.
2. Accuracy of georeferencing of building contours: the window must be large enough to catch the pixels of contour, even if the position of vector and raster contours differ.
3. Image resolution.

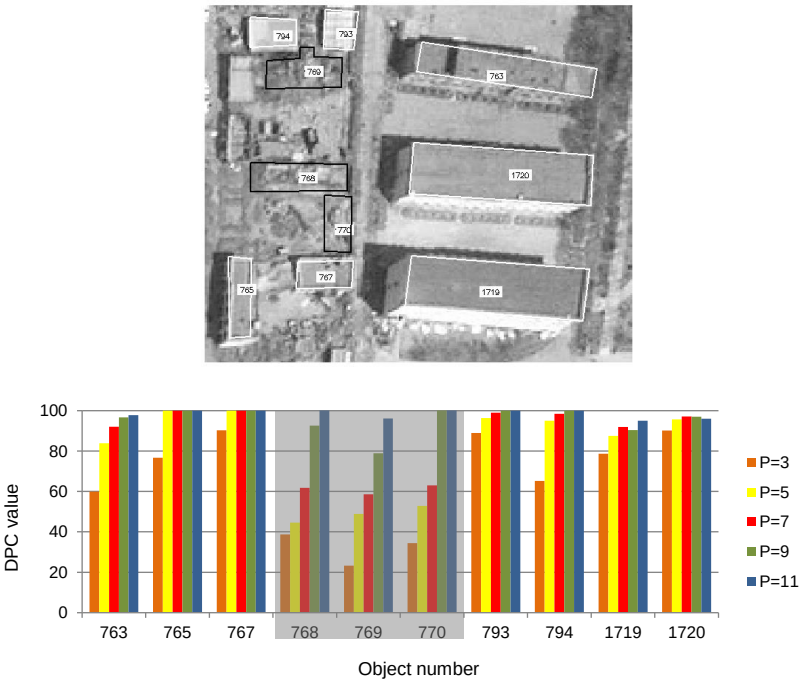


Figure 4.15: Effectiveness of DPC calculation depending on the size of the search area P. Objects with black contours (i.e. 768, 769, 770) are destroyed. Test image is overlaid with the vector map. Satellite image courtesy of Digital Globe (©Digital Globe 2010).

According to the experiments performed on the DPC calculation with different windows (3x3, 5x5, 7x7, 9x9 and 11x11 pixels), the optimal results are provided with window sizes of five

and seven pixels (Figure 4.15). The search window with the size of 3x3 pixels is too small, given that it only provides sufficient results if the center of the window is directly located on the investigated raster contour. If the center of the window lies outside the contour, a 3x3 window is incapable of covering it, thus leading to a DPC underestimation.

As expected, window sizes of 9x9 and more pixels yield a DPC overestimation, eventually reaching the maximum value of 100%. This is because they cover numerous pixels belonging to the nearby contours, thus providing unreliable results and eliminating the differences between intact and destroyed buildings. Therefore, window sizes of 5x5 and 7x7 pixels were selected as the most suitable for further calculation.

4.5.1.4 Correction of search window position

On a remotely sensed image, the detected contour of a building often does not pass through the center of a search area, which can lead to a loss of the contour pixels. To remedy this situation, the position of the windows is corrected when the number of found pixels is less than P. First, the center of mass for the found pixels of the building contour is calculated as follows:

$$X = \frac{\sum x_i}{N} \qquad Y = \frac{\sum y_i}{N} \tag{4.38}$$

where N – number of detected contour pixels; and x, y their coordinates, respectively.

The center of the search area is moved to the position of a pixel that is the nearest to the center of mass and has the suitable direction. Thus, this ensures that the detected edge of the contour passes through the center of search area (see Figure 4.16).

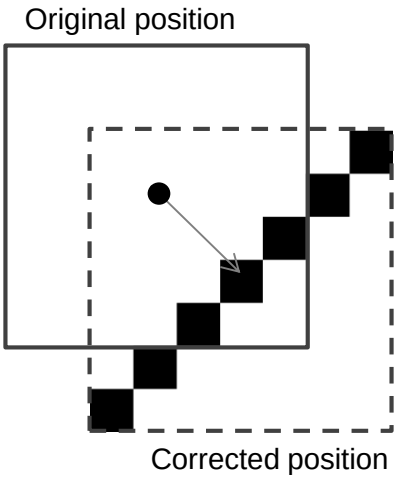


Figure 4.16: Correction of the search window position.

4.5.1.5 Calculation of DPC

The part of the contour is assumed to be intact if the providing search area contains at least P pixels of the contour (see Figure 4.17).

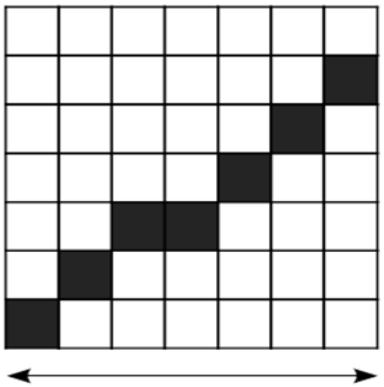


Figure 4.17: Raster contour inside the study area.

In order to reduce the influence of excess pixels on the final result, maximum number of P pixels per search area is taken into account. Considering the whole contour, DPC is defined as:

$$DPC = \frac{\sum_i^{N_P} min(P|N_i)}{N_P P}\ 100\% \qquad (4.39)$$

Here, N_i is the number of pixels found in the i-th search area; P is the size of search area in pixels, and N_P is number of search areas.

The search area can cover the pixels that do not belong to the contour of the building. In order to reduce the influence of such pixels, pixels with direction value close to the direction of the considered control points are counted. The pixel is considered suitable if it has a value of one of two angle segments closest to the angle of the studied contour part (see Figure 4.18).

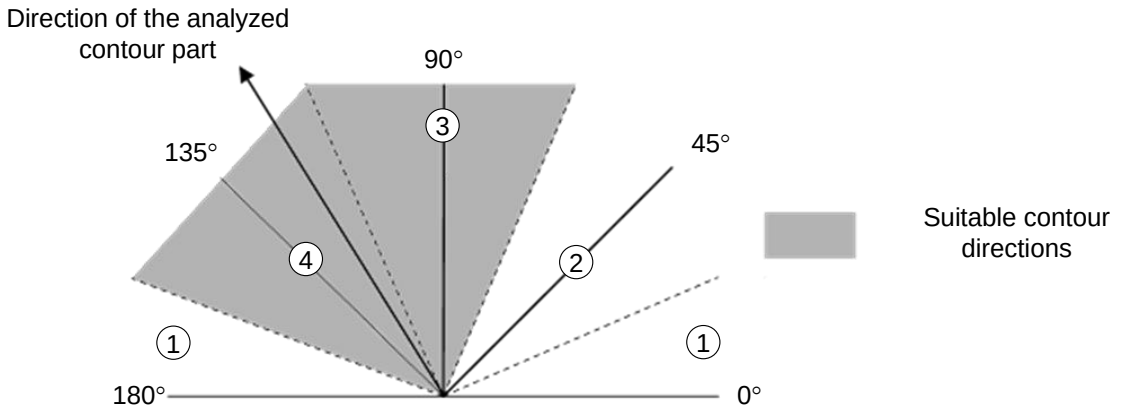


Figure 4.18: Definition of the suitable direction of the contour pixels.

4.5.2 Calculation of textural features

Various applications of pattern and object recognition use per-pixel image classification based on the analysis of spectral information. However, as a rule, it is difficult to obtain reliable results solely by means of spectral analysis. Spectral information of objects on the Earth's surface is changeable and depends on different factors, such as relief, soils, climate and geographical position. Besides, for an object represented by a restricted image area, it is necessary to have parameters describing the area in its entirety. Such parameters can be obtained by means of texture analysis. Texture is one of the most powerful characteristics for understanding and interpreting pictorial data. The features obtained by texture analysis describe an image's structure, based on the positional relationship of tonal variation within a considered area. Textural features play a crucial role in many applications of computer image analysis for classification and segmentation based on information concerning the spatial distribution of intensity variations in the images. Methods of texture analysis have been widely used in a variety of application fields, including remote sensing, medical image processing, the automatic detection of defects in images of textiles, carpet wear and automobile paint, etc. The major issues can be divided into the following four categories (Materka & Strzelecki, 1998):

1. Feature extraction – calculation of different characteristics, providing a numerical description of image texture properties.
2. Texture discrimination – image segmentation, separating the original image into homogeneous areas corresponding to different texture classes.
3. Texture classification – assigning an investigated sample image to one of the predefined categories.

4. Shape from texture – reconstruction of a 3D scene by using texture information from a 2D image as reference data.

A wide variety of techniques describing image texture has been discussed in numerous reviews and comparative studies (Van Gool et al., 1985; Zhang & Tan, 2002; Tuceryan & Jain, 1998; Materka & Strzelecki, 1998; Randen & Husoy, 1999). A technique for efficiently distinguishing textures must be chosen based on the type of textures considered. Figure 4.19 provides some examples of different textures. The types of texture can be defined based on the approaches given below:

1. By nature: artificial – graphical patterns on a neutral background; natural – sand, water, grass, forest, etc.
2. By surface structure: structural – formed by geometrically correct repeated elements; stochastic – formed by random tonal variation over the area.
3. By a shape of structural elements: wavy, spotted, linear, etc.

Among the known methods of textural image analysis, certain commonly used approaches can be emphasized: structural (geometrical), statistical, model-based and signal processing methods.

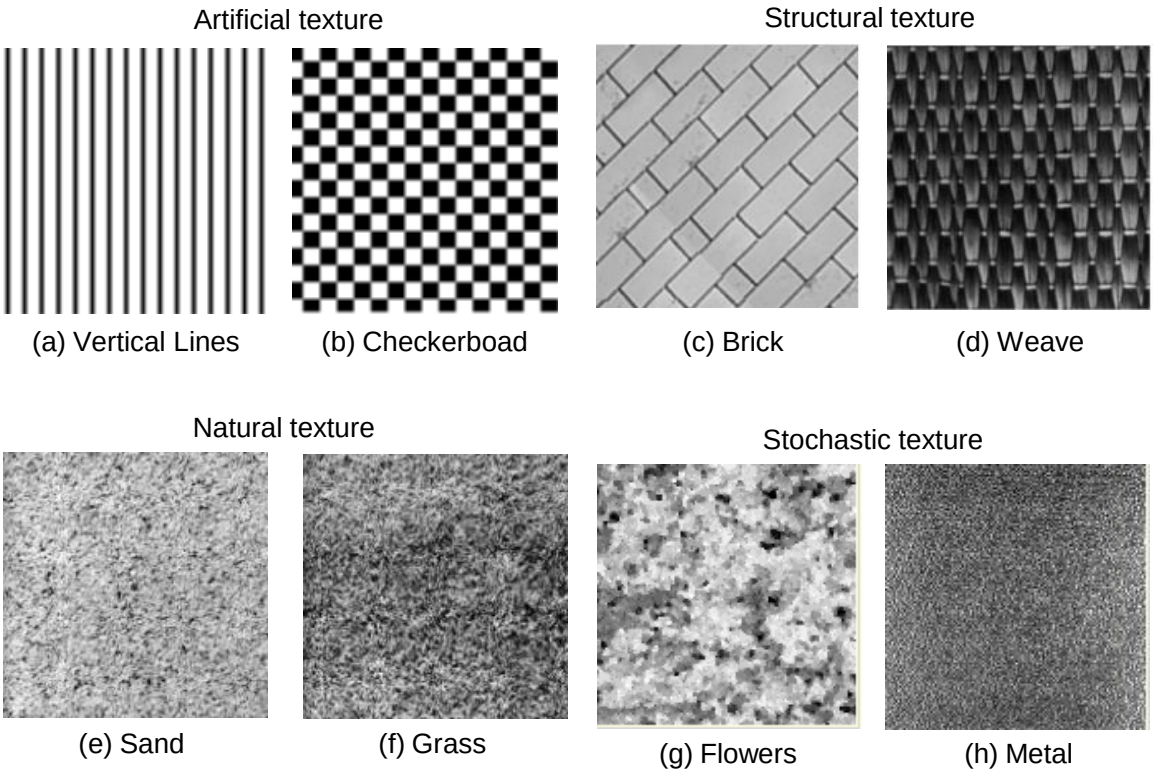


Figure 4.19: Examples of different textures (Brodatz, 1966).

Structural methods deal with the arrangement of image primitives, attempting to describe these primitives and the rules defining their spatial organization. The primitive elements can be extracted by edge detection, adaptive region extraction or mathematical morphology. After the identification of the primitives, the computation of statistics is performed by analyzing the placement dependencies of the elements with specified properties, such as intensity (see Horn, 1975), area, elongation, orientation, magnitude, compactness, etc. (Baheerathan et al., 1999).

The advantage of the structural methods is based upon their applicability to texture analysis tasks and texture synthesis. Additionally, this approach yields a good symbolic description of the image. However, the main disadvantage of the structural methods is their inefficiency for image possessing of texture elements without clear geometric shapes.

Statistical approaches represent the texture indirectly by computing non-deterministic properties embodying the distribution and relationships between the grey levels of an image. The local feature statistical methods can be classified into first-order (one pixel), second-order (two pixels) and higher-order (three or more pixels) statistics. The first-order statistics evaluate texture properties based on single pixel values without taking the positional relationship of image pixels (average values or variance) into account. The second- and higher-order statistics estimate texture properties based on two or more pixel values and their relations.

The most popular features in this group are based on the Grey-Level Co-occurrence Matrices (GLCM) suggested by Haralick (Haralick et al., 1973). Methods such as autocorrelation function, grey level run lengths, grey level difference method and Fourier power spectrum analysis are reported as relatively poor with respect to automatic texture classification (Conners & Harlow, 1980, Weszka et al., 1976). The detailed description of the algorithms mentioned above can be found in Sonka (1999) and Pratt (1991).

The statistical methods have the advantage of being a simple calculation approach with a high computing rate. Methods based on the generation of a co-occurrence matrix provide more accurate results of image segmentation compared to other methods that belong to this category.

Model-based methods including the approaches based on Gaussian Markov random fields (Cohen et al., 1991), Gibbs random fields (Derin & Elliot, 1987) and Wold model (Fang & Picard, 1996), involve the construction of an image model representing the essential characteristics of texture. The model can be generated as a probability model or as a set of basic functions, with the obtained parameter values used to characterize texture images.

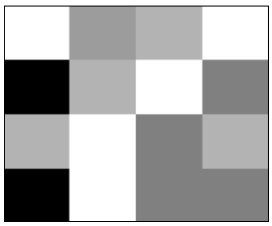
One such example of a model-based approach is the usage of a fractal decomposition in a method proposed by Mandelbrot (Mandelbrot, 1983). This method has become popular in

modeling a statistical quality of natural surface roughness and self-similarity in different scales. In the context of texture analysis, the fractal dimension is generally very efficient for decrypting the naturally occurring textures such as water surface, mountains or clouds.
Model-based methods are the most effective in terms of noisy image processing.

Signal processing methods focus on techniques of texture extraction using image filtering in the spatial or frequency domain. Examples of such methods are spatial domain filters, including Laws masks (Laws, 1980), local linear transforms and various masks for edge detection (e.g. Sobel, Roberts operators e.g.), or a set of moment-based approaches proposed by Tuceryan (Tuceryan, 1994).
The most commonly known instrument for signal processing methods is Fourier transformation (Weszka et al., 1976), illustrating the global frequency content of the image. However, this provides a poor performance due to the lack of spatial localization. A classical way of incorporating the spatial dependency into the Fourier analysis is through the use of a window function (Bajcsy & Lieberman, 1976).
The application of statistical approaches to remote sensing images has become particularly popular in the classification of built-up regions. Second-order statistics are the most effective measurements and are based on the generation of the GLCM that yields valuable information for the classification of urban areas. The efficiency of the GLCM features in respect to change detection in crisis areas has been proven in detail by Tomowski et al. (2011) and Klonus et al. (2012). Based on these investigations, the GLCM features were chosen for texture description within the change detection procedure developed in the framework of the performed study.
The following section provides a more detailed description of the general procedure for GLCM generation.

4.5.2.1 Grey-level co-occurrence matrix

Originally introduced by Haralick (Haralick et al., 1973), the spatial grey level co-occurrence matrix (GLCM) is a well-known statistical tool for extracting second-order texture information from images. GLCM describes the probability of the grey levels i and j of two neighboring pixels at a certain distance and angle. The following example illustrates the calculation of a GLCM for a 4×4 image (Figure 4.20(a)) with four grey-tone values 0-3 (Figure 4.20(b)) for vertical direction (90°).



(a)

0	2	1	0
3	1	0	2
1	0	2	1
3	0	2	2

(b)

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

(c)

2	2	2	1
2	0	3	2
2	3	2	0
1	2	0	0

(d)

Figure 4.20: Calculation of a GLCM. (a) 4×4 input image; (b) four grey-tone values 0-3. General form of all grey-tone spatial-dependences for image with grey-tone values 0-3 (c). Calculation of a grey-tone spatial-dependence matrix for vertical direction (90°) with inter-pixel distance 1 (d).

The size of a calculated co-occurrence matrix depends on the value range of the image analyzed. According to the matrix containing all possible grey level combinations (Figure 4.20(c)), a frequency of two neighboring elements i and j is line-by-line counted at the specified inter pixel distance d and angles (in this case 90°), resulting in the directionally dependent symmetric co-occurrence matrix (Figure 4.20(d)).The orientations used for calculation of the co-occurrence are presented in Figure 4.21.

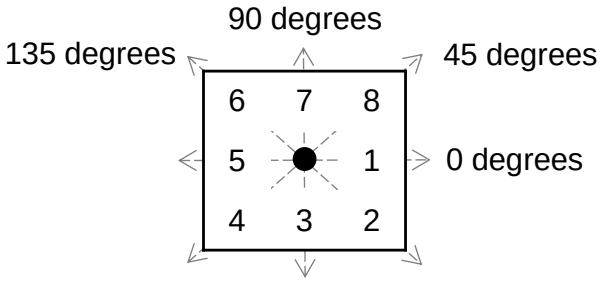


Figure 4.21: GLCM directions for calculating the Haralick textural features (Haralick et al., 1973).

However, the matrices obtained are not invariant to the size of the image, which can adversely affect the resulting performance. Additionally, the total number of grey level pairs compared is different due to the angular relationships. To overcome such problems, the normalization of the co-occurrence matrix is indispensable, whereby each value in the matrix is divided by the sum of all the values:

$$p(i,j) = \frac{V_{r,c}}{\sum_{r,c=0}^{G-1} V_{r,c}} \tag{4.40}$$

where V is a value in the GLCM; r and c are rows and columns in the GLCM; G is the total sum of the values; and $p(i,j)$ is the normalized GLCM (Table 4.3).

0,0833	0,0833	0,0833	0,0417
0,0833	0	0,125	0,0833
0,0833	0,125	0,0833	0
0,0417	0,0833	0	0

Table 4.3: Normalized GLCM.

Various texture measures can be directly computed from the GLCM, with the most frequently used set of 14 features proposed by Haralick including Angular Second Moment, Contrast, Correlation, and Entropy. Table 4.4 shows a representative example of some textural features calculated for the image from Figure 4.20(a). The commonly adopted approach for the employment of texture analysis for image interpretation, classification or segmentation does not utilize the angularly dependent features directly, but rather their average values (Haralick et al., 1973).

Angle	ASM	IDM	Inertia
0°	0,1146	0,3833	2,8333
45°	0,1358	0,7	1,6666
90°	0,0903	0,45	2,5
135°	0,1235	0,4444	2,4444
Average	0,1160	0,4944	2,3610

Table 4.4: Textural features obtained from grey level spatial-dependence matrices.

4.5.2.2 Object-based GLCM calculation

The conventional approach for computing co-occurrence matrices and other texture measures, local variance of brightness values, as well as performing image filtering, edge detection and image classification exploits a fixed size rectangular sliding window. Figure 4.22 illustrates a general routine to derive the textural features from a GLCM, where the window moves from left to right with a start position at the first image pixel. Thereby, the corresponding probabilities of grey level pair distributions are extracted from GLCM for the related image pixel in the defined window. As a result, a new image containing the texture measures for each pixel (at the center of the window) is calculated.

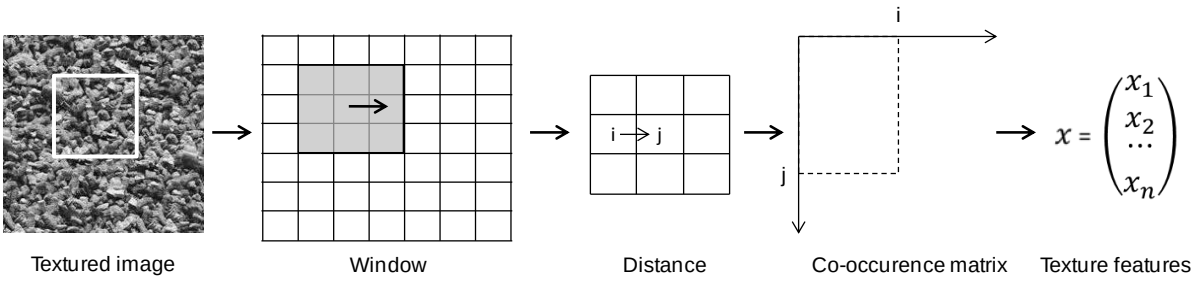


Figure 4.22: General procedural steps for deriving textural features. Figure adapted from Lehmann et al.,1997.

A window-based technique is a frequently used in digital image processing to define the local information content around a single pixel. The determination of the optimum window size reflects a crucial point for obtaining valuable information and texture representation within the window. A number of contradictory conclusions have been reported regarding the window size and shape properties, including minimum, maximum and optimal window size that can effectively provide essential information to sufficiently characterize a sub-area of interest from a larger image region. Hsu (Hsu, 1978) and Dutra (Dutra & Mascarenhas, 1984) argued for the small window size, which is more effective for extracting micro textures, noting the negative effects of using large windows, including that image classification can be adversely affected by the impact of edge pixels. Nellis and Briggs (Nellis & Briggs, 1989) proposed large window sizes for investigating homogeneous landscapes. It has generally been reported that the appropriate size for per-pixel image classification ranges from a 3×3 to a 9×9 matrix of pixels (Irons & Petersen, 1981; Jensen J. R., 1979; Gong et al., 1992).

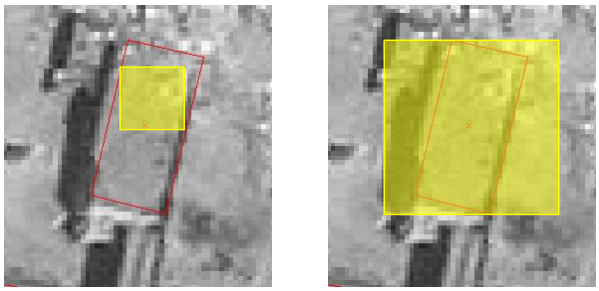
Thus, the ultimate objective of the analysis should also be considered in the choice of the window size. Image classification and the following separation into individual homogeneous regions require larger windows, given that a rather small window can cause significant errors within the homogeneous areas due to the small primitives detected. Additionally, a rather small window can contain insufficient statistical information in terms of providing a reliable description of the objects represented in the image. On the other hand, large windows can lead to unreliable results because they can be significantly larger than objects of interest, i.e. they cover redundant information outside the studied area.

In the case of building state detection, special interest is usually given to building roofs, given that they are more visible in remotely sensed images. Naturally, if the building is damaged or destroyed, the texture of its roof is usually changed, which makes it detectable by texture analysis. The roof of an intact building possesses a rather homogeneous texture and a low variation in the image brightness, whereas the image texture becomes more heterogeneous due to the fragments of destruction (see Figure 4.23).

(a) (b)

Figure 4.23: Image areas corresponding to (a) the roof of an intact building with a homogeneous texture and (b) the roof of a destroyed building with a heterogeneous image texture. Images are from Port-au-Prince, Haiti. Satellite image courtesy of Digital Globe (©Digital Globe 2010) and RIT.

Thus, the texture analysis in such a case should focus on the characterization of a specified image area corresponding to the building investigated and neighborhood within this area, rather than a single pixel centered in a sliding window and its neighborhood. Therewith, the exact positioning, shape and size of each window analyzed become known through the use of vector data. Given that the shape and size of buildings are not fixed and differ for each individual construction, the utilization of the conventional rectangular sliding window can be considered inappropriate and inefficient. A small window does not include the whole area of the building (Figure 4.24(a)), thereby providing insufficient information concerning the object investigated. A rather large window can also produce unreliable results due to the negative impact of edge effects, as well as covering image areas outside the object to be investigated (Figure 4.24(b)).



Window 9x9 pixels (a) Window 40x40 pixels (b)

Figure 4.24: Placement of a rectangular window over the building in the image: (a) small 9×9 window and (b) large 40×40 window. 9×9 window is too small and it does not cover the entire building. 40×40 window is too large and it covers redundant pixels outside the building.

In order to investigate an image area corresponding to an individual building, an etirely different approach to window configuration for defining the sample size of the image is required. Therewith, the size and shape of the window should be initially non-fixed and able

to be dynamically computed and adapted to each building, with the out-of-building information truncated by the window boundary.

Dillworth et al. (1994) deviate from the conventional window transformations with a rectangular ‘geometric’ window and employ the ‘geographic’ window originally proposed by Merchant (Merchant, 1984). A geographic window is able to adapt the ‘size and shape in response to changes in local landscape characteristics, so that any patch within the original window dimensions is completely included in the dynamically changing geographic window’ (Dillworth et al., 1994). The experimental study was performed with the Spatial Measurements Package (Whistler, 1989), providing the spatial characterization of classified data by means of a geographic and geometric window for a test area in northeastern Colorado. Given that the software initially requires a classified image as its input, Merchant suggested a ‘region-growing’ algorithm for defining areas with similar land-cover characteristics, which are further used for the configuration of geographic windows. From their experiments, they conclude that the geographic window represents a new approach to extract statistical measurements and provides more complete and accurate information concerning the landscape structure in comparison to employing the prevailing fixed-size rectangular window.

Hodgson (1991) also argued for an automatically adapted window size, with multiple windows of various shapes and sizes proposed to characterize the homogeneity of a landscape.

A modern object-based GIS approach can considerably simplify the procedure of generating the window due to the accurate definition of object geometry and location. Given that an application of vector data containing information concerning the original urban layout is an initial prerequisite in this work, the extraction of the object dimensions and precise positioning can be relatively effectively implemented. The generation of the automatically adapted window for calculating the statistic texture measurements corresponding to each vector object can be described as follows (Figure 4.25). To analyze the image area related to a specific building, a small rectangular fragment equal to the size of a vector object under investigation is overlaid on the image. A binary mask of the building footprint is subsequently created by converting the related vector object in a raster format, which enables selecting the pixels belonging to the studied area, with the image part representing the building patch that falls entirely within the specified constraints.

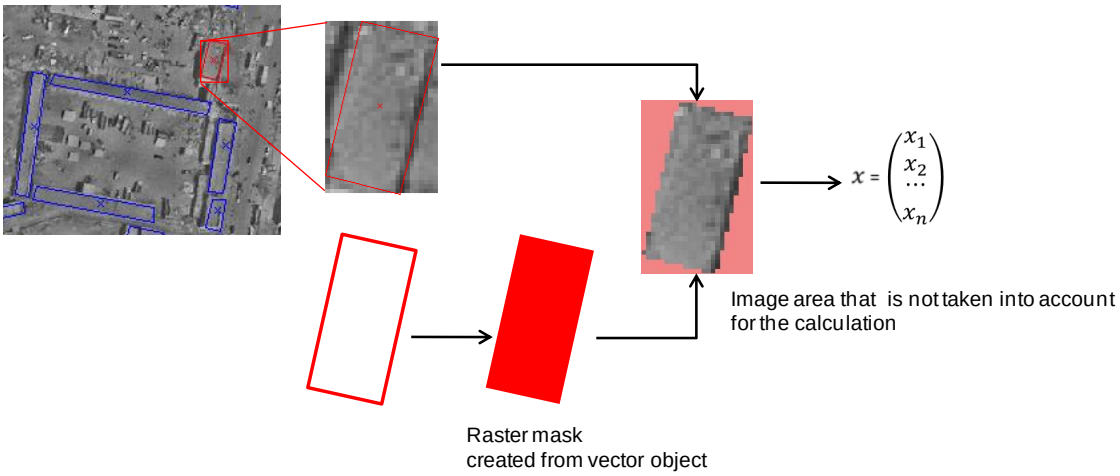


Figure 4.25: Generation of automatically adapted window with a size equal to an individual vector object.

Assuming that only pixels corresponding to the building area have to be used in the calculation of textural features, the equations for grey-tone spatial-dependence matrices presented by Haralick et al. (1973) can be modified as follows:

$$P(i,j,d,0°) = \#\{((k,l),(m,n)) \in (B \times B) \mid k-m=0, |l-n| = d, I(k,l)=i, I(m,n)=j\} \quad (4.41)$$

$$P(i,j,d,45°) = \#\{((k,l),(m,n)) \in (B \times B) \mid (k-m=d, l-n=-d) \text{ or } (k-m=-d, l-n=d), I(k,l)=i, I(m,n)=j\} \quad (4.42)$$

$$P(i,j,d,90°) = \#\{((k,l),(m,n)) \in (B \times B) \mid |k-m|=d, l-n=0, I(k,l)=i, I(m,n)=j\} \quad (4.43)$$

$$P(i,j,d,135°) = \#\{((k,l),(m,n)) \in (B \times B) \mid (k-m=d, l-n=d) \text{ or } (k-m=-d, l-n=-d), I(k,l)=i, I(m,n)=j\} \quad (4.44)$$

where # denotes the number of elements in the set; and B is the set of pixels from the building area selected by the mask.

The matrix has to be normalized to remove dependency on the building size, as follows:

$$R = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P(i,j) \tag{4.45}$$

$$p(i,j) = \frac{P(i,j)}{R}$$

where R is a normalization constant; and N_g the number of grey levels in the input image (in this study $N_g = 256$).

Furthermore, it should be noted that the object-based procedure to generate the windows for the extraction of statistical measurements enables calculating a single feature value for each individual vector object (i.e for each individual building in the map), as opposed to an array of feature values being computed for every image pixel, which would be provided by a geometric window.

4.5.3 Feature assessment

A general problem for data classification is the selection of the most informative features. Unfortunately, there is no complete theory for an optimal synthesis of textural features that can produce a minimum number of classification errors. In practice, a set of features is still empirically selected for each specific study case. A frequently used approach to define the most informative features is the calculation of a large variety of statistic measurements that are further tested for their information content.

In order to improve the efficiency of the final classification, it is desirable to analyze the information content of the features calculated and to optimize the feature set, determining how many and which members should be selected accordingly. Moreover, the appropriately selected features enable reducing data processing time and performing the successful classification by means of even simple classifiers.

The calculation of all possible texture measurements can increase the amount of information, but in practice, a redundancy of initial knowledge does not cause a positive effect due to a probable data correlation and the unnecessary duplication of information. The optimization of the feature set can be performed by eliminating and combining of several features into one (Markovitch & Rosenstein, 2002). Another way to overcome this problem is the selection of features that originally depict the differing information. The key points in generating an optimum feature set are: (1) identification of the most effective characteristic that is proper to all objects belonging to a specified class, as well as the collection of features pointedly presenting this characteristic; and (2) rules or algorithms for the selection and assessment of

the most informative features.

As previously mentioned, an image area corresponding to a roof of an intact building is usually homogeneous, with a low variation of the image brightness. On the other hand, fragments of destructions can cause significant differences in the image grey values, thereby adversely affecting the image homogeneity. In order to extract information concerning the state of buildings from a remotely sensed image, image homogeneity is used as the basic indicator for developing a system of features.

Therefore, textural features that characterize the image homogeneity are taken into account for the generation of a feature set in this study, with four angular grey-tone spatial-dependency matrices computed through an inter-pixel distance of 1 (Table 4.5).

Textural feature	Equation	Description
Angular Second Moment (ASM)	$ASM = \sum_i \sum_j \{p(i,j)\}^2$	Measure of uniformity. High values correspond to very similar image texture.
Inertia	$Inertia = \sum_i \sum_j (i-j)^2 \, p(i,j)$	Characterizes availability of sharp borders and contours.
Inverse Difference Moment (IDM)	$IDM = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j)$	Measure of local homogeneity. High values indicate a high homogeneous image texture.

Table 4.5: Textural features extracted from grey-level co-occurrence matrices (where $p(i,j)$ is an entry in a normalized grey-tone spatial dependence matrix).

The roofs of intact buildings possess a rather regular texture orientation, mainly concentrated along the edges. Figure 4.26 shows the significant angular dependency of the IDM. On the other hand, there are no prevalent directions in the image texture related to destroyed buildings. The orientation of the texture is mainly randomly distributed due to variations in the image brightness, caused by fragments of destruction represented in the image.

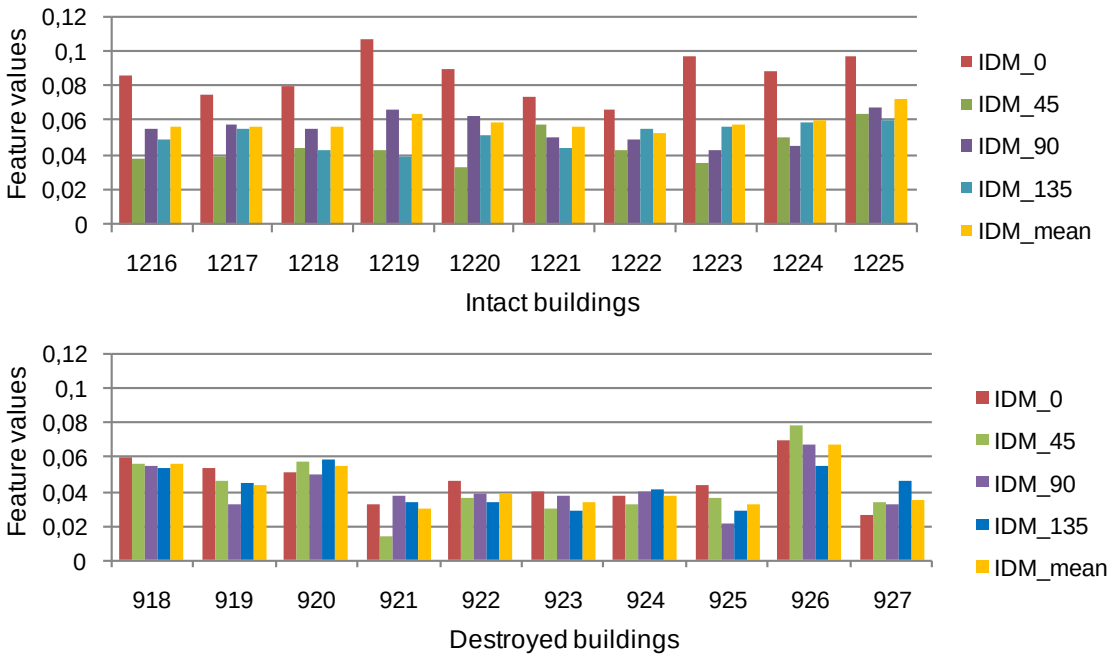


Figure 4.26: Example of distribution of angular IDM values for intact and destroyed buildings.

Using the maximum of the angular features, the presence of the textural orientation can be exploited for the identification of the building states. Calculation of average values results in loss of information concerning the texture orientation, consequently providing inaccurate results. Besides maximum and average values, minimum of the angularly dependent textural features is also investigated as indicator in this study. The feature capability to separate objects into two classes (destroyed and intact buildings) reflects a decision criterion of the feature selection.

The individual feature relevance for the separation of objects into two classes (positive and negative) can be estimated by the receiver operating characteristic method (ROC method). The ROC curve is plotted as a function of true positive rate (TPR) versus false positive rate (FPR) by changing the threshold value separating the classes (see Figure 4.27):

$$TPR = \frac{TP}{TP + FN} \tag{4.46}$$

$$FPR = FP/(FP + TN)$$

Here, FP = number of false classified objects of first class; TP = number of correctly classified objects of first class; FN = number of false classified objects of second class; and TN = number of correctly classified objects of second class.

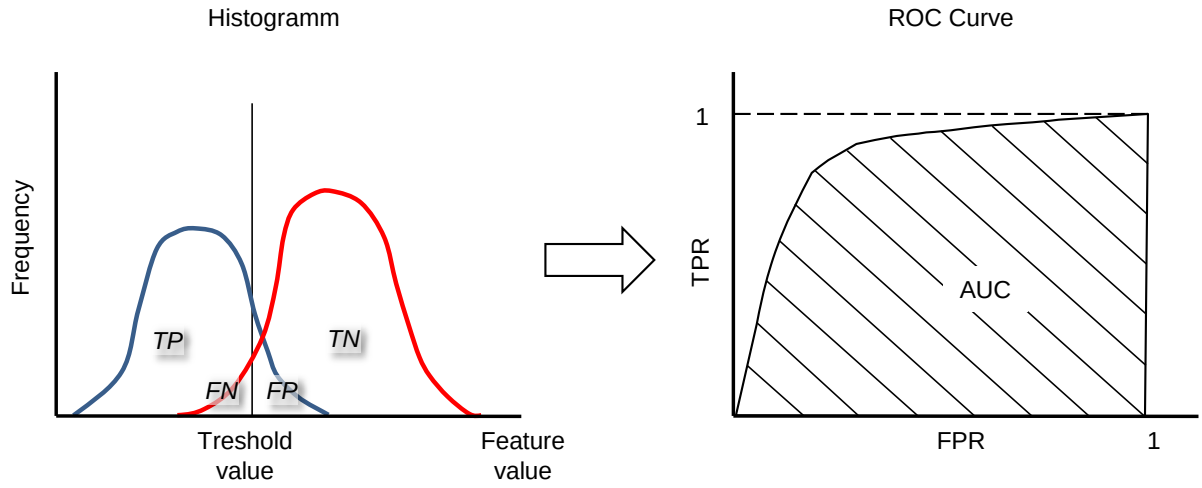


Figure 4.27: Calculation of area under receiver operating characteristic (ROC) curve.

The value of an area under the ROC curve (AUC) reflects its ability to separate classes based on a single feature. The value of the area can take values from 0.5 to 1, whereby a value of 0.5 corresponds to the worst possible separation and 1 to the best possible separation. Thus, the larger the AUC value, the better the separation that can be reached.
In this research, the effectiveness of the features was also analyzed by frequency distribution histograms which visually demonstrate the capability of the features to separate the object into different classes. The detailed analysis of the above considered features is provided in Chapter 5.

4.6 Classification

The amount of data is constantly growing, with processing often reflecting a challenge as a result. The relatively new data mining techniques are intended for extracting useful information from a large amount of data. Data mining gained its name from a pair of concepts: retrieving information from databases (data) and the extraction of different geological materials from the earth (mining). The term ‘mining’ reflects sifting a huge amount of raw material for the extraction of valuable information.
The term ‘data mining’ appears increasingly commonly on book covers and in scientific papers concerning data analysis. Due to their methods of artificial intelligence for multi-dimensional data analysis, data mining techniques support one- and two-dimensional statistics, which are widely used for data analysis. A comprehensive description of data mining techniques can be found in Fayyad et al. (1996).
The majority of analytical methods applied in data mining are based on long-known mathematical algorithms. They involve visual data analysis, demanding the participation of a

human expert, as well as automated analysis. In the practice of data analysis, the combination of visual and analytical data mining approaches is often most efficient. Especially for the processing of geoinformation, a visual data analysis is commonly indispensable for the post processing of data mining results.

Data mining implies data receiving, knowledge extraction, intelligent data analysis, pattern and relationship discovering. Accordingly, it is a multidisciplinary approach (Figure 4.28) that came into existence based upon different branches, including statistics, pattern recognition, artificial intelligence and database systems (Sumathi & Sivanandam, 2006).

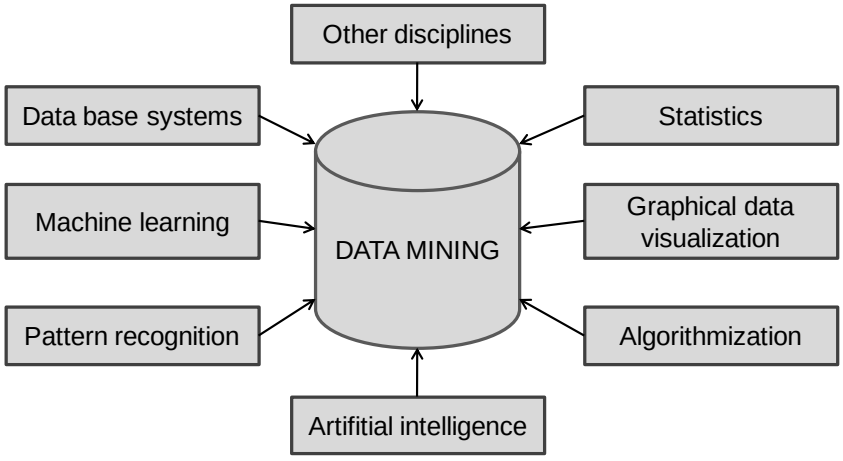


Figure 4.28: Data mining as a multidisciplinary approach (Sumathi & Sivanandam, 2006).

It is generally supposed that data mining tools are easy-to-use and it can be sufficient to start such a tool for a large terabyte database in order to instantly extract useful knowledge from the input data. In fact, there are, however, no universal methods of data mining. The development of a data processing procedure requires a good understanding of the input data, data acquisition techniques and the entire process of data analysis. However, even the best technologies of artificial intelligence cannot replace an expert; rather, they provide powerful tools and techniques to increase the efficiency of his work. Moreover, due to the interdisciplinary of this approach, a collaborative work of several specialists with different backgrounds is often necessary.

Nowadays, numerous data mining tools have been developed and successfully applied in industry and research. The technique has proved very efficient for problems of different branches such as business, medicine, security services, or satellite imagery processing.

A wide variety of data mining methods and algorithms (neural networks, decision trees, rule induction, k-nearest neighbor, support vector machine, regression, cluster analysis, as well as different techniques of data visualization) can be categorized into two groups:

unsupervised learning and supervised learning methods. The detailed discussion of these methods is provided in the following sections.

4.6.1 Unsupervised classification

Even without any information about the data structure, it can be possible to identify groups of similar objects within a given data set. The objects can be characterized by the properties typical for the members of its group. The identification of groups with similar objects within a data set is called unsupervised learning or clustering. Figure 4.29 shows a scatter plot of a data set containing objects described by two features. In the presented example, the objects have been combined into three groups.

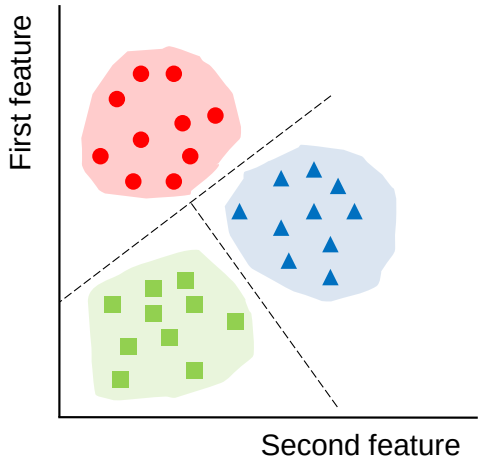


Figure 4.29: Example of unsupervised learning for a data set of objects described by two features.

Clustering provides an analysis of a data structure without any conclusion concerning the cluster properties. The result of clustering requires further interpretation based upon the properties of the considered objects, in order to describe the generated clusters. Assessment of clustering quality can be manually performed by setting control points or comparing the clusters provided by different methods. Depending on the data set, different clustering techniques can lead to differing results. A correspondence of similar clusters by means of different approaches indicates a high reliability of the achieved results.

As a rule, each method of clustering possesses a set of parameters that can be configured for a considered individual problem. The appropriate choice of parameters can significantly improve the performance of the methods. One of the most important parameters is a distance measure for any pair of objects, which defines a difference between the objects and can be represented by any scalar function $d(x, y) \geq 0$ satisfying the following four axioms:

1. $d(x,y) \geq 0$ (non-negativity).
2. $d(x,y) = d(y,x)$ (symmetry).
3. $d(x,y) + d(y,z) \geq d(x,z)$ (triangle inequality).
4. $if\ d(x,y) = 0, then\ x = y$ (identity of indiscernibles).

Scientific publications relating to the cluster analysis of different data types provide numerous methods for the calculation of a distance parameter (Deza & Deza, 2009), with those most frequently used presented in Table 4.6.

Distance metric	Equation
Euclidean	$d(x,y) = \left(\sum_{i=1}^{m}(x_i - y_i)^2\right)^{\frac{1}{2}}$
Manhattan	$d(x,y) = \left(\sum_{i=1}^{m}\lvert x_i - y_i\rvert\right)$
Chebychev	$d(x,y) = max_{i=1}^{m}\lvert x_i - y_i\rvert$
Minkowsky	$d(x,y) = \left(\sum_{i=1}^{m}\lvert x_i - y_i\rvert^r\right)^{\frac{1}{r}}$
Canberra	$d(x,y) = \left(\sum_{i=1}^{m}\frac{\lvert x_i - y_i\rvert}{\lvert x_i + y_i\rvert}\right)$

Table 4.6: The commonly used approaches to define the distance between instances (x and y).

where $d(x,y)$ - the distance between x and y objects;

1. x_i – the value of i-th feature of x object.
2. y_i – the value of i-th feature of y object.
3. m– the total number of features describing the objects in the set.

All features in the data set should be presented ‘’at the same scale”, i.e. normalized. Otherwise, a feature with the largest numerical values will dominate the metric calculation and all other features will have much less influence on the classification results.

Nowadays, numerous clustering algorithms have been developed, including connectivity-based clustering, also known as hierarchical clustering, centroid-based clustering or *k*-means clustering if the number of clusters is fixed to k, distribution-based clustering, density-based clustering, etc. (Hartigan, 1975; Clarke et al., 2009). In this research, the k-means clustering

was employed for the classification of building states. Accordingly, this method is described in further detail below.

The k-means algorithm is a popular clustering method. The k-parameter in the name of the approach stands for the number of expected clusters given as an input parameter. The main idea is to group the sample based on the distance to the predefined cluster's center. The algorithm can be summarized in the following steps:

1) k initial cluster centers are chosen;
2) assign to the each sample the cluster corresponding to the closest cluster center;
3) recalculate the cluster centers as the average of positions of samples belonging to the cluster; and
4) if at least one cluster position is changed, proceed to the step 2.

The result of a k-means algorithm is nondeterministic, i.e. different results can be obtained by specifying different initial positions of cluster's centers. For this reason, it is desirable to repeat the k-means algorithm several times and compare the results. In order to improve the convergence and result quality, the algorithm is often used with various extensions, such as different approaches for initialization of cluster center, different convergence criteria, etc.

In comparison with other clustering techniques, the k-means algorithm is fast, relatively robust and surprisingly modest in its resource consumption. Due to its simplicity, it is easy to be augmented, implemented and extended. However, the results of clustering can depend on the input order of the data, as well as the initial choice of center positions.

Cluster analysis methods are easy to use and do not require a training sample set, as it is necessary for algorithms of the supervised learning. The latter advantage is particularly important for the processing of remotely sensed data, given that the parameters and quality of the images significantly differ depending on the acquisition conditions. Indeed, it is often difficult to find a universal training data set optimal for all study cases. Figure 4.30 shows remotely sensed images presenting different types of built-up areas. Naturally, such a robust, fast and unsupervised method is very promising for a change detection technique and especially in terms of initial damage assessment.

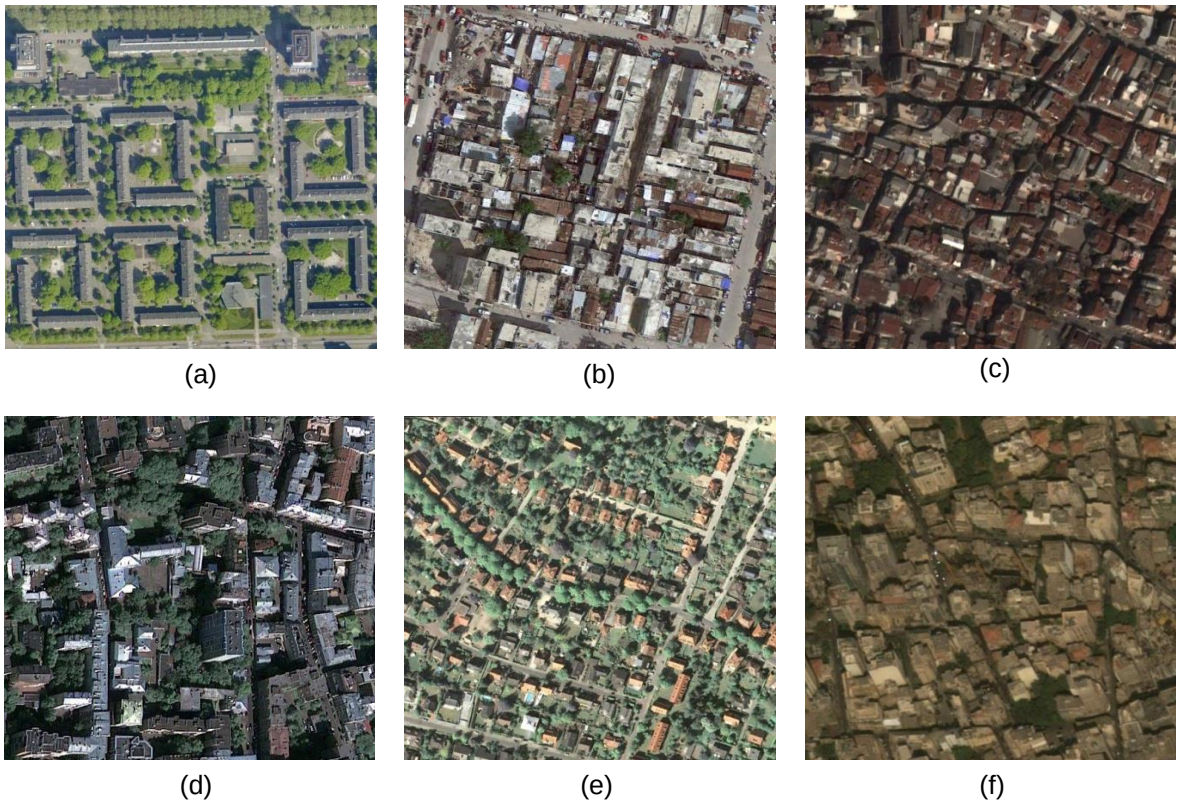


Figure 4.30: Examples of remotely sensed images corresponding to various types of build-up areas. Images courtesy Google Earth™ mapping service.

4.6.2 Supervised classification

Supervised learning or classification is a process that assigns one of the prespecified classes to the object in accordance with the object features. Construction of classification rules is based on the training set of objects for which the true classes are known. A general process of data classification can be divided into two steps (see Figure 4.31):

1. Model construction (learning) – construction of classification rules based on the training data set.
2. Accuracy assessment and model usage: the performance of classifiers is tested by using the test data set with the known classes. If the model yields an acceptable accuracy, it is used for the further classification of new objects with unknown classes.

As already noted, the supervised learning requires a set of samples with known classes. This set is randomly split into training and test data sets in a prescribed ratio. The former is used for model construction, and the latter for the assessment of the model accuracy.

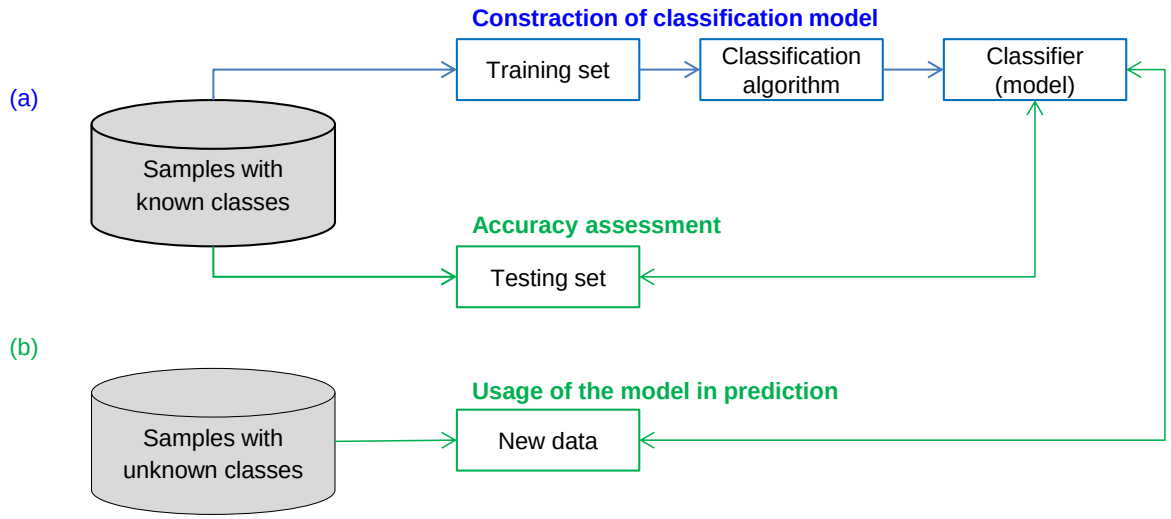


Figure 4.31: A two-step classification process. (a) the first step in the data classification process (construction of classification model); (b) the second step of data classification (accuracy assessment and further usage of the model in prediction).

A large number of techniques for supervised learning has been established, including logic-based algorithms (decision trees, learning set of rules), perceptron-based techniques (single layered perceptrons, multilayered perceptrons, radial basis networks), statistical learning algorithms (Naïve Bayer classifiers, Bayesian networks), instance-based learning (nearest neighbor algorithm), and support vector machines. (Olson & Delen, 2008; Tso & Mather, 2009). The description provided below only focuses on the techniques used in this research, including the k-Nearest Neighbor algorithm (k-NN) and Support Vector Machines (SVM).

The Nearest Neighbors method is one of the simplest machine learning algorithms, based on an assumption that an object must have the same class as majority of objects from its neighborhood. In order to define the neighborhood of the object, the distance measure must be chosen. Usually, the Euclidean measure is used to measure the distance between objects, while the other possible distance measures are described in section 4.5.1.

The decision about the class of a considered object is taken based on the classes of k nearest objects from the training set. The object class is defined by a majority vote of its neighbors. The learning phase is very simple, and is only devoted to the acquisition of a training set. The methods with such a simple learning phase are called lazy learners. Naturally, the learning approach allows for an easy improvement of the classification results by extension or modification of the training set. On the other hand, the classification procedure of k-NN method is resource consuming and can be extremely slow. For instance, it requires storing all training samples and many calculations for each classified object.

Figure 4.32 shows an example of k-NN classification with k=5. The training samples for class 1 are marked with pluses, and the objects of the class 2 with minuses. The considered object

is assigned to the second class, given that three of the five nearest neighbors belong to the second class.

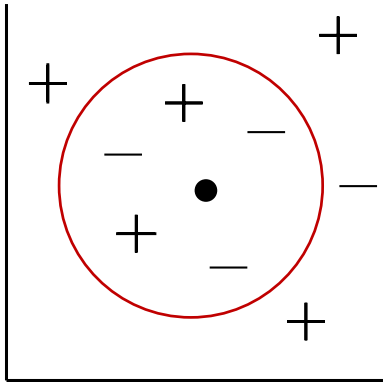
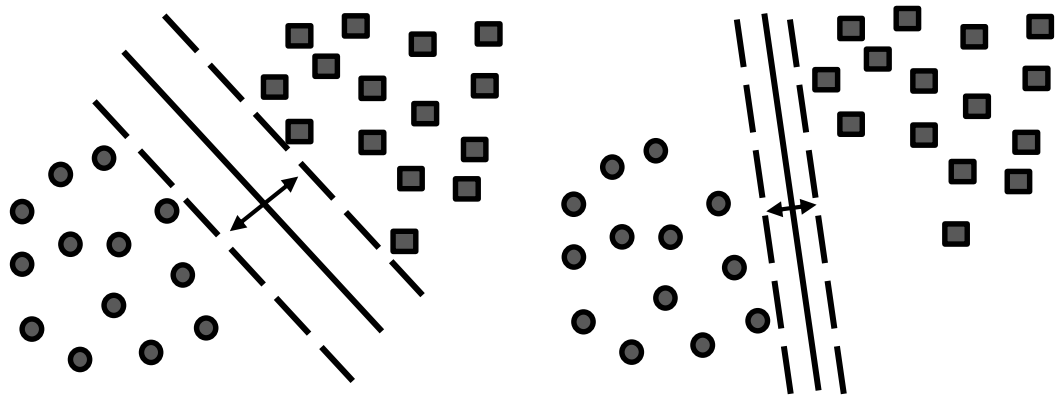


Figure 4.32: Example of k-NN classification. The considered sample is represented by filled black circle, the objects with first class by pluses and the objects with second class by minuses. The red circle around the sample shows the neighborhood with five nearest neighbors.

The best choice of the k value depends upon the considered data. If k=1, the classification accuracy is very sensitive to the noise in the training data set. With increasing k value, the classification reliability is improved; however, the boundaries between the classes become less distinct. In the practical applications, a suitable k value can be found experimentally, as values corresponding to the best classification accuracy (Hand & Vinciotti, 2002; Ghosh, 2006; Batista & Silva, 2009). In case of a binary classification, an odd integer number of neighbors is usually used to avoid ambiguity, i.e. when an equal number of the nearest neighbors belongs to different classes. The ambiguity can be resolved by a more general strategy suitable for a multiclass classification, by counting the neighbor votes with the certain weights. The most commonly used method of weight definition is to apply the distances between the found neighbors and the considered object.

The k-NN method possesses the methodological and conceptual simplicity, although it also reveals two significant disadvantages: a high memory consumption for storage of the entire training data set and the high time consumption for the search of the nearest neighbors in the training data set.

Originally proposed by Vapnik & Chervonenkis (1974) and Vapnik (1995), Support Vector Machines (SVM) construct a separating hyperplane that is equidistant to the convex hulls of classes. The hyperplane is defined as the most distant plane surface from the nearest instances of the training data set (Figure 4.33).



(a) (b)

Figure 4.33: Separating hyperplanes with large margin (a) and small margin (b). The hyperplane with larger margin provides the optimal separation capability.

The main principles of SVM can be best explained in terms of a two-class problem. In this case, each i-th object of training data set corresponds to the feature vector $\overrightarrow{x_i} \in R^N$ and the scalar value y_i equal to 1 or -1 indicating the object class. The SVM learner looks for a hyperplane separating the areas of the different classes. The hyperplane can be described by the following equation:

$$\vec{w} \cdot \vec{x} = b \tag{4.47}$$

where $\vec{w}$ is a weight vector and b is a scalar, often referred to as bias. The vector $\vec{w}$ and scalar b must be chosen in order that:

$$y_i = \begin{cases} +1, if\ \vec{w} \cdot \vec{x}_i \geq b \\ -1, if\ \vec{w} \cdot \vec{x}_i\ < b \end{cases} \tag{4.48}$$

Clearly, there are infinitely many appropriate values of $\vec{w}$ and b in a general case (Figure 4.33). In order to compare the different hyperplanes, a margin is used, which is defined as the distance between two parallel hyperplanes fulfilling the condition and plotted as close as possible to the samples of the first and second classes. The separating hyperplane used for classification is chosen between these two hyperplanes, whereby the hyperplane with the larger margin enables a better classification accuracy. The SVM method searches for the hyperplane with the largest margin for the surface yielding the best separation. Assuming that the hyperplane is defined by the unit weight vector $\widetilde{\vec{w}}$ and scalar value $\tilde{b}$ with the margin size ε, the expression can be rewritten as:

$$y_i = \begin{cases} +1, if\ \tilde{\vec{w}} \cdot \vec{x}_i \geq \tilde{b} + \dfrac{\varepsilon}{2} \\ -1, if\ \tilde{\vec{w}} \cdot \vec{x}_i \leq \tilde{b} - \dfrac{\varepsilon}{2} \end{cases} \quad (4.49)$$

Multiplying both parts of inequalities by $\frac{2}{\varepsilon}$ and substituting $\vec{w} = \frac{2}{\varepsilon}\tilde{\vec{w}}$ and $b = \frac{2}{\varepsilon}\tilde{b}$, one obtains:

$$y_i = \begin{cases} +1, if\ \vec{w} \cdot \vec{x}_i \geq b + 1 \\ -1, if\ \vec{w} \cdot \vec{x}_i \leq b - 1 \end{cases} \quad (4.50)$$

The hyperplanes $\vec{w} \cdot \vec{x} = b + 1$ and $\vec{w} \cdot \vec{x} = b - 1$ define a buffer area between classes. The hyperplanes pass through one or several points from the training set, called support vectors. The search of support vectors and hyperplane parameters can be formulated as a quadratic constrained optimization problem. The vector $\vec{w}$ is obtained as a superposition of the support vectors:

$$\vec{w} = \sum_{i=1}^{l} y_i a_i \vec{s}_i \quad (4.51)$$

Here, $\{\overrightarrow{s_1} \dots \overrightarrow{s_l}\} \in \{\overrightarrow{x_1} \dots \overrightarrow{x_n}\}$ are support vectors, y_i is the class value corresponding to support vectors, a_i and b are numerical parameters defined by solution of optimization problem.
The hyperplane is defined by:

$$\sum_{i=1}^{l} y_i a_i \vec{s}_i \cdot \vec{x} = b \quad (4.52)$$

As a result, the following expression for the classification of a sample with $\vec{z}$ feature vector is obtained:

$$y(\vec{z}) = sign\left(\sum_{i=1}^{l} y_i a_i \vec{s}_i \cdot \vec{z} - b\right) \quad (4.53)$$

Unfortunately, the majority of real-world problems involve linearly non-separable data for which no hyperplane can be created. One solution to the inseparability problem is to transform the initial data into a higher-dimensioned data space and construct the hyperplane there. Referring to Figure 4.34, a data set is not linearly separable in a two-dimensional space (left), but it can be separated in a non-linear “bent” space (right), through a data transformation into a three-dimensional space.

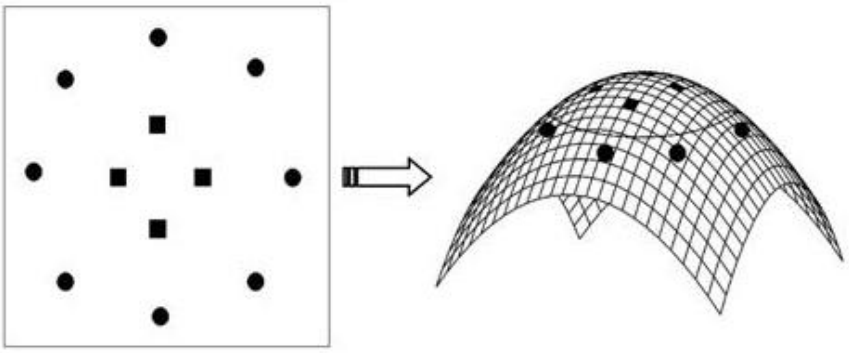


Figure 4.34: Linearly non-separable data transformed into a higher-dimension space (Lande et al, 2009).

Mapping the data to some other dimensional space $\vec{x} \rightarrow \phi(\vec{x})$ enables separating any consistent training set. Substituting the mapping function into the expression yields the scalar product of input variables $k(\vec{x},\vec{z}) = \phi(\vec{x}) \cdot \phi(\vec{z})$. By calculation, the product is not calculated explicitly; instead, a function $k(\vec{x},\vec{z})$, called the kernel function, is used. Genton (2001) described several classes of kernel functions, with the most commonly used ones summarized in Table 4.7.

Type of kernel	Kernel function
Linear kernel	$k(\vec{x},\vec{z}) = \vec{x} \cdot \vec{z} + c$
Polynomial kernel of degree d	$k(\vec{x},\vec{z}) = (\alpha\vec{x} \cdot \vec{z} + c)^d$
Gaussian kernel (radial basis function)	$k(\vec{x},\vec{z}) = exp\left(-\frac{\|\vec{x}-\vec{z}\|^2}{2\sigma^2}\right)$
Exponential (radial) kernel	$k(\vec{x},\vec{z}) = exp\left(-\frac{\|\vec{x}-\vec{z}\|}{2\sigma^2}\right)$
Sigmoid Kernel	$k(\vec{x},\vec{z}) = tanh(\alpha\vec{x} \cdot \vec{z} + c)$

Table 4.7: Commonly used kernel functions for SVM classifiers.

There is no general rule for the choice of the kernel function. Indeed, the choice of the kernel function best suited to the specified problem is very difficult.

The SVM is commonly used as a binary algorithm, although some work has been dedicated to its extension to multiclass classification (Duan & Keerthi, 2005). Learning SVM provides a unique solution for classification rules. This is an advantage compared to Neural Networks, which have multiple solutions for the same set of training data. The SVM are highly accurate, due to their ability to build complex nonlinear decision boundaries.

However, due to the application of kernel functions, it is difficult to understand the obtained classification rules. Moreover, the learning of SVM becomes very time consuming, especially for large data sets.

4.6.3 Assessment of the classification accuracy

As a rule in practice, a 100% accurate building classification is not possible, because of a number of different factors, including both an imperfection of the generated feature set and an insufficiency of the applied mathematical methods. Thus, a certain amount of misclassification is often inevitable.

The accuracy estimation requires a set of samples with known classes. In case of a supervised classification, the samples with known classes are also required to build classification rules. There are several strategies for using the samples for both learning and accuracy estimation.

The simplest method is to randomly split the set of the samples into training and testing sets. The training set is used for the building of classifiers, and the testing set for the accuracy estimation. This approach is called a holdout method. For improved accuracy, the holdout method can be repeated k times, with the accuracy calculated as an average value of k accuracy calculations. This approach is called random subsampling.

A more sophisticated method is cross validation, in which the set of samples is randomly divided into k parts with equal number of samples. Subsequently, k iterations are performed, whereby each part is successively used as a test set and the remaining samples as training set. The resulting accuracy is computed based on the total numbers of correctly and incorrectly performed classifications for k iterations. The final classification model is built by using the entire set of samples as the training set.

Classification performance refers to the classifier's ability to correctly predict a class for a new instance, namely the probability of correct classification. The quantitative performance estimation can be done by a predictive accuracy. The predictive accuracy of a classification is defined as the percentage of correctly classified samples. However, such a measure of accuracy is not always suitable. For example, two buildings of hundred are destroyed due to an earthquake and have to be detected by change detection analysis. By using the classifier that always predicts intact building states, very high accuracy of 98% can be obtained; however, the classifier is unable to find any destroyed building of interest. Naturally, a comprehensive assessment of classification results requires additional accuracy measurements.

The detailed results of classification can be represented in the form of a confusion matrix (Stehman, 1997; Congalton & Green, 2009). The confusion matrix or error matrix represents

actual and predicted class values for each class, whereby columns contain information concerning the predicted classes and rows represent the correct ones. A general form of the confusion matrix for the case of classification into two classes (e.g. intact and destroyed buildings) is given in Table 4.8. For the sake of generality, the classes are denoted as positive and negative.

		Predicted classes	
Actual classes		Positive	Negative
	Positive	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

Table 4.8: Confusion matrix of results of building states classification.

True Positive (TP) – number of correctly classified positive objects.

True Negative (TN) – number of correctly classified negative objects.

False Positive (FP) – number of incorrectly classified negative objects.

False Negative (FN) – number of incorrectly classified positive objects.

Based on the confusion matrix, the different accuracy measurements given in Table 4.9 can be derived.

In practice, the combined analysis of two accuracy measures TPR and FPR (see Table 4.9) is preferable to an estimation based solely on predictive accuracy. The following example shows the advantages of TPR and FPR measures. In the considered objects set, two buildings of ten are destroyed. A classifier identified one intact and one destroyed buildings as being destroyed, with the results of the classification shown in the following confusion matrix:

		Predicted classes	
Actual classes		Intact	Destroyed
	Intact	9	1
	Destroyed	1	1

The accuracy measures are:

$Predictive\ Accuracy\ = {}^{10}/_{12} = 0.833$

$TPR\ = {}^{1}/_{10}\ = 0.1$

$FPR = {}^{1}/_{2} = 0.5$

Measure	Calculation	Description
Overall accuracy	$\frac{TP + TN}{N}$	The portion of objects that are correctly classified.
Error rate	$\frac{FP + FN}{N}$	The portion of objects that are incorrectly classified.
Precision	$\frac{TP}{TP + FP}$	The portion of objects classified as positive that are correctly classified.
Sensitivity or True Positive Rate (TPR)	$\frac{TP}{TP + FN}$	The portion of positive objects that are correctly classified ($TPR = 1 - FNR$).
False Positive Rate (FPR)	$\frac{FP}{TN + FP}$	The portion of negative objects that are incorrectly classified ($FPR = 1 - TNR$).
Specificity or True Negative Rate (TNR)	$\frac{TN}{TN + FP}$	The portion of negative objects that are correctly classified ($TNR = 1 - FPR$).
False Negative Rate (FNR)	$\frac{FN}{TP + FN}$	The portion of positive objects that are incorrectly classified ($FNR = 1 - TPR$).

Table 4.9: The commonly used classification accuracy measures. Here, N is the total number of samples (Bramer, 2013).

Extending the analysis to the addition area with 20 intact buildings, it is possible that 10% of the intact buildings will be classified incorrectly, as for the first set. Thus, a new updated confusion matrix is:

		Predicted classes	
Actual classes		Intact	Destroyed
	Intact	27	3
	Destroyed	1	1

The new accuracy measures are as follows:

$Predictive\ Accuracy = {}^{28}/_{32} = 0.875$

$TPR = 0.1$

$FPR = 0.5$

Thus, values of TPR and FPR remained the same for the same classifier, whereas the predictive accuracy increased. Moreover, as was shown in the example at the beginning of this section, the predictive accuracy can be completely useless if the classes are strongly

unbalanced. The following section provides a description of a comparative analysis of classifiers based on the measures TPR and FPR.

4.6.4 Selection of classifiers

A more sophisticated classification strategy does not necessarily lead to better classification results, as one might assume. By increasing the complexity of classification rules, it is possible to reach an error-free classification for the samples of the training set; however, this generally leads to an insufficient performance for the samples from the testing set. This phenomenon is known as overfitting. On the other hand, extremely simple classification rules provide poor results for both the training and testing set, which is known as underfitting.

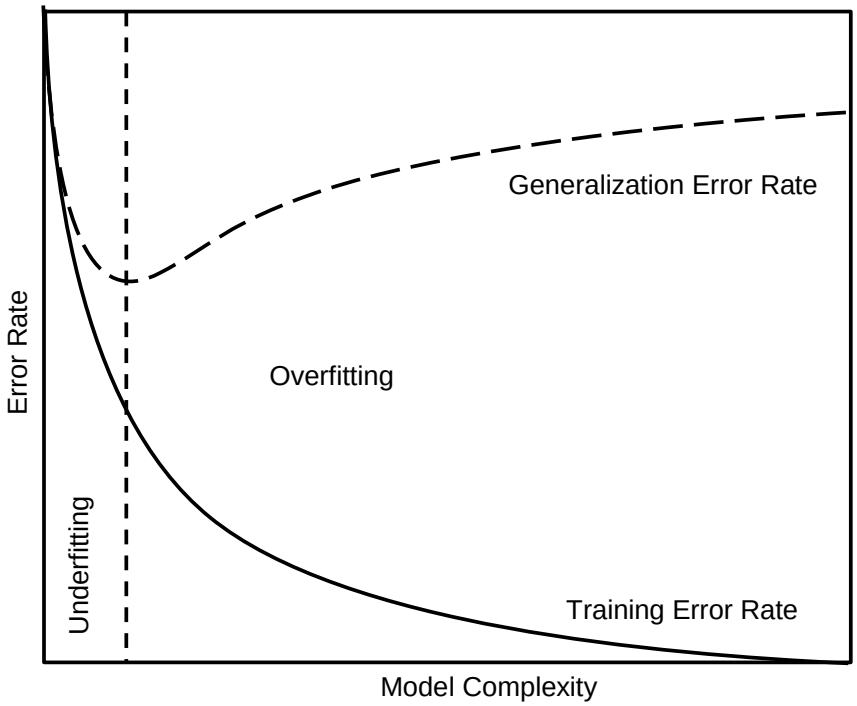


Figure 4.35: Dependency of error rate on the complexity of classification model (Janert, 2010).

Figure 4.35 shows the dependency of classification accuracy on the complexity of the classification model. In order to avoid the influence of overfitting on the calculation of classification accuracy, the testing set used for accuracy calculation should not include the samples from training set. On the plot, it can be seen that there is an optimum complexity, which corresponds to the minimum of the generalization error rate.

The figure below shows an example of underfitting and overfitting phenomena in a space of two features x_1 and x_2. The complexity of chosen classification model is reflected in the form of the boundary separating the classes (Figure 4.36).

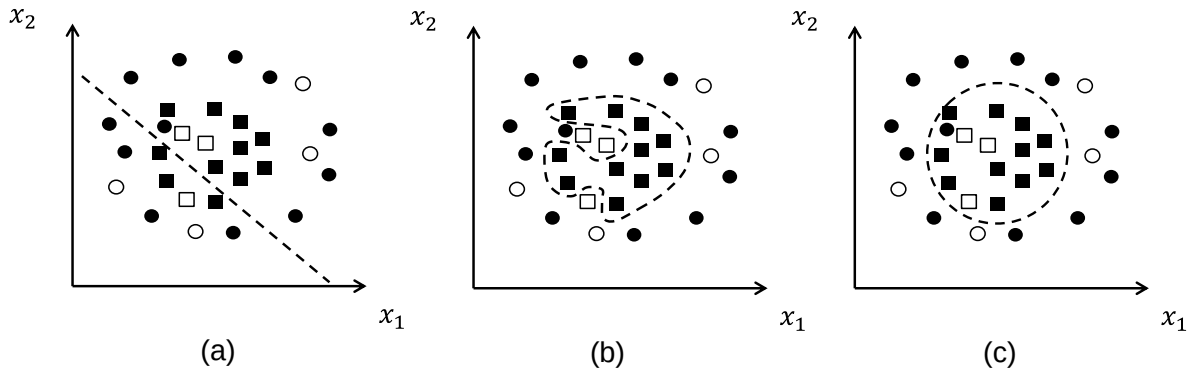


Figure 4.36: Comparison of classifier complexity based on the 2D scatter plot of two features x_1 and x_2. The filled and empty figures correspond to the training and testing sets, respectively. The circles and squares denote the positive and negative objects respectively. (a) Underfitting: a simple decision rule provides a low classification accuracy even for a training set; (b) Overfitting: an abundance of complexity leads to a poor classification accuracy for testing set; and (c) Optimal complexity choice.

A simple linear boundary form cannot provide sufficient accuracy, even for the training set. A complicated boundary yields a very accurate classification of samples from the training set, yet poorly suits the classification of samples from the testing set. A boundary in a form of circle cannot separate the error-free samples of training set, but yields sufficient results for the testing set. Thus, the classifier yielding the circle separation boundary is the most suitable of the considered examples.

A comparison of classifiers can be also performed based on the ROC graph, applying TPR and FPR accuracy measurements (see Figure 4.37).

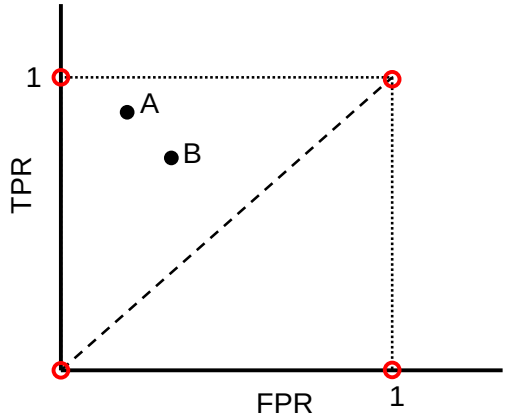


Figure 4.37: Comparison of classification performance based on ROC graph. Comparison of two classifiers A and B.

The points (1,1), (0,0), (1,0) and (0,1) correspond to the four respective special cases: a perfect classifier, the worst possible classifier, an ultra-liberal classifier and an ultra-conservative classifier. The perfect classifier provides the correct classification for each sample. Denoting the number of positive samples as P and the number of negative samples as N, the accuracy measures are calculated as follows:

		Predicted classes	
Actual classes		Positive	Negative
	Positive	P	0
	Negative	0	N

$TPR = 1, \quad FPR = 0$

The worst possible classifier provides an incorrect prediction for all samples.

		Predicted classes	
Actual classes		Positive	Negative
	Positive	0	P
	Negative	N	0

$TPR = 0, \quad FPR = 1$

The ultra-liberal classifier labels all samples as positive.

		Predicted classes	
Actual classes		Positive	Negative
	Positive	P	0
	Negative	N	0

$TPR = 1, \quad FPR = 1$

The ultra-conservative classifier labels all samples as negative.

		Predicted classes	
Actual classes		Positive	Negative
	Positive	0	P
	Negative	0	N

$TPR = 0, \quad FPR = 0$

The points lying on the line (0,0) - (1,1) correspond to the random classification. For example, a classifier randomly results in a positive class with a probability of 0.8, i.e. correctly classifies 80% positive and 20% negative samples. The corresponding confusion matrix comes to the following modification:

		Predicted classes	
Actual classes		Positive	Negative
	Positive	0.8*P	0.2*P
	Negative	0.8*N	0.2*N

$TPR = 0.8, \quad FPR = 0.8$

The closer the point of the classifier to (1,0), the better its performance. Naturally, the classifier corresponding to point A yields a better performance than the classifier corresponding to the point B (see Figure 4.37). The points of the classifiers, which are better than random guessing, have to be in the upper left-hand triangle. A classifier with prediction performance worse than random guessing can be converted into one with a prediction better than random guessing by reversing its predictions.

The classifiers can be compared based on the Euclidian distance to the point (1,0):

$$Distance = \sqrt{FPR^2 + (1 - TPR)^2} \tag{4.54}$$

To take into account the different importance of the FPR and TPR values, the weighted Euclidian distance with the weight $w \in [0,1]$ is used:

$$Weighted\ Distance = \sqrt{(1 - w)FPR^2 + w(1 - TPR)^2} \tag{4.55}$$

To choose the best suitable classifier, additional properties have to be taken into account; for example, the required computation resources and the possibility of training set extension.

5. Experiments and Result Discussion

In this chapter, an empirical evaluation of the proposed approach to detect building changes is performed using data from the following test areas: Kamaishi, Japan (2011 Tohoku earthquake), Yushu, China (2010 Yushu earthquake) and the city of Osnabrueck, Germany. The goal of this chapter is to demonstrate how this framework can be used for buildings that have undergone changes caused by catastrophic events, as well as discussing the particular advantages and weakness of the technique. Based upon the example city of Osnabrueck (Germany), it will be additionally shown that this method can be applied not only for the detection of destroyed buildings but also for reconstructed buildings.

5.1 First study case: Kamaishi, Japan (the 2011 Tohoku earthquake)

On 11 March 2011, a massive earthquake with a magnitude over 9,0 M_w occurred in the north-western Pacific Ocean, with the epicenter located approximately 130 km from Sendai and 373 km from Tokyo. The hypocenter of the most devastating quake was registered at an underwater depth of approximately 30 km. The earthquake triggered enormous tsunami waves, causing extensive destruction along the coast of the northern islands of Japan. The tsunami spread through the Pacific coastline, including North and South America from Alaska to Chile, which were under a flood warning with evacuation activities. However, the real effect was insignificant in these areas.

The height of the tsunami waves in Japan varied, with the maximum observed at the coast of Miyagi Prefecture up to 10 meters. The huge waves flooded Sendai Airport, sweeping away anything and everything in their path, including planes, cars, constructions, etc.

The earthquake and resulting tsunami caused an accident at the Fukushima nuclear power plant, leading to a radioactive leak. Several reactors were corrupted at different levels, causing subsequent large explosions and problems with storing the spent nuclear fuel. People in the local neighborhood were evacuated due to the risk of radioactive contamination. Some workers of the nuclear power plants were injured and irradiated to different degrees of severity. A high radiation level was registered in Tokyo's water supply, vegetables and milk in the area nearby Fukushima.

The nuclear crisis and deficit of electricity had a large negative impact on the economic situation in the country, prompting enormous economic and human losses. The National Police Agency of Japan reported about 28 000 people being dead or missing across the affected prefectures (Mimura et al., 2011; Takeuchi & Chavoshian, 2011). The earthquake and tsunami caused extensive infrastructural damage of roads, bridges, railways and dikes, as well as over 300 000 buildings being entirely, half or partially collapsed and burned. The total economic losses estimated by the World Bank were US$235 billion, excluding five years

prognosticated costs for the required postdisaster reconstruction (The World Bank, 2011).

To perform the experiments, a scene of the small city of Kamaishi, located on the Sanriky rias coast of Iwate (Japan), was selected. The city was considerably damaged with about 1,250 people dead or missing due to the catastrophic incident, notwithstanding the Kamaishi Tsunami Protection Breakwater built in 2009, which was easily surmounted by 4.3 m high waves.

For the first study case, an area of 686×1722 pixels was selected from the post-disaster image dated 1 April 2011. Figure 5.1 shows a part of the study area overlaid with vector data presenting the building footprints corresponding with the pre-disaster layout. The vector information was obtained from OpenStreetMap (http://www.geofabrik.de) and contains 696 vector objects for the selected study image. In order to verify the final result of the change detection procedure, the information containing the post-event states of buildings was prepared in advance as a reference for each vector object through a visual comparison of the source data.

Figure 5.1: The first study case: post-event image of Kamaishi, Japan overlaid with vector data containing building footprints. The satellite image is courtesy of Digital Globe (©Digital Globe 2013); vector data are provided by OpenStreetMap (© OpenStreetMap contributors).

5.1.1 Calculation of DPC

The calculation of the DPC feature was performed according to the algorithm described in detail in section 4.5.1. Figure 5.2 depicts the separate steps of the calculation procedure for

intact and destroyed buildings. The intact building was selected in the center-to-top part of the test image and the destroyed building is located in the center part (see Figure 5.1). First, image enhancement is performed by means of homomorphic filtering with a Gaussian highpass filter (Figure 5.2 (b,f)).

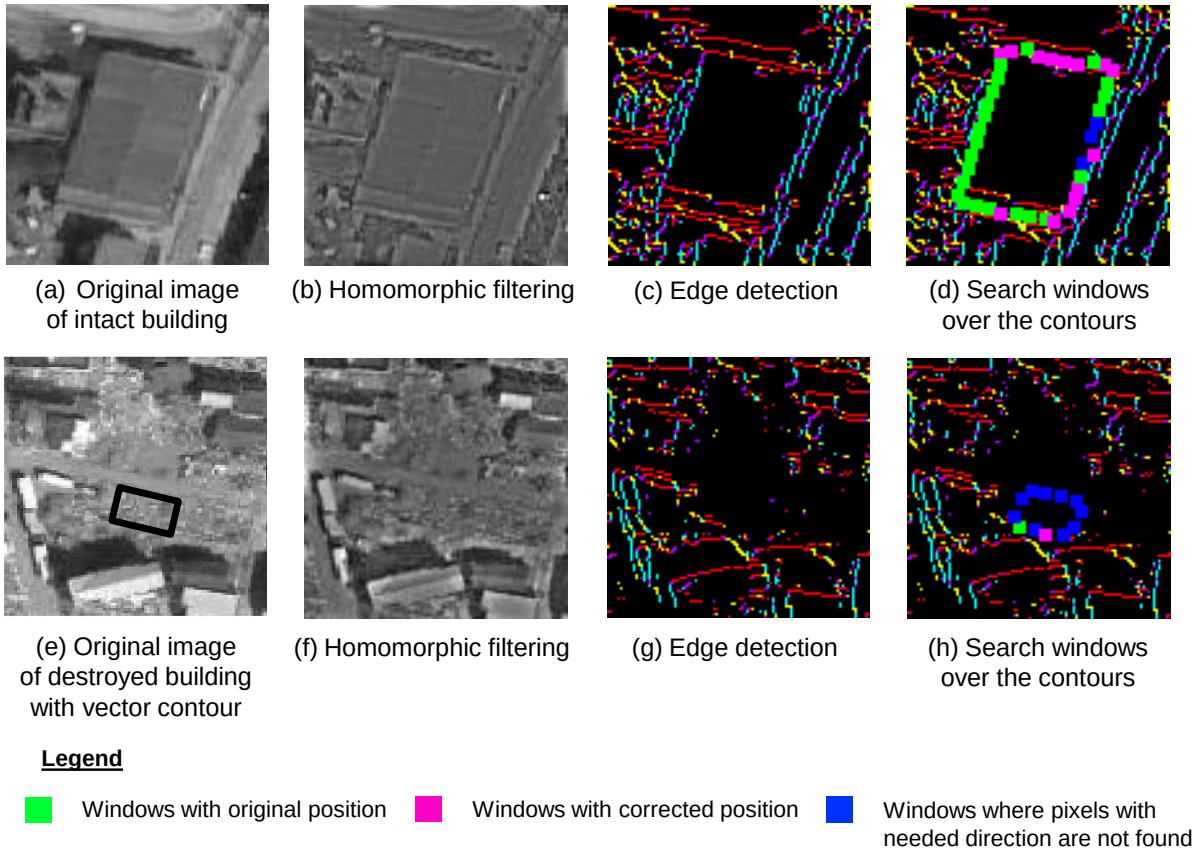


Figure 5.2: Calculation of DPC. (a) Original image of the intact building; (b) image of the intact building enhanced using homomorphic filtering; (c) image of the intact building after edge detection with colors of contours corresponding with their orientations; (d) image of the intact building after edge detection with the search windows, where pixels with a needed direction of contour are counted; (e) original image of the destroyed building; (f) image of the destroyed building after homomorphic filtering; (g) image of the destroyed building after edge detection; (h) image of the destroyed building after edge detection with the search windows, where pixels with a needed direction of contour are counted.

The filtered images are further used to calculate textural features, as well as edge detection by using the Canny algorithm. Figure 5.2 (c,g) shows the resulting contours detected, colored according to their orientations (the colors of pixels are explained in Figure 4.9). Around each control point extracted from the vector contour (see section 4.5.1.2), the pixels with an appropriate direction of the contour are sought in the area with a size of 5x5 pixels. Figure 5.2 (d,h) shows the search windows: in the windows depicted in green color, at least 5 pixels of contour were found without correction of window position; the position of the pink windows were corrected to improve the accuracy of calculation; at the blue windows, no contour points

were detected at all. Due to the window width and position correction procedure, the cases in which the raster and vector contours do not exactly coincide can be treated correctly. Of course, the algorithm fails if the difference in contour positions is larger than half of the search window size.

The histogram in Figure 5.3 demonstrates how many pixels of contour were detected in the search windows. The green graphic corresponds to the intact building and the red to the destroyed one. In the case of the intact building, it can be seen that the maximal number of contour pixels was found in most search windows (the number of counted pixels per window is restricted by 5). The maximal numbers of contour pixels were found in 34 out of 41 search windows, four pixels were found in four windows and the pixels with the appropriate contour directions were not detected in only three windows. For the destroyed building, the opposite situation can be seen, where most of windows do not contain any contour pixels, whereas the maximal number of pixels was only found in one window.

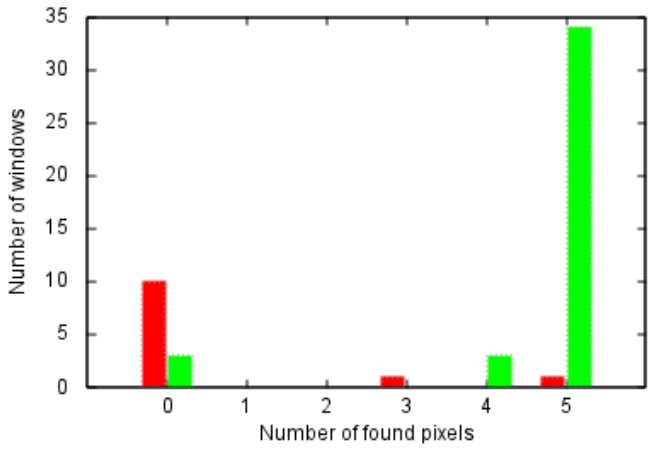


Figure 5.3: Number of contour pixels found in the search windows. Green boxes correspond to intact building and red to destroyed one respectively.

The procedure for evaluating the contour integrity produces a DPC value of 91% for the intact building, indicating that 91% the building contour was detected, and a DPC value of 13% for the destroyed building.

5.1.2 Calculation of textural features

Besides the DPC feature, texture measures describing the homogeneity of the image part corresponded to the building were calculated. A general approach to object-oriented calculation of textural features is presented in detail in section 4.5.2.2 and schematically demonstrated in Figure 4.25. The textural features were calculated based on the enhanced image produced by the homomorphic filtering. The considered feature set included Angular

Second Moment (ASM), Inertia, Inverse Difference Moment (IDM) as well as mean, maximal and minimal values of their angular dependent measures.

As previously mentioned in Chapter 4, employing a large number of different features makes the classification more complex and may negatively affect the final result. Generating an optimal feature set is a key point of the data analysis. The information content of the features can be evaluated by the ROC analysis, using AUC values that evaluate the separability of objects based on their feature values (see section 4.5.3). AUC takes values from 0.5 to 1.0, whereby the larger values correspond to a better performance of the feature.

ASM_0	**ASM_45**	**ASM_90**	**ASM_135**	**ASM_min**	**ASM_max**	**ASM_mean**
0,50	0,51	0,51	0,51	0,51	0,51	0,51
Inertia_0	**Inertia_45**	**Inertia_90**	**Inertia_135**	**Inertia_min**	**Inertia_max**	**Inertia_mean**
0,52	0,53	0,54	0,53	0,53	0,54	0,53
IDM_0	**IDM_45**	**IDM_90**	**IDM_135**	**IDM_min**	**IDM_max**	**IDM_mean**
0,90	0,70	0,69	0,68	0,66	0,90	0,78

Table 5.1: AUC values for GLCM features (study case Kamaishi, Japan (the 2011 Tohoku earthquake)).

Table 5.1 shows the values achieved for each textural feature. Of course, only mean, maximal and minimal values are rotationally invariant and can be used for classification, while the angular dependent measures (IDM_0, ASM_90 etc.) are only presented here for comparison sake. The IDM features show much better results than ASM and Inertia features. ROC analysis of DPC feature yields an AUC value of 0.89. Thus, the IDM_max with an AUC value of 0.90 and DPC with an AUC value of 0.89 are the two best features.

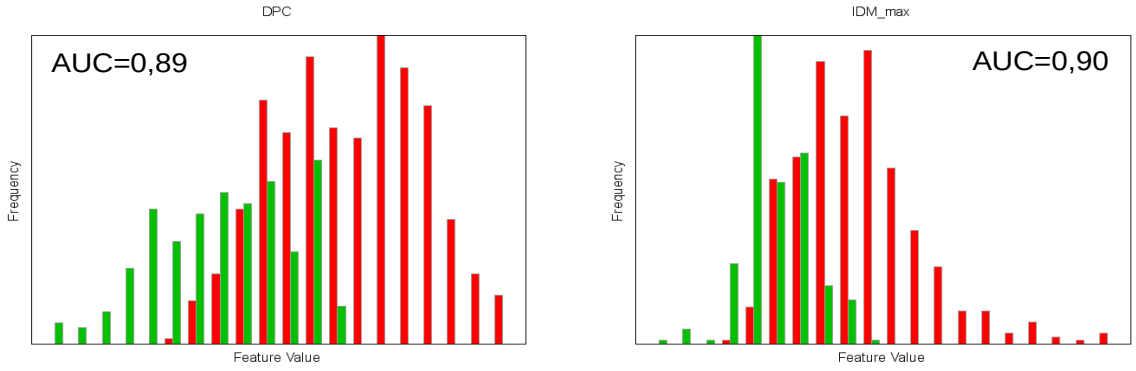


Figure 5.4: Frequency distribution histograms of selected features (DPC and IDM_max). Red rectangles correspond to destroyed buildings, green to intact buildings (study case Kamaishi, Japan (the 2011 Tohoku earthquake)).

A more vivid way of demonstrating a capability of features to separate test objects into the known number of classes is through the construction of frequency distribution histograms. The visual interpretation of the histograms (see Appendix 5.1) indicates that DPC and IDM_max provide the best separation of the objects. Figure 5.4 demonstrates the frequency distribution histograms for the two best selected features.

In the scatter plot below (Figure 5.5), it can be seen that vector information expressed by DPC feature and raster information expressed by IDM_max feature supplement each other, allowing the separation of objects that have different states but similar DPC or IDM_max values.

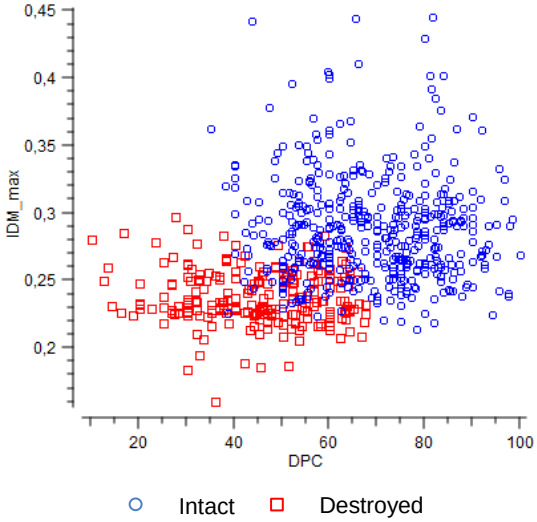


Figure 5.5: Scatter plot demonstrating the two-class separation of objects (study case Kamaishi, Japan (the 2011 Tohoku earthquake)).

In the following sections, classification of the buildings under investigation will be performed using the DPC and IDM_max features.

5.1.3 Classification and results

This section deals with analysis of the building states classification by means of an unsupervised classification (k-means) and two supervised classification techniques (k-NN and SVM) using DPC and IDM_max features. The tests include the study of different parameter values used for the configuration of classification rules.

5.1.3.1 Unsupervised classification

Unsupervised classification does not require a training data set, splitting objects into groups/clusters, whose state is defined based on the feature values corresponding to cluster centers. An unsupervised classification warrants proper attention due to its capability to

cluster without a learning phase, and thus the training set is not required for the classification procedure.

The **k-means clustering algorithm** is a rather easy yet efficient method of unsupervised classification (see section 4.6.1). For this test, the number of clusters is essentially set to 2: intact and destroyed buildings. For initialization, the deterministic method similar to KKZ algorithm (Katsavounidis et al., 1994) is applied. First, the centre of mass is calculated for all objects, whereby the object most distant from the centre of mass is taken as the first cluster centre. The centre of second cluster is chosen at the position of the object most distant from the first cluster centre.

The choice of the distance function was performed based upon the analysis of overall accuracies computed for the following distance metrics: Euclidean, Manhattan and Chebychev. Referring to Table 5.2, it can be seen that the accuracy values do not significantly vary depending on the distance metrics; however, the classification using Euclidean metric provides a more accurate result. Accordingly, for this reason the Euclidean distance measure was chosen for the further data classification.

	Euclidean	Manhattan	Chebychev
Overall Accuracy	0,7730	0,7691	0,7707

Table 5.2: Testing classification efficiency by means of k-means clustering depending on different distance measures for the study case Kamaishi, Japan (the 2011 Tohoku earthquake).

In Figure 5.6, the analysis of classification results based on the confusion matrix is shown, highlighting that good results with the overall accuracy of 77% can be reached, even for unsupervised classification. The scatter plot shows linear separation typical for k-means classification with the Euclidian metric. The quality of the individual class detection is assessed by means of user and producer accuracies. Producer's accuracy indicates how well the objects of the given class are classified and is a rate of correctly classified objects regarding all objects of the class. For the destroyed buildings, the producer's accuracy shows that 93% of destroyed objects are detected as being destroyed and only 7% (12 objects) are mistakenly classified as intact. User's accuracy is an indication of the probability that an object classified into a certain class truly represents this class, thus reflecting the rate of correctly classified objects regarding all objects classified as this class. The value of 55% for destroyed buildings means that only 55% of buildings classfied as destroyed are really destroyed. Thus, for this study case, the unsupervised classification identified almost all destroyed buildings correctly, although 45% of objects marked as destroyed were actually intact.

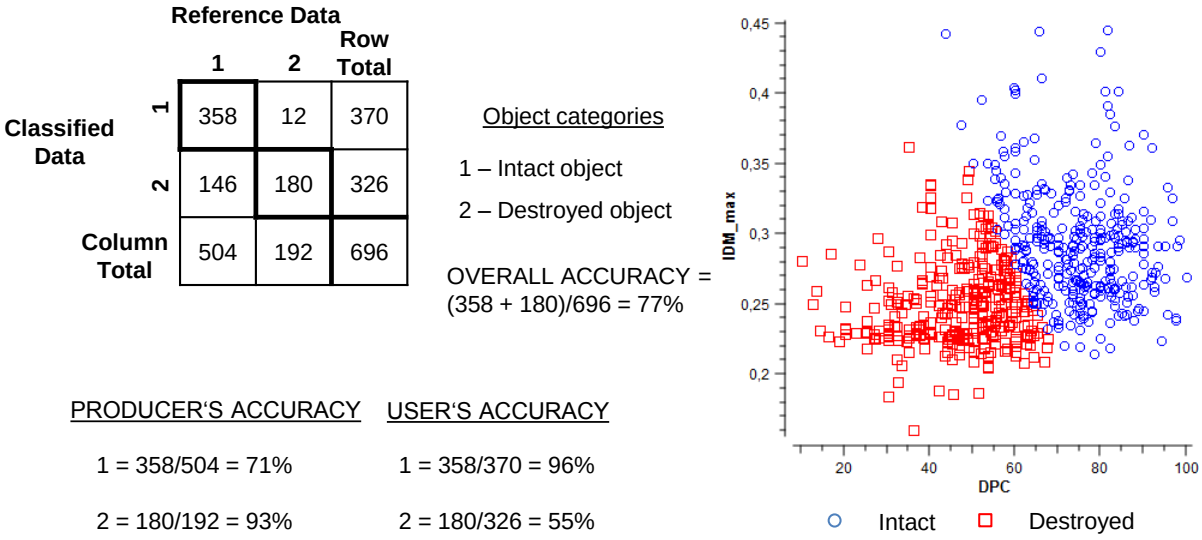


Figure 5.6*:* Classification result calculated for the study case Kamaishi, Japan (the 2011 Tohoku earthquake) using the k-means clustering algorithm: confusion matrix (left) for accuracy assessment and scatter plot (right) illustrating the separation of objects.

5.1.3.2 Supervised classification

Supervised classification was described in detail in section 4.6.2. Its application includes two steps: learning and classification. During the learning phase, the inner structure and parameters of the classification rules are defined based on a training set of objects with known states. The obtained classification rules are used during the classification phase to define the state of new objects. For the assessment of classification efficiency for both the learning and classification phase, the object set with known condition was used. The classification accuracy was estimated by comparison of known state values with values from classification. The study of classification efficiency involved the following steps:

1. Splitting the whole data set into training and testing data sets. For the training set, a near equal number of destroyed and intact objects were randomly selected. The remainder data was used in the testing set.
2. Learning phase and the choice of optimal classification parameters based on training set data. The selection of optimal classification parameters was performed using a grid-search method that includes calculations on the grid of different parameter combinations and the choice of settings yielding the best accuracy (Hsu et al, 2003). The classification accuracy is therewith evaluated by means of n-folds cross validation (see Chapter 4).
3. Classification and accuracy assessment based on testing data set.

The first experiment was performed with **k-Nearest Neighbours** classifier, whose theoretical description is provided in section 4.6.2. In this study, the k-NN classifier is used with a distance-based weigting of neighbour's votes. The classification rule has one parameter k defining the number of neighbours voting for the the class of the considered object. The k-value was determined with values of k from 5 to 30 and different sizes of a training set equal 5%, 10%, 15% and 30% objects (Figure 5.7). The classification accuracy was therewith estimated by 5-folds cross validation on the training set. Figure 5.7 below shows the classification accuracy presented by three measures: overall accuracy, sensitivity and specificity (see Table 4.9). It can be seen that the accuracy does not significantly vary with the variation of the k-value and the number of objects in the training set.

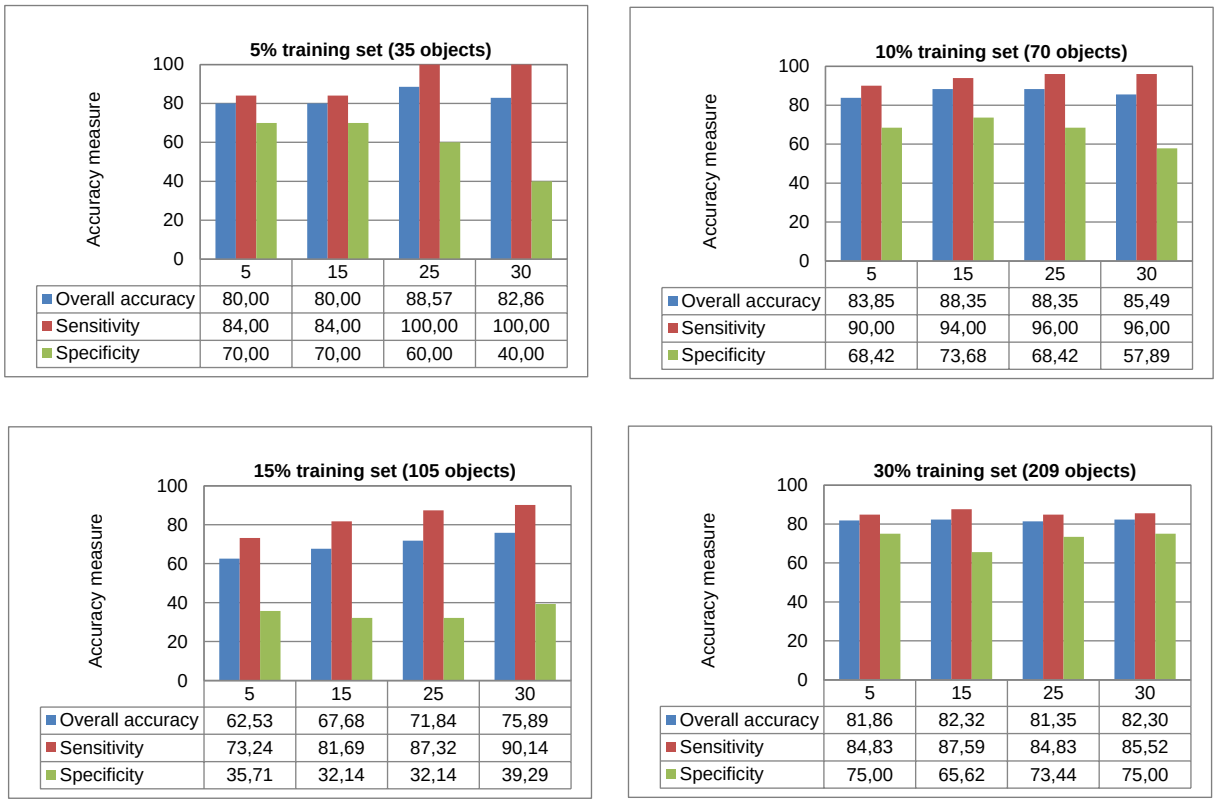


Figure 5.7*:* Accuracy of k-NN classification depending on different number of k-neighbours and training samples calculated for the study case Kamaishi, Japan (the 2011 Tohoku earthquake*).*

As it follows from the results obtained, the best accuracy measures were reached using the 10%-training set containing 70 objects from the original data set and the k-value of 25. Therefore, this combination was selected for a more detailed analysis. Figure 5.7 illustrates the classification results obtained for the first study case, using a k-NN classifier with the chosen parameters. The relationship between the actual and predicted classes is summarized in the confusion matrix (Figure 5.8 (left)). The accuracy of classification was

estimated based on the overall accuracy, the user's and producer's accuracies for intact and damaged buildings. The overall accuracy reached a value of 84%. The classification provided a wrong state for 22 out of 163 destroyed buildings, corresponding to a producer's accuracy of 86%. The set of objects detected as destroyed contained 215 objects, 74 of which were classified erroneously, yielding a user's accuracy of 66%. The possible reasons for classification mistakes are discussed in section 5.4.

The scatter plot (Figure 5.8 (right)) demonstrates that a separation of the test objects is possible. It can be seen that the k-NN method allows generating of a quite complex rule to separate areas of intact and destroyed objects in the IDM_max/DPC feature space.

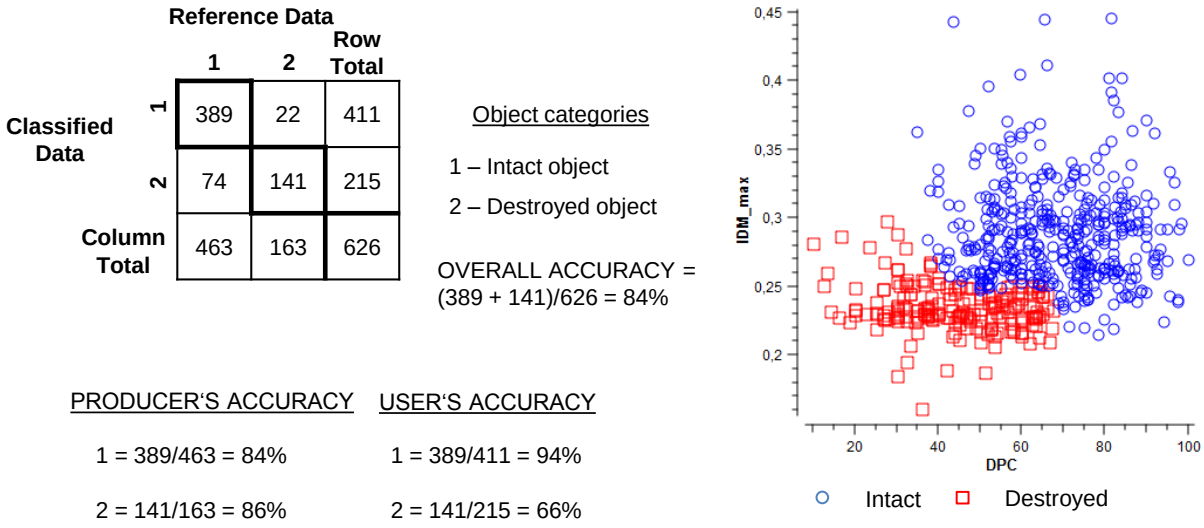


Figure 5.8: Classification result calculated for the study case Kamaishi, Japan (the 2011 Tohoku earthquake) using the k-NN algorithm: confusion matrix (left) for accuracy assessment and scatter plot (right) illustrating the separation of objects.

The next experiment was performed with the **Support Vector Machine (SVM)** classification, described in detail in section 4.6.2. SVM has often been shown to have great potential, especially for the classification of remotely sensed images involving high-dimensional data with small training sets (Lizarazo, 2008; Melgani & Bruzzone, 2004). However, this powerful classification approach has a significant drawback, namely that the learning phase becomes computationally very expensive with increasing training set size. In contrast to pixel-based approaches, the object-oriented classification requires a small set of training samples, which is why the application of SVM is very efficient (Tzotsos & Argialas, 2008).

The SVM classification can be used with different kernels that have several configuration parameters. In the study, three kernel types were tested: linear, polynomial and RBF. This experiment was performed with four training sets constituting 5%, 10%, 15% and 30% of objects from the total number of considered objects. For each training set, the parameters of

the SVM kernel function were defined by the grid-search algorithm. The classification accuracy for the grid-search algorithm were estimated by 5-folds cross validation. The results are provided in Figure 5.9, illustrating a dependence of the classification performance on different sizes of training sets and kernel functions. As can be observed, high overall accuracies were obtained for all kernel functions. However, the portion of destroyed objects that a correctly classified (Specificity) is rather low for training sets containing 5% and 15% of the test data. The training set containing 10% of the objects (=70) and RBF kernel function provided more accurate result. Therefore, this parameter configuration was chosen for further detailed analysis.

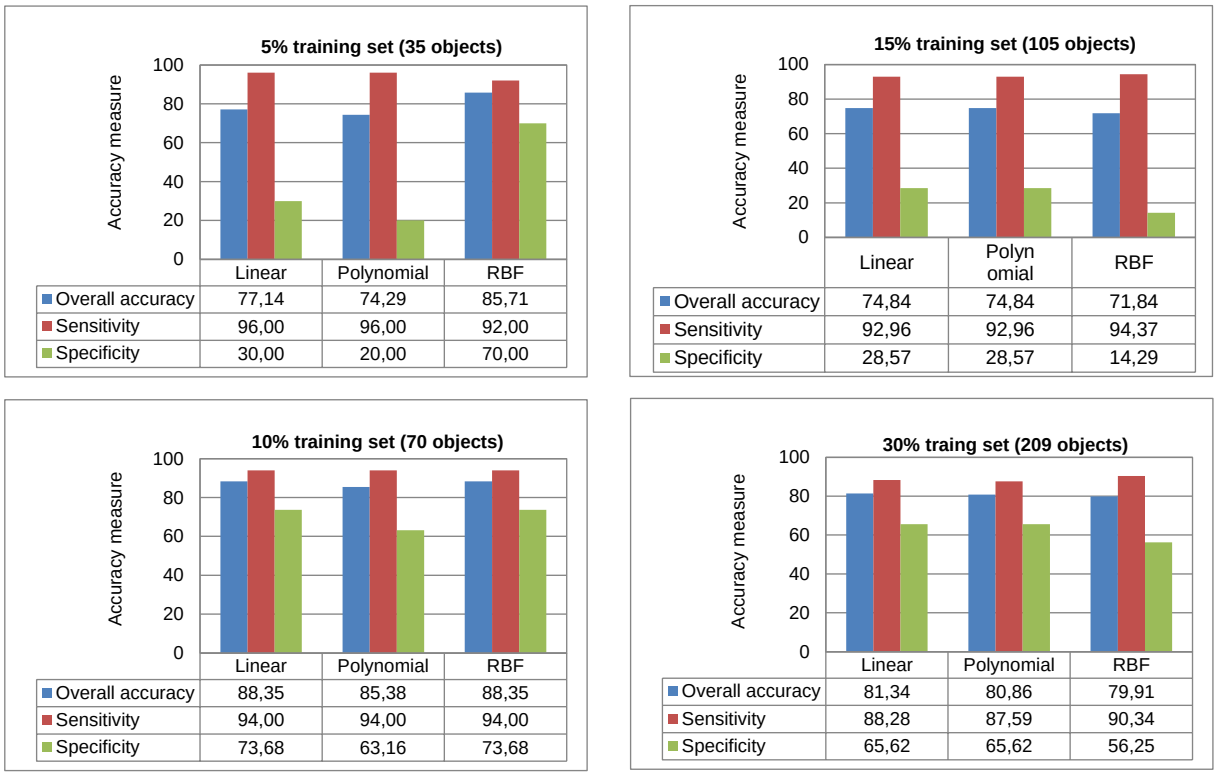


Figure 5.9: Selection of optimal classification parameters for SVM depending on different kernel function and different number of training samples calculated for the study case Kamaishi, Japan (the 2011 Tohoku earthquake).

The results of classification in the form of the confusion matrix and computed accuracy measures are presented in Figure 5.10. The scatter plot (Figure 5.10 (right)) illustrates the obtained separation of the objects. The classification was performed for the testing set containing 626 objects, 163 of which were destroyed. 9 destroyed objects were wrongly classified, corresponding to a producer's accuracy of 94%. The user's accuracy of 68% signifies that 32% of buildings classified as destroyed (73 objects) are actually intact. The reasons for the classification errors are discussed further in section 5.4.

Classified Data \ Reference Data	1	2	Row Total
1	390	9	399
2	73	154	227
Column Total	463	163	626

Object categories

1 – Intact object

2 – Destroyed object

OVERALL ACCURACY = (390+ 154)/626 = 87%

PRODUCER'S ACCURACY	USER'S ACCURACY
1 = 390/463 = 84%	1 = 390/399 = 97%
2 = 154/163 = 94%	2 = 154/227 = 68%

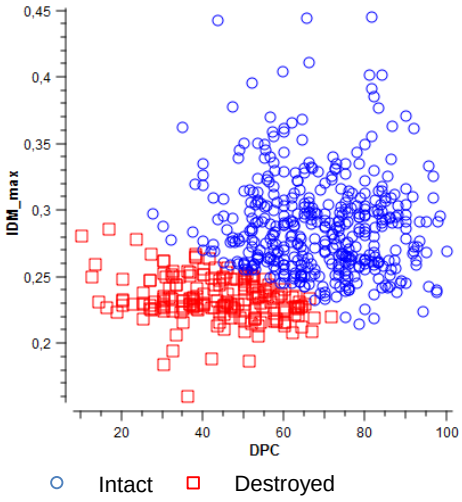
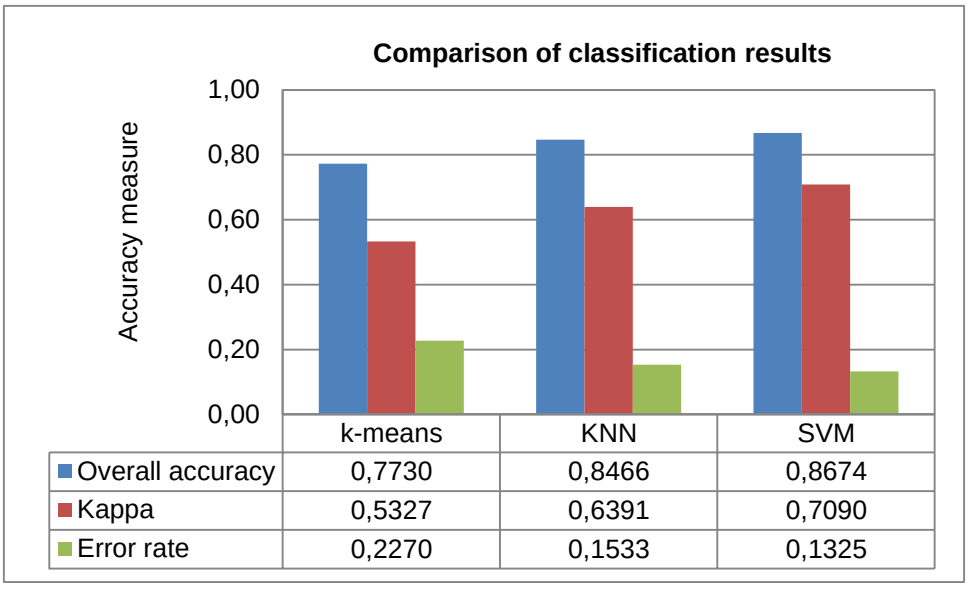


Figure 5.10: Classification results calculated for the study case Kamaishi, Japan (the 2011 Tohoku earthquake) using the SVM algorithm: confusion matrix (left) for accuracy assessment and scatter plot (right) illustrating the separation of objects.

Figure 5.11 graphically summarizes the accuracy measures for the classification techniques described above, which are compared based upon three accuracy measures: overall accuracy, kappa and error rate. Assessment based on the kappa measurement developed by Cohen (Cohen, 1960) is often preferred for the comparison of different classification techniques. Kappa can be characterized as a measure of the agreement between prediction and reality (Congalton R., 1991).



	k-means	KNN	SVM
Overall accuracy	0,7730	0,8466	0,8674
Kappa	0,5327	0,6391	0,7090
Error rate	0,2270	0,1533	0,1325

Figure 5.11: Comparison of classification results obtained using different classifiers for the study case Kamaishi, Japan (the 2011 Tohoku earthquake).

The best performance was reached using the SVM classifier, which provides the lowest error rate and the highest values of overall accuracy and kappa. The resulting damage map shown

in Figure 5.12 was generated using data obtained from the SVM classification, yielding the most accurate results.

Figure 5.12: Resulting damage map generated from the results obtained by SVM classification for the study case Kamaishi, Japan (the 2011 Tohoku earthquake): white objects represent intact buildings, red – destroyed, and green objects correspond to the incorrectly classified objects.

5.1.3.3 Comparison with texture-based change detection

In this section, the efficiency of the DPC feature in contrast to single-parameter texture-based change detection is experimentally demonstrated. In this context, the classification assessment with different combinations of textural features is performed. Four groups of features were considered: the first group includes all angularly dependent features; the second group contains mean values of the angularly dependent features; the third group comprises only minimum values of the angularly dependent features; and the fourth group includes maximum values of the angularly dependent features. For classification, the

parameters determined for the above experiments were used. For the k-means clustering, the Euclidean distance metric was used. The supervised classification was performed with the training set containing 70 objects from the total data set and RBF kernel function for the SVM classifier; and a k-value equal to 25 was defined for the k-NN classifier. Figure 5.13 presents the experimental results with classification accuracies calculated for k-means, k-NN and SVM techniques. This experiment results in low specificity values, meaning that a significant part of the destroyed objects was incorrectly classified. The worst results were obtained for minimum values of the angularly dependent features, while the best overall accuracies were reached for the fourth group, maximum feature values.

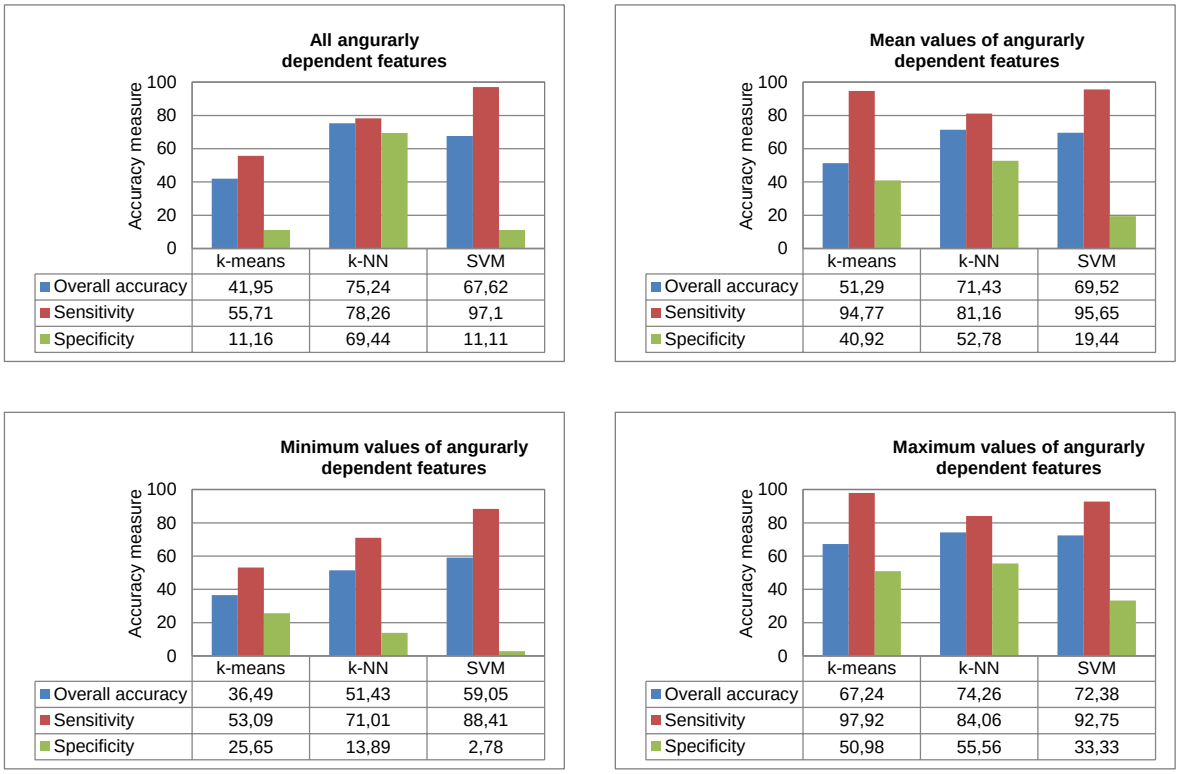


Figure 5.13: Classification performance depending on feature sets containing different combinations of textural features calculated for the study case Kamaishi, Japan (the 2011 Tohoku earthquake).

The classification performance obtained using the feature set containing mean values of the angularly dependent features is lower in contrast to the classification of the maximum values. The comparison of results obtained for different feature sets leads to the following conclusions:

1. The application of redundantly large feature sets (the first group) does not lead to better classification results.
2. The application of maximal values of the angularly dependent features is more efficient than the commonly used mean values, for reasons discussed in section 4.5.3.

This experiment demonstrates that the exclusion of DPC from the feature set leads to less accurate classification results. Accuracies increase by 15-25% with the DPC as second parameter.

5.2 Second study case: Yushu, Qinghai, China (the 2010 Yushu earthquake)

China is the most populous country in the world and has many areas of active seismicity. A list of historical earthquakes in China includes numerous devastating catastrophes, such as the 1556 Shaanxi earthquake (more than 800 000 people were killed), the 1920 Haiyuan earthquake (200 000 people were killed), the 1975 Haicheng earthquake (2000 people were killed), the 1976 Tangshan earthquake (over 240 000 people were killed), the 2008 Sichuan earthquake (over 80 000 people were killed), the 2010 Yushu earthquake (2 200 people were killed) and a series of earthquakes in 2011 (source: see http://earthquake.usgs.gov/earthquakes/world/world_ deaths.php).

The second study case presents a part of the Yushu City, the autonomous prefecture of south-western Qinghai province located in the northwest of the People's Republic of China. On 14 April 2010, a strong 7.1-magnitude earthquake occurred in Qinghai Province, with the epicentre registered about 31.5 kilometres northwest of the town of Gyegu. In the heart of both Yushu Tibetan Autonomous Prefecture and Yushu City, the worst-affected area in the region, most buildings were entirely demolished, leaving thousands of people homeless. This natural disaster left 2,700 people dead, 270 missing and 12,135 injured (UNICEF, 2011).

The test study site was selected from a QuickBird satellite image dated 15 April 2010. From the available data set, an 843 x 467 pixel subset was selected as the study area. In the absence of initial cadastral vector data, a vector map containing building footprints was manually created from a pre-event image corresponding to the scenario dated November 6, 2004. The test area containing 610 vector objects overlaying the raster map is presented in Figure 5.14. The reference information containing the post-event state of buildings was obtained for each vector object by a visual comparison of pre- and post-event data.

Figure 5.14: Second study case: post-event image of city Yushu, China overlaid with vector map corresponded to the pre-event layout. Satellite image courtesy of Digital Globe (©Digital Globe 2010).

5.2.1 Unsupervised classification

For this study case, analogously to the previous study case, the **k-means clustering** with Euclidean metric was used for unsupervised classification. The resulting confusion matrix is presented in Figure 5.15. The classification allows for recognition of the destroyed objects with quite high values of producer's and user's accuracies of 85% and 87%, respectively. These values reflect that 58 of 395 destroyed buildings were wrongly classified and 49 intact buildings were erroneously included in the set of 386 destroyed objects. The obtained overall accuracy of 82% also confirms the high potential of the unsupervised classification approach.

Classified Data	Reference Data 1	Reference Data 2	Row Total
1	166	58	224
2	49	337	386
Column Total	215	395	610

Object categories

1 – Intact objects

2 – Destroyed objects

OVERALL ACCURACY = (166+ 337)/610 = 82%

PRODUCER'S ACCURACY	USER'S ACCURACY
1 = 166/215 = 77%	1 = 166/224 = 74%
2 = 337/395 = 85%	2 = 337/386 = 87%

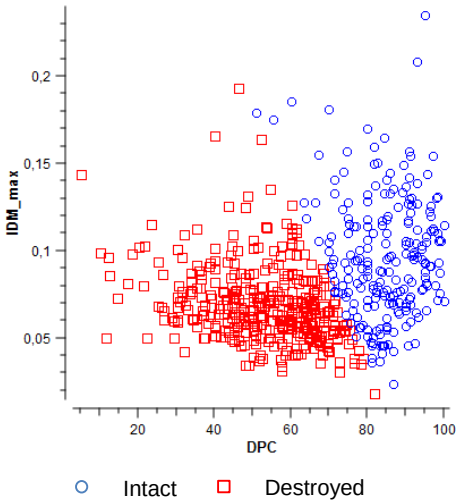


Figure 5.15: Classification results produced for the study case Yushu, Qinghai, China (the 2010 Yushu earthquake) using the k-means clustering algorithm: confusion matrix (left) for accuracy assessment and scatter plot (right) illustrating the separation of objects.

5.2.2 Supervised classification

As in the first study case, the supervised classification was performed using k-NN and SVM classification techniques. Based on the experiments performed for the first study case, 92 objects, constituting 15% of the total number objects, were extracted as training set for classifier learning.

Classified Data	Reference Data 1	Reference Data 2	Row Total
1	134	28	162
2	59	328	387
Column Total	193	356	549

Object categories

1 – Intact object

2 – Destroyed object

OVERALL ACCURACY = (134 + 328)/549 = 84%

PRODUCER'S ACCURACY	USER'S ACCURACY
1 = 134/193 = 69%	1 = 134/162 = 83%
2 = 328/356 = 92%	2 = 328/387 = 85%

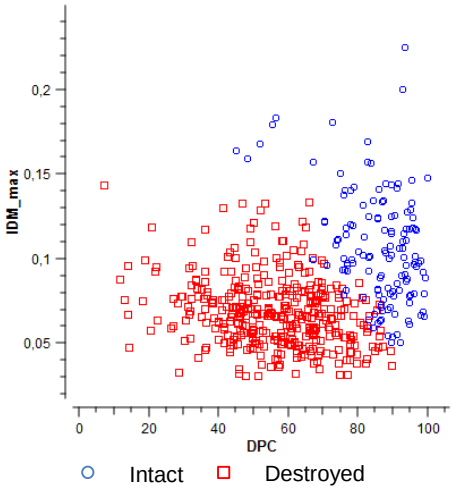


Figure 5.16: Classification results produced for the study case Yushu, Qinghai, China (the 2010 Yushu earthquake) using the k-NN algorithm: confusion matrix (left) for accuracy assessment and scatter plot (right) illustrating the separation of objects.

For the **k-NN classification**, the value of k was also defined as equal to 25. The result of this classification is provided in Figure 5.16. The obtained producer's accuracy for the destroyed buildings was better than for k-means classification and reached 92%, meaning that only 28

of the 356 destroyed buildings were incorrectly classified. However, the user's accuracy of 85% was worse than for k-means classification, showing that 59 objects of 387 were mistakenly classified as destroyed. The overall accuracy reached 84%.

The results of **SVM classification** are shown in Figure 5.17. The SVM classification was performed using the same training set as for the k-NN classification, with the classification rules built based on the RBF kernel function. The kernel parameters were determined using the grid-search method, with the classification accuracy evaluated by the 5-fold cross validation. As can be seen, 19 objects out of 356 destroyed reference building were incorrectly classified, reflecting a producer's accuracy of 95%. The obtained user's accuracy of 85% for destroyed building shows that the object set classified as destroyed contains 15% intact objects. Accordingly, the classification approach yields an overall accuracy of 86%.

		Reference Data 1	2	Row Total
Classified Data	1	133	19	152
	2	60	337	397
Column Total		193	356	549

Object categories

1 – Intact object

2 – Destroyed object

OVERALL ACCURACY = (133 + 337)/549 = 86%

PRODUCER'S ACCURACY	USER'S ACCURACY
1 = 133/193 = 69%	1 = 133/152 = 87%
2 = 337/356 = 95%	2 = 337/397 = 85%

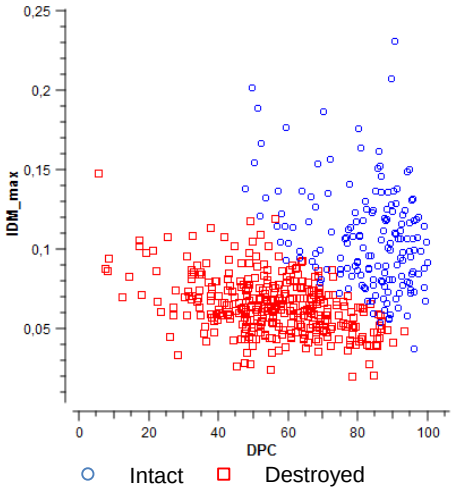
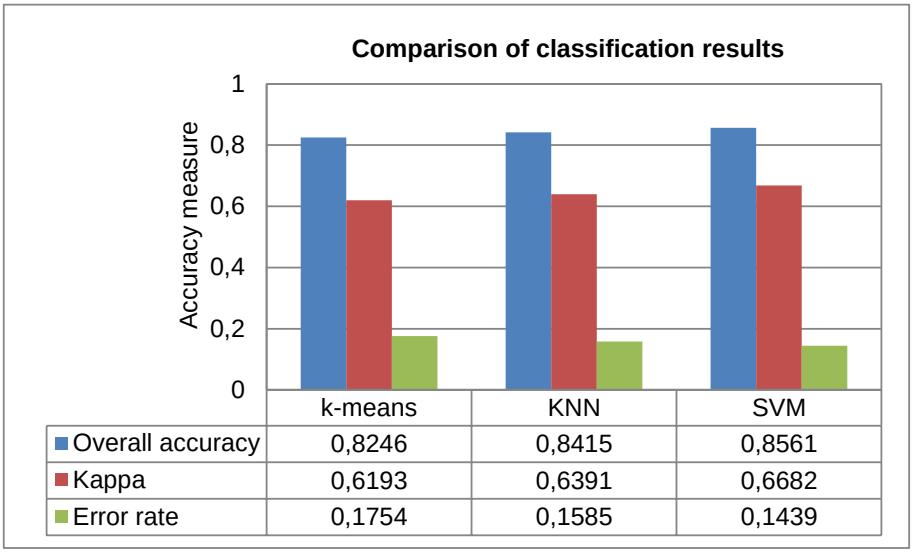


Figure 5.17: Classification results produced for study case Yushu, Qinghai, China (the 2010 Yushu earthquake) using the SVM algorithm: confusion matrix (left) for accuracy assessment and scatter plot (right) illustrating the separation of objects.

A comparative table with classification accuracies expressed by overall accuracy, kappa coefficient and error rate is shown in Figure 5.18. While all three approaches provide similar classification accuracy, the SVM classification leads to the best results.



	k-means	KNN	SVM
Overall accuracy	0,8246	0,8415	0,8561
Kappa	0,6193	0,6391	0,6682
Error rate	0,1754	0,1585	0,1439

Figure 5.18*:* Comparison of classification results obtained using different classifiers for the study case Yushu, Qinghai, China (the 2010 Yushu earthquake).

The resulting damage map was generated from the data obtained from the SVM classification, providing the more accurate results. Figure 5.19 illustrates the final map, with the intact buildings pictured by white polygons, destroyed buildings by red polygons and mistakenly classified buildings by green polygons.

Figure 5.19: Resulting damage map generated from the results obtained by SVM classification for the study case Yushu, Qinghai, China (the 2010 Yushu earthquake): white objects represent intact buildings, red – destroyed; and green objects correspond to the incorrectly classified buildings.

5.3 Third study case: Osnabrueck (Germany)

Finally, the proposed method was tested on a data set from the city of Osnabrueck, Germany. This experiment was intended to demonstrate the possibility to adapt the proposed method to non-catastrophic applications, for detecting and documenting urban changes such as building demolishing or reconstruction. Figure 5.20 illustrates this study case, showing a

scene from Eversburg, in the north-west district of the city of Osnabrueck. The data set contains 319 vector objects extracted from a cadastral GIS database (Automated Land Registration Map (ALK), Osnabrueck, Germany) and a 649 × 417 pixel image area from an aerial photograph. The reference information containing the original building state was collected by the visual image interpretation, with the objects divided into two classes, namely changed and unchanged objects.

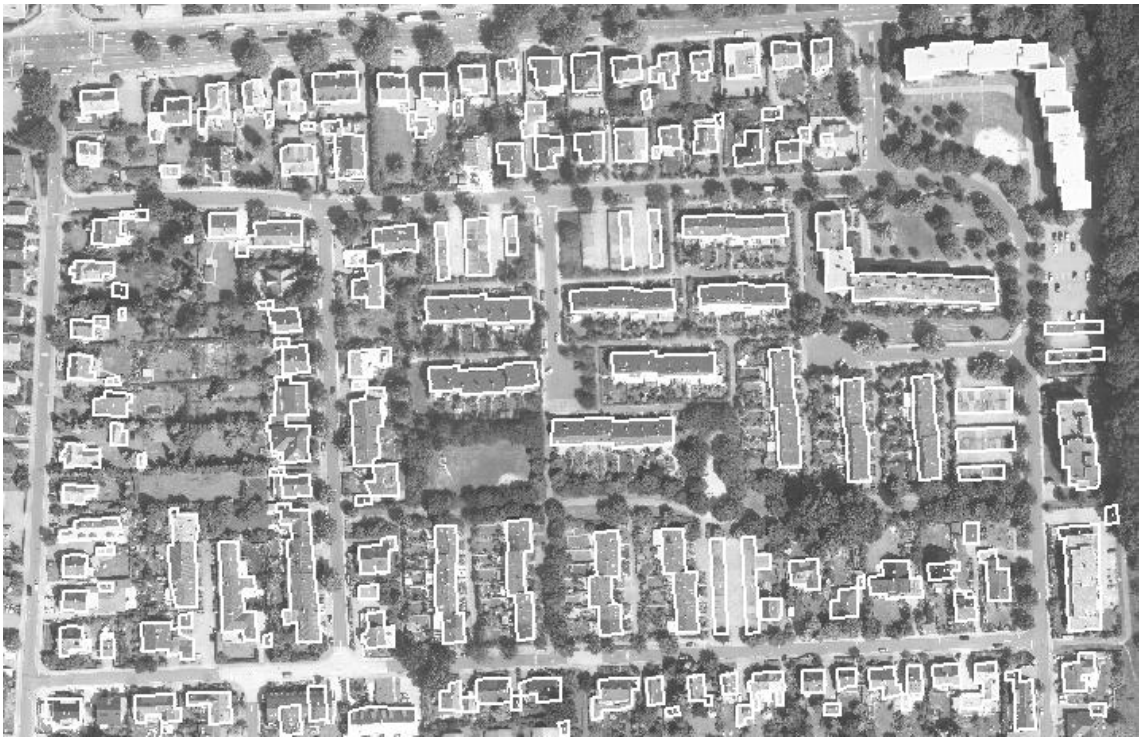


Figure 5.20: Third study case: a part of aerial photograph of the city Osnabrueck, Germany (1994), overlaid with the related vector data (data courtesy University of Osnabrueck and Automated Land Registration Map (ALK), City of Osnabrueck).

Figure 5.21 demonstrates a scatter plot where the separation of the objects can be clearly observed.

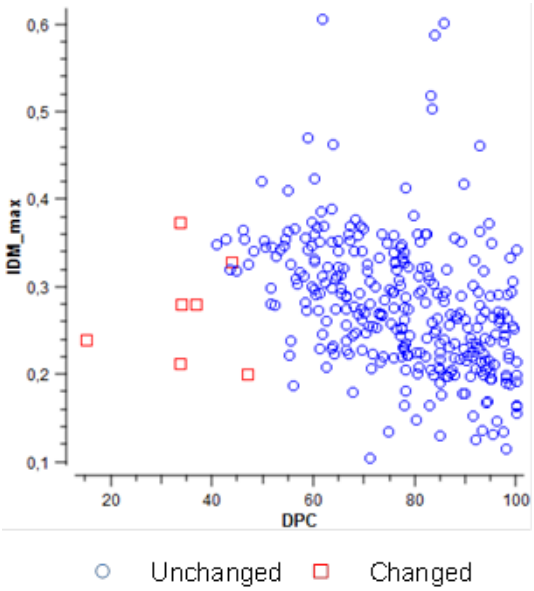


Figure 5.21*:* Scatter plot demonstrating the good separation of the changed and unchanged object sets generated for the study case Osnabrueck (Germany)*.*

The studied area does not contain any destroyed buildings; the roof images possess a rather homogeneous texture for all changed and unchanged objects. Due to change of object location or geometry, the image corresponding to the original vector contour of changed objects contains fragments of vegetation and constructions surrounding objects. This image tends to have a heterogeneous structure, as well as the images corresponding to destroyed objects. Thus, for this case, the textural features describing the homogeneity of an area also allow detecting changed objects. However, it is obvious that the explicit measure of the contour integrity provided by the DPC feature must be a more reliable parameter for the identification of building demolition or reconstruction.

Due to the architectural design, the study area includes buildings consisting of a series of adjacent vector objects joined by a common wall (Figure 5.22). Such joined objects were united by dissolving boundaries between objects sharing common attributes. Through the merging procedure, the vector data set was reduced to 157 objects.

(a) (b) (c)

Figure 5.22: Representation of buildings in the data set Osnabrueck (Germany): (a) buildings in the original image; (b) original image overlaid with the cadaster data, where the buildings really consist in the series of single buildings joined by the common wall; (c) buildings with the dissolved boundaries.

The object set contains only few changed objects, thus making it difficult to apply the classification strategies described above. However, it can be seen from the scatter plot presented in Figure 5.23 that the two groups - changed and unchanged buildings - can be well separated.

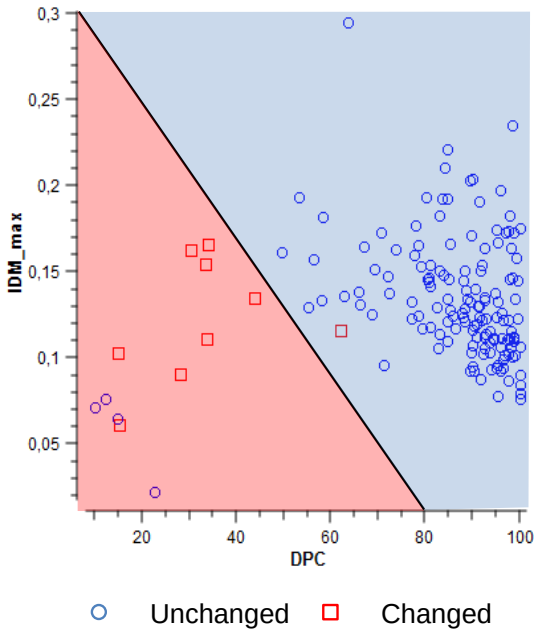


Figure 5.23: Classification of building for the study case Osnabrueck (Germany).

The classification rule can be easily derived based on the separating line $80\,IDMmax + 0.3\,DPC - 24 = 0$ shown on the figure:

$$Building\ State = \begin{cases} Unchanged, & if\ 80\,IDMmax + 0.3\,DPC - 24 \geq 0, \\ Changed, & if\ 80\,IDMmax + 0.3\,DPC - 24 < 0. \end{cases}$$

Classified Data	Reference Data 1	Reference Data 2	Row Total
1	144	1	145
2	4	8	12
Column Total	148	9	157

Object categories

1 – Intact object

2 – Destroyed object

OVERALL ACCURACY = (144 + 8)/157 = 97%

PRODUCER'S ACCURACY	USER'S ACCURACY
1 = 144/148 = 97%	1 = 144/145 = 99%
2 = 8/9 = 89%	2 = 8/12 = 67%

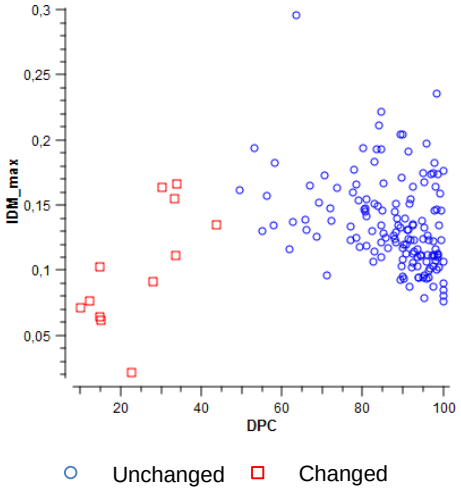


Figure 5.24: Classification result produced for the study case Osnabrueck (Germany): confusion matrix (left) for accuracy assessment and scatter plot (right) illustrating the separation of objects.

Figure 5.25: Resulting damage map generated for the study case Osnabrueck (Germany): white objects represent intact buildings, red – destroyed; and green objects correspond to the incorrectly classified buildings.

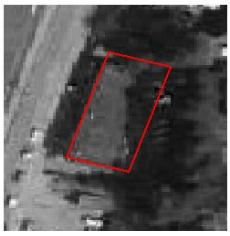
5.4 Discussion

For the aforementioned study cases, the method proposed in this thesis yields reliable classification results (see Table 5.3).

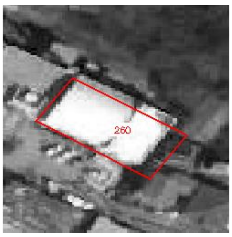
Study case	Total number of objects	Number of destroyed/ changed objects	Overall accuracy	Error rate
Japan	696	192	86,74%	0,13
China	610	395	85,61%	0,14
Osnabrueck	157	9	96,81%	0,03

Table 5.3: Classification summary table.

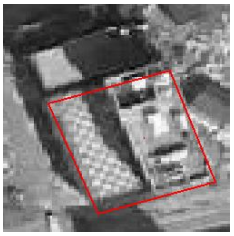
However, the method fails to identify the state of some objects. Accordingly, the possible reasons for such failures are discussed below.



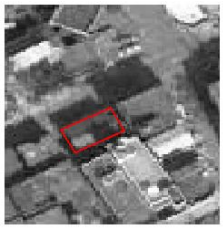
1) Building is occluded with trees. Building occlusions by trees cause inhomogeneous appearance on a roof image. Moreover, the contour of a building also cannot be fully detected by the DPC.



2) Deviation between the vector and raster contours. It should be noted that the application of this change detection method implies an accurate georeferencing of used maps, i.e. a precise matching of the vector and raster contours of the objects. In fact, it is rather difficult to achieve such a high accuracy. In practice, there are always objects with the mismatching contours on vector and raster maps, which can be caused by different acquisition angles of the remotely sensed images, poor georeferencing, roof design etc. While the large contour mismatching can lead to wrong DPC values, for most objects with mismatched contours the DPC feature is calculated correctly due to the certain width of search windows and the procedure of correction of search window positions.



3) Objects on building roofs. The texture measures are sensitive to an appearance of heterogeneity caused by objects located on building roofs (pipes, air vents, antennas, etc.). However, in most cases, the DPC successfully recognized the contour of the building.



4) Building is entirely or partially covered by shadow. Shadows from the neighbouring objects can significantly worsen the results of the edge detection and the texture analysis.

To overcome such problems, the following approaches can be considered, with the new information being added in the form of additional features:

1. Taking into account additional spectral information (Tomowski, 2011; Awrangjeb et al., 2012; Dong & Shan, 2013).
2. Using approaches of 3D building change detection (Turker & Cetinkaya, 2005; Tong, et al., 2012).
3. Inclusion of an algorithm for shadow elimination in an image enhancement step (Das & Aery, 2013; Shahtahmassebi et al., 2013).

Despite the drawbacks listed above, the integrated analysis of the raster- and vector-based information applied in this work benefits the extraction of valuable data from the image due to the precise information concerning the geometry and location of each studied object. It enables filtering out meaningless information such as seasonal changes and movable objects (cars or peoples in the image), thus focusing the analysis solely on the objects of interest, collecting meaningful information in the form of features for further analysis through different data mining techniques. The easy integration of GIS information significantly simplifies the process of data interpretation with the powerful processing tools provided by GIS.

6. Conclusion and Further Research

The main goal of the thesis is a change detection technique for urban areas, based upon the combined analysis of vector and raster information. The study includes an overview of current change detection methods, the choice of OSS as implementation environment, development of a new methodology of change detection, application of a new technique for several study cases and assessment of its advantages and disadvantages.

The technique is supposed to be applied for a rapid assessment of destruction in urban areas affected by a catastrophic event. The developed technique can be applied as an independent change detection tool or in addition to the conventional methods. The field of application advances the special demands such as robustness, a high degree of automation, the utilization of full available information and efficient presentation of results, which are taken into account during the development. Moreover, the extensibility of the technique is also addressed. The structure of the developed change detection technique and its Open Source Software based implementation allow for easy extension of the technique. The development involves a combination of different techniques in the field of data mining, image enhancement, image processing, GIS, and visualization. For the treatment of vector information, a new feature, the DPC, is developed and the conventional approach to calculate GLCM textural features is adapted for object-based analysis.

Three independent study cases show the high potential of the presented technique, for both rapid mapping of building damage and new developments. However, some drawbacks were also recognized, including the strong influence of surrounding vegetation and shadows, deviation between raster and vector contours of a building and the appearance of some add-ons on the building roofs.

6.1 Considered problems

The study starts by providing an overview and analysis of conventional change detection techniques, vector and raster data structures, GIS application and the combined analysis of vector and raster data. Vector data is shown as effective supplement to raster data and their combination allows for improved change detection analysis, especially in urban areas. The GIS is chosen as the essential implementation platform for integrated analysis of vector and raster data.

The technique of change detection is implemented based on several Open Source packages (GRASS GIS, Python (programming language) and Orange (data mining software). Through the use of Python programming language, it becomes possible to integrate all parts of the algorithm into a single system, which enables easily automating the developed method.

The use of Open Source Software as implementation platform allows the fast inclusion and validation of new ideas and algorithms. The study of Open Source Software feasibilities proves that Open Source Software provides almost the same functionality as commercial software. Moreover, due to the flexible licences and no costs, the obtained implementation of the method can be easily distributed and customized.

A methodological concept for a change detection procedure based upon integrated information extracted from a vector and raster data is developed and described in detail. In the proposed change detection technique, several main steps can be realized: raster data enhancement, extraction of features describing objects and classification of object state. For each step, appropriate methods are selected. It is suggested to perform the image enhancement by means of the homomorphic filtering. The features are split into two groups, extracted from vector and raster data. As a feature representing vector data, the new feature Detected Part of Contour (DPC) is developed and used, which represents a detectable part of a building contour. The features corresponding to raster data are obtained based on Haralick's texture measures characterizing image homogeneity: Angular Second Moment (ASM), Inverse Difference Moment (IDM) and Inertia. The calculation of textural features is adapted for object-oriented analysis. It is found that the application of maximal values of the angularly dependent features is more efficient than the commonly used mean values. For the classification of building states, both the supervised and unsupervised approaches are studied, with methods proposed: k-means (unsupervised classification), k-NN and SVM (supervised classification).

The experimental part of this thesis demonstrates the effectiveness of the developed change detection algorithm based upon three study cases. The first study case illustrates destruction after the devastating tsunami caused by the 2011 Tohoku earthquake in Japan, the second data set was obtained after the 2010 Yushu earthquake in China and the aim of the third experiment was to demonstrate a possibility to adapt the suggested method to a non-catastrophic scenario involving reconstructed or demolished buildings.

From the feature set, the features DPC and IDM_max are selected as being most informative and are used for the classification. The experiments produce an overall accuracy of over 80% for all study cases. The analysis proves the efficiency of the integrated analysis of vector and raster data. Despite the high performance of the developed algorithm, the studies considered reveales several drawbacks, which are described in section 5.4.

6.2 Recommendations and further research

Based on the findings and aforementioned limitations, the study concluded with the following recommendations for further investigation:

1) While the technique of change detection is developed for the classification of buildings into destroyed and intact, in certain cases, buildings are only partially destroyed. In such a case, the algorithm can be adapted to evaluate the integrity of a building's individual segments/walls. In turn, this requires the application of classification approaches to solve multi-class problems given the various damage grades of buildings.
2) The usage of features extracted from spectral information and data of stereo photogrammetry. The new features can help to overcome the drawback of the algorithm related to the objects covered by crowns of trees or shadows.
3) The individual steps of the developed method can be improved depending on the particular situation. For example, new approaches to image enhancement, edge detection or classification techniques can be evaluated and implemented.
4) The efficiency of the change detection analysis can be studied for different catastrophic scenarios, such as floods, landslides, hurricanes, or fires.

Reference

Adams, J. B., Sabol, D., Kapos, V., Filho, R. A., Roberts, D. A., Smith, M. O., & Gillespie, A. R. (1995). Classification of multispectral images based on fractions of endmembers: aplication to land-cover change in the Brazilian Amazon. Remote Sensing of Environment, 52, pp. 137-154.

Agouris, P., Stefanidis, A., & Gyftakis, S. (2003). Differential snakes for change detection in road segments. Photogrammetric Engineering and Remote Sensing, 67(12), pp. 1391-1399.

Albertz, J. (2001). Einführung in die Fernerkundung: Grundlagen der Interpretation von Luft- und Satellitenbildern (2 ed.). Darmstadt: Wissenschaftliche Buch gesellschaft.

Al-Khudhairy, D.H.A., Caravaggi, I., Giad, S., 2005. Structural damage assessments from Ikonos data using change detection, object-oriented segmentation, and classification techniques. Photogrammetric Engineering & Remote Sensing, 71, pp. 825–837.

Alves, D. S., & Skole, D. L. (1996). Characterizing land cover dynamics using multitemporal imagery. International Journal of Remote Sensing, 17(4), pp. 835-839.

Armour, B., Tanaka, A., Ohkura, H., & Saito, G. (1998). Radar interferometry for environmental change detection. In R. S. Lunetta, & C. D. Elvidge, Remote Sensing change detection: environmental monitoring methods and applications (pp. 245-279). Chelsea, MI: Ann Arbor Press.

Awrangjeb, M., Zhang, C., & Fraser, C. S. (2012). Building detection in complex scenes thorough effective separation of buildings from trees. Photogrammeric Engineering and Remote Sensing, 78(7), pp. 729-745.

Baheerathan, S., Albregtsen, F., & Danielsen, H. E. (1999). New texture features based on the complexity curve. Pattern Recognition, 32, pp. 605-618.

Bajcsy, R., & Lieberman, L. (1976). Texture gradient as a depth cue. Computer Graphics and Image Pricessing, 5, S. 52-67.

Batista, G. E., & Silva, D. F. (2009). How k-Nearest Neighbor parameters affect its performace. 38º JAIIO - Simposio Argentino de Inteligencia Artificial (ASAI 2009), (pp. pp. 95-106). Mar del Plata, Argentina.

Beaurepaire, L., Chehdi, K., & Vozel, B. (1997). Identification of the Nature of Noise and Estimation of its Statistical Parameters by Analysis of Local Histograms. Proc. IEEE Int. Conf. Acoustics, Speech Signal Process. 4, pp. 2805 - 2808. Munich: IEEE Computer Society Washington, DC, USA.

Blaschke, T. (2010). Object based image analysis for remote sensing. ISPRS Journal of Photogrammetry and Remote Sensing, 65, pp. 2-16.

Bovolo, F., Bruzzone, L., Marconcini, M., 2008. A novel approach to unsupervised change detection based on a semisupervised SVM and a similarity measure. IEEE Transactions on Geoscience and Remote Sensing, 46, pp. 2070-2082.

Bourgeau-Chavez, L. L., Kasischke, E. S., Brunzell, S., Mudd, J. P., & Tukman, M. (2002). Mapping fire scars in global boreal forest using imaging radar data. International Journal of Remote Sensing, 23, pp. 4211-4234.

Bramer, M. (2013). Principles of Data Mining. London: Springer.

Briger, J. (2002). Multisensorale und multitemporale Fernerkundungsdaten zur Erfassung, Differenzierung und Veränderungsanalyse ausgewählter Vegetationsstrukturen der Bergbaufolgelandschaft Mitteldeutschlands. Dissertation.

Brodatz, P. (1966). Textures: A Photographic Album for Artists and Designers. Dover, NewYork, NY,USA: Dover Publications Inc.

Bruzzone, L., Prieto, D.F., 2000. Automatic analysis of the difference image for unsupervised change detection. IEEE Transactions on Geoscience and Remote Sensing, 38, pp. 1171-1182.

Burges, C. J. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. In Data Mining and Knowledge Discovery (pp. 121–167). Netherlands: Kluver Academic Publishers.

Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(6), pp. 679-698.

Centeno, J. S. (2000). Integration of satellite imagery and GIS for land-use classification purposes. International Archives of Photogrammetry and Remote Sensing, XXXIII, Part B7, pp. 1389-1394. Amsterdam.

Chandra, A. M., & Ghosh, S. K. (2006). Remote Sensing and Geographical Information System. Alpha Science International Ltd.

Chen, G., Hay, G. J., Carvalho, L. M., & Wulder, M. A. (2012). Object-based change detection. International Journal of Remote Sensing, 14(33), pp. 4434-4457.

Chesnel, A., Binet, R., & Wald, L. (2008). Urban damage assessment using multimodal QuickBird images and ancillary data: the Bam and the Boumerdes earthquakes. 6th International Workshop on remote Sensing for Disaster Management Applications. Pavia, Italy.

Cihlar, J., Pultz, T. J., & Gray, A. L. (1992). Change detection with synthetic aperture radar. International Journal of Remote Sensing, 13, pp. 401-414.

Clarke, B., Fokoue, E., & Zhang, H. H. (2009). Principles and theory for data mining and machine learning. Dordrecht Heidelberg London New York: Springer.

Cohen, F., Fan, Z., & Patel, M. A. (1991). Classification of rotated and scaled textured images using Gaussian Markov random field models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(2), S. 192 - 202.

Cohen, J. (1960). A coefficient of agreement for niminal scales. Educational and Psychological Measurement, 20(1), pp. 37-46.

Congalton, R. (1991). A review of assessing the accuracy of classifications of remotely sensed data. Remote sensing of environment, 37, pp. 35-46.

Congalton, R. G., & Green, K. (2009). Assessing the Accuracy of Remotely Sensed Data: Principles and Practices. CRC Pr Inc.

Conners, R. W., & Harlow, C. A. (1980). A theoretical comparison of texture algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-2, pp. 204-222.

Coppin, P., Jonckheere, I., Nackaerts, K., Muys, B., & Lambin, E. (2004). Review article. Digital change detection methods in ecosystems monitoring a review. International Journal of Remote Sensing, 25, pp. 1565-1596.

Coppin, P., Nackaerts, K., Quenn, L., & Brewer, K. (2001). Operational monitoring of green biomass change for forest management. Photogrammentric Engineering and Remote Sensing, 67, pp. 603-611.

Cuomo, V., Lasaponara, R., & Tramutoli, V. (2001). Evaluation of new satellite-based method for forest fire detection. International Journal of Remote Sensing, 22, pp. 1799-1826.

Das, S., & Aery, A. (2013). A review: shadow detection and shadow removal from images. International Journal of Engineering Trends and Technology (IJETT), 4(5), pp. 1764-1767.

de Lange, N. (2006). Geoinformatik: in Theorie und Praxis (2 ed.). Springer.

Delac, K., Grgic, M., & Kos, T. (2006). Sub-image homomorphic filtering technique for improving facial identification under difficult illumination conditions. International Conference on Systems, Signals and Image Processing. Budapest, Hungary.

Dellaert, F. (2002). The Expectation Maximization Algorithm. Technical Report number GIT-GVU-02-20. Georgia Institute of Technology.

Demsar, J., & Zupan, B. (2012). ORANGE: Data mining fruitful and fun. http://ailab.ijs.si/dunja/TuringSLAIS-2012/Papers/Demsar_Orange.pdf.

Derin, H., & Howard, E. (1987). Modeling and segmentation of noisy and textured images using Gibbs Random Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9(1), pp. 39 - 55.

Dewidar, K. M. (2004). Detection of land use/land cover changes for the northern part of the Nile delta (Burullus region), Egypt. International Journal of Remote Sensing, 25(20), pp. 4079-4089.

Deza, M. M., & Deza, E. (2009). Encyclopedia of distances. Berlin, Heidelberg: Springer.

Dillworth, M. E., Whistler, J. L., & Merchant, J. W. (1994). Measuring landscape structure using geographic and geometric windows. Photogrammetric Engineering and Remote Sensing, 60(10), pp. 1215-1224.

Dong, L., & Shan, J. (2013). A comprehensive review of earthquake-induced building damage detection with remote sesning techniques. ISPRS Journal of Photogrammetry and Remote Sensing, 84, pp. 85-99.

Duan, K.-B., & Keerthi, S. (2005). Which Is the Best Multiclass SVM Method? An Empirical Study. Proceedings of the Sixth International Workshop on Multiple Classifier Systems (pp. 278-285). Springer-Verlag, Berlin Heidelberg.

Dutra, L. V., & Mascarenhas, D. A. (1984). Some experiments with spatial feature extraction methods in multisüectral classification. International Journal of Remote Sensing, 5(2), pp. 303-313.

Ehlers, M., Edwards, G., Bedard, Y. (1989). Integration of remote sensing with Geographic information Systems: a necessary evolution. Photogrammetric Engineering and Remote Sensing, 55(11), pp. 1619-1627.

Ehlers, M., & Schiewe, J. (2012). Geoinformatik. Darmstadt: Wissenschaftliche Buchgesellschaft.

Ehlers, M., Klonus, S., Tomowski, D., Michel, U., & Reinartz, P. (2010). Automated change detection from high-resolution remote sensing images. A special joint sysmposium of ISPRS Technical Cimmission IV and AutoCarto in conjunction with ASPRS/CaGIS 2010 Fall Specialty Conference. Orlando, Florida.

Elvidge, C. D., Pack, D. W., Prins, E., Kihn, E. A., Kendall, J., & Baugh, K. E. (1998). Wildfire detection with meteorologica satellite data:results from New Mexico during June of 1996 using GOES, AVHRR, and DMSP-OLS. In R. S. Lunetta, & C. D. Elvidge, Remote sensing change detection: environmental monitoring methods and applications (pp. 103-121). Chelsea, MI: Ann Arbor Press.

Fang, L., & Picard, R. W. (1996). Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(7), S. 722 - 733.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge doscovery in databases. AI Magazine, 17(3), pp. 37-54.

Franklin, S. E., Dickson, E. E., Farr, D. R., Hansen, M. J., & Moskal, L. M. (2000). Quantification of landscape change from satellite remote sensing. Forestry Chronicle, 76, pp. 877-886.

Fuller, D. O. (2000). Satellite remote sensing of biomass burning with optical and thermal sensors. Progress in Pysical Geography, 24, pp. 543-561.

Garcia-Haro, F. J., Gilabert, M. A., & Melia, J. (2001). Monitoring fire-affected areas using Thematic Mapper data. International Journal of Remote Sensing, 22, pp. 533-549.

Gautam, N. C., & Chennaiah, G. C. (1985). Land-use and land-cover mapping and change detection in Tripura using satellite Landsat data. International Journal of Remote Sensing, 6, pp. 517-528.

Gautama, S., Haeyer, D. J., & Philips, W. (2006). Graph-based change detection in geographic information using VHR satellites images. International Journal of Remote Sensing, 27(9-10), pp. 1809-1824.

Genton, M. G. (2001). Classes of kernels for machine learning: a statistical perspective. Journal of Machine Learning Research, 2, pp. 299-312.

Ghosh, A. (2006). On optimum choice of k in nearest neighbor classification. Computational Statistics & Data Analysis, 50, S. 3113 – 3123.

Goldberg, D. E. (1989). Genetic algorithms in search, optimization and machine learning. Addison-Wesley Lingman.

Gong, P. (1992). A comparison of spatial feature extraction algorithms for land-use classification with SPOT HRV data. Remote Sensing of Environmental, 40, pp. 137-151.

Gonzalez, R. C., & Woods, R. E. (2002). Digital Image Processing. New Jersey: Printice Hall.

Gopal, S., & Woodcock, C. E. (1996). Remote sensing of forest change using artificial neural networks. IEEE Transactions on Geoscience and Remote Sensing, 34, pp. 398-403.

Graetz, R. D., Pech, R. P., & Davis, A. W. (1988). The assessment and monitoring of sparsely vegetated rangelands using calibrated Landsat data. International Journal of Remote Sensing, 8, pp. 1201-1222.

Green, K., Kempka, D., & Lackey, L. (1994). Using remote sensing to detect and monitor land-cover and land-use change. Photogrammentic Engineering and Remote Sensing, 60, pp. 331-337.

Gupta, R. P., Saraf, A. K., Saxena, P., & Chander, R. (1994). IRS detection of surface effects of the Uttarkashi earthquake of 20 October 1991, Himalaya. International Journal of Remote Sensing, 15, pp. 2153-2156.

Gutman, G. (2004). Land change science: Observing, monitoring and understanding trajectories of change on the Earth's surface. Dordrecht: Kluwer Academic Publishers.

Hall, F. G., Strebel, D. E., Nickeson, J. E., & Goetz, S. J. (1991). Radiometric rectification: toward a common radiometric response among multidate, multisensor images. Remote Sensing Environment, 35, pp. 11-27.

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. IEEE Transactions on Systems, Man and Cybernetics, SMC-3(6), pp. 610-621.

Hartigan, J. A. (1975). Clustering algorithms. New York London Sydney Toronto: John Wiley & Sons Inc.

Henebry, G. M. (1993). Detecting change in grasslands using measures of spatial dependence with Landsat TM data. Remote Sensing of Environment, 46, pp. 223-234.

Hodgson, M. E. (1998). What size window for image classification? A cognitive perspective. Photogrammetric Engineering and Remote Sensing, 64(8), pp. 797-807.

Horn, B. P. (1975). Image intensity understanding. Artificial Intelligence Laboratory. Massachusetts Institute of Technology .

Howarth, P. J., & Wickware, G. M. (1981). Procedures for change detection using Landsat digital data. 2(3), pp. 277-291.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). A practical guide to support vector classification. National Taiwan University, Taipei: Department of Computer Science and Information Engineering.

Hsu, S. Y. (1978). Texture-tone analysis for automated land-use mapping. Photogrammetric Engineering and Remote Sensing, 44(11), pp. 1393-1404.

Huertas, A., & Nevatia, R. (1988). Detecting buldings in aerial images. Computer vision, Graphics, and Image Processing, 41, pp. 131-152.

Hussain, M., Chen, D., Cheng, A., Wei, H., & Stanley, D. (2013). Change detection from remotely sensed images: from pixel-based to object-based approaches. ISPRS Journal of Photogrammetry and Remote Sensing(80), pp. 91-106.

Hussin, Y. A., De Gier, A., & Hargyono. (1994). Forest cover change detection analysis using remote sensing: a test for the spatially resolved area production model. Fifth European Conference and Exhibition on Geographic Information Systems, EGIS'94 Proceedings. II, pp. 1825-1834. Paris, France: Utrecht: EGIS Foundation.

Ingram, K., Knapp, E., & Robinson, J. W. (1981). Change detection technique development for improved urbanized area delineation. (S. Springs, Ed.) Maryland, USA.

International Atomic Energy Agency. (2006). Chernobyl's legacy: health, environmental and socio-economic impacts and recommendations to the governments of Belarus, the Russian Federation and Ukraine. Vienna, Austria: International Atomic Energy Agency (IAEA).

Irons, J. R., & Petersen, G. W. (1981). Texture transforms of remote sensing data. Remote Sensing of Environment, 11, pp. 359-370.

Irvin, R. B., & McKeown, D. M. (1989). Methods for exploiting the relationship between buildings and their shadows in aerial imagery. IEEE Transactions on Systems, Man, and Cybernetics, 19, pp. 1564-1575.

Jakubauskas, M. E., Kamlesh, P. L., & Mausel, P. W. (1990). Assessment of vegetation change in a fire altered forest landscape. Photogrammetric Engineering and Remote Sensing, 56, pp. 371-377.

Janert, P. K. (2010). Data analysis with Open Source Tools. O'Reilly Media.

Jayaraman, S., Esakkirajan, S., & Veerakumar, T. (2011). Digital image processing. West Patel Nagar, New Delhi: Tata McGraw-Hill Education.

Jensen, J. R. (1979). Spectral and textural features to classify elusive land cover at the urban fringe. The Professional Geographer, 31(4), pp. 400-409.

Jensen, J. R. (2007). Remote sensing of the environment: An earth resouce perspective. Upper Saddle River, NJ: Pearson Prentice Hall.

Jensen, J. R., Cowen, D. J., Althausen, J. D., Narumalani, S., & Weatherbee, O. (1993). An evaluation of the Coastwatch change detection protocol in South Carolina. Photogrammetric Engineering and Remote Sensing, 59, pp. 1039-1046.

Johnson, R. (2011). Japan's 2011 Earthquake and Tsunami: food and agriculture implications. Congressional Research Service.

Katsavounidis, I., Kuo, C.-C. J., & Zhang, Z. (1994). A new initialization technique for generalization Lloyd iteration. IEEE Signal Processing Letters, 1(10), pp. 144-146.

Kaufmann, R. K., & Seto, K. C. (2001). Change detection ,accuracy, and bais in a sequential analysis of Landsat imagery in the Pear River Delta, China: econometric techniques. Agriculture Ecosystems and Environment, 85, pp. 95-105.

Kauth, R. J., & Thomas, G. S. (1976). The Tasseled Cap - a graphic description of the spectral-temporal development of agricultural crops as seen by Landsat. Proceedings of the Symposium on Machine Processing of Remotely Sensed Data (pp. 4B41-4B51). West Lafayette, USA: Purdue University.

Keshava, N., & Mustard, J. F. (2002, January). Spectral Unmixing. IEEE Signal Processing Magazine, 1053(5888), pp. 44-57.

Klonus, S., Tomowski, D., Ehlers, M., Reinartz, P., & Michel, U. (2012). Combined edge segment texture analysis for the detection of damaged buildings in crisis areas. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 5(4), pp. 1118-1128.

Koch, M., & Ei-Baz, F. (1998). Identifying the affects of the Gulf War on the geomorphic features of Kuwait by remote sensing and GIS. Photogrammetric Engineering and Remote Sensing, 07, pp. 739-747.

Köhl, M., Magnussen, S., & Marchetti, M. (2006). Sampling methods, remote sensing and GIS multisource forest inventiry. Berlin, London: Springer.

Kumar, A., Chingkhei, R. K., & Dolendro, T. (2007). Tsunami damage assessment: a case study in Car Nicobar Island, India. International Journal of Remote Sensing, 28(13-14), pp. 2937-2959.

Lambin, E. F., & Strahler, A. H. (1994). Indicators of land cover change for change-vector analysis in multitemporal space at coarse spatial scales. International Journal of Remote Sensing, 15, pp. 2099-2119.

Lande, D. V., Snarskii, A. A., & Bezsudnov, I. V. (2009). Internetica. Navigating in complex networks: models and algorithms. Moscow: Librocom (russian).

Lawrence, R. L., & Ripple, W. J. (1999). Calculating change curves for multitemporal satellite imagery: Mount St. Helens 1980-1995. Remote Sensing of Environment, 67, pp. 309-319.

Laws, K. I. (1980). Rapid texture identification. SPIE Image Processing for Missile Guidance, 238, pp. 376-380.

Lehmann, T., Oberschelp, W., Pelikan, E., & Repges, R. (1997). Bildverarbeitung für die Medizin: Grundlagen, Modelle, Methoden, Anwendungen. Berlin: Springer.

Li, D. (2010). Remotely sensed images and GIS data fusion for automatic change. International Journal of Data Fusion, 1(1), pp. 99-108.

Li, X., & Strahler, A. H. (1985). Geometric-optical modeling of conifer forest canopy. IEEE Transactions on geoscience and remote sensing, GE-23(5), pp. 705-721.

Li, X., & Yeh, A. G. (1998). Principal component analysis of stacked multitemporal images for the monitoring of rapid urban expansion in the Pearl River Delta. International Journal of Remote Sensing, 19, 1501-1518.

Ling, H., & Jacobs, D. W. (2007). Shape classification using the Inner-Distance. IEEE Transaction on Pattern Analysis and Machine Intelligence, 29, pp. 286-299.

Liu, X., & Lathrop, R. G. (2002). Urban change detection based on an artificial neural network. International Journal of Remote Sensing, 23, pp. 2513-2518.

Lo, C. P., & Shipman, R. L. (1990). A GIS approachto land-use change dynamics detection. Photogrammetric Engeneering and Remote Sensing, 56(11), pp. 1483-1491.

Lu, D., Mausel, P., Brondizio, E., & Moran, E. (2004). Change detection techniques. International Journal of Remote Sensing, 25(12), pp. 2365-2407.

Lu, D., & Weng, Q. (2004). Spectral mixture analysis of the urban landscape in Indianapolis with Landsat ETM+ imagery. Photogrammetric Engineering and Remote Sensing, 70(9), pp. 1053-1062.

Lupo, F., Reginster, I., & Lambin, E. F. (2001). Monitoring land-cover changes in West Africa with SPOT vegetation: impact of natural disasters in 1998-1999. International Journal of Remote Sensing, 22(13), pp. 2633-2639.

Macleod, R. D., & Congalton, R. G. (1998). A quantitative comparison of change-detection algorithms for monitoring eelgrass from remotely sensed data. Photogrammetric Engineering and Remote Sensing, 64(3), pp. 207-216.

Macomber, S. A., & Woodcock, C. E. (1994). Mapping and monitoring conifer mortality using remote sensing in the Lake Tahoe Basin. Remote Sensing of Environment, 50, pp. 255-266.

Manavalan, P., Kesavasamy, K., & Adiga, S. (1995). Irrigated crops monitoring through seasons using digital change detection analysis of IRD-LISS 2 data. International Journal of Remote Sensing, 16, pp. 633-640.

Mandelbrot, B. B. (1983). The Fractal Geometry of Nature. Henry Holt and Company.

Markovitch, S., & Rosenstein, D. (2002). Feature generation using general construction functions. Machine Learning, 49, pp. 59-98.

Mas, J.-F. (1999). Monitoring land-cover changes: a comparison of change detection techniques. International Journal of Remote Sensing, 20(1), pp. 139-152.

Materka, A., & Strzelecki, M. (1998). Texture analysis methods - a review. Technical University of Lodz, Institute of Electronics, Brussels.

Mattikalli, N. M., Devereux, B. J., & Richards, K. S. (1995). Integration of remotely sensed satellite images with a geographical informationsystem. Computers and Geosciences, 21(8), pp. 947-956.

Melgani, F., & Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. IEEE Transactions on Geoscience and Remote Sensing, 42, pp. 1778-1790.

Melgani, F., Moser, G., Serpico, S. B. (2002). Unsupervised change-detection methods for remote-sensing images. Opt. Eng. 41(12), pp. 3288-3297.

Merchant, J. W. (1984). Using spatial logic in classification of Landsat TM data. Proceedings of the Pecora IX Symposium, (pp. 378-385). Sioux Falls, South Dakota.

Michener, W. K., & Houhoulis, P. F. (1997). Detection of vegetation associated with extensive flooding in a forested ecosystem. Photogrammetric Engineering and Remote Sensing, 63, pp. 1363-1374.

Mimura, N., Yasuhara, K., Kawagoe, S., Yokoki, H., & Kazama, S. (2011). Damage from the Great East Japan Earthquake and Tsunami - a quick report. Mitig Adapt Strateg Glob Change.

Morisette, J. T., Khorram, S., & Mace, T. (1999). Land-cover change detection enhanced with generalized linear models. International Journal of Remote Sensing, 20, pp. 2703-2721.

Nayak, S., & Zlatanova, S. (2008). Remote Sensing and GIS technologies for monitoring and Prediction of Disasters. Berlin Heidelberg: Springer-Verlag.

Nellis, M. D., & Briggs, J. M. (1989). The effect of spatial scale on Konza landscape classification using textural analysis. Lanscape Ecology, 2(2), pp. 93-100.

Nelson, R. F. (1983). Detecting forest canopy change due to insect activity using Landsat MSS. Photogrammetric Engineering and Remote Sensing, 49, pp. 1303-1314.

Neteler, M., & Mitasova, H. (2005). Open Source GIS: A GRASS GIS Approach. Springer.

Neteler, M., Bowman, H. M., Landa, M., & Metz, M. (2012). GRASS GIS: A multi-purpose open source GIS. Environmental Modelling and Software, 31, pp. 124-130.

Niemeyer, I., & Nussbaum, S. (2006). Automatisierte Detektion, Klassifizierung und Visualisierung von Veränderungen auf der Grundlage von fernerkundungsdaten. Tagungsband des 7. Geokinematischen Tages vom 11. und 12. Mai 2006 in freiberg (pp. 248-257). Essen: Glückauf.

Niemeyer, I., Marpu, P.R., Marpu, P.R., 2008. Change detection using object features. In: Blaschke, T., Lang, S., Hay, G.J. (Eds.), Object-Based Image Analysis: Spatial Concepts for Knowledge-Driven Remote Sensing Applications. Springer Verlag, Berlin Heidelberg, pp. 185–201.

Nordberg, M.-L., & Evertson, J. (2005). Vegetation index differencing and Linear regression for change detection in a Swedish mountain range using Landsat TM and ETM+ imagery. Land degradation & Development, 16, pp. 139-149.

Olson, D. L., & Delen, D. (2008). Advanced Data Mining Techniques. Springer-Verlag Berlin Heidelberg.

Peters, A. J., Walter-Shea, E. A., Lei, J., Vina, A., Hayes, M., & Svoboda, M. D. (2002). Drought monitoring with NDVI-based standardized vegetation index. Photogrammetric Engineering and Remote Sensing, 68, pp. 71-76.

Petit, C. C., & Lambin, E. F. (2001). Integration of multi-source remote sensing data for land cover change detection. International Journal of Geographical Information Science, 15, pp. 785-803.

Pratt, W. K. (1991). Digital Image Processing. New York, NY, USA: John Wiley & Sons, Inc.

Price, K. P., Pyke, D. A., & Mendes, L. (1992). Shrub dieback in a semiarid ecosystem:the integration of remote sensing and GIS for detecting vegetation change. Photogrammetric Engineering and Remote Sensing, 58, pp. 455-463.

Quintano, C., Fernandez-Manso, A., Shimabukuro, Y. E., & Pereira, G. (2012, September). Spectral Unmixing. International Journal of Remote Sensing, 33(17-18), pp. 5307-5340.

Radeloff, V. C., Mladenoff, D. J., & Boyce, M. S. (1999). Detecting jack pine budworm defoliation using spectra lmixture analysis: separating effects from determinats. Remote Sensing of Environment, 69, pp. 156-169.

Randen, T., & Husoy, J. (1999). Filtering for texture classification: a comparative study. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(4), pp. 291 - 310.

Raymond, E. S. (1999). The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. O'Reilly Media, p. 279.

Read, J. M., & Lam, N. S.-N. (2002). Spatial methods for characterizing land cover and detecting land cover changes for the tropics. International Journal of Remote Sensing, 23, pp. 2457-2474.

Republic of the Marshall Islands Maritime Administrator (2011). Deepwater Horizont marine casuality investigation report. Republic of the Marshall Islands Maritime Administrator.

Rigina, O., Baklanov, A., Hagner, O., & Olsson, H. (1999). Monitoring of forest damage in the Kola Peninsula, Northern Russia due to smelting industry. Science of the Total Environment, 229, pp. 147-163.

Roberts, D. A., Batista, G. T., Pereira, J. L., Waller, E. K., & Nelson, B. W. (1998). Change identification using multitemporal spectral mixture analysis: applications in eactern Amazonia. In L. a. Elvidge, Remote Sensing Change Detection: environmental Monitoring Methods and Applications, Chelsea: Ann Arbor Press. pp. 137-161.

Roberts, D.A., Gardner, M., Church, R., Ustin, S., Scheer, G., Green, R.O. (1998). Mapping chaparral in the Santa Monica mountains using multiple endmember spectral mixture models. Remote Sensing of Environment, 65, pp. 265-279.

Saksa, T., Uuttera, J., Kolström, T., Lehikoinen, M., Pekkarinen, A.& Sarvi,V. (2003). Clear-cut detection in boreal forest aided by remote sensing. Scandinavian Journal of Forest Research, 18(6), pp. 537–546.

Samadzadegan, F., & Rastiveisi, H. (2008). Automatic detection and classification of damaged buildings, using high resolution satellite imagery and vector data. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXVII, pp. 415-420. Beijing.

Saraf, A. K., Sinvhal, A., Sinvhal, H., Ghosh, P., & Sarma, P. (2002). Satellite data reveals 26 January 2001 Kutch earthquake-induced ground changes and appearance of water bodies. International Journal of Remote Sensing, 23, pp. 1749-1756.

Shahtahmassebi, A., Ning, Y., Ke, W., Moore, N., & Zhangquan, S. (2013). Review of shadow detection and de-shadowing methods in remote sensing. Chinese Geographical Science, 23(4), pp. 403-420.

Shi, W., & Hao, M. (2012). A method to detect earthquake-damaged buildings. (internal report)

Singh, A. (1984). Tropical forest monitoring using digital Landsat data in northeastern India. Ph.D. thesis.

Singh, A. (1989). Review Article digital change detection techniques using remotely-sensed data. International Journal of Remote Sensing, 10(6), pp. 989-1003.

Sofina, N., Ehlers, M., & Michel, U. (2011). Object-based analysis of destroyed buildings based on remotely sensed image and GIS. Proc. SPIE 8181, Earth Resources and Environmental Remote Sensing/GIS Applications II. Prague, Czech Republic.

Sohn, Y., Moran, E., & Gurri, F. (1999). Deforestation in North-Central Yucatan (1985-1995): mapping secondary succession of forest and agricultural land use in Sotuta using the Cosine of the Angle Concept. Photogrammetric Engineering and Remote Sensing, 65(8), pp. 947-958.

Sommer, S., Hill, J., & Megier, J. (1998). The potential of remote sensing for monitoring rural land use change and their effects on soil conditions. Agriculture Ecosystems and Environment, 67, pp. 197-209.

Sonka, M. V. (1999). Image Processing Analysis and Machine Vision. International Thomson Publishing.

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. Remote Sensing of Environment, 62(1), pp. 77-89.

Stow, D. A., Tinney, L. R., & Estes, J. E. (1980). Deriving land use/land cover change statistics from Landsat: A study of prime agricultural land. Proceedimgs of the 14th International Symposium on Remote Sensing of Environment held in Ann Arbor in 1980 (pp. 1227-1237). Ann Arbor, Michigan: Environmental Research Institute of Michigan.

Stramodo, S., Bignami, C., Chini, M., Pierdicca, N., & Tertulliani, A. (2006). Satellite radar and optical remote sensing for earthquake damage detection: results from different case studies. International Journal of Remote Sensing, 27(18-20), pp. 4433-4447.

Sumathi, S., & Sivanandam, S.N. (2006). Introduction to data mining principles. Studies in Computational Intelligence (SCI), 29, pp. 1-20.

Sumer, E., & Turker, M. (2006). An integrated earthquake damage detection system. ISPRS. Salzburg: Salzburg University, Austria.

Takeuchi, K., & Chavoshian, A. (2011). Quick report of Great Eastern Japan Earthquake and Tsunami (As of 22 April 2011). Tsukuba 305-8516, Japan: ICHARM.

Taylor, J. C., Brewer, T. R., & Bird, A. C. (2000). Monitoring landscape change in the national parks of England and Wales using aerial photo interpolation and GIS. International Journal of Remote Sensing, 21, pp. 2737-2752.

Theau, J. (2012). Change detection. In W., Kresse & D.M., Danko, Springer Handbook of Geographic Information (pp. 75-94). Springer Berlin Heidelberg.

Tobler, W. R. (1959). Automation and Cartography. Geographical Review, 49(4), pp. 526-534.

Tomlinson, R. F. (1985). Geographic information Systems - the new frontier. The Operational Geographer(5), pp. 31-36.

Tomowski, D. (2011). Kombiniert integrierte Veränderungsanalyse zur Schadenskartierung in urbanen Gebieten auf Basis sehr hoch auflösender multispektraler Bilddaten. Dissertation, p. 212.

Tong, X., Hong, Z., Liu, S., Zhang, X., Xie, H., Li, Z., et al. (2012). Building-damage detection using pre- and post-seismic high-resolution satellite stereo imagery: a case study of the May 2008 Wenchuan earthquake. ISPRS Journal of Photogrammetry and Remote Sensing, 68, pp. 13-27.

Tso, B., & Mather, P. M. (2009). Classification methods for remotely sensed data. Taylor & Francis Group, LLC.

Tuceryan, M. (1994). Moment-based texture segmentation. Pattern Recognition Letters, 15, pp. 659-668.

Tuceryan, M., & Jain, A. K. (1998). Texture analysis. In C. H. Chen, L. F. Pau, & P. P. Wang, The Handbook of Pattern Recognition and Computer Vision (pp. 207-248). World Scientific Publishing Co.

Turker, M., & Cetinkaya, B. (2005). Automatic detection of earthquake-damaged buildings using DEMs created from pre- and post-earthquake stereo aerial photographs. International Journal of Remote Sensing, 26(4), pp. 823-832.

Turker, M., & San, B. T. (2004). Detection of collapsed buildings caused by the 1999 Izmir, Turkey earthquake through digital analysis of post-event aerial photographs. International Journal of Remote Sensing, 25(21), pp. 4701-4714.

Two Crows Corporation. (2009). Introduction to Data Mining and Knowledge Discovery, Third Edition. Two Crows Corporation, 10500 Falls Road, Potomac, MD 20854 (U.S.A.): Two Crows Corporation.

Tzotsos, A., & Argialas, D. (2008). Support vector machine classification for object-based image analysis. In T. Blaschke, S. Lang, & G. Hay, Object-based image analysis - spatial concepts for knowledge-driven remote sensing applications (pp. 663-677). Berlin: Springer Berlin Heidelberg.

UNICEF. (2011). Yushu Earthquake One Year Report. Beijing, China: UNICEF .

Van Gool, L., Dewaele, P., & Oosterlinck, A. (1985). Texture analysis Anno 1983. Computer Vision, Graphics, and Image Processing, 29(3), pp. 336–357.

Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. New York Berlin Heidelberg: Springer-Verlag.

Vapnik, V. N., & Chervonenkis, A. Y. (1974). Theory of pattern recoqnition. Moscow: Nauka (russian).

Vogelmann, J. E. (1988). Detection of forest change in the Green Mountains of Vermont using multi-spectral scanner data. International Journal of Remote Sensing, 9, pp. 1187-1200.

Vogelmann, J. E. (1989). Use of Thematic Mapper data for detection of forest damage caused by the Pear Thrips. Remote Sensing of Environment, 30, pp. 217-225.

Wang, F. (1993). A knowledge-based vision system for detecting land change at urban fringes. IEEE Transactions on Geoscience and Remote Sensing, 31, pp. 136-145.

Ward, D., Phinn, S. R., & Murray, A. T. (2000). Monitoring growth in rapidly urbanizing areas using remotely sensed data. 52, pp. 371-386.

Weng, Q. (2002). Land use change analysis in the Zhujian Delta of China using satellite remote sensing, GIS and stochastic modelling. Journal of Environmental Management, 64, pp. 273-284.

Westen, V. C. (2000). Remote sensing for natural disaster management. International Archives of Photogrammetry and Remote Sensing, XXXIII, Part B7, pp. 1609-1617. Amsterdam.

Wezska, J. S., Charles, R. D., & Rosenfeld, A. (1976). A comparative study of texture measures for terrain classification. IEEE Transactions on System, Man, and Cybernetics, SMC-6(4), pp. 269-285.

Whistler, J. L. (1989). Spatial Measurements Package Programm. Lawrence, Kansas: Kansas Applied Remote Sensing Programm.

Wikantika, K., Sinaga, A., Hadi, F., & Darmawan, S. (2007). Quick assessment on identification of damaged building and land-use changes in the post-tsunami disaster with a quick-llok image of IKONOS and Quickbird (a case study in Meulaboh City, Aceh). International Journal of Remote Sensing, 28(13-14), pp. 3037-3044.

Yu, C., Shen, S., Huang, J., Yi, Y., 2010. An object-based change detection approach using high-resolution remote sensing image and GIS data. In: 2010 International Conference on Image Analysis and Signal Processing (IASP), pp. 565–569.

Yue, T. X., Chen, S. P., Xu, B., Liu, Q. S., Li, H. G., Liu, G. H., & Ye, Q. H. (2002). A curve-theorem based approach for change detection and its application to Yellow River Delta. International Journal of Remote Sensing, 23, pp. 2283-2292.

Zhang, J., & Tan, T. (2002). Brief review of invariant texture analysis methods. Pattern Recognition, 35, pp. 735-747.

Zhang, Q., Wang, J., Peng, X., Gong, P., & Shi, P. (2002). Urban build-up land change detection with road density and spectral informationfrom multitemporal Landsat TM data. International Journal of Remote Sensing, 23, pp. 3057-3078.

Zhen, D. L., Wallin, D. O., & Hao, Z. Q. (1997). Rates and patterns change between 1972 and 1988 in the Changbai Mountain area of China and North Korea. Landscape Ecology, 12, pp. 241-254.

Appendix 5.1

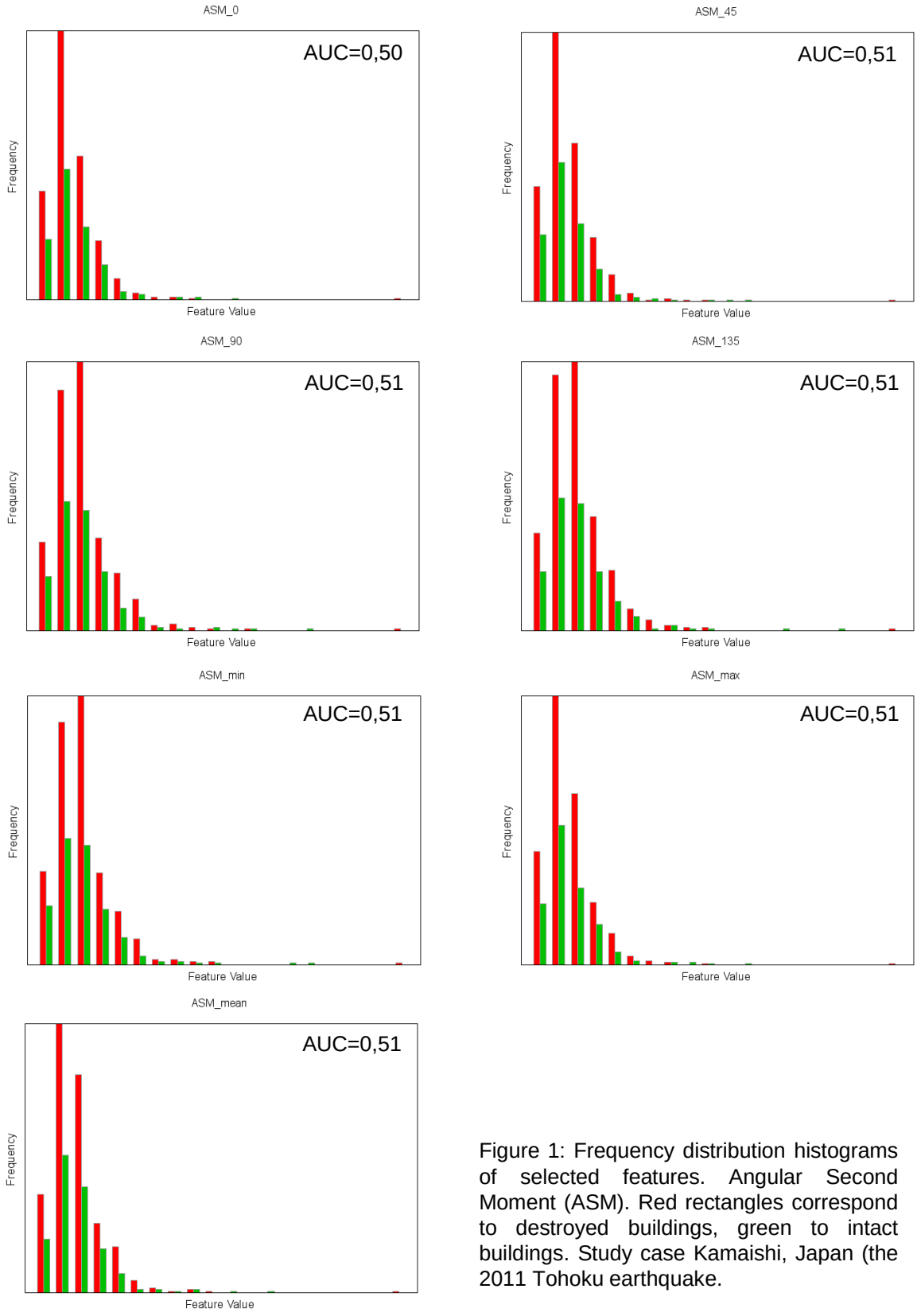


Figure 1: Frequency distribution histograms of selected features. Angular Second Moment (ASM). Red rectangles correspond to destroyed buildings, green to intact buildings. Study case Kamaishi, Japan (the 2011 Tohoku earthquake.

Appendix 5.1

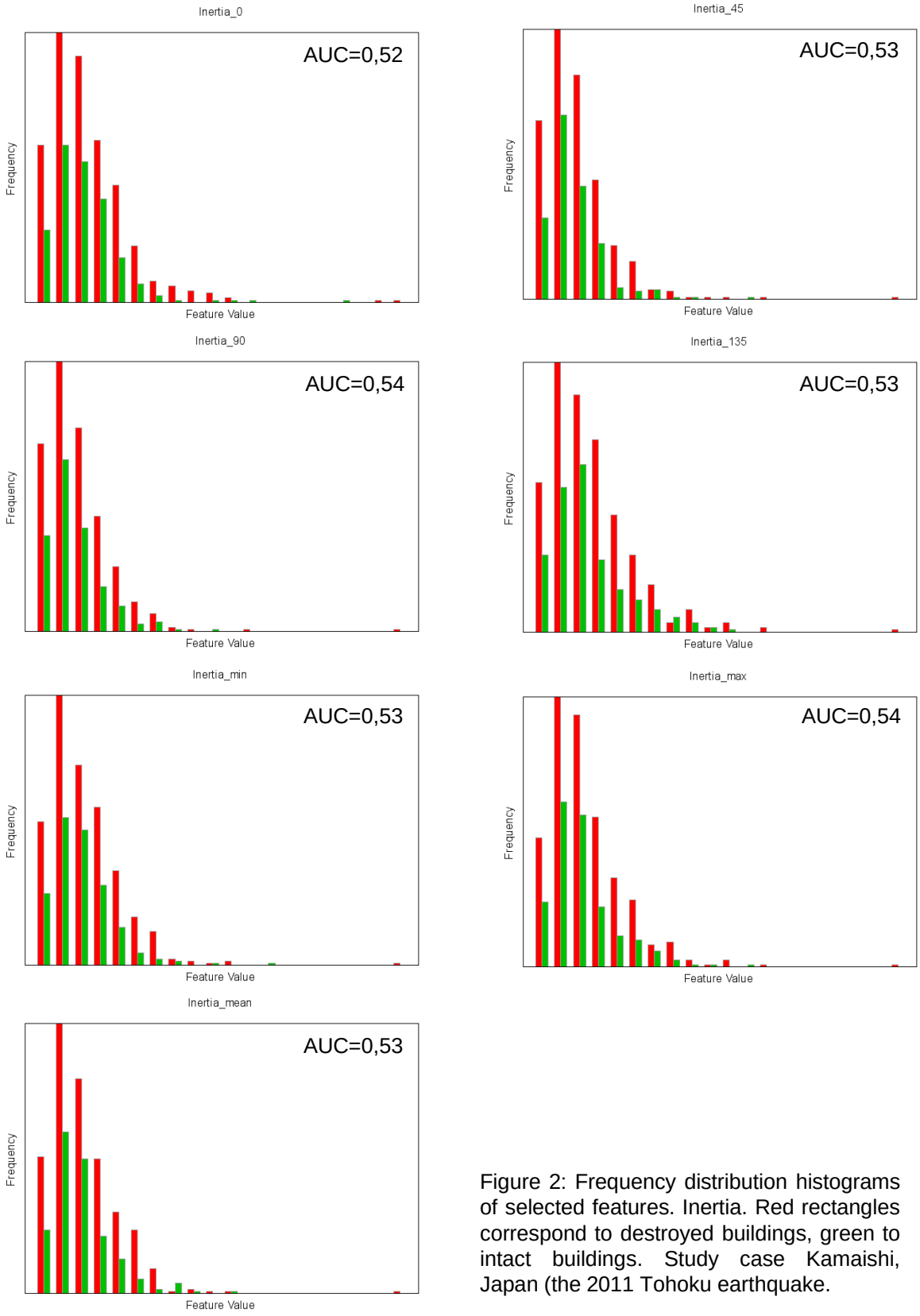


Figure 2: Frequency distribution histograms of selected features. Inertia. Red rectangles correspond to destroyed buildings, green to intact buildings. Study case Kamaishi, Japan (the 2011 Tohoku earthquake.

Appendix 5.1

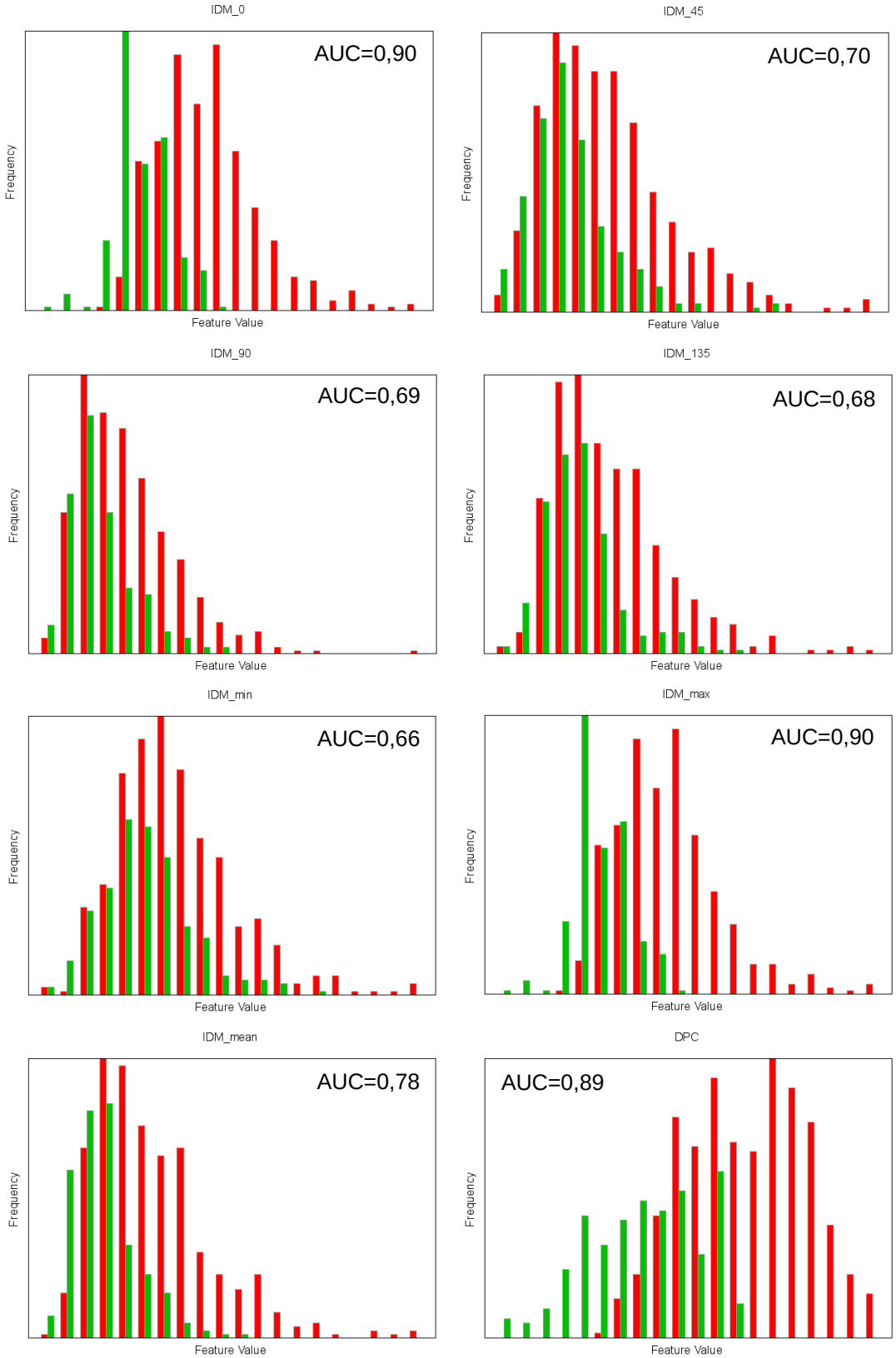


Figure 3: Frequency distribution histograms of selected features. Inverse Difference Moment (IDM) and Detected Part of Contour (DPC). Red rectangles correspond to destroyed buildings, green to intact buildings. Study case Kamaishi, Japan (the 2011 Tohoku earthquake.