Alexander Reiterer
Uwe Egly (Eds.)

Application of Artificial Intelligence in Engineering Geodesy

First International Workshop (AIEG 2008)
Vienna, Austria, December 2008
Proceedings

First International Workshop on

Application of Artificial Intelligence in Engineering Geodesy (AIEG 2008)

Vienna (Austria) – December 2008

Editors:

Alexander Reiterer
Research Group of Engineering Geodesy
Institute of Geodesy and Geophysics
Vienna University of Technology (Austria)

Uwe Egly
Knowledge Based Systems Group
Institute of Information Systems
Vienna University of Technology (Austria)

First International Workshop on
Application of Artificial Intelligence in
Engineering Geodesy (AIEG 2008)
ISBN: 978-3-9501492-4-1

Local Organizing Committee:
Alexander Reiterer, Uwe Egly, Martin Lehmann, Tanja Vicovac

The Workshop has been supported by:

- when it has to be right Leica Geosystems

FWF Der Wissenschaftsfonds.

Preface

In the last years, Artificial Intelligence (AI) has become an essential technique for solving complex problems in Engineering Geodesy. AI is an extremely broad field – the topics rage from the understanding of the nature of intelligence to the understanding of knowledge representation and deduction processes, eventually resulting in the construction of computer programs which act intelligently. Especially the latter topic plays a central role in applications.

In 2008, the Working Group 4.2.3 of the IAG Sub-Commission 4.2 ("Applications of Geodesy in Engineering") was reorganized and extended from "Application of Knowledge-Based Systems" to "Application of Artificial Intelligence". The reason behind this restructuring was to open the working group to researchers working on all sorts of problems concerning AI-techniques and engineering geodesy. Current applications using AI methodologies in engineering geodesy are geodetic data analysis, deformation analysis, navigation, deformation network adjustment, optimization of complex measurement procedure and others. Simultaneously with the restructuring of the working group, we decided to initiate a first workshop to bring together researcher from different research fields, mainly from informatics and geodesy. „The Workshop on Application of Artificial Intelligence in Engineering Geodesy – AIEG" will give an overview of the state of the art and recent developments in AI application in engineering geodesy. The aim of AIEG is to bring together the members of the Working Group 4.2.3 in order to share and to transfer experience for the development of applications of artificial intelligence in engineering geodesy.

The workshop on AIEG was held in December 2008 in Vienna and was a platform for around twenty experts from informatics and geodesy for presentation, discussion and networking.

Vienna, December 2008

Alexander Reiterer
Uwe Egly

Table of Contents

Application of Artificial Intelligence in Personal Navigation

Dorota A. Grejner-Brzezinska

Satellite Positioning and Inertial Navigation (SPIN) Laboratory
The Ohio State University, USA
email: dbrzezinska@osu.edu

Keywords: Artificial Intelligence, Personal Navigation, Global Positioning System, Wireless Local Area Networks.

The ability to determine one's position in absolute or map-referenced terms, relative to objects in the environment, and to navigate to a desired destination is an everyday necessity. Recent years brought up an explosion in the development of portable devices that support this functionality as well as provide a basis for the development of the Location Based Services (LBS). Most of the time, a Personal Navigation Device (PND) which combines the positioning and navigation capabilities, usually provided by the Global Positioning System (GPS), is used.

Pedestrian and personal navigation systems require continuous positioning and tracking of a mobile user with a certain positioning accuracy and reliability. However, navigating in urban and other GPS-impeded environments is a very challenging task. Thus, personal navigation systems require multiple navigation technologies to be integrated together, in order to serve as many different environments as possible for seamless and reliable navigation. Example technologies suitable for multisensor solutions supporting personal navigation include (aside from GPS/GNSS) ground-based RF systems, such as pseudolites suitable for confined and indoor environs, as well as cellular phone positioning for absolute position determination, dead reckoning sensors (e.g., gyroscopes, accelerometers, barometers or magnetic compass) to determine orientation, distance traveled and height. For location determination of a pedestrian in multi-storey buildings the Wireless Local Area Networks (WLAN), transponders or beacons installed in the buildings are increasingly used. Other indoor positioning systems include so-called Active Badge Systems (ABS). These methods can provide few-meter accuracy for indoor tracking and positioning. Robustness of the ultra wideband (UWB) signal to multipath fading and its high penetration capability makes it another technique suitable for indoor positioning. Other methods considered in indoor navigation are based on optical tracking systems, also referred to as image-based systems, and laser ranging systems, which provide range measurements to active or passive targets.

Recent years brought new developments in Artificial Intelligence (AI) techniques leading to an exponential increase in the number of applications in numerous areas, such as engineering, social and biomedical. AI is the study and design of intelligent agents, where an intelligent agent is a system that perceives its environment and takes actions which maximize its chances of success. In particular, AI techniques are very suitable in applications related to human motion modeling, and are being increasingly used for this purpose, due mainly to the complexity of the biological systems as well as the limitations of the existing quantitative techniques in modeling. Using AI methods allows for better process control and more reliable prediction/modeling of the processes under consideration. Examples of algorithms and methods used in AI are Artificial Neural Networks (ANNs) and Fuzzy Logic (FL). Both methods are suitable for modeling human locomotion dynamics that can be considered as a source of navigation information, where step length (SL) and step direction (SD) are the primary measurements that support navigation in a dead reckoning (DR) mode. The human dynamics model is calibrated while other sensors, primarily GPS, provide continuous navigation solution, and the human-based sensors are used in situation where other sensors cease to operate.

This presentation provides a review of the navigation techniques suitable for personal navigation, including some AI methods, and follows with the design, implementation and performance assessment of a system prototype, with a special emphasis on DR navigation supported by the human locomotion model handled by the AI techniques. An adaptive knowledge system (KBS) based ANN and FL has been implemented at The Ohio State University Satellite Positioning and Inertial Navigation (SPIN) Laboratory as an integral component of a multisensor personal navigator. The current target accuracy of the system is 3-5 m CEP (circular error probable, 50%). System design, as well as a summary of the performance analysis in the mixed indoor-outdoor environments, with the special emphasis on the DR performance, are provided.

Fuzzy Technology and ANN for Analysis of Deformation processes

Katja Heine

Chair of Land Surveying
Brandenburg University of Technology at Cottbus
email: Katja.Heine@tu-cottbus.de

Abstract

The paper presents methods of experimental deformation analysis. Artificial Neural Networks (ANN) can describe process connections without assumptions about the type of connections. By means of training process the so called synaptic weights of the connections between the processing unit (neurons) can be adapted so that at the end of training all the weights guarantee an optimal mapping of the connection between input- and output data.

Fuzzy techniques also are able to describe non-linear and complex connections without assumptions about their general character. In addition, they are suitable for processing data with non-probabilistic uncertainty. The connections can be described by so-called fuzzy rules, which are linguistic expressions similar to those of human language. In contrast, Artificial Neural Networks are pure black-box models.

Keywords: Dynamic Deformation Analysis, Fuzzy Rules, Artificial Neural Networks (ANN).

1 Description of Dynamic Deformation Processes

Modelling of deformation processes demands the consideration not only of the geometric changes in the object being monitored but also of the physical influences on the object. Beyond that, it has to be considered that the monitored object has a memory. Hence the deformation process is dynamic. That means that changes of input quantities release a time-dependent process of system-adaptation and thus a time-delay of the deformation signal.

In general a dynamic deformation process can be described by

$$\mathbf{y}(k\Delta t) = f(\mathbf{x}(k\Delta t), \mathbf{x}([k-1]\Delta t), ..., \mathbf{x}([k-m]\Delta t), \mathbf{z}(k\Delta t)) \tag{1}$$

where:

$\mathbf{z}(k\Delta t)$ values of deformation quantities at the point of time $k\Delta t$

$\mathbf{x}(k\Delta t), \mathbf{x}([k-1]\Delta t), ..., \mathbf{x}([k-m]\Delta t)$ values of influence quantities at the point of time $k\Delta t$ and at previous points of time $(k-l)\Delta t$ with $l = 1...m$

$\mathbf{z}(k\Delta t)$ disturbance quantities

The relation (1) can be described in different ways, by parametric model or by non-parametric model. The first (parametric) can be described by a differential equation system, the second one (nonparametric) by a multiple regression function e.g. (Heunecke et al., 1998).

The description of a deformation process by a parametric model is the most comprehensive and precise method. However it presupposes that the natural laws and rules which determine the process and the physical structure of the system are sufficiently known.

In practice this condition often is not realized or is realized only in part, so that the process can be modelled only by a so-called experimental analysis method. This method allows for description of the process-relations by measuring the time-series of input- and output-data and adjusting from these the coefficients of a special modelling equation, for instance regression functions or weight-functions. For describing connections by input-output-models, normally assumptions about the general character – especially about the degree of the functional connection – have to be made. The search for a suitable formulation for the model can be very difficult, particularly in the case of a high degree of non-linearity and complexity of the process. Artificial Neural Networks are an alternative to classical "input-output" models.Between the parametric and non-parametric process identification the cause-effect relation of the deformation process can also be modelled by a (fuzzy) rule-base.

2 Modelling the deformation process by Artificial Neural Networks

2.1 Architecture and Characteristics of ANN

For modelling the process connection given in equation (1) by classical regression functions, assumptions about the order of non-linearity and the term of system memory have to be made. Artificial Neural Networks (ANN) are able to learn this kind of process connection from given examples of input and output data. ANN are networks that consist of independent processing units (neurons) and simulate the processing principles of biological networks like the human brain. ANN are characterized by a high computation rate and,

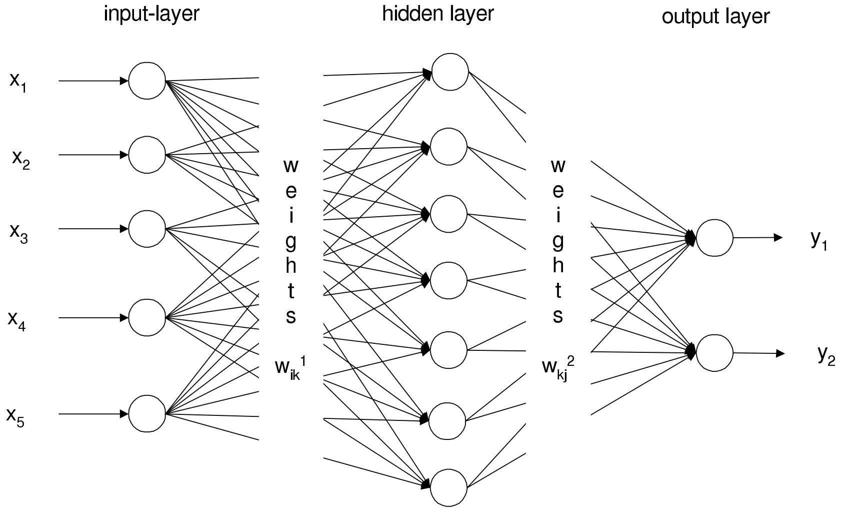


Figure 1: Multilayer-Feed-Forward Network (MLFF)

at the same time, a high degree of robustness and failure tolerance. ANN have the ability to generalize and to learn adaptively (Nelles, 2001).

Neurons are combined in layers and the neurons of one layer are normally connected with all neurons of the following layer by weighted links. There are several kinds of topology of ANN. The best-known and most commonly used are Multilayer-Feed-Forward Networks (fig. 1). Multilayer-feed-forward (MLFF) networks consist of an input-layer, one or more hidden layers and an output layer. Therefore, ANN can be described as a black-box, which maps an input space domain to an output space domain. The model or the mapping function is intrinsically defined by the adjusted weight matrix at the end of a training process, comparable with the parameters of a regression function.

A dynamic deformation model according to equation (1) can be represented by a MLFF network as follows. To the neurons in the input-layer the values of the influence quantities and their time-shifts were led. The output-layer contains only one neuron for the deformation. The principal structure of the network is shown in figure 2.

The principle mapping process from input-signals to output-signals is, briefly, the following: All neurons of a layer are linked according to their weights by an input-function

$$s_j = T(w_{1j} \cdot x_1, w_{2j} \cdot x_2, w_{3j} \cdot x_3, ..., w_{nj} \cdot x_n) \tag{2}$$

where T is an operator (i.e. maximum, minimum, sum, product).

The linear or non-linear activation function

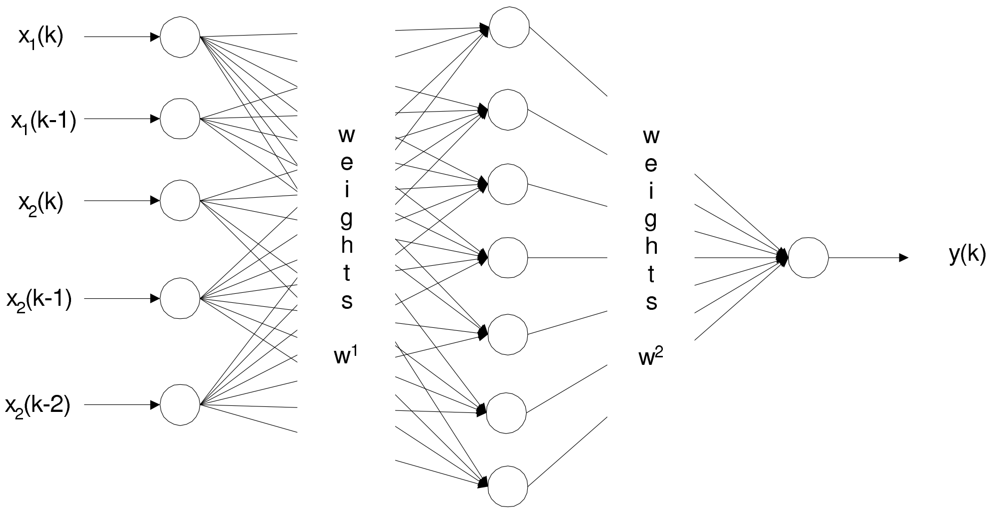


Figure 2: MLFF of two influence quantities with time-shift of one respectively two days in the input-layer and deformation quantity in the output-layer

$$y_j = f(s_j - w_0) \tag{3}$$

where w_0 is the so-called "threshold" determines the output of the corresponding neuron of the next layer. Hence the values of the output-neurons are a function g of the input-vector $\mathbf{x}^p$ and the weight matrix, which is non-linear if the activation-function is non-linear. Usually the activation function of the output-layer is linear and for the hidden layer is non-linear (e. g. Sigmoid-transfer function, tanh-function, Gaussian function). The process of determination of weights is called training process. A very popular algorithm for training MLFF networks is the Backpropagation-algorithm. The error calculated from the desired and the achieved network output is propagated backwards through the layers in order to update weights, when each input vector (pattern) is presented. The weight adjustment is proportional to the negative derivative of the cost function (often MSE) with respect to the weight. A detailed description of this algorithm is given in the standard literature, i.e. (Patterson, 1996).

2.2 Determination of an Optimal Network Structure

The number of input- and output-signals gives the number of neurons in the input- and output-layer. However, it is not a simple problem to find the ideal number of hidden layers and hidden neurons. There is no strict rule for defining this number, but some useful rules of thumb can be found in the literature that might be helpful for choosing optimal network architecture. As a rule for data and system analysis only one hidden layer is enough for the mapping task ((Otto, 1995), (Patterson, 1996)). >From the theorem of Kolmogorov it can be detected that, theoretically at least, MLFF networks with one hidden layer and $2n + 1$ hidden neurons can realize every mapping function from n-dimensional to m-dimensional space domain (Patterson, 1996).

Otto (Otto, 1995) suggests the following rule for the maximum number:

$$m = \frac{p}{5 \cdot (n + s)} \tag{4}$$

where:

p	number of values in the training-set
n	number of input-neurons
s	number of output-neurons

The number of hidden neurons is not so much a problem of computer time, but of mapping quality and generalization ability. If the hidden layer contains too few neurons, it leads to underfitting – the training and the generalization error and thus the statistical bias are high because the small number of neurons is not able to map the function sufficiently. The reverse effect is overfitting, where the network fits the noise, not just the signal, comparable with a polynomial with too many parameters. A sign of overfitting is that the training error is small, while the generalization error is high. To prove this, a data set generally has to be divided into a training set and a validation set, the latter of which is used only for validation of the network but not for training. In (Sarle, 1997) one can find valuable advice for finding the optimal network architecture. The best method is always to test several networks with different architectures and to choose the best one. Finally, the number of neurons should be as high as necessary, but as small as possible.

3 Fuzzy Rule Based Description

3.1 Uncertainty of deformation processes and Fuzzy Logic

The motivation for using fuzzy models for the analysis of deformation processes is that natural and even technical processes are influenced by a certain degree of uncertainty. It has to be emphasized that uncertainties of data and models of deformation processes are not only probabilistic. Non-probabilistic uncertainties occur mainly for the following reasons:

- It is not possible to include all causes of a deformation in the model, because some quantities are not measurable, not observable or even cannot be identified.

- Some causes cannot be represented adequately by numerical values (i.e. weight by traffic).

- The measured input quantities are not always identical with the acting forces. They only represent them as accurately as possible.

- The connections of the process can be very complex, so that it is very difficult or even not possible to describe them by explicit expressions and closed mathematical formulas.

Fuzzy set theory is the extension of classical set theory and was founded by L. Zadeh in 1965 (Zadeh, 1965). The basic idea is that the membership of a value in a set cannot assume only the two values "yes" or "no", but can be expressed by a gradual membership function within a range from zero to, normally, "1" in case of full membership degree. Membership function can assume several forms. In practice triangular or trapezium forms are often used. But membership function of Gaussian form is also quite possible if the uncertainty is purely random. Adequate to fuzzy set theory, classical bivalent logic was also extended to fuzzy logic. This logic enables calculation with fuzzy sets. It is the basis of fuzzy inference and thus of the so-called fuzzy rules. Fuzzy rules are implications $(A \Rightarrow B)$, which means that they determine which conclusion has to be drawn if certain conditions are realized:

If x_1 *is* A_1 *and* x_2 *is* A_2 *and ... then* y *is* B.

Corresponding to the membership of real facts or observations concerning this condition, the degree of certainty of the rule and the fuzzy value of the conclusion can be determined by special operators. The output-value for the whole rule base can be accumulated, for instance, by an "and"-operator.The theory of fuzzy set and fuzzy logic has been explained by a great deal of specialized literature, e.g. (Gottwald, 1993), (Zimmermann, 1996), (Seising, 1999), and therefore is not described in detail in this paper.

3.2 Rule-bases for Process Description

The cause-effect relation of the deformation process can be modelled by a rule-base. Rule-bases are well-known in connection with knowledge-based systems and indeed a deformation model can also be considered as knowledge based system of the deformation behaviour of an object or of a class of objects. An important advantage of knowledge-based systems is that knowledge or information can be represented not only by equations and algorithms as in classical information processing but also in unstructured forms like rules, frames and networks. Thus a model can be set up which is parametric, non-parametric or hybrid. But even if it is non-parametric, it is not a pure "black-box" model, because the physical process connections can be expressed by the rules.

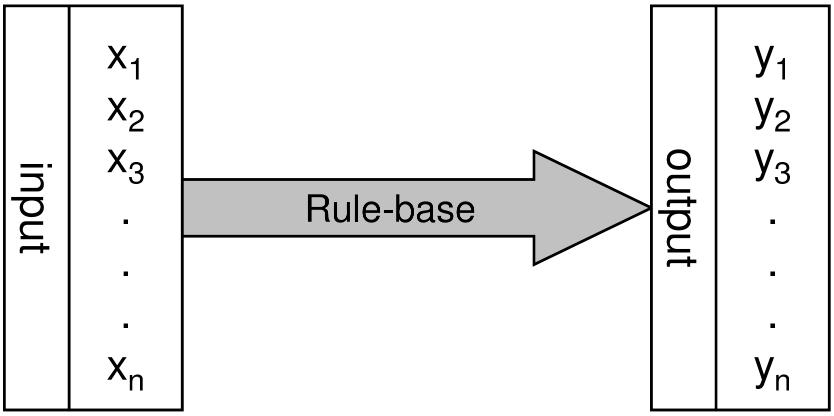


Figure 3: Description of a deformation process by a rule-base

Fuzzy rule based models consist of the above described two parts: fuzzy sets and fuzzy rules, which have to be set up.

This can be done either from knowledge and experience, from measured data or also from a hybrid base. The extraction of fuzzy rules from measured data can be realized by different methods:

1. Inductive machine learning algorithm: The so-called ID-3-algorithm by Quinlan (Quinlan, 1992) is an inductive learning algorithm that establishes a so-called tree of decisions. The criterion of this algorithm is conditional entropy. The ID3-algorithm will be described further on a practical example (see chapter 4).

2. Clustering: Clustering is an important method of finding common features and natural groupings of data from a large data set. Special versions of the classical clustering algorithm have been developed for fuzzy sets, i.e. Fuzzy C-Means (FCM) algorithm, Gustafson-Kessel-algorithm (Zimmermann, 1996), (Nelles, 2001).

3. Neuro-fuzzy-systems: There are several methods for setting up fuzzy sets and fuzzy rules by means of neural networks (see e.g. (Wang, 1994), (Nelles, 2001)).

4 Modelling deformation process of a creeping slope

4.1 Monitoring object

The examined slope is situated on a reservoir in the southern part of Thueringer Wald-mountains. The first explorations of the mass movement, which covers an area of 120.000 m^2, were carried out before the beginning of construction (1967-1975) of the reservoir, which presently contains 23.2 hm^3 of drinking water (Peters, 1997). The reason for this exploration is the fear of a sudden landslide on this slope that could threaten the dam stability. Therefore it was decided to conduct long-term-monitoring in addition to building a toe accretion for stabilizing the slope.

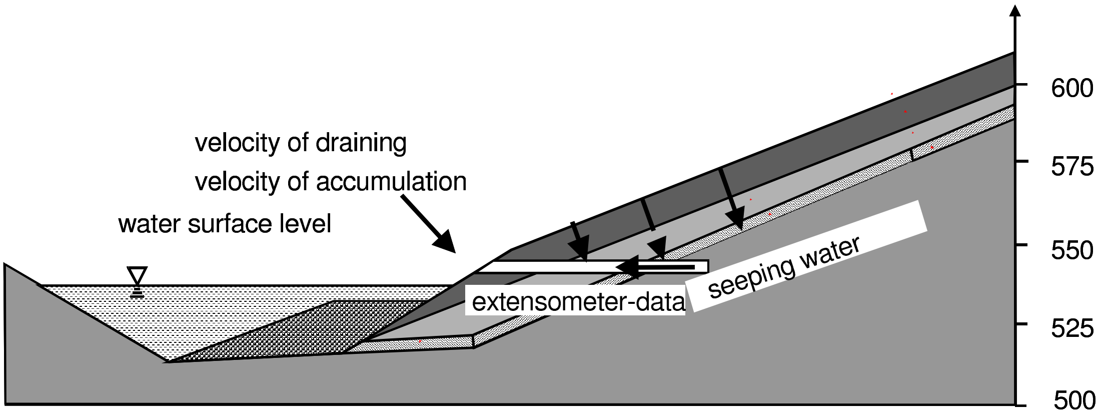


Figure 4: Monitoring object

The monitoring system included geological explorations, hydrometrical and meteorological measurements of the influence quantities (e.g. water surface level, precipitation) as well as geodetical and geophysical observations.

The main installation for monitoring is a gallery that cuts through the mass movement and reaches the stable rock. From this it is possible to measure displacement rates directly at the boundary between the sliding and the stable rock mass. The installed eight extensometers in the gallery at the base of vibrating-wire strain made it possible to continually monitor the mass movement. Furthermore the draining of seeping water, which is an indicator for the moisture content of the moving mass, was recorded in the gallery. For modelling the daily measured values of seeping water, water surface level and velocity of water surface level alteration were selected as influences and the daily recorded extensometer values were chosen as deformation.

The deformation behaviour of the creeping slope was modelled by classical Volterra-function, ANN and Fuzzy rule based models.

4.2 Modelling the slope's behaviour by Artificial Neural Networks

For the modelling, Multi-Layer-Feed-Forward-Networks with one hidden layer were used exclusively. The neurons in the input-layer correspond to the values of the influence quantities and their time-shifts, similar to the other modelling approaches shown in equation (1). It is not a simple problem of how long the considered time-shift has to be. Usually there is no accurate knowledge about the reaction time of the object to the influences. The problem was solved through consideration of the results of the classical model and by sensitive analysis supported by the used software.

The sensitive analysis and classical modelling had shown that eight neurons have to be arranged in the input-layer: three for seeping water, one for water surface level and four for velocity of water surface level alteration. With consideration to equation (4) and 616 values in the training-set it follows that the maximum number of neurons in the hidden layer is 14. It is known that too many hidden neurons may lead to overfitting, whereas too few may lead to the result that the respective relations cannot be mapped adequately. Thus several networks with different numbers of hidden neurons were tested (see table 2). For comparing the networks, the following was determined: the correlation coefficient between the time-series of the training-set (system) and the time-series calculated by ANN (model), the standard deviation between the two time-series and the standard deviation between time-series of a test-set and of a training-set. The test-set contains values which are not contained in the training-set. Thus a test-set is necessary to test the ability to generalization.

influence-quantity	time-shift [d]
seeping water	0,1,2
water surface level	2
velocity of accumulation	3,4
velocity of draining	6,7

Table 1: Input-quantities for ANN (Heine, 1999)

number of hidden neurons	6	10	14
correlation coefficient between system-output and model-output	0.985	0.985	0.985
standard deviation for the training-set [mm/d]*	±0.0027	±0.0027	±0.0026
standard deviation for the test-set [mm/d]	±0.0028	±0.0026	±0.0029

* in comparison: the values of system-output amount from 0.000 to 0.180 mm/d

Table 2: Comparison of networks with different number of hidden neurons (Heine, 1999)

Comparison shows that all networks are of nearly the same quality. The network with 14 neurons has a higher standard deviation for the test-set than the network with ten neurons; perhaps this is the first sign of overfitting.

All networks were calculated by Backpropagation. A similarly important problem as the number of hidden neurons is the rate of the period of training. If it is too long, then there is the risk of overfitting as well. For that reason, the training was broken off when the quality criteria shown in table 2 hadn't improved significantly from one pass to the next. The training should be broken off at the latest when the standard deviation of the test-set begins to increase.

The training was carried out in such a way that in the beginning a high learning rate and tolerance threshold were selected, in order to achieve a rapid learning improvement and then they were lowered to ensure the stability of the network and the precision of the resulting model. Due to the higher convergence of training-process and the better capability for generalization a random noise was added and the training values were presented in random order.

4.3 Fuzzy rule based model for the creeping slope

The fuzzy modelling of the deformation process was realized by Fuzzy-Opt (Dung et al., 1997), a commercial software-tool, into which the ID-3 algorithm is integrated.

4.3.1 Fuzzification of input- and output-data

Before the ID3-algorithm can be employed, the input- and output-quantities usually have to be fuzzified because most influence quantities and the deformation are available as numerical values and not as fuzzy sets. To accomplish

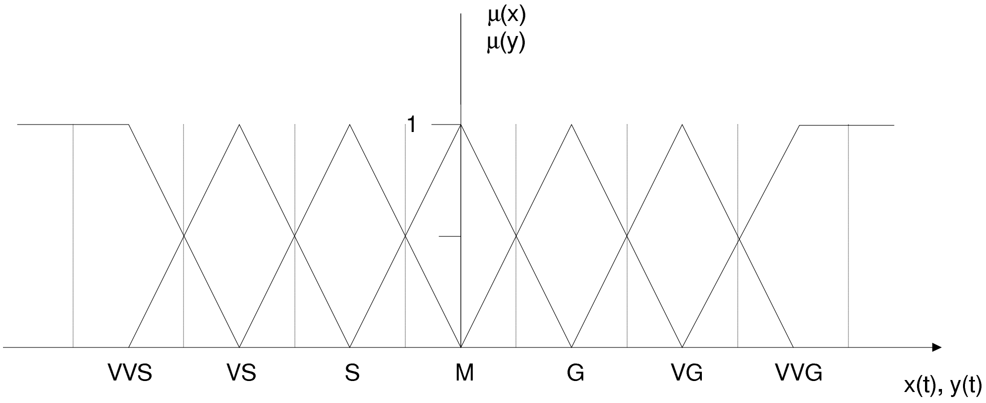


Figure 5: Fuzzification of input- and output-data

this, the whole range of every input- and output-quantity has to be divided into fuzzy terms, which are named, for instance, "very very small", "very small", "small", "middle", "great", "very great", "very very great" (see figure 3).

Usually the terms are represented by a triangular membership function or singletons. Every crisp value of the influence quantities and the deformation now has a membership degree of the fuzzy terms and the sum of these membership degrees always will be "1". The accuracy of the fuzzy rule based model depends mainly on the number of fuzzy terms into which input- and output-quantities are divided. The more terms used, the more accurate the model will be. But on the other hand, as the complexity of the model increases, interpretation becomes more difficult.

4.3.2 ID3-algorithm

The central problem of fuzzy rule based deformation models is the acquisition of fuzzy rules by a machine-learning algorithm from data sets of influence and deformation quantities. The ID3-algorithm developed by Quinlan (Quinlan, 1992) is an inductive algorithm because it realizes a conclusion from the special case (examples, data) to the general (If-Then-rules). The ID3-algorithm establishes a so-called "decision tree,"from which, in turn, a fuzzy rule base can be deduced. The nodes of the tree contain subsets of learning examples, which consist of values (for instance very small, small, middle, great, very great) of attributes (influence quantities including the time-shifts) and values of the decision (deformation). Table 3 contains an example of attribute values ignoring time-shifts of a dynamic model due to a better clarity.

water surface level (WSL)	Temperature	Deformation
S	S	M
M	S	M
G	M	G
...	...	...
...	...	...

Table 3: Examples for attributes

The tree is established in the following way:

As long as not all nodes are homogeneous – that means that all included examples have the same value for the influence quantities and for the deformation – the nodes are split into new groups of attributes. The actual node of attributes is split up into all groups of possible values.

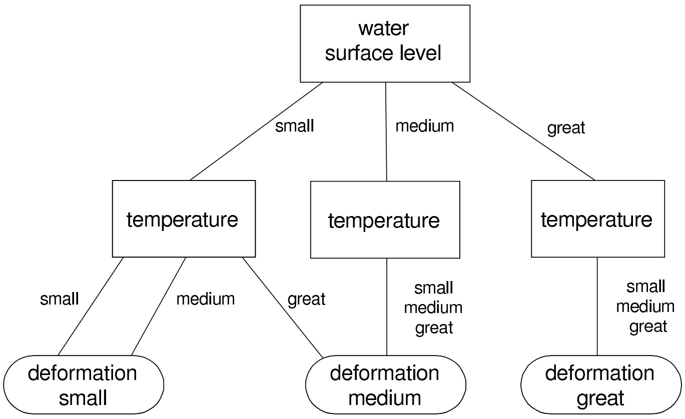


Figure 6: Example for decision tree

To get a complete and consistent tree the attributes have to be chosen in the right order. The criterion of choice is conditional entropy. Entropy

$$H(X) = -\sum_{i=1}^{m} p(X_i)\log_2 p(X_i) \tag{5}$$

is a measure of information content. The information with the highest possibility has the lowest entropy. From the conditional probability the conditional entropy can be deduced:

$$H(Y/x_j) = \sum_{i=1}^{N} P(C_i)H(Y/C_i) \tag{6}$$

where

$$H(Y/C_i) = -\sum_{j=1}^{n} P(Y_j/C_i)\log_2 P(Y_j/C_i) \tag{7}$$

where:

C_i	class with the same term (value) for an input-quantity
Y_j	term of deformation
n	number of terms of deformation
N	number of classes of an input-quantity
$P(C_j)$	Probability of occurrence of a class
$P(Y_j/C_i)$	conditional probability of a term of deformation within a special class of term of an input-quantity
$H(Y/C_i)$	entropy of deformation within a special class
$H(Y/x_j)$	entropy of deformation for a special input-quantity

In table 4 an example is given of data-sets and of the calculation of the conditional entropy. Under the condition that influence quantities have the value given in column 2, the numbers of data sets of deformation are listed in column 3.

Class	Influence quantity	Deformation			$P(Y_j/C_i)$			$H(Y/C_i)$	$P(C_i)$	$P(C_i)\cdot H(Y/C_i)$
		S	M	G						
1	WSL S	25	15		0,63	0,37		0,286	0,258	0,074
2	WSL M	5	48	2	0,09	0,87	0,04	0,203	0,354	0,071
3	WSL G		3	57		0,05	0,95	0,086	0,386	0,033
4	Temp. S	15	12	18	0,33	0,27	0,40	0,472	0,288	0,136
5	Temp. M	10	26	19	0,18	0,47	0,35	0,448	0,354	0,159
6	Temp. G	5	28	22	0,09	0,51	0,40	0,402	0,354	0,142

Entropy of water surface level (WSL): 0,074 + 0,071 + 0,033 = 0,178
Entropy of temperature: 0,136 + 0,159 + 0,071 = 0,437

Table 4: Examples for calculation of the conditional entrophy

The influence quantity with the lowest entropy– that means the most dominant one – is the first node of the decision tree. For the following nodes, new sub-classes are established and the entropy is calculated again until there is no more influence quantity left. From the example in table 4 the decision tree in figure 6 can be deduced. The rules can be deduced just from the decision tree, for instance, corresponding to figure 6:

If WSL = small and Temperature = small then Deformation = small
If WSL = small and Temperature = middle then Deformation = small
If WSL = small then Deformation = middle
If WSL = small then Deformation = great

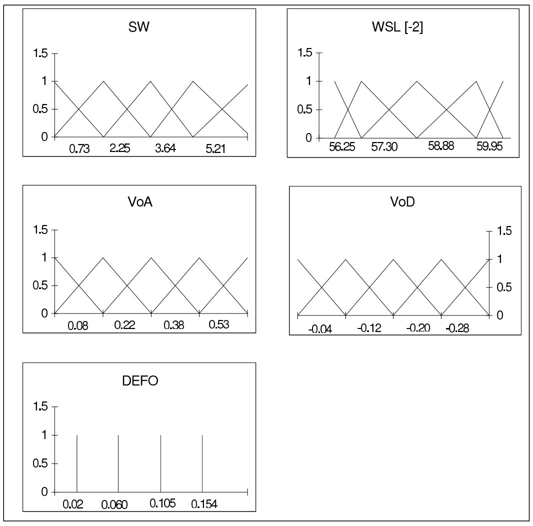
Figure 7 shows the optimised fuzzy sets of the input- and output-quantities and the rule base of the deformation process model.

The rule base (figure 7) is relatively complex. The elimination of less relevant input-quantities reduces the rule base, but on the other hand of course increases the standard deviation of model- and system-output. The elimination of non-relevant input quantities (velocity of accumulation and velocity of draining) does not have an influence on modelling accuracy as expected.

4.4 Results of process modelling

The modelling accuracy can be evaluated by the standard deviation of residue between system output (measured displacement rates) and model output (calculated deformation).

The standard deviation of ANN is quite a bit smaller than those of the classical model and the fuzzy model, where the same influence quantities



```
(1)  IF (SW_1 = G) AND (SW = M) THEN DEFO:=M ;
(2)  IF (SW_1 = G) AND (SW = G) THEN DEFO:=G ;
(3)  IF (SW_1 = VG) AND (SW_2 = M) THEN DEFO:=G ;
(4)  IF (SW_1 = VG) AND (SW_2 = VG) THEN DEFO:=VG ;
(5)  IF (SW_1 = M) AND (SW_2 = S) THEN DEFO:=S ;
(6)  IF (SW_1 = M) AND (SW_2 = M) THEN DEFO:=M ;
(7)  IF (SW_1 = M) AND (SW_2 = G) THEN DEFO:=M ;
(8)  IF (SW_1 = S) AND (SW = VS) AND (WSL_2 = VG) THEN DEFO:=VS;
(9)  IF (SW_1 = S) AND (SW = VS) AND (WSL_2 = G) THEN DEFO:=S ;
(10) IF (SW_1 = S) AND (SW = VS) AND (WSL_2 = M) THEN DEFO:=S ;
(11) IF (SW_1 = S) AND (SW = M) THEN DEFO:=S ;
(12) IF (SW_1 = S) AND (SW = S) THEN DEFO:=S ;
(13) IF (SW_1 = VS) AND (WSL_2 = VS) THEN DEFO:=VS ;
(14) IF (SW_1 = VS) AND (WSL_2 = S) THEN DEFO:=VS;
(15) IF (SW_1 = VS) AND (WSL_2 = M) AND (SW_2 = S) THEN DEFO:=S ;
(16) IF (SW_1 = VS) AND (WSL_2 = M) AND (SW_2 = VS) THEN DEFO:=VS ;
(17) IF (SW_1 = VS) AND (WSL_2 = G) AND (SW = S) THEN DEFO:=S ;
(18) IF (SW_1 = VS) AND (WSL_2 = G) AND (SW = VS) AND (SW_2 = VS) THEN DEFO:=VS ;
(19) IF (SW_1 = VS) AND (WSL_2 = G) AND (SW = VS) AND (SW_2 = S) THEN DEFO:=S ;
(20) IF (SW_1 = VS) AND (WSL_2 = VG) THEN DEFO:=VS ;
     END.
```

Figure 7: Fuzzy rule-base and optimised fuzzy sets (SW= seeping water, _1,_2=time-shift of 1 resp. 2 days; WSL=Water surface level; VoA=velocity of accumulation; VoD= velocity of draining; DEFO=deformation)

were considered. >From this it can be concluded that ANN-model is more precise than the other models.

Figure 8 represents system-output and model outputs of the ANN- and the fuzzy model. Nevertheless natural processes always include uncertainty and it is impossible, and often unnecessary too, that the model be an identical mapping of the process. Rather the model should be a generalization of the main relations.

	classical VOLTERRA-model	ANN	Fuzzy model
standard deviation a posteriori	±0.0038 mm/d	±0.0027 mm/d	±0.0035 mm/d

Table 5: Modelling accuracy

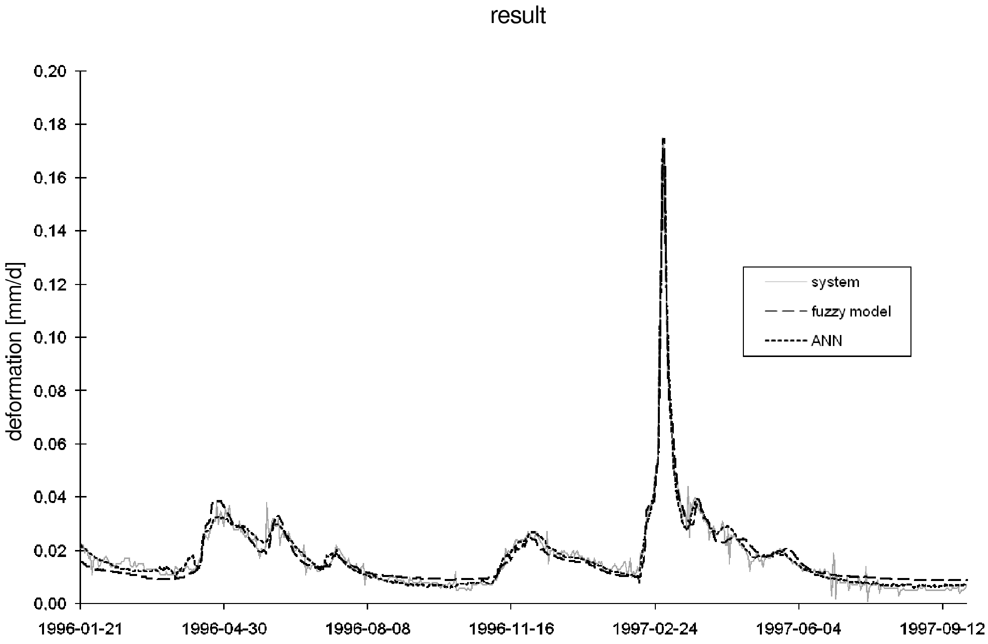


Figure 8: Comparison between system and model output

5 Conclusions

Although the modelling accuracy of ANN-model is higher than those of the fuzzy rule based model, it must not be concluded that ANN modelling on principle is the better method. The fuzzy rule base that can be interpreted by experts is a serious advantage in comparison with the pure black-box ANN. Thus the decision as to which method is to be preferred depends on the purpose of modelling. If a high modelling accuracy has to be reached, for instance, for prediction models, ANN can provide for the better result. If on the other hand process connections have to be detected or theoretical presumptions or calculations have to be evaluated, the fuzzy rule based model is rather the appropriate approach.

Both approaches – ANN and ID3-algorithm – are methods of learning from examples. Therefore an important condition for using this method has to be fulfilled: There must be enough examples available and the examples have to represent the whole system's behaviour; that means the input-quantities have to cover the whole possible input space domain.

References

Dung, L. T. and Otto, P.: Fuzzy-Modellbildung mit maschinell gelernten Regeln. Proceedings of Internationales Wissenschaftliches Kolloquium der TU Ilmenau 1997, Vol. 3, pp. 209-215. Ilmenau, 1997.

Dung, L. T., Koch, M., Otto, P.: FuzzyOpt - ein Werkzeug zum Entwurf optimaler Fuzzy Systeme. at-Automatisierungstechnik 45 (1997), pp. 555-556. München, 1997.

Gottwald, S.: Fuzzy Sets and Fuzzy Logic. Vieweg, Braunschweig, 1993.

Heine, K.: Beschreibung von Deformationsprozessen durch Volterra- und Fuzzy-Modelle sowie Neuronale Netze. Deutsche Geodätische Kommission, Reihe C, Vol. 516. Munich, 1999.

Heine, K.: Deformation Analysis by Fuzzy Rule Based Models. In: Kahmen, H., Niemeier, W., Retscher, G.: Symposium on Geodesy for Geotechnical and Structural Engineering II at Berlin 2002, pp. 177-185. Wien, 2002.

Herrmann, J.: Maschinelles Lernen und Wissensbasierte Systeme. Springer, Berlin, 1997.

Heunecke, O., Pelzer, H., Welsch, W.: On the Classification of Deformation Models and Identification Methods in Engineering Surveying. Proceedings of XXI. International Congress, Developing the Profession in a Developing World, Com. 6 Engineering Surveys, pp 230-245. Brighton, 1998.

Nelles, O.: Nonlinear System Identification. Springer, Berlin, 2001.

Niemeier, W., Miima, J.: A Neural Network Approach for Modelling Geodetic Deformations. In: Carosio, A., Kutterer, H.(ed.): First International Symposium on Robust Statistics and Fuzzy Techniques in Geodesy and GIS, pp 111-116. Zurich, 2001.

Otto, P.: Identifikation nichtlinearer Systeme mit Künstlichen Neuronalen Netzen. at – Automatisierungstechnik 43 (1995) 2, pp. 62-68. R. Oldenbourg Verlag, München, 1995.

Patterson, D. W.: Artificial neural networks. Prentice Hall, Singapore, 1996.

Peters, J.: Überwachung bewegungsaktiver Stauraumhänge an der Talsperre Schönbrunn. DVWK-Seminar, Heyda, 1997.

Quinlan, J. R.: C 4.5. Programs for Machine Learning. San Mateo, California, 1992.

Rojas, R.: Neural Networks. Springer, Berlin, 1996.

Sarle, W. S. (ed.): Neural Networks FAQ. URL: ftp://ftp.sas.com/pub/neural/FAQ.html. 1997

Seising, R.: Fuzzy Theorie und Stochastik. Vieweg, Braunschweig, 1999.

Wang L.-X.: Adaptive fuzzy systems and control. Prentice Hall, Englewood Cliffs, 1994.

Zadeh L. A.: Fuzzy sets. In: Information and Control 8 (1965), pp. 338-353. 1965.

Zimmermann H. J.: Fuzzy Set Theory - and its Applications. Kluwer, Boston, 1996.

Application of Fuzzy Clustering in Deformation Analysis

Rainer Fletling

Department of Engineering Surveying
Faculty of Civil Engineering
University Kassel
email: fletling@uni-kassel.de

Abstract

An approach adopted from the field of pattern recognition is used to tackle two typical problems occuring in the area of geodetic deformation measurements. Automatic clustering methods are used that, based on formal criteria, classify data into homogeneous classes. Distinctive to the work at hand is the fuzzy clustering method employed that does not assign a data point to a single unique class but to all classes, computing gradually varying degrees of membership of the point to each of the classes. The first example problem that is given deals with the identification of point groups with similar movement patterns in a two-dimensional control network. The second example treats the automatic classification of settlement time series.

Keywords: Fuzzy, Clustering, Deformation Analysis.

1 Introduction

When analysing deformation measurements, the following question usually arise: are there significant point movements?

If yes,

1. Which points have moved?
2. In which way have these points moved?
3. Can particular movement patterns be identified that describe the movement of individual points or groups of points?

A conventional way of answering these questions is by means of statistical tests working with a two-valued logic (i.e. there are only two possible truth values, "true" and "false"). One problem of this approach is that a point having been displaced by a value slightly below a significance threshold is classified as "stable", while a point with a displacement slightly above this threshold is regarded as "has moved". Of course, the actual displacements in this case differ only marginally. Similar problems arise when groups of points following similar movement patterns shall be found, or if the movement of an individual point or a number of points should be assigned to a particular movement pattern. In such cases, well-defined boundaries between patterns are hard to establish.

An alternative to the two-valued logic is fuzzy logic. Here, elements are not assigned to only one particular set but to all sets, allowing for gradual assessment of an element's membership to a certain set by a so-called degree of membership value (permitting values between 0 and 1, in contrast to two-valued logic, which permits only 0 and 1). Fuzzy logic beares its name from this "fuzzy" assignment of elements to sets. Viewing a problem with fuzzy logic corresponds much better to the real world and the human notion thereof. For example, we cannot only see the two colours black and white, but also arbitrarily many intermediate shades of grey.

The work at hand introduces an approach to solving the above problems that is also widely used in the pattern recognition field.

Clustering methods belong to the group of classification methods. Such methods classify data into homogeneous groups or classes by using formal criteria and procedures. Here, data within a particular class should be as similar as possible with respect to the issue of interest, while data in different classes should be distinct from each other. For pattern recognition problems, clustering methods are used, for example, to find unknown structures in large data sets, or to assign new data to known structures. Both will be demonstrated in examples below.

Objects (represented by data) to be classified usually possess a number of distinct properties. Those properties that are of particular importance for the issue of interest are called features. From the mathematical viewpoint, these features span a feature space, in which actual classification takes place.

As mentioned above, objects within a particular class should be similar to each other, objects in distinct classes should differ. The similarity of the objects is thus crucial for their class membership and has to be computed by a suitable distance function. Since most features are real-valued (or can be transformed into such), the Eucleadian metric is normaly used.

The clustering method then estimates the centre of each cluster (i.e. the position of a class in feature space) as well as its shape (i.e. class boundaries or lines of similar class membership). Depending on the application these parameters can be prescribed or be determined by the method:

1. If the structures in the data are completely unknown, the parameters can be automatically estimated as unknowns by the classification process.

2. If repeating classification tasks of similar kind are considered, the parameters usually are estimated in advance based on a representative random sample of the data. This is often called the "learning" or "training" (Duda et al, 2000) of the classifier. The parameters are then held constant in the subsequent classification process.

3. If expert knowledge about the position and shape of the expected classes is available at the beginning of the classification, the paramters can be directly prescribed.

"Hard" clustering methods assign an object to exactly one particular class. However, such a strict classification is not appropriate in many cases. Consider, for instance, a situation in which only uncertain or imprecise knowledge about the objects is available. Also, objects that are located in the immediate surroundings of a class boundary can often be assigned similarly well to both classes on either side of the boundary. Fuzzy clustering methods assign an object not to exactly one cluster but compute gradual degree of membership values for each class. Objects whose features exhibit a high similarity with the features of a particular cluster centre yield a high degree of membership for this cluster, while objects with a low similarity correspondingly are assigned a low value. If c clusters are present at the end of the clustering process, the clustering result for every object will be a vector of c degrees of membership, as opposed to a unique assignment.

2 Fuzzy-Clustering

The most well-known algorithm for the fuzzy clustering of data is the Fuzzy C-Means algorithm (FCM) (Bezdek, 1973). It has been well-proven as being a robust and stable clustering method in a great number of applications. In the FCM, a cost function J_{FCM} is minimized under consideration of the constraint $\sum_{i=1}^{c} \mu_{i,k} = 1$, where $\mu_{i,k} \in [0, 1]$.

$$J_{FCM} = \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{i,k})^{w} \cdot d_{i,k}^{2}(\boldsymbol{\nu}_i, \boldsymbol{x}_k) \tag{1}$$

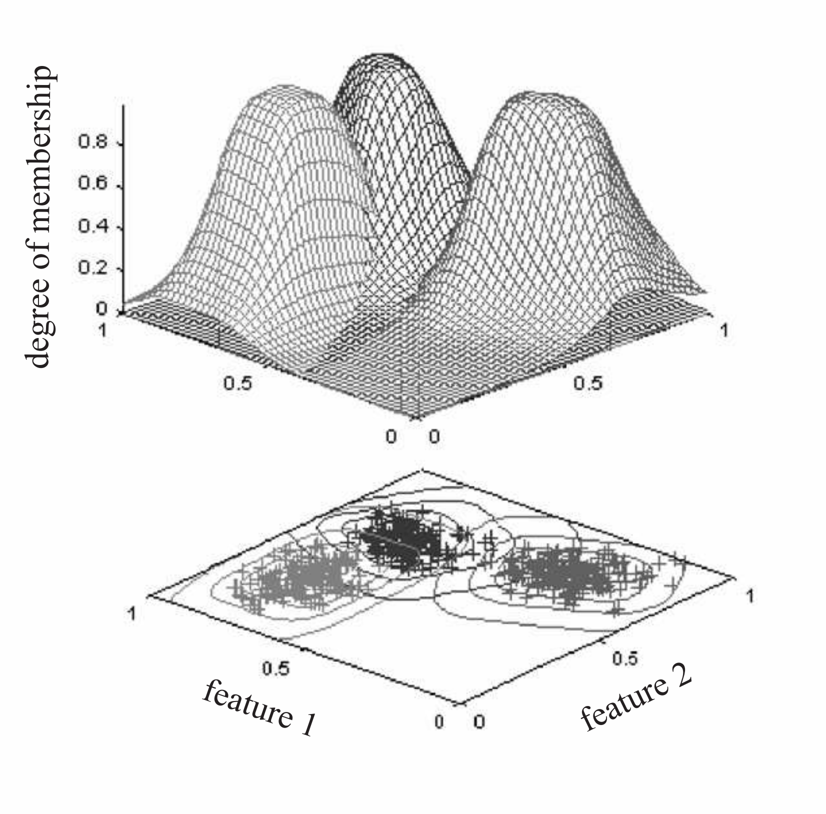


Figure 1: Example of membership functions of three fuzzy classes in a two-dimensional feature space. (Burmeister, 1997).

c	:	number of clusters
n	:	number of objects (data-points)
$\mu_{i,k}$	:	degree of membership of the k th object to the i th class
w	:	weighting exponent
$d_{i,k}$	:	(Eukleadian) distance
$\boldsymbol{\nu}_i$	:	data vector of the i th clustercenter
$\boldsymbol{x}_k$	:	data vector of the k th object

The weighting exponent w determines the degree of fuzziness of the clustering prozess. For $w = 1$ one obtaines a unique assignment. If w becomes larger, the results become more fuzzy. In practice, $w = 2$ is usually chosen.

Unfortunately, there is no analytical solution to the optimisation problem. After specification of c, w and initial values for either the clustercenter points or the degrees of membership, the cost function is minimized iteratively. $\mu_{i,k}$ and $\boldsymbol{\nu}_i$ are alternately re-computed until the change in $\boldsymbol{\nu}_i$ between two iterations falls below a given threshold.

$$\mu_{i,k} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d^2_{i,k}(\boldsymbol{\nu}_i, \boldsymbol{x}_k)}{d^2_{j,k}(\boldsymbol{\nu}_j, \boldsymbol{x}_k)} \right)^{\frac{1}{w-1}}} \,, \quad \forall\, i = 1, \ldots, c\,; \;\; \forall\, k = 1, \ldots, n \tag{2}$$

$$\boldsymbol{\nu}_i^{new} = \frac{\sum_{k=1}^{n}(\mu_{i,k})^w \cdot \boldsymbol{x}_k}{\sum_{k=1}^{n}(\mu_{i,k})^w} \,, \quad \forall\, i = 1, \ldots, c \tag{3}$$

The largest disadvantage of this algorithm is that when using the Eucleadian distance it forms equally large clusters shaped as hyperspheres, a behaviour independent of the actual distribution of the objects in the feature space. Höppner et al (1997) writes that the FCM reaches its limits for clusters of distinct shape, size and density. From the geodetic viewpoint, another disadvantage is that the FCM considers the object's features as being error-free. It hence cannot consider the fuzziness of the objects to be clustered.

In order to compensate for the FCM disadvantage of forming hyperspherical clusters several modifications have been developed over the time. The most quoted examples of these are the Gustafson-Kessel algorithm (Gustafson and Kessel, 1979) as well as the Gath-Geva algorithm (Gath and Geva, 1989). The iterative approach of alternatingly optimising the degrees of membership $\mu_{i,k}$ and the cluster centers $\boldsymbol{\nu}_i$ is kept in both algorithms. However, in contrast to the FCM, different distance functions are employed. As a result, the Gustafson-Kessel algorithm forms hyperelliptical clusters in the feature space, that, however, are still constant in size. When applying the Gath-Geva algorithm, these clusters can also vary in size. A disadvantage of the modified algorithms is their sensitivity to prescribed initial values for the cluster centres, which in practise should be very close to the final values.

3 Identifying point groups with similar movement patterns.

In this section, the identification of point groups with similar movement patterns shall be demonstrated by using the two-dimensional test network Delft as an example.

The test network Delft (figure 2) consists of a two-dimensional control network with simulated measurements of several observational epochs that were published within the context of the seminar "Deformationsanalysen '83" (Welsch, 1983). The data were used as a basis for a comparitive investigation of several analysis strategies. Here, we will have a closer look at changes between the epochs 1 and 3b.

Methods for the analysis of measurements from geodetic surveys that are used in the geodetic practice usually lead to the specification of displacement vectors and a test for significance. They are often combined with a graphical representation of the object points and their displacement vectors. The interpretation with respect to identifying groups of similar displacement vectors is then left to the user, who can try to identify movement patterns by

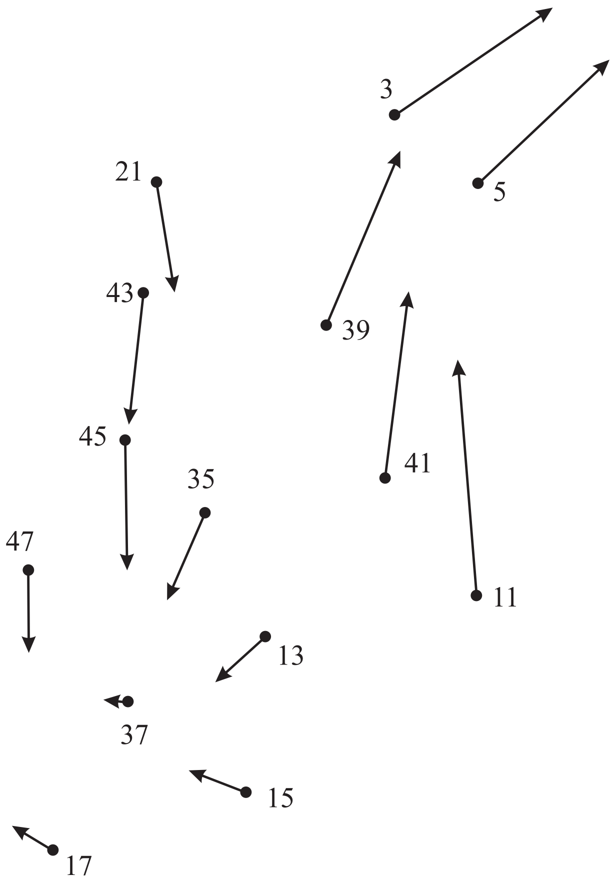


Figure 2: Test network Delft: Object points with their displacement vectors between the epochs 1 and 3b.

visual inspection. While this approach does not pose a significant problem for trained staff and problems with a reasonable amount of data, the need for an automated interpretation of the displacement vectors is quickly increased for problems with a large number of object points such as, for instance, the surveillance of large mountain slides or open-cut mining. Such monitoring systems with automatic alarm should be able to function without human staff. It is hence necessary to incorporate the knowledge of a trained expert into the system. There are several possibilities to achieve this task. A particular approach making use of fuzzy rules was introduced by Haberler (2005), however, the approach differs fundamentally from the procedure presented here.

3.1 Clustering of object points

The length and the azimuth of the displacement vectors are those features that a human observer would use to infer groups of points with a similar movement behaviour in a two-dimensional control network. These features are also used as input variables for the approach presented here. In order to prevent the point of discontinuity at 0 [gon] in the azimuth of the displacement vectors from running directly through cluster 1 in the example, the network was rotated by 10 [gon]. [1].

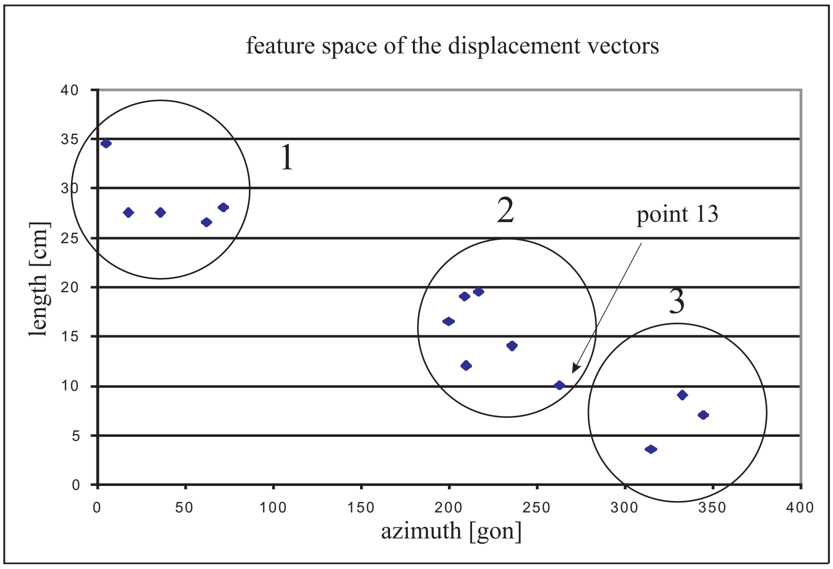


Figure 3: Test network Delft (rotated): Displacement vectors between the epochs 1 and 3b in the feature space and the approximately position of the three clusters.

In order to obtain meaningful results from the clustering process, features on different axes should be of the same order of magnitude. This can be achieved by normalising the input data to lie in the range $]0, 1[$.

The number of clusters to which the FCM should determine the degrees of membership of the individual object points has to be specified in advance. The "right" number is usually unknown, hence it has to be determined during the analysis process as well. This is done by the following procedure: The algorithm is run with different numbers of clusters, afterwards the results are evaluated based on several criteria, a list of which is given by Windham (1981) but will not be detailed here. For the example in this paper, an optimal number of clusters was found to be three.

Table 1 lists the results of the clustering using the FCM algorithm. Relatively high degrees of membership were found for almost all object points, indicating a good classification of the objects into the corresponding clusters. Only point 13, having a maximum degree of membership value of 0.57 and a second-largest value of 0.40, would be difficult to classify uniquely. The point, belonging to cluster 2, is specially marked in Figure 3. It can be seen that it is also located close to cluster 3.

[1] By introducing the sine and the cosine of the azimuth as features, one can avoid rotating the network. However, for the example at hand this would have lead to a less intuitive three-dimensional feature space.

	cluster		
point	1	2	3
3	**0.97**	0.02	0.01
5	**0.97**	0.02	0.01
11	**0.91**	0.06	0.03
13	0.03	**0.57**	0.40
15	0.01	0.06	**0.93**
17	0.01	0.01	**0.98**
21	0.02	**0.96**	0.02
35	0.01	**0.97**	0.02
37	0.01	0.04	**0.95**
39	**1.00**	0.00	0.00
41	**0.99**	0.01	0.00
43	0.05	**0.89**	0.06
45	0.05	**0.90**	0.05
47	0.02	**0.92**	0.06

Table 1: Computed degrees of membership of the object points to the three clusters.

3.2 Consideration of neighborhood

The analysis process now shows three clusters into which the objects points can be well classified. These three clusters nicely represent the three blocks of points that exhibit a similar movement pattern. However, so far the proximity of the points to each other has been neglected in the analysis procedure. Theoretically, two points that, judged from their maximal degree of membership values, belong to the same cluster, could be located at spatially seperated boundaries of the network. In between, a group of points with a very different movement pattern could be situated. It is thus necessary to account for the proximity relations of the points. This is achieved by generating a triangular irregular network (TIN) between the object points using a Delauny triangulation. The resulting line connections between the object points are considered in the further analysis step.

Each line is delimited by two end points. Due to the preceeding clustering of the object points each of these end points possesses degrees of membership to the corresponding clusters. By means of the arithmetic mean, a degree of membership can also be computed for the line. If both end points exhibit maximal membership values to a particular cluster, the line will also be assigned a high degree of membership to this cluster. If the maximal membership values of the end points belong to different clusters the line will get a relatively low value, indicating an intersection of the line with a class boundary. This is the advantage of the fuzzy approach, since the degree-of-membership-based fuzzy classification of an object point into a cluster can be carried over to the second analysis step without having to make a preliminary decision due to an enforced unique classification. Always think of those points that would lie in close proximity to a sharp class boundary.

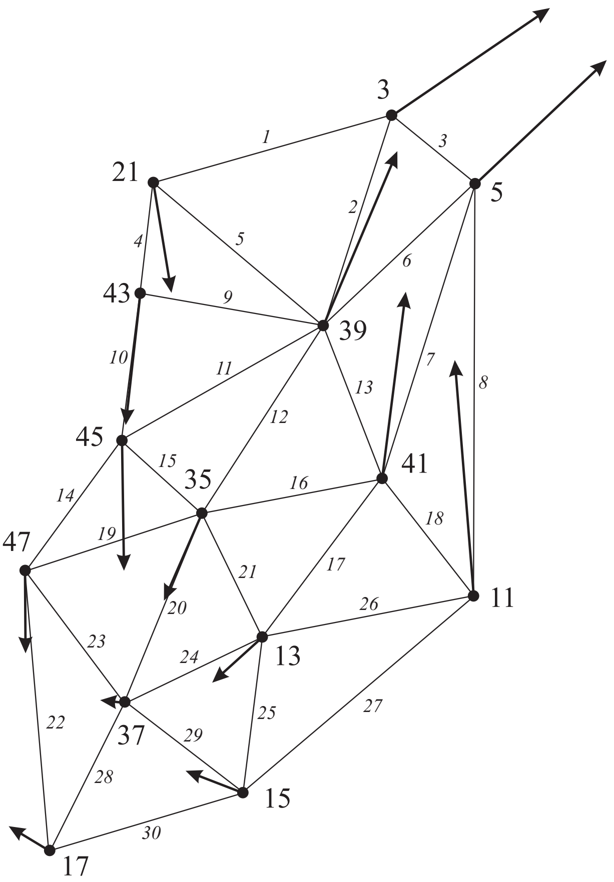


Figure 4: Test network Delft: Object points with their displacement vectors between the epochs 1 and 3b connected by a triangular irregular network (TIN). The small numbers show the numbers of the lines.

Table 2 lists the lines with their corresponding end points and average membership values. In the last column the ratio q between the largest and second largest degree of membership is given as a quality criterion. Large values of q indicate good classifications into a cluster, small values uncertain classifications. Here, the empirically determined value of $q = 4$ was chosen as the smallest value for a reliable classification. In table 2, lines with $q > 4$ are highlighted with bold font. All lines exhibiting values of q of less than four are assumed to intersect with a class boundary. Figure 5 also highlights lines with $q > 4$ as bold, whereas all lines intersecting a cluster boundary are plotted as thin lines. In addition the found boundaries between blocks of points with similar point movements are plotted in bold dashed lines.

In this example three groups of consistent point movements can be distinguished. If the boundary for the quality criterion q would be slightly increased, line 21 would also be classified as unreliable and point 13 would become the sole representative of a fourth group.

For the example at hand, analyses that were carried out with both the modified fuzzy clustering procedures after Gustafson-Kessel and Gath-Geva lead to identical results for the cluster boundaries.

			cluster			
line	point1	point 2	1	2	3	q
1	3	21	0.49	0.49	0.02	1
2	3	39	**0.98**	0.02	0.00	**76**
3	3	5	**0.97**	0.02	0.01	**44**
4	21	43	0.04	**0.92**	0.04	**21**
5	21	39	0.51	0.48	0.01	1
6	5	39	**0.99**	0.01	0.00	**99**
7	5	41	**0.98**	0.02	0.00	**76**
8	5	11	**0.94**	0.04	0.02	**24**
9	39	43	0.53	0.44	0.03	1
10	43	45	0.05	**0.89**	0.06	**16**
11	39	45	0.52	0.45	0.03	1
12	35	39	0.50	0.49	0.01	1
13	39	41	**0.99**	0.01	0.00	**199**
14	45	47	0.03	**0.91**	0.06	**16**
15	35	45	0.02	**0.94**	0.04	**27**
16	35	41	0.50	0.49	0.01	1
17	13	41	0.51	0.27	0.22	2
18	11	41	**0.95**	0.04	0.01	**27**
19	35	47	0.01	**0.95**	0.04	**24**
20	35	37	0.00	0.51	0.49	1
21	13	35	0.02	**0.77**	0.21	**4**
22	17	47	0.01	0.47	0.52	1
23	37	47	0.01	0.48	0.51	1
24	13	37	0.02	0.30	0.68	2
25	13	15	0.02	0.31	0.67	2
26	11	13	0.47	0.31	0.22	2
27	11	15	0.46	0.06	0.48	1
28	17	37	0.00	0.03	**0.97**	**37**
29	15	37	0.01	0.05	**0.94**	**19**
30	15	17	0.01	0.04	**0.95**	**26**

Table 2: Computed degrees of membership of the lines to the three clusters (rounded).

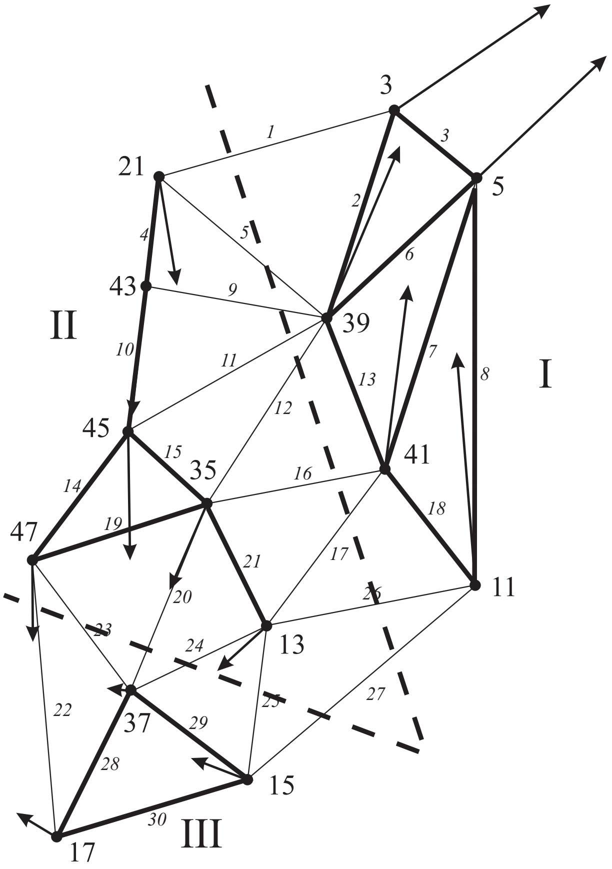


Figure 5: Test network Delft: Object points with their displacement vectors between the epochs 1 and 3b.

4 Automated clustering of settlements

The measurements[2] of settlements i.e. of buildings are usually carried out over several measurement epochs and their results are presented as a time series in numerical and/or graphical form. From a graphical representation of such a time series an experienced user is able to relatively easily infer the movement behaviour of a measurement point, which can be attributed to a particular pattern (e.g. linear, accelerating, consolidating or no settlement). However, in large present-day projects, settlements are commonly measured in an automated way at much more than 1,000 points (e.g. (Heinz and Pönitz, 2006)). Such measurements can be manually analysed only under great effort. Especially fuzzy clustering methods lend themselves to the automated classification of the movement patterns, since unique assignments to a specific pattern are often not possible or appropriate.

Features with which an experienced user would assign the time series to specific patterns are, for example, the average slope of the time series between the initial epochs and the average slope between the last epochs. This

[2] This example was already published in german language in Fletling (2007).

procedure is emulated with the automatic approach presented here. Theoretically, arbitrarily many features are possible. However, since a two-dimensional feature space can be more easily plotted than a higher dimensional space only two features are used in this example: The avarage slope of the time series in the first half of the time and the average slope in the second half.

The following enumeration shows the course of the automatical clustering prodedure.

1. At the beginning of the analysis, an expert supplies prototypes of expected movement patterns in a normalized time and measurement range (e.g. 0..1, see figure 6). These prototypes 1, 2, 3, 4 are then mapped into the feature space (figure 8), where they form the cluster centres to which the actually measured time series (e.g. A, B, C, D in figure 7) will be assigned to later.

2. After each measurement epoch, the measured time series (figure 7) are normalized to lie within the same range as the prototypes in order to get comparable conditions. Subsequently, the same features are computed from the time series as were from the prototypes. The time series are then mapped into feature space.

3. By computing the distances between the position of the time series in feature space and the cluster centres of the prototypes, the corresponding degrees of membership can be obtained from equation (3).

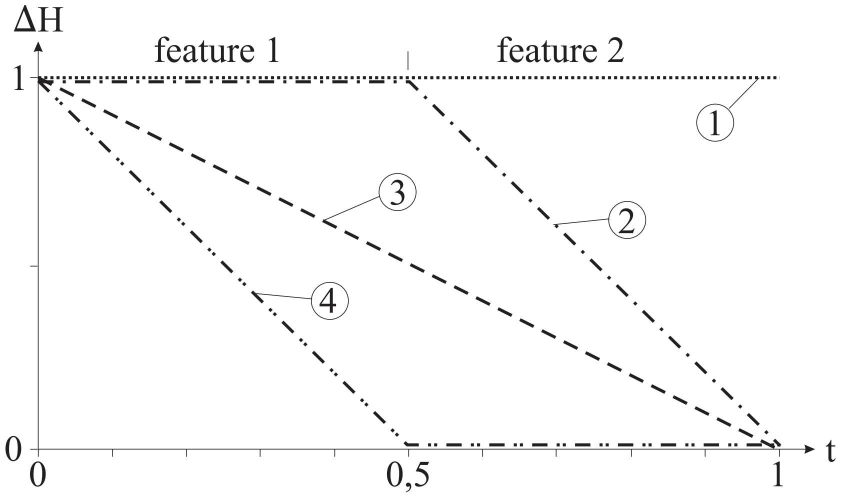


Figure 6: Simple prototypes of expected settlement pattern.

1	:	no settlement
2	:	accelerated settlement
3	:	linear settlement
4	:	consolidating settlement

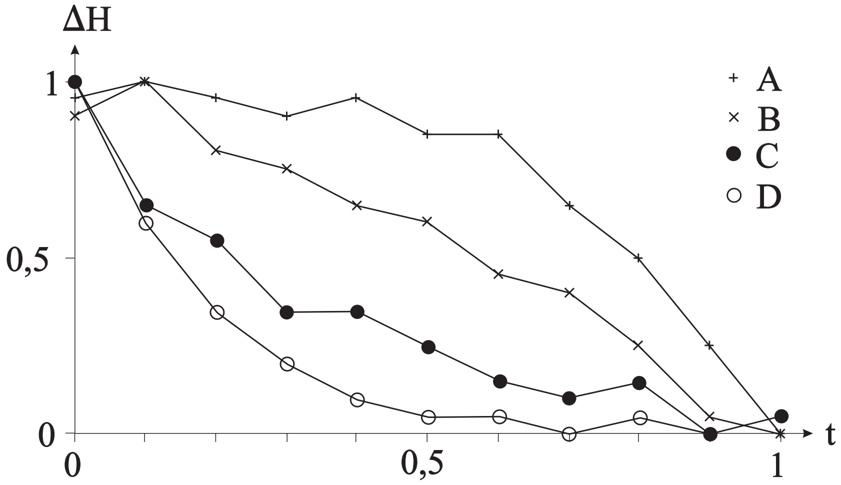


Figure 7: Normalized time series of measured settlements.

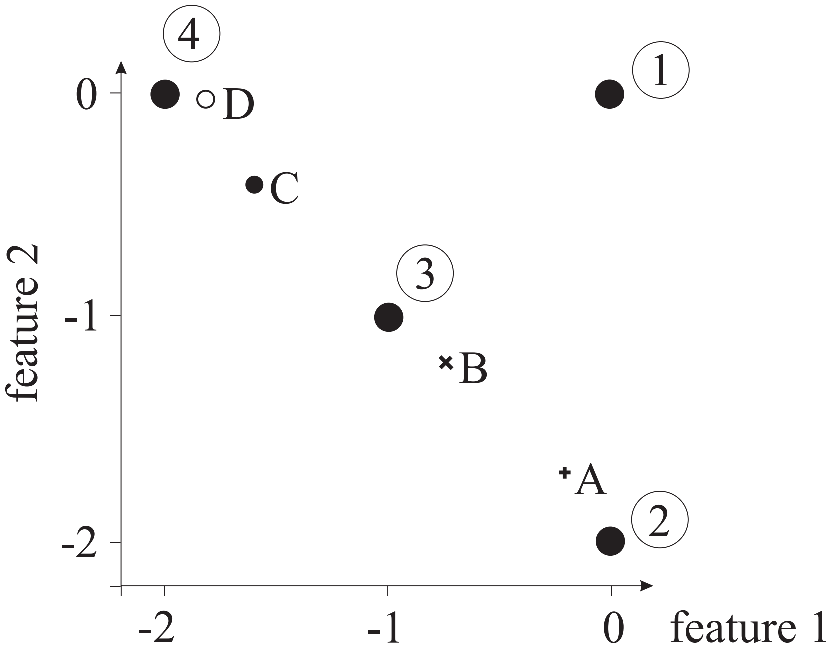


Figure 8: Two-dimensional feature space with the cluster centres 1 - 4 of the prototypes and the time series A - D.

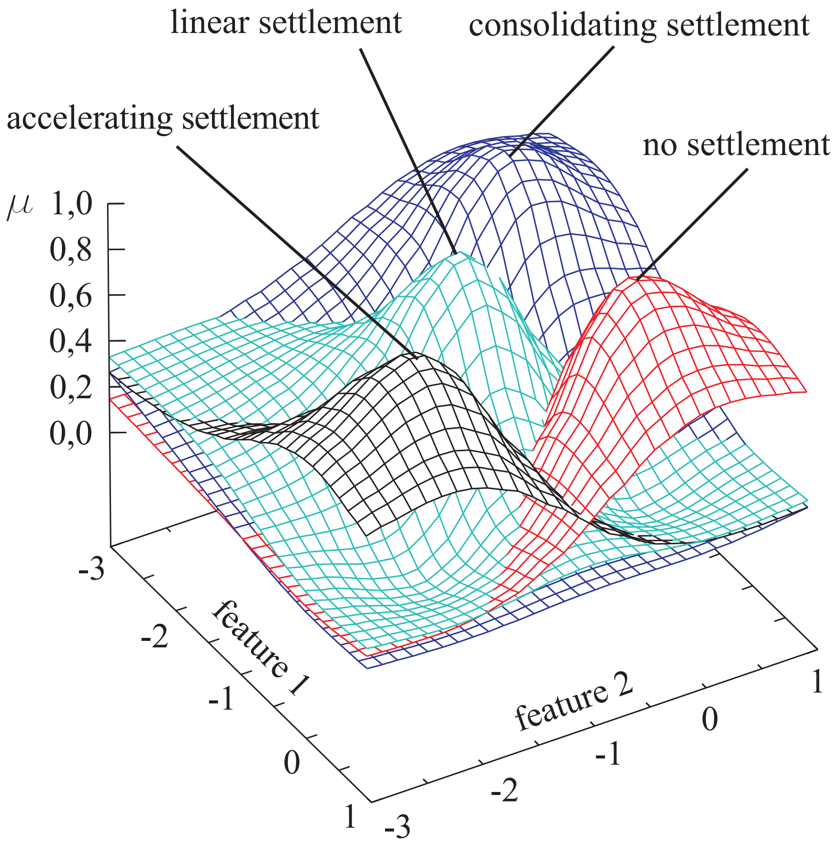


Figure 9: The four membership functions of the four prototypes.

	settlements			
	no	accel.	linear	consol.
A	0.03	**0.89**	0.07	0.01
B	0.05	0.08	**0.85**	0.02
C	0.11	0.05	**0.41**	**0.43**
D	0.01	0.00	0.02	**0.97**

Table 3: Computed degrees of membership of the measured time series to the prototypes.

By means of the computed membership values (Table 3) the fuzzy classification of the measured time series to the prototypes can be identified. In the example, the series of measurements A, B and D are clearly assigned to the corresponding patterns. Only series C possesses approximately equal degrees of membership to the patterns of a linear and a consolidating settlement. The decision made by the algorithm would certainly have been made similarly by a human expert. The method shows that with such a classifier the manual interpretation of time series can largely be replaced by an automated system.

Only few, hard to classify series of measurements (e.g. the series C) have to be classified by a human expert if necessary.

Problematic in such an automated classification are also measurement series that equally possess relatively large distances to all prototypes. The constraints $\sum_{i=1}^{c} \mu_{i,k} = 1$ and $\mu_{i,k} \in [0,1]$ of the employed fuzzy methods lead to that in this case the membership values of the corresponding measurement series approach $1/c$. This is slightly discernible in Figure 9 in the left corner. Remedy could be found in so-called possibilistic clustering methods (e.g. after Krishnapuram and Keller (1993)) that perform without the above mentioned constrains.

The presented approach is arbitrarily extensible. Possible would be further prototypes that, for instance, account for elevations or, as mentioned above, more features. Extending the approach to pattern recognition of point movements in two- or in three-dimensional control networks is feasible.

5 Conclusion

The presented examples showed that two typical problems from the field of geodetic deformation analysis can be tackled by making use of fuzzy clustering methods. It was shown that the methods yielded good results. The up to now in practise commonly used procedure of having an expert manually analyse graphical representations of the data can be automated to a large extent by the fuzzy approach.

References

Bezdek, J.C.: Fuzzy Mathematics in Pattern Classification. Cornell University, Ph.D. Thesis, Ann Arbor, 1973

Burmeister, J.: Tutorial Grundlagen der Fuzzy-Pattern-Klassifikation. TAT Transfer von Automatisierungstechnologien GmbH, Chemnitz, 1997

Duda, R.O., Hart, P.E., Strork, D.G.: Pattern Classification (2nd ed.). Wiley & Sons, New York, 2000

Fletling, R.: Fuzzy-Clusterung zur Analyse von Überwachungsmessungen. In: Brunner, F.K. (Ed.): Ingenieurvermessung 07, Beiträge zum 15. Internationalen Ingenieurvermessungskurs, Wichmann Verlag, Heidelberg, 2007

Gath, I., Geva, A. B.: Unsupervised Optimal Fuzzy Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 11, 1989

Gustafson, E. E., Kessel, W. C.: Fuzzy Clustering with a Fuzzy Covariance Matrix. IEEE CDC, San Diego, Kalifornien, 1979

Haberler, M.: Einsatz von Fuzzy Methoden zur Detektion konsistenter Punktbewegungen. Veröffentlichung des Instituts für Geodäsie und Geophysik der TU Wien, Heft Nr. 73, Wien, 2005

Heinz, D., Pönitz, S.: Bewegungs- und Deformationsmonitoring am Projekt City-Tunnel Leipzig. In Schwarz, W. (Ed.): Interdisziplinäre Messaufgaben im Bauwesen - Weimar 2006. Schriftenreihe des DVW, Band 50, Wissner Verlag, Augsburg, 2006

Höppner, F., Klawonn, F., Kruse, R.: Fuzzy-Clusteranalyse. Vieweg Verlag, Braunschweig, 1997

Krishnapuram, R., Keller, M. (1993): A possibilistic approach to clustering. IEEE Transactions on Fuzzy Systems, 1(2), S. 98-110, 1993

Welsch, W. (Ed.): Deformationsanalysen ’83. Schriftenreihe Studiengang Vermessungswesen Hochschule der Bundeswehr München, Heft Nr. 9, München, 1983

Windham, M. P.: Cluster Validity for Fuzzy Clustering Algorithms. Fuzzy Sets and Systems, 5, 1981, pp. 177 - 185, 1981

Artificial neural networks – how to open the black boxes?

Michael Heinert

Institut für Geodäsie und Photogrammetrie
Technische Universität
Carolo-Wilhelmina zu Braunschweig
e-mail: m.heinert@tu-bs.de

Abstract

A combination of several simple measures enables the analysis of a trained neural network and thus the possibility for an improvement of it. Such way it is possible to open the black box of an artificial neural network (ANN). These are simple in set up, easy to train and deliver quickly well-fitted model results. But an ANN faces increasing problems with the complexity of the systems it has to model. While the fit of the training data can be infinitely improved by adding more neurons, the quality of the prediction becomes worse.

The black box character that was firstly the big advance in the use of ANN – the supervisor must not think about the model set up – becomes the crucial disadvantage. Finally, there seems to be no possibility to improve the networks by using simple rules up to now.

Accordingly it is necessary to understand how an ANN projects the behaviour of a system into its weight matrix: the original system function will be approached by an n-dimensional polygon. Thus, beside the necessary normalisation of the input and output data a kind of pre-linearisation of the input-output relation is very helpful. Sophisticated supervisors feed their network not only with the original input. They use additionally for example derivates, squares, square roots of the original input to train a network. The network has now to decide on its own which input is the most suitable one. Therefore it is a straighter way to use from the beginning – if already known – the real physical relations between input and output.

An ANN also uses the weights in different ways: if there are more neurons available than necessary, one and the same linear relation can be expressed in the weight matrix with more than one neuron per layer. It would be more simple to sum up their weights and use them in only one neuron. In the

case of less neurons than necessary several similar input-output relations will pass one and the same group of neurons. It becomes clear that the next important step has to be the determination of a suitable size of the network architecture. This needs not to take place heuristically: rather ideal measures for a model capacity already exist. The most common is the Vapnik-Chervonenkis-dimension that has been developed within the statistical learning theory.
The trained functional relations can be extracted using a alternating sensitivity analysis of the input data: one input channel gets linearly increasing inputs while all the other input channels are set on a mean value. Each output curves can be investigated by the techniques of the functional analysis.

Keywords: System Analysis, Artificial Neural Networks, Statistical Learning Theory, Semi-parametric Models, Model Inference.

1 Introduction

The artificial neural network technique was originally designed for pattern recognition purposes (Turing, 1948; Farley and Clark, 1954; Rosenblatt, 1958). Nevertheless it can be used as a regression as well. This becomes interesting if we look at geodetic observations like a process and therefore as a system's output (Heine, 1999).

On one side it is the big advantage of an artificial neural network (ANN) that it does not ask for physical pre-information to create the possibility of modelling. Furthermore it projects the system's function into a weight matrix without any physical relevance. But this *black box character* is on the other side a crucial problem: if the complexity of the system increases, an ANN needs much more projection capacity and becomes complex itself. If the network output fits the training patterns including the system's output, but stops delivering plausible predictions, it is not easy to find a way to improve the solution.

Within the following discussions the view should be focussed onto types of supervised learning feed-forward networks. This paper does not deal with unsupervised or reinforced algorithms like for example self-organizing maps (SOM) or agents which are counted among the neural networks as well (Haykin, 1999, chap. 9).

2 How does an artificial neural network model a system?

2.1 A biological approach

Let us remember once more that the artificial neural network technique is based on a crude abstraction of a biological brain. Therefore we remember once more the way how a biological brain works. A natural neural network consists of a group of cells – called the neurons. A neuron receives at its synapses chemo-electric signals from their neighbours (Haykin, 1999, pp. 9). If the combined electric current at a discrete time exceeds the cell-owned threshold, it starts "firing" itself. It sends out an own chemo-electric signal to its neighbours (Fig. 1). All these synapses are getting strengthened which cumulated stimuli makes our neuron "firing". All the inactive synapses atrophy until this connections even vanish. Such way only needed connections are established. The mathematical copy of this process has been done like this: every synapse is represented by a weight w_i. This weight multiplies the signals x_i that reaches our neuron from its neighbour (Spitzer, 2002, pp. 55). The cumulated stimulus at the time t can be expressed

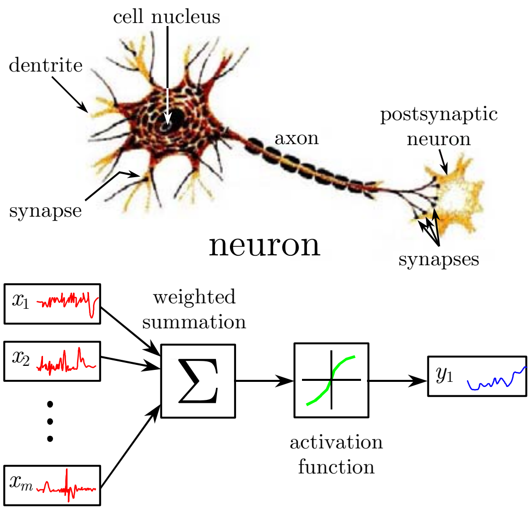


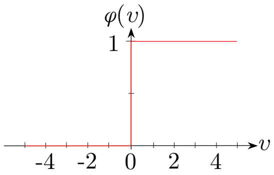
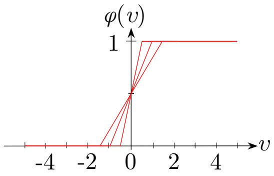
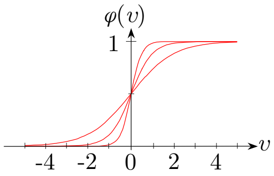
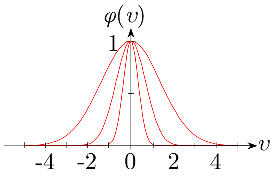
Figure 1: The biological neuron (Sitter, 2001) as the model for the set-up of an artificial neuron (Heinert, 2008).

$$\bar{\bar{y}}(t) = \sum_{i=1}^{n} w_i \cdot x_i(t) = \mathbf{w}^{\mathsf{T}} \cdot \mathbf{x}(t) \ \forall \ \mathbf{x} \in \mathbb{X} \subset \mathbb{R}^n, t \in \mathbb{T} \subset \mathbb{R}, \mathbf{w} \in \mathbb{W} \subset \mathbb{R}^n. \tag{1}$$

The reaction of a stimulated neuron is described by an activation function. The most similar to a natural one is the Heaviside-function

$$\Theta(\upsilon) = \lim_{\varepsilon \to 0} -\frac{1}{2\pi i} \int_{-\infty}^{\infty} \frac{1}{\tau + i\varepsilon} e^{-i\upsilon\tau} \, \mathrm{d}\tau \Longleftrightarrow \varphi_{\Theta}(\upsilon) = \frac{1}{2}\left(\frac{\upsilon}{|\upsilon|} + 1\right). \tag{2}$$

Table 1: Commonly used activations. The first three are suitable for perceptrons, the last more for RBF-networks.

threshold-activation (Heaviside)	$\varphi_\Theta(v) = \frac{1}{2}\left(\frac{v}{\lvert v\rvert} + 1\right)$	
bounded linear activation	$\varphi_{[/]}(v) = \begin{cases} 0 & \forall \quad v \leq \frac{1}{2m} \\ mv + \frac{1}{2} & \forall \quad -\frac{1}{2m} < v \leq \frac{1}{2m} \\ 1 & \forall \quad v > \frac{1}{2m} \end{cases}$	
sigmoid activation	$\varphi_\Sigma(v) = \frac{1}{1+e^{-av}}$	
Gaussian activation	$\varphi_N(v) = e^{-\left(\frac{v}{\sigma}\right)^2}$	

To refer to the fact that every cell got its individual reaction

$$\bar{y}(t) = \varphi_\Theta\left(\bar{\bar{y}}(t)\right) = \varphi_\Theta\left(N + \sum_i w_i \cdot x_i(t)\right) = \varphi_\Theta\left(N + \mathbf{w}^\mathsf{T} \cdot \mathbf{x}(t)\right). \quad (3)$$

on the different stimuli x, the mathematical neuron is biased by an additive N.

An other suitable network type is the so-called RBF-network. Several activated radial baseline functions

$$\Phi_i(\mathbf{w}) = \varphi\left(\lVert \mathbf{x}_i - \boldsymbol{\chi}_k \rVert\right) + N \quad (4)$$

are placed within one hidden layer (Patterson, 1996, sect. 7.2.1). This network type was 1965 well-founded by Cover's theorem (Cover, 1965). Beside this pure theoretical approach, there exist biological reasons for this network type. For the navigation through their habitat the rat's hippocampus uses so-called *place cells* (Wilson and McNaughton, 1993). Their activity depends on the distance to well-known places. If the rat is nearing such a well-known place that one neuron "fires", that represents this place. The summation of the place cells signals tells the rat its position between all these well known places (Spitzer, 2002, pp. 24). Accordingly, RBF networks are quite suitable non-linear interpolation algorithms.

2.2 A comparative approach

Each artificial neuron contains of a high model capacity. This becomes clear by assorting an artificial neuron into its algorithm family. Therefore one has to forget for a while the original consideration that the artificial neuron is a copy of a biological one. A recursive algorithm that is able to predict an element of a time series using the weighted information of former epochs is the well-known autoregressive model

$$y_t = \sum_{\tau=1}^{p} u_\tau y_{t-\tau} \tag{5}$$

with p parameters u, abbreviated as AR[p] (Fig. 2a). This algorithm has been extended to refer to impacts that acting onto this process. It is possible to include an exogenous process by the multiplication of its values x with weights w into the summation (Fig. 4b). Analysing the structure of a non-linear activated recursive artificial neuron one can find that this is the same like a so-called non-linear autoregressive exogenous model (NARX) (Bunke, 1997; Heinert and Niemeier, 2004). A more detailed comparison of the algorithms and their designations supplies new realizations about them. It is possible to find results of investigations carried out in different specializations. Several present investigations emphasize the benefits they took from the enormous model capacity of this algorithm (Ghosh and Maka, 2008; Mellit and Kalogirou, 2008; Menezes and Barreto, 2008, et al.). In an artificial neural network are more than one of these powerful algorithms combined, what multiplies their model capacity. Therefore, we should compare:

- unbiased recursive neuron with unbounded linear activation

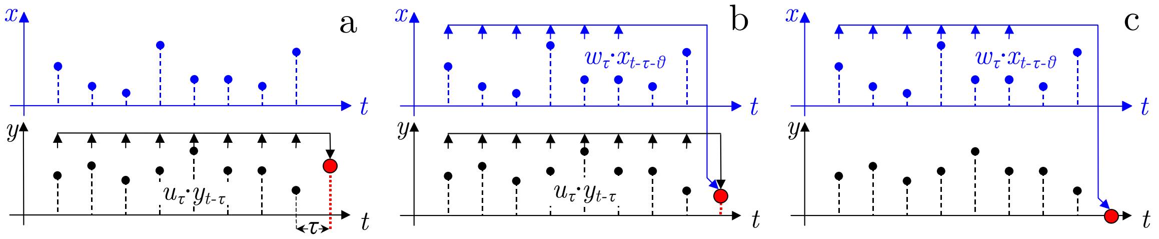

x
t
y
t
a
$u_\tau \cdot y_{t\text{-}\tau}$
τ
x
t
y
t
b
$w_\tau \cdot x_{t\text{-}\tau\text{-}\vartheta}$
$u_\tau \cdot y_{t\text{-}\tau}$
x
t
y
t
c
$w_\tau \cdot x_{t\text{-}\tau\text{-}\vartheta}$

Figure 2: Comparison of different algorithms: a) the autoregressive model, b) the autoregressive model with one exogenous input and c) the non-recursive neuron with unbounded linear activation.

$$\Longleftrightarrow\ y_t = \sum_{\tau=1}^{p} u_\tau y_{t-\tau} + \sum_{\theta=0}^{e} w_\theta x_{t-\theta}\ \Longleftrightarrow$$
autoregressive model with exogenous input (ARX[p, e]),

- biased recursive neuron with unbounded linear activation
$$\Longleftrightarrow\ y_t = \sum_{\tau=1}^{p} u_\tau y_{t-\tau} + N + \sum_{\theta=0}^{e} w_\theta x_{t-\theta}\ \Longleftrightarrow$$
autoregressive model with biased exogenous input,

- recursive neuron with non-linear activation
$$\Longleftrightarrow\ y_t = \varphi\left(\sum_{\tau=1}^{p} u_\tau y_{t-\tau} + N + \sum_{\theta=0}^{e} w_\theta x_{t-\theta}\right)\ \Longleftrightarrow$$
non-linear autoregressive model with exogenous input (NARX[p, e]),

- non-recursive neuron with non-linear activation
$$\Longleftrightarrow\ y_t = \varphi\left(N + \sum_{\theta=0}^{e} w_\theta x_{t-\theta}\right)\ \Longleftrightarrow$$
non-linear trivial autoregressive model with exogenous input (ARX[$0, e$]).

2.3 A theoretical approach

Another explanation of the high model capacity of neural networks is surprising as well. Let us imagine that all the patterns, namely input and output data, can be transformed into a n-dimensional feature space $\mathcal{H}$. Its dimension n is significantly higher than the number m of the input-output channels. Within this feature space would exist a $n - 1$ dimensional hyperplane that shatters – for pattern recognition purposes – or fits – for regression purposes – the patterns. Such a transformation would need a lot of computational efforts. So, let us imagine that this hyperplane could be re-transformed with the data into the original space of the patterns. It can be shown mathematically, that an ANN can be the result of a re-transformation of a best fitting hyperplane of the n-dimensional feature space[1] (Haykin, 1999, p. 333).

To understand this theoretical approach, let us imagine that our patterns are defined to be to a plane in a three-dimensional feature space (Fig. 3, left). Accordingly, a three-dimensional linear regression would fit the patters immediately. If these patterns are projected for example into a plane along two of the three Cartesian axes, a linear regression will fail within this projected space (Fig. 3, right). After all this is exactly the situation in which a user has to start the analysis in the data space.

[1] The support vector machines (SVM) are based on this special idea as well (Vapnik, 1998; Haykin, 1999; Schölkopf and Smola, 2001; Riedel and Heinert, 2008)

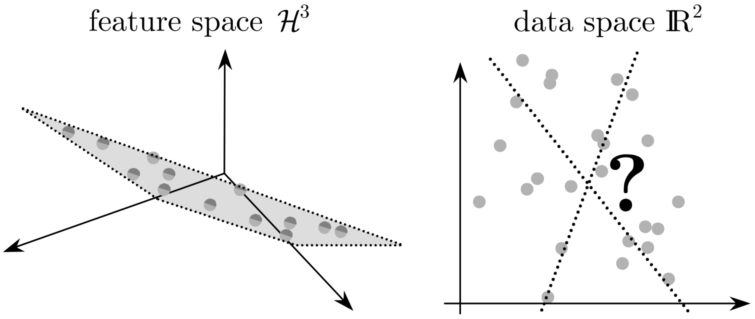


Figure 3: A set of patterns is defined to be on a plane in a three-dimensional feature space (left). After a projection of the patterns into e.g. the x-z-plane of the coordinate system a regression is impossible (right).

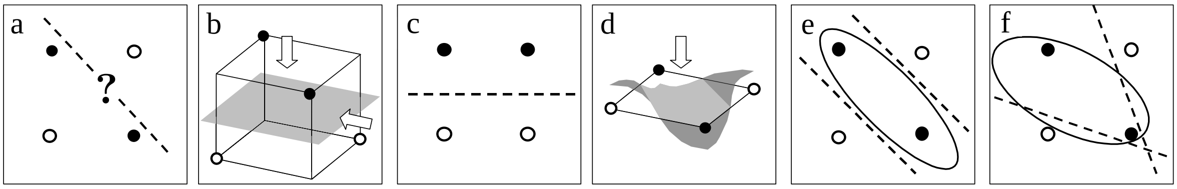


Figure 4: XOR-problem: a) linearly non-separable patterns, b) transformation into a 3D feature space (view on the original space: arrow upside down), c) possible transformation into an orthogonal data space (arrow from the right in b)), d) transformation of the hyperplane into the data space, e) theoretical classification by a perceptron (dashed lines) or an radial baseline function net (ellipse) and f) typical actual ANN approaches.

Example: As a last repetition of this approach let us look at the often used pattern recognition example, namely the so called XOR-problem (Haykin, 1999, pp. 175 and pp. 259). The results of the Boolean function *exclusive or* have to be linearly shattered. Therefore a *right* can be labelled by the value 1 and a *false* vice versa by a -1. It can be shown that there exists no line is able to separate the patterns referring to their results (Fig. 4a). If we create vectors by extending each input vector x by its pattern label, these vectors would be part of the feature space $\mathcal{H}^3$ (Fig. 4b). Within this space exists a hyperplane that shatters the patterns. This simple example shows that there must exist another data space in which it should be possible to shatter the patterns linearly (Fig. 4b, arrow from the right). In this very case it is every orthogonal data space (Fig. 4c). As the feature space containing the shattering hyperplane is re-transformed into the original data space – it is not possible as depicted in Fig. 4d, but a helpful experiment of thought – it can be shown that this re-transformation is approached by different network solutions (Haykin, 1999, p. 333).

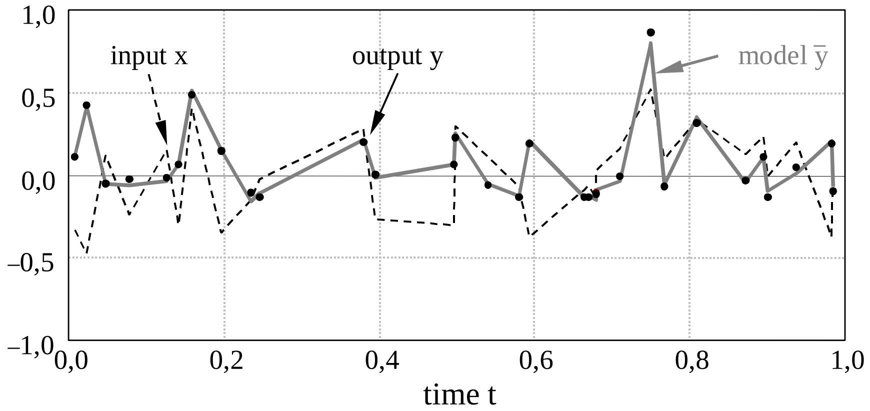


Figure 5: An 2-1 layered ANN models a time series. The observed system's output y (dots) is the squared input x (dashed line). The modelled output $\bar{y}$ (grey line) seems to fit well.

2.4 A practical approach

Nevertheless, how a network with its neuron acts is quite simple. It becomes easy to understand, if we look at neurons containing a bounded piecewise-linear activation-function

$$\varphi_{[/]}(\upsilon) = \begin{cases} 0 & \forall \quad \upsilon \leq \frac{1}{2m} \\ m\upsilon + \frac{1}{2} & \forall \quad -\frac{1}{2m} < \upsilon \leq \frac{1}{2m} \\ 1 & \forall \quad \upsilon > \frac{1}{2m} \end{cases} . \tag{6}$$

This function has got the slope m and the value 0 for the lower and the value 1 for the upper bound (Tab. 1). Alternatively one can find the definition that it is bounded between -1 and 1 (Haykin, 1999, Fig. 14.22).

Example: In a neural network with only one hidden layer, each neuron uses its weights to create a secant along the real output function. To demonstrate this, let the system's output $y(t)$ be the squared input of $x(t)$ (Fig. 5). The best neural network fit with two neurons will create a continuous trough-shaped model function with three bounded linear segments (Fig. 6a). How the slopes of the segments are and whether the model function is symmetric like the system's function depends on the interval and the distribution of the output values (Fig. 6b). Imagine that there are only four patterns, the network should be able to fit them without any residuals (Fig. 6c). These four patterns can easily be modelled by another combination of activations, what means that the weights can be totally different, especially that activation function that creates the right part of the approach.

If one now uses a non-linear activation function it is possible to remove the typical edges of polygonal curves. It is quite common to use a hyperbolic

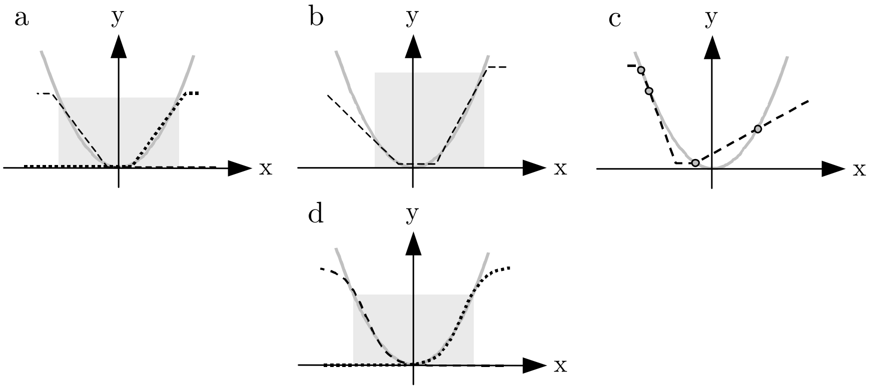


Figure 6: Same ANN as in Fig. 5, but with output $\bar{y}$ sorted by x: a) two bounded linear activated neurons with a symmetrical input interval (grey field), b) with an asymmetrical input interval, c) by only using four single patterns and d) two sigmoid activated neurons with a symmetrical input interval.

tangent

$$\varphi_\Sigma(v) = \frac{1}{1+e^{-av}}. \tag{7}$$

Within the interval of output values we may expect a very smooth approach to our system's output (Fig. 6d).

2.5 What makes an ANN a black box?

The basic problem of every ANN is its black box character, that prevents any analyses of the reliability and the plausibility of modelled output. Accordingly it happens, that every new computational approach ends up with a different new weight matrix (Miima, 2002). Surprisingly, the results of all the approaches can be nearly or totally the same. This phenomenon is caused by the start matrix, that has to contain small random values. In most of the software solutions the start matrices are different in the beginning of every new programme run. Therefore the trained weight matrices differ basically. These differences occur in both the position of a matrix element and its value. This is a typical effect forced by what is called an *underdetermined network* (Vapnik, 1998; Haykin, 1999). That means that the model capacity of the chosen network architecture is to high in comparison with the number of patterns. This term of underdetermination is not the same to that one which is commonly used for adjustments (Niemeier, 2008): even if the number of free degrees is very high an ANN can be underdetermined by far.

It is necessary to understand, what is going on inside an underdetermined network. One and the same numerical input-output relation can be expressed

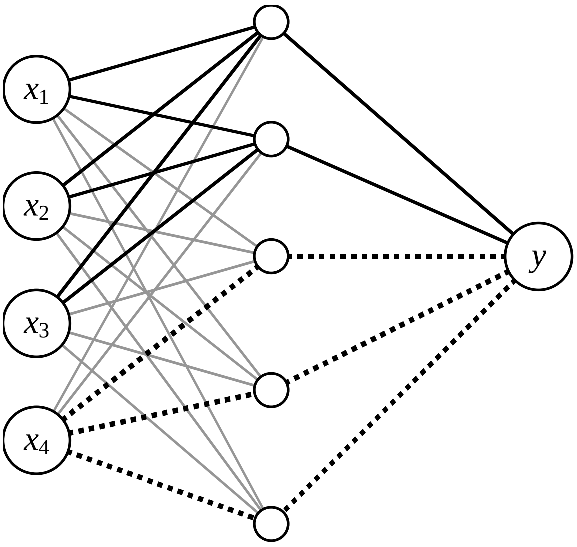


Figure 7: The undertermined network: implicit weight sharing causes that three inputs use only two neurons (black lines) and the remaining input may use three neurons. The other synapses can be insignificant (grey).

by separated groups of neurons. But the optimization of the network during the training phase forces a fast approach by a training of all neurons. While separated groups of neurons would have been able to project the input-output relations, this projection is spread over all neurons now. Easily spoken, if the user sums up the weights of the homologues neurons and deletes the redundant ones afterwards, the modelled result would be exactly the same. The new reduced ANN would be much smaller instead.

Example: Let us imagine that we want to model a process y by four inputs x_i. Our network contains five neurons within the hidden layer. Let us further assume, that the input-output relation between the first three inputs with the output are the same, e.g. the input has to be squared to model the output. As we already know: two neurons are sufficient for modelling this (Fig. 6). Accordingly, one possible solution could be that the first three inputs come to an implicit *weight sharing* using the first two neurons. Now there are three neurons left to model the last input x_4. In the case of a linearly relationship the last two neurons are without substantial input: they will immediately start to model any noise.

3 How to prepare the data?

3.1 Why input and output have to be semi-parameterised?

ANNs are not only fed with the original input. Furthermore it is quite common to use derivates, squares, square roots and many other functions of

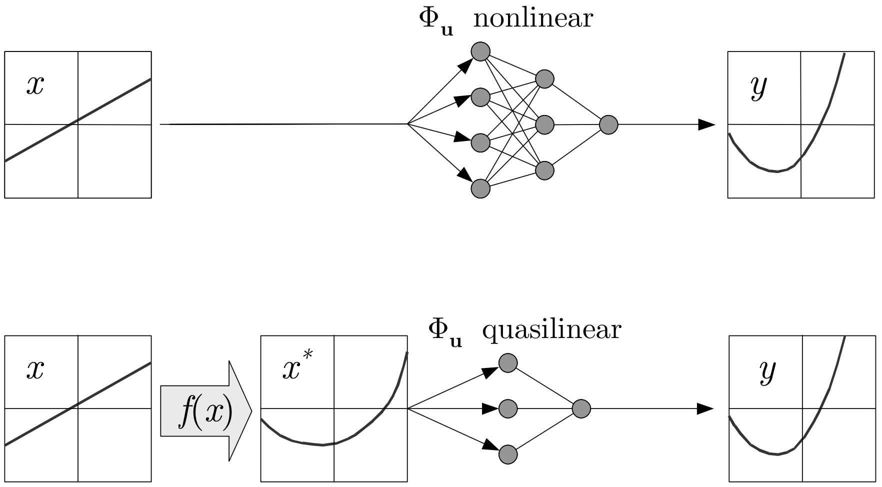

$\Phi_{\mathbf{u}}$ nonlinear
x
y
$\Phi_{\mathbf{u}}$ quasilinear
x
$f(x)$
x^*
y

Figure 8: A pure non-parametric model transforms e.g. a linear input into a quadratic output and uses many neurons. The input data of a semi-parametric model are already functionally pre-treated (Heinert, 2008).

the original input to train a network. It is now upon the network to decide on its own which of them produces the best model output to fit the data. This successful method can be improved. Often there exits some pre-information on the physical relationship between single input and output channels. In such a case it is useful to feed the ANN with the functional output of the well-known physical relationship.

$$x_i^*(t) = f(x_i(t)) \tag{8}$$

Such networks using a pre-treated input are not non-parametrical anymore, therefore the term *semi-parametrical* will be used for such networks within this article.

3.2 Why input and output have to be normalised?

A first important step before the training phase of any ANN is the normalisation of the input and output data. At first all input and output data should be reduced by their expected value μ_x, μ_y. At second the range of values has to be bounded to be between -1 and 1. The necessary scale factor is usually related to the multiple of the standard deviation σ_x, σ_y. Such kind of limitation

$${}^n x_i^*(t) = \frac{x_i^*(t) - \mu_{x^*}}{k_i \sigma_{x_i^*}}, \; {}^n y(t) = \frac{y(t) - \mu_y}{k' \sigma_y} \tag{9}$$

has the practical advantage that neurons with a bounded activation function are not saturated immediately and therefore lost for further modelling purposes.

Table 2: Upper and lower bound of the Vapnik-Chervonenkis-dimension for different activations in dependence of the number of weights n or neurons H within one single hidden layer. This compilation can be found in Heinert (2008).

activation	$\varphi(\upsilon)$	υ	$\mathcal{H}_{min}$	$\mathcal{H}_{max}$
linearly	$\varphi_{/}(\upsilon)$	$\forall\upsilon$	$\Theta(n)$	$\mathcal{O}(n^2)$
bounded linearly	$\varphi_{[/]}(\upsilon)$	$\lvert\upsilon\rvert > \frac{1}{2m}$ $\lvert\upsilon\rvert < \frac{1}{2m}$	$\Theta(n)$	$\mathcal{O}(n\mathrm{lb}(n))$ $\mathcal{O}(n^2)$
threshold	$\varphi_H(\upsilon)$	$\forall\upsilon$	$\Theta(n)$	$\mathcal{O}(n\mathrm{lb}(n))$
combinatorial	$\varphi_H(\upsilon) \oplus \varphi_{/}(\upsilon)$	$\lvert\upsilon\rvert > \frac{1}{2m}$ $\lvert\upsilon\rvert < \frac{1}{2m}$	$\Theta(n)$	$\mathcal{O}(n\mathrm{lb}(n))$ $\mathcal{O}(n^2)$
sigmoidal	$\varphi_\Sigma(\upsilon)$	$\lvert\upsilon\rvert < \frac{1}{a}$ $\lvert\upsilon\rvert > \frac{1}{a}$	$\Theta(n)$	$\mathcal{O}(n^2)$ $\mathcal{O}(n\mathrm{lb}(n))$
RBF	$\varphi_N(\lVert\upsilon\rVert)$	$\forall\upsilon$	$\Theta(n)$	$\mathcal{O}(n\mathrm{lb}(H))$

4 Which size of an ANN is suitable?

What does it mean, if one speaks about underdetermined networks as it has been done in section 2.5 already. How it is possible to describe underdetermination? As also already mentioned in section 2.2 an ANN contains neurons, where each is an algorithm with a high model capacity. In combination with other models this capacity increases rapidly (Tab. 2). The critical point is reached when the algorithm starts memorising single patterns without any degree of generalization. Accordingly, an ANN has to be set up with the smallest possible number of neurons to create intelligence instead of memory.

Vice versa a model is categorized as to be *overdetermined* if the number of neurons is too low. Then the ANN looses his model capacity and is only able to imitate the input-output relation within the range of values.

At this point the necessity of a numerical measure of a model capacity becomes clear. A meanwhile quite common measure is the coefficient h which describes the capacity of a set of functions with logarithmic bounded growth function (Vapnik and Chervonenkis, 1974; Vapnik, 1998). In honour of the developers it is called the Vapnik-Chervonenkis-dimension or abbreviated VC-dimension (Haykin, 1999, p. 95).

> **Definition**: The VC-dimension of an ensemble of dichotomies $\mathcal{F} = \{\Phi_w(x) : w \in \mathbb{W}, \Phi : \mathbb{R}^m\mathbb{W} \to 0, 1\}$ is the cardinality $h = |\mathcal{L}|$ of the largest set $\mathcal{L}$ that is shattered by $\mathcal{F}$.

Vapnik himself gave 1998 a more explanatory version of this quite compact definition (p. 147):

> The VC-dimension of a set of indicator functions $\Phi_w(y), w \in \mathbb{W}$ is equal to the largest number h of vectors that can be separated into two different classes in all the 2^h possible ways using this set of functions.

Within this definition the set of indicator functions $\Phi_w(x)$ describes that very set from which a specific model function $\Phi_{w^*}(x)$ is taken. It is defined by the optimal parameter $w = w^*$.

Example: This concept has to be explained less theoretical: We already heard about the XOR-problem with its four patterns, but imagine now an example with any three given patterns in a 2D-space. Maximum 3 straight lines are able to shatter the 3 patterns in 2^3 ways (Fig. 9). Therefore the largest number h of vectors is 3. Further the definition $h = 3$ fulfils in our case the condition of 2^3 permutation, that we may state the VC-dimension is 3. By the way, such a straight line is just the result of a single threshold activated neuron.

The principle concept to determine the suitable model size with the optimal VC-dimension is to look for the minimum upper bound on the generalisation error (Fig. 10). Therefore two risks have to be combined: the decreasing training error with increasing model capacity and the – at the same time – increasing confidence interval. The summation

$$R(h) = R_{\text{emp}} + \epsilon_1(N, h, \alpha, R_{\text{emp}}) \tag{10}$$

of the empirical Risk R_{emp} and the confidence interval

$$\epsilon_1(N, h, \alpha, R_{\text{emp}}) = 4\sqrt{\frac{h}{N}\left(\log\left(\frac{2N}{h}+1\right) - \frac{1}{N}\log\alpha\right)} \tag{11}$$

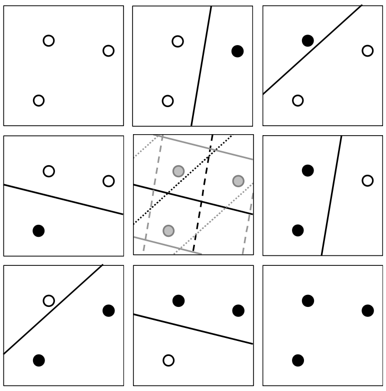


Figure 9: An example for the VC $\dim(\Phi(x)) = 3$: the patterns can be shattered by maximum $h = 3$ lines according to the 2^3 permutations of the patterns.

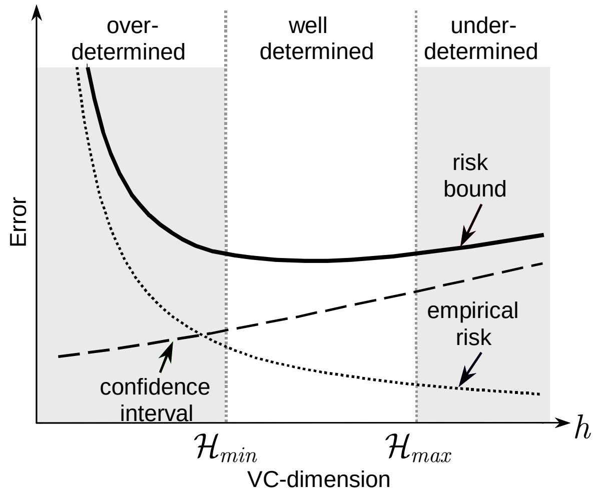


Figure 10: Relationship between the VC-dimension and the upper bound on the generalisation error.

builds the upper bound on the generalisation error referring to the chosen VC-dimension h (Haykin, 1999, pp. 99). Herein the probability is defined as

$$\alpha = \left(\frac{2eN}{h}\right)^{h} e^{-\eta^2 N} \tag{12}$$

with η as the precision of the approximation.
Meanwhile there are more practicable derivations of these theoretical equations (Elisseeff and Paugam-Moisy, 1997): accordingly, the optimal number of nodes $\mathcal{H}$ is within multi-layered feed-forward perceptron is defined as

$$\left\lceil \frac{\mathcal{P}\dim(\mathbf{y})}{\dim(\mathbf{x}) + \dim(\mathbf{y})} \right\rceil < \mathcal{H} < 2 \left\lceil \frac{\mathcal{P}\dim(\mathbf{y})}{\dim(\mathbf{x}) + \dim(\mathbf{y})} \right\rceil . \tag{13}$$

Nodes summarizes neurons (3) as well as RBF-neurons (4). In this equation denotes $\mathcal{P}$ the number of patterns of input and output vectors, $\dim(\mathbf{x})$ the dimension of the input vector and $\dim(\mathbf{y})$ the dimension of the output vector. This rule may be expected to be improved by all the investigations presently carried out within the *advanced statistical learning theory* .

5 How to extract the input-output relations?

After a long list of necessary operations – as demonstrated before: normalising and semi-parameterising the data, choosing a suitable model type and optimal size of the network architecture – we may focus onto the most crucial point of this investigation: the inference of a neural network. Therefore, we start with several assumptions:
❶ the network has got an optimal VC-dimension,

❷ the model converged successfully with different randomly chosen start weights to same solution,
❸ the different successfully converged solutions deliver – beside the exact position within a layer – always the same weight relationships.

We can state that such a stabilized neural network represents a model function

$$\Phi_{\mathbf{u}}(\mathbf{x}) \in \mathcal{F}_{\mathbf{u}} \ni \mathcal{T}_{\mathbf{u}}(\mathbf{x}) \ \forall \ \mathbf{x} \in \mathbb{X} \subset \mathbb{R}^n. \tag{14}$$

The model function $\Phi_{\mathbf{u}}(\mathbf{x})$ is an element of the same set of functions $\mathcal{F}_{\mathbf{u}}$ like the actual system function $\mathcal{T}_{\mathbf{u}}(\mathbf{x})$, what means that we now found an *identifying model* instead of several possible *imitating models* (Heinert, 2008).

The basic idea to extract the learned input-output relations is to feed the trained and converged neural network with time series of well-determined input vectors

$$^{n,0}\mathbf{x}^*_\iota(t) \ni^{n,0} x^*_{\iota,i}(t) = \begin{cases} \nabla\left({}^n x^*_i(t) | {}^n x^*_i(t) \in \mathbb{X}^{(i)}\right) & \forall \ i = \iota \\ \mu_{{}^n x^*_i} & \forall \ i \neq \iota \end{cases} \tag{15}$$

with

$$\nabla\left({}^n x^*_i(t)\right) = \frac{(\max({}^n x^*_i(t) - \min({}^n x^*_i(t))) \cdot t}{t_{\max} - t_{\min}}. \tag{16}$$

These n input vector series containing only means $\mu_{{}^n x^*_i}$ of the normalised input vectors ${}^n\mathbf{x}^*(t)$. Only at the i-th position of every ι-th of the n vectors ${}^n\mathbf{x}^*(t)$ are put linearly increasing values corresponding to the actual interval of input values. Accordingly, we may write for the ι-th vector series:

$$^{n,0}\mathbf{x}^*_\iota(t) = \begin{bmatrix} \mu_{{}^n x^*_1} \\ \vdots \\ \nabla\left({}^n x^*_i(t)\right) \\ \vdots \\ \mu_{{}^n x^*_n} \end{bmatrix} \tag{17}$$

The output as reaction of every ι-th input channel corresponds with the estimated impact of the input acting on the actual system. The resulting output

$$^{n,0}\bar{y}_\iota(t) = {}^n\Phi^*_{\mathbf{u},\mathbf{w}}({}^{n,0}\mathbf{x}^*_\iota(t)) \tag{18}$$

describes now the modelled functional relationship between one single input and the output. Therefore we remember that we normalised in (9) the system's output that the de-normalised model output referring to the standardised input vector can be written as

$$^0\bar{y}_\iota(t) = k'\sigma_y \cdot {}^n\Phi^*_{\mathbf{u},\mathbf{w}}({}^{n,0}\mathbf{x}^*_\iota(t)) + \mu_y. \tag{19}$$

That means for the single i-th input

$$ {}^{0}\bar{y}_{\iota}(t) = k'\sigma_y \cdot {}^{n}\phi^{*}_{\mathbf{u},\mathbf{w}}(\nabla({}^{n}x_i^{*}(t))) + \mu_y. \tag{20}$$

We assort to this input the de-normalised and "de-semi-parameterised" standardised input

$$ {}^{0}x_{\iota}(t) = k_{\iota}\sigma_{x_{\iota}^{*}}{}^{-}f^{*}(\nabla^{n}x_i^{*}(t)) + \mu_{x_{\iota}}, \tag{21}$$

whereby ${}^{-}f^{*}(\,\cdot\,)$ acts as the inverse function in (8) of the i-th input. So, we may define that there exists a function $F(\,\cdot\,)$ which fulfils

$$ {}^{0}\bar{y}_{\iota} \models F({}^{0}x_{\iota}). \tag{22}$$

This function has to be defined by the user in an inductive way and can be fitted to this input-output relation by using the techniques of the non-linear optimization (Domschke and Drexl, 2002; Rardin, 1998; Winston, 1994). This empirical-physical function contains parameters that should be able to be assigned to physical constants.

In the very case that

$$ {}^{n,0}\mathbf{x}^{*}_{\iota}(t) \ni^{n,0} x^{*}_{\iota,i}(t) = \mu_{{}^{n}x_i^{*}} \;\; \forall \;\; i,\iota \tag{23}$$

only the trend of the recursion will remain. This result is the most robust regression without any leverage effects at the ends of the output time series.

6 Exemplary practical model inference

Since nearly ten years the Institut für Geodäsie und Photogrammetrie monitors the bridge instable "Fallersleber Tor" in Brunswick, Germany (Heinert and Niemeier, 2007). To understand the climate impact on the single motions the author used a non-recursive artificial neural network (ANN) with three N-biased neurons in the hidden layer and one output neuron for every coordinate component. The output estimation of one coordinate component

$$ \hat{y}_{i,k} = \varphi_{0/}\left(N_0 + \sum_{\iota=0}^{3}\left(w_{\iota,i}\varphi_{\iota/}\left(\hat{y}^{*}_{\iota,k}\right)\right)\right) \;\; \text{with} \;\; \varphi_{/}(\nu) = \nu \tag{24}$$

and with respect to

$$ \hat{y}^{*}_{\iota,k} = N_{\iota} + \sum_{j=0}^{2}\left(w_{\iota,j}x_{\iota,k-j}\right) $$

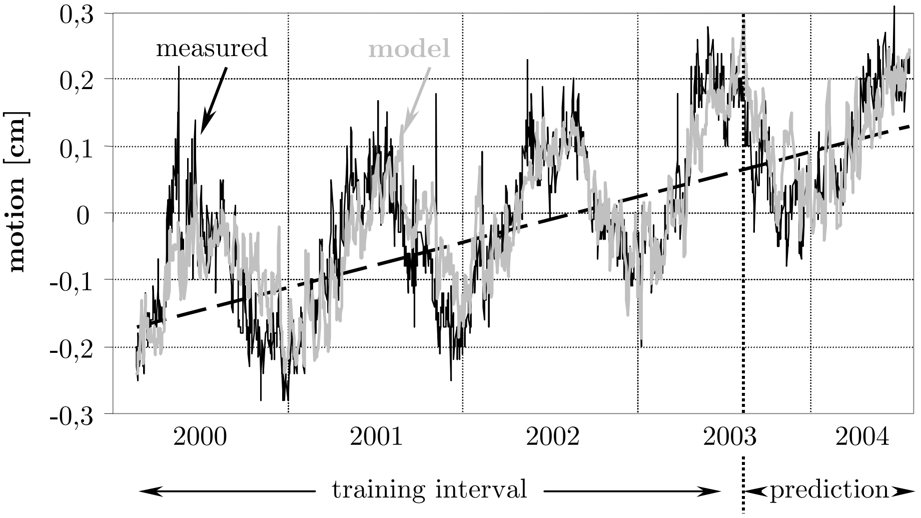


Figure 11: One example of an observed displacement (black) of one target and the successful ANN prediction (grey) of one year. The dashed line shows the linear trend estimation of the ANN.

uses the input vector at epoch k

$$\mathbf{x}_k = \begin{bmatrix} x_{0,k} \\ x_{1,k} \\ x_{2,k} \\ x_{3,k} \end{bmatrix} = \begin{bmatrix} t_k \\ T_k \\ P_k \\ H_k \end{bmatrix}$$

that contains the present information of the time t_k, temperature T_k, pressure P_k and relative humidity H_k. A neuron within the hidden layer combines each the input of one component $x_{\iota,k}$ of the epochs $k - j$ using the weights $w_{\iota,j}$ and the output vector

$$\hat{\mathbf{y}}_k = \begin{bmatrix} \hat{y}_{0,k} \\ \hat{y}_{1,k} \\ \hat{y}_{2,k} \end{bmatrix} = \begin{bmatrix} X_k \\ Y_k \\ H_k \end{bmatrix}$$

that contains the present information of the estimation of each output component $\hat{y}_{i,k}$. The output neuron with its weights $w_{\iota,i}$ decides which coordinate component has to be estimated. This simple network architecture uses the phenomenon of weight sharing (Haykin, 1999, p. 28). That way the number of neurons and accordingly the number of free parameters can be reduced significantly. All neurons of this specific network have unbounded linear activation functions $\varphi_l(\,\cdot\,)$.

Thinking in systems and their inputs it has to be stated that the model *must not* fit during the training phase: several existing inputs beside the weather are not known and not used for modelling. Their impact on the output is neglected and this lack of the model has to result in short-comings of the model

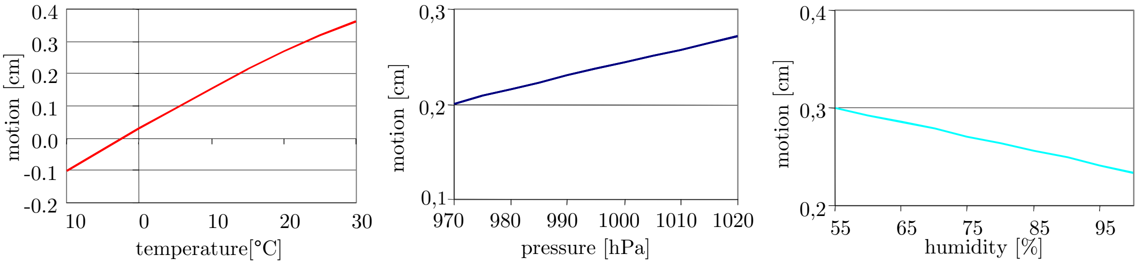


Figure 12: The functional relationships between input and output after model inference.

fit. Forcing the convergence of the risk function by extending the architecture means that we try to create artificial relationships between the input-output pairs. In a most optimum case such artificial relationships represent pseudo-correlations between used and unused inputs.

In spite of the short-comings that our model does not refer to alternative inputs, like traffic load, water level of the river, sunlight intensity and several others, the model does still fit well and the prediction is quite stable (Fig. 11).

By using the technique demonstrated in the previous section we obtain a temperature gradient of about 0.12mm/K, an barometric pressure gradient of about 0.14mm/10hPa and an gradient of the relative humidity of about -0.15mm/10% (Fig. 12). Nevertheless, there still exists a phase lag between the model output and the system's output. Here we assume an aliasing effect due to a neglected diurnal period. The neural network as described above works with pairs of one daily input and output at the same epoch and with the same observation rate. While the air warms fast up during spring days, the structure keeps colder and vice versa while the air cools fast down during autumn nights, the structure keeps warmer. The structure does not only act on the actual temperature, but on the temperatures hours before. This artificial phase lag within the model function creates an additional annual pseudo-wave. Accordingly, a next improved network has to use an input that represents the diurnal temperature course.

Acknowledgements

I am most grateful to my colleagues Dr.-Ing. John-Bosco Miima and Dr.-Ing. Björn Riedel for a lot of very useful discussions on modelling. My special thank is directed to ir. Jeroen Meeus (formerly University of Ghent), who introduced me to the basics of operations research and statistical learning theory. The initions of this study were partly supported by research the grants NI220/8-1 and NI220/8-2 of the German Research Council (DFG).

References

Bothe, H.-H.: Neuro-Fuzzy-Methoden – Einführung in Theorie und Anwendung. Springer-Verlag, Berlin- Heidelberg- New York- Barcelona- Budapest- Hong Kong- London- Mailand- Paris- Santa Clara- Singapur- Tokyo, 1998.

Bunke, J.: Künstliche Neuronale Netze zur Systemidentifikation aus gestörten Messwerten. Improvement-rep. VDI-series 8, No 667, VDI-Verlag Düsseldorf, 1997.

Cover, T. M.: Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. IEEE Transactions on Electronic Computers, EC-14 (3), 1965, 326–334.

Domschke, W. and Drexl, A.: Einführung in Operations Research. 5th rev. and ext. edition, Springer Berlin-Heidelberg, 2002.

Elisseeff, A. and Paugam-Moisy, H.: Size of multilayer networks for exact learning: analytic approach. NeuroCOLT Techn. Rep. Series, NC-TR-97-002, 1997.

Farley, B. and Clark, W. A.: Simulation of self-Organizing Systems by Digital Computer. IRE Transactions on Information Theory 4 (1954): 76–84.

Ghosh, S. and Maka, S.: A NARX modeling-based approach for evaluation of insulin sensitivity. Biomedical Signal Processing and Control, in press.

Haykin, S.: Neural Networks – A Comprehensive Foundation. 2nd edition, Prentice Hall, Upper Saddle River NJ, 1999.

Heine, K.: Beschreibung von Deformationsprozessen durch Volterra- und Fuzzy-Modelle sowie Neuronale Netze. PhD thesis, German Geodetic Commission, series C, issue 516, München, 1999.

Heinert, M. and Niemeier, W.: Zeitreihenanalyse bei der Überwachung von Bauwerken. in Schwarz, W. (Eds.): DVW-Fortbildungsseminar Interdisziplinäre Messaufgaben im Bauwesen – Weimar 2004. DVW-Rep. series No 46, 157–174.

Heinert, M. and Niemeier, W.: From fully automated observations to a neural network model inference: The Bridge "Fallersleben Gate" in Brunswick, Germany, 1999 - 2006. J. Appl. Geodesy 1 (2007), 71–80.

Heinert, M.: Systemanalyse der seismisch bedingten Kinematik Islands. PhD thesis, Univ. of Technology Brunswick (Germany), in press.

Mellit, A. and Kalogirou, S. A.: Artificial intelligence techniques for photovoltaic applications: A review. Progress in Energy and Combustion Science 34 (2008), 574–632.

Menezes Jr., J. M. and Barreto, G. A.: Long-term time series prediction with the NARX network: An empirical evaluation. Neurocomputing 71 (2008), 3335–3343.

Miima, J. B.: Artificial Neural Networks and Fuzzy Logic Techniques for the Reconstruction of Structural Deformations. PhD thesis, Geod. rep. series Univ. of Technology Brunswick (Germany), issue 18, 2002.

Minkler, G. and Minkler, J.: Theory and Application of KALMAN Filtering. Magellan Book Comp, Palm Bay, U.S.A., 1993.

Niemeier, W.: Ausgleichsrechnung – Eine Einführung für Studierende und Praktiker des Vermessungs- und Geoinformationswesens. 2nd rev. and ext. edition Walter de Gruyter, Berlin-New York, 2008.

Patterson, D. W.: Künstliche neuronale Netze: das Lehrbuch. Prentice Hall. München- London- Mexiko- New York- Singapur- Sydney- Toronto, 1996.

Rardin, R.-L.: Optimization in Operation Research. Prentice Hall, Upper Saddle River, USA, 1998.

Riedel, B. and Heinert, M.: An adapted support vector machine for velocity field interpolation at the Baota landslide. In this issue.

Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Reviews 65 (1958), 386–408.

Schlittgen, R. and Streitberg, B. H. J.: Zeitreihenanalyse. 7th edition, R. Oldenburg Verlag, München-Wien, 1997.

Schölkopf, B. and Smola, A. J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning). MIT Press, 2001.

Sitter, R.: Neuronen. `http://home.arcor.de/ralf.sitter/kyb/neuro/neur.htm`.

Spitzer, M.: Lernen: Gehirnforschung und die Schule des Lebens. Spektrum Akad. Verl., Heidelberg-Berlin, 2002.

Intelligent Machinery. Collected Works of A. M. Turing: Mechanical Intelligence. Edited by D. C. Ince. Elsevier Science Publishers, 1992.

Vapnik, V. N. and Chervonenkis, A. Ya.: Theory of Pattern Recognition. (in Russian) Nauka, Moscow, 1974, (German translation: Wapnik, W. N., Tschervonenkis, A. Ja.), Theorie der Zeichenerkennung, Akademia, Berlin, 1979.

Vapnik, V. N.: Statistical Learning Theory. in Haykin, S. (Ed.): Adaptive and Learning Systems for Signal Processing, Communications and Control. John Wiley & Sons, New York- Chichester-Weinheim- Brisbane-Singapore-Toronto, 1998.

Wilson, M. A. and McNaughton, B. L.: Dynamics of the hippocampal ensemble code for space. Science 261 (1993), 1055–1058.

Winston, W. L.: Operations Research: Applications and Algorithms. 3rd edition, Duxbury Press, Belmont, California, 1994.

Personal Navigator Supported by Machine Learning Techniques: Human Dynamics as Navigation Sensor

D. A. Grejner-Brzezinska, C. K. Toth, S. Moafipoor

Satellite Positioning and Inertial Navigation (SPIN) Laboratory
The Ohio State University, USA
email: dbrzezinska@osu.edu

Keywords: Artificial Intelligence, Personal Navigation, Global Positioning System, Wireless Local Area Networks.

The prototype of a personal navigator, which integrates the Global Positioning System (GPS), inertial measurement unit (IMU), digital barometer, magnetometer, and human pedometry to facilitate navigation and tracking of military and rescue ground personnel has been developed at The Ohio State University Satellite Positioning and Inertial Navigation (SPIN) Laboratory. The goal of this system (see Figure 1) is to provide precise and reliable position/velocity/heading information of the individuals in various environments. In the open sky environment, either GPS alone, or a GPS/IMU system can facilitate the basic navigation functionality with the accuracy depending on the choice of GPS and IMU sensors. In confined and GPS-denied environments, however, the main challenge for a personal navigator is to implement a backup plan to maintain the navigation information in the absence of GPS signals.

A basic human locomotion model, considered as navigation sensor, works with the step length (SL) and step direction (SD) as primary parameters. The major focus of this research is on dead reckoning (DR) navigation supported by human dynamics during GPS signal outages. It is demonstrated that in the absence of GPS signals, the sensors used in the current prototype can sense the body locomotion in terms of its dynamics and geometry that represent an implicit function of SL and SD.

The OSU SPIN implementation of the DR system based on human dynamics implemented in the prototype navigator is based on Fuzzy Logic (FL) and Artificial Neural Network (ANN) Knowledge-Based System (KBS); see

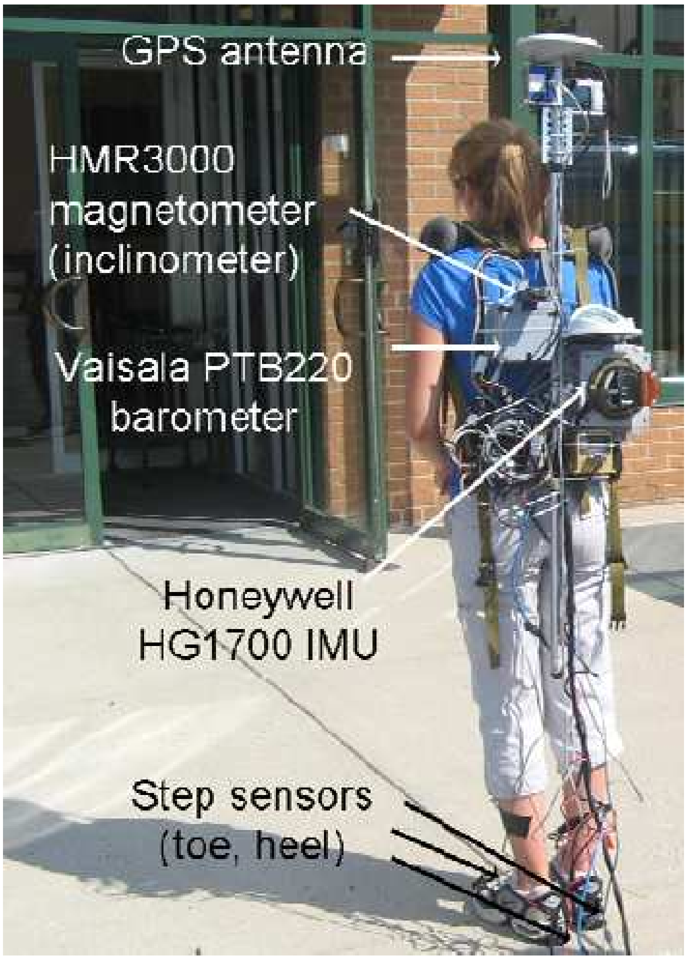


Figure 1: PN sensor configuration.

Figure 2 for the design architecture of the system. The knowledge based system is trained a priori using sensory data collected by various operators in various environments during good GPS signal reception, and is used to support navigation under GPS-denied conditions. SL is correlated with several sensory and environmental data types, such as acceleration, acceleration variation, SF, terrain slope, operatorŠs height, etc. that constitute the input parameters to the KBS system. The KBS-predicted SL, together with the heading information provided by the magnetometer and/or gyroscope, calibrated by the Dead-Reckoning Kalman filter (DR-KF) supports the DR navigation. DR-KF is an additional knowledge-based module used to improve heading information available from the magnetometer and gyroscope under GPS-denied conditions, where GPS-supported sensor calibration cannot be accomplished. The example performance evaluation of the DR module is summarized in Figure 3.

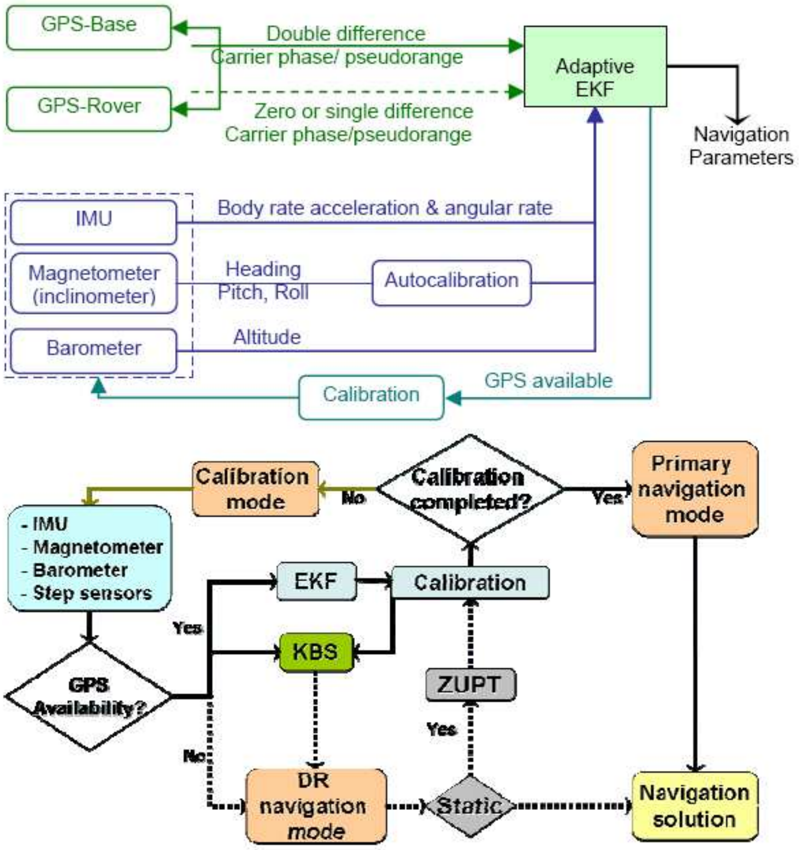
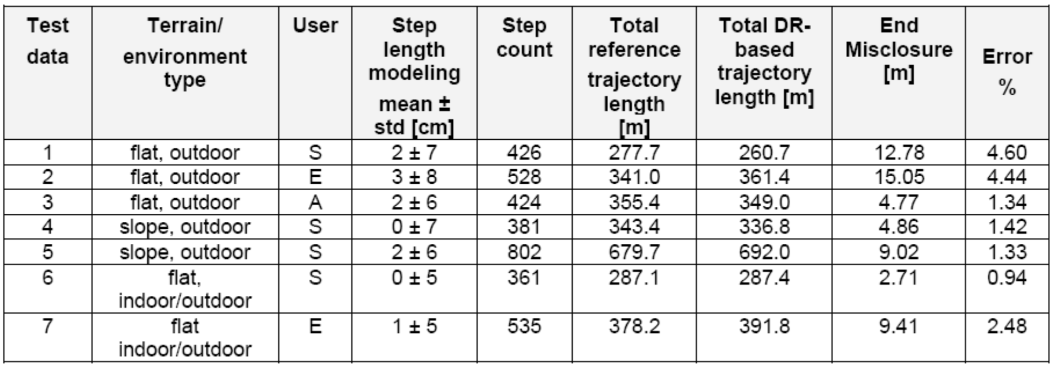


Figure 2: Adaptive Extended Kalman Filter structure for navigation and sensor calibration (top) and Operational modes: initial calibration, primary navigation, and dead-reckoning navigation (bottom).



Test data	Terrain/ environment type	User	Step length modeling mean ± std [cm]	Step count	Total reference trajectory length [m]	Total DR-based trajectory length [m]	End Misclosure [m]	Error %
1	flat, outdoor	S	2 ± 7	426	277.7	260.7	12.78	4.60
2	flat, outdoor	E	3 ± 8	528	341.0	361.4	15.05	4.44
3	flat, outdoor	A	2 ± 6	424	355.4	349.0	4.77	1.34
4	slope, outdoor	S	0 ± 7	381	343.4	336.8	4.86	1.42
5	slope, outdoor	S	2 ± 6	802	679.7	692.0	9.02	1.33
6	flat, indoor/outdoor	S	0 ± 5	361	287.1	287.4	2.71	0.94
7	flat indoor/outdoor	E	1 ± 5	535	378.2	391.8	9.41	2.48

Figure 3: Step length prediction and DR trajectory accuracy: summary statistics.

This short review provided the basic configuration design and the summary of performance evaluation, while the details are provided in the enclosed publication list.

ACKNOWLEDGMENT: This research is supported by the NGA 2004 NURI grant.

References

Grejner-Brzezinska, D.A., C.K. Toth, S. Moafipoor (2008). Body of Movement. Designing a Navigator Using Human Locomotion and Machine Learning, GPS World, November, pp. 34-41.

Grejner-Brzezinska, D.A., C.K. Toth, S. Moafipoor, and Y. Jwa (2006a). Multi-Sensor Personal Navigator Supported by Human Motion Dynamics Model, Proceedings of the 3rd IAG Symposium on Geodesy for Geotechnical and Structural Engineering/12th FIG Symposium on Deformation Measurements, May 2006, Baden, Austria, CD ROM.

Grejner-Brzezinska, D.A., C. K. Toth, S. Moafipoor, Y. Jwa and J. Kwon (2006b). A Low Cost Multi- Sensor Personal Navigator: System Design and Calibration. Presented at IEEE/ION PLANS Meeting, April 25-27, San Diego, CA.

Grejner-Brzezinska, D.A., C.K. Toth, Y. Jwa, and S. Moafipoor (2006c). Seamless and Reliable Personal Navigator. Proceedings of the ION Technical Meeting, January 18-20, Monterey, CA, CD ROM, pp. 597-603.

Grejner-Brzezinska, D.A., C.K. Toth, and S. Moafipoor, (2007a). Adaptive Knowledge-based System for Personal Navigation in GPS-Denied Environments, Proceedings of the ION National Technical Meeting, January 22-24, San Diego, CA, CD ROM, pp.517-521.

Grejner-Brzezinska, D.A., C.K. Toth, S. Moafipoor, (2007b). Pedestrian Tracking and Navigation Using Adaptive Knowledge System Based on Neural Networks and Fuzzy Logic, Journal of Applied Geodesy, Vol. 1, No. 3, pp. 111-123.

Moafipoor, S., D.A. Grejner-Brzezinska and C.K., Toth, (2008). Multi-Sensor Personal Navigator Supported by Adaptive Knowledge Based System: Performance Assessment, Proceeding of the IEEE/ION PLANS 2008 Meeting, May 5-8, Monterey, California, CD ROM.

Moafipoor, S., D.A Grejner-Brzezinska, and C.K. Toth, (2007a). A

Fuzzy Dead Reckoning Algorithm for a Personal Navigator, Proceedings of the ION GNSS Meeting, September 2007, Fort Worth, Texas, USA, CD-ROM, 2007, pp.48-59, also in Navigation, in press.

Moafipoor, S., D.A Grejner-Brzezinska and C.K. Toth, (2007b). Adaptive Calibration of a Magnetometer Compass for a Personal Navigation System, Proceedings of the International Global Navigation Satellite Systems Society (IGNSS) Symposium, University of New South Wales, December 2007, Sydney, Australia, CD ROM.

Toth, C.K., D.A. Grejner-Brzezinska, S. Moafipoor, (2007). Pedestrian Tracking and Navigation using Neural Networks and Fuzzy Logic, Proceedings, IEEE International Symposium on Intelligent Signal Processing, Alcala De Henares, October 2007, Madrid, Spain, CD ROM, pp. 657-662.

Knowledge-Based System for Deformation Analysis and Interpretation

Miloš Miljanović

Knowledge-Based Systems Group
Institute for Information Systems
Vienna University of Technology
email: milos@kr.tuwien.ac.at

Abstract

A great deal of effort has been put into eliciting knowledge and reasoning strategies from engineering experts with the aim of building up a computer model of their expertise in order to assist engineers in their decision-making processes. The main challenge has been to develop computer support systems that induce changes and improvements to the way that engineers solve their everyday problems. In this paper, a specific knowledge-based system for the analysis and interpretation of deformations found in the construction of buildings is developed and evaluated. The system uses data gathered from the information, such as Interest Operators, and Regions of Interest which are extracted from images of facades. This information is then used to keep track on the deformations over a period of time, and to make forecasts of risks and potential hazardous problems.

Keywords: knowledge-base systems, interest operators, deformation analysis

1 Introduction

The ongoing advances in information and communication technologies are changing many aspects of engineering problem-solving as well as the decision-making processes of experts. The end users imposing new approaches to civil engineering problems through the use of variety of innovative tools, which are now becoming widely available.

Towards the end of the eightes, intense activity in KBS technology led to the development of a variety of systems in a number of engineering fields. The main challenge of KBS applications is to elicit knowledge and reasoning skills from engineering experts, so that a computer model of such expertise can be developed to assisting engineers in a variety of tasks.

In this paper, a knowledge base system for the interpretation of rigid body movements during the construction of a building is developed. The main information being used are the so-called Interest-Operators (IOPs), which are used to describe striking points in an image. These points are calculated from a 2D image of a facade, but their 3D equivalent is used in order to deal with the deformations. The purpose of the KBS is to identify the movements (if present), monitor their behaviour in order to detect possible future hazards.

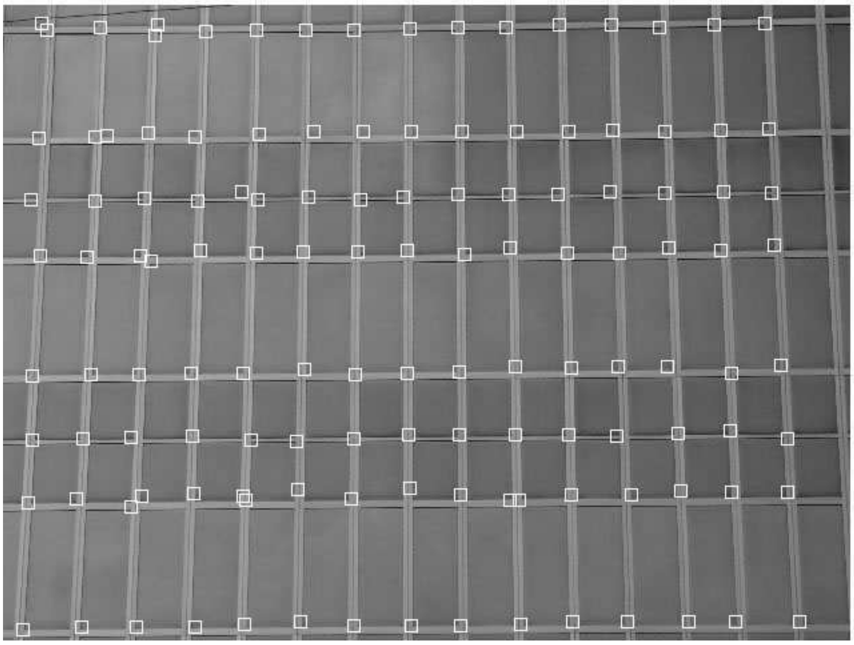


Figure 1: Example of Harris operator

2 Interest Operators and Regions of Interest

Interest operators are used in variety of computer vision applications such as motion tracking and object recognition. Ideally an interest operator finds salient points within an image that contain information that is strongly invariant to change. These interest points can be tracked across images or used to categorize them. A well known and widely used interest operator proposed in [2] also known as a Harris corner detector, is described. For every pixel p

in an image I we calculate

$$Z(p) = \begin{bmatrix} \sum_{p \in W} \frac{\delta^2 I}{\delta x^2} & \sum_{p \in W} \frac{\delta I}{\delta x}\frac{\delta I}{\delta y} \\ \sum_{p \in W} \frac{\delta I}{\delta x}\frac{\delta I}{\delta y} & \sum_{p \in W} \frac{\delta^2 I}{\delta y^2} \end{bmatrix} \quad (1)$$

where W is a window around p and $\frac{\delta I}{\delta x}$ and $\frac{\delta I}{\delta y}$ are the partial derivatives of the image in the horizontal and vertical directions. The eigenvalues λ_1 and λ_2 of $Z(p)$ are calculated and the point is considered valid if

$$min(\lambda_1, \lambda_2) > \lambda \quad (2)$$

for some predefined threshold λ.

The Foerstner interest operator [1], which is very similair to the popular Harris operator, is an example of a gradient-based grey-scale corner detection algorithm. It works by measuring the curvature of an edge as it passes through a given image neighbourhood with the strength of a corner dependent on the magnitude of the edge and rate of change in edge direction. The Foerstner operator defines 5 criteria for detecting corners and interest points [1]:

- Distinctness - corners should be different from neighbouring points in all directions.
- Invariance - the detection of corners should be unaffected by variations in viewing geometry and radiometry.
- Stability - the same corner should appear in other images.
- Seldomness - an interest point (or corner) should be interesting, i.e. not a repeating distinct point.
- Interpretability - the selection principle should permit quantitative intepretation.

After the IOPs have been extracted from the image, their 3D equivalents are also calculated. If there is a match between two subsequent images, then the corresponding points are being used as a reference point for the KBS. The 3D coordinates of the operators are described using two vectors, one for rotation, and the other for translation.

Once the interest operators have been calculated, they are combined to form a region of interest (ROI). These regions need to be sorted first because they have been inserted in the database without any particular order. To do this, a simple rule is being asserted to the system which finds two adjacent points (either vertically or horizontally). Once the sorting of ROIs is done, the system is then presented with a table of ROIs, with the first row having ROIs which are all located at the top of the observed object. This will serve as a good reference point to the KBS, so that consistency checks can be done easily.

3 Consistency Checks

The system is fed by a series of ROI, which hold the information such as the (x, y) coordinates of the ROI, together with their 3D equivalents (two vectors describing the translation (tx, ty, tz) and rotation (α, β, γ):

```
(deftemplate ROI
(slot id (type NUMBER))
(slot WA_x (type NUMBER)) (slot WA_y (type NUMBER))
(slot tx) (slot ty) (slot tz)
(slot alpha) (slot beta) (slot gamma))
```

Before any interpretation can be done, the system needs to check whether the ROIs differ from one another. Some of the consistency checks are rather simple, for example, if one region of interest is moving in one direction, then the adjacent ROIs need to be moving in the same direction, and at the same magnitude. If this is not the case, the system will inform the user about that ROI, so that it can be re-calculated (or ignored).

The only type of translation alowed is the translation along the z-axis. This is because the object being observed cannot move to the left, right, or to and away from the camera. Thus, the main concern for the KBS is to observe the values of the tz parameter. If there is a translation along the z-axis, then it is possible that there is also a translation at the x or y axis as well (which would mean that the ROI is moving diagonally). However, a lonely translation at the x and y axis is not possible.

As for the rotations, a negative rotation cannot be adjacent next to a positive, and vice-versa. A combination of a couple of rotations is possible, but again, if such a rotation is present, then all the adjacent ROIs need to be rotating in the same direction, at a similar magnitude. In total, 40 consistency checks are performed before the interpretation.

4 Deformations

The KBS deals with a simple, yet important deformation, which is present in building construction. A settlement can occur, if for example, one of the underground stations is being built, or if the soil underneath the building is moist. When this happens, one of the columns will have a negative tz parameter, and the adjacent columns will be rotating left, and right. The first thing the KBS needs to do is locate the position of the ROI which has a negative parameter tz and then check that all the ROIs which are located on the left have a positive rotation, whereas those which are on the right have a negative

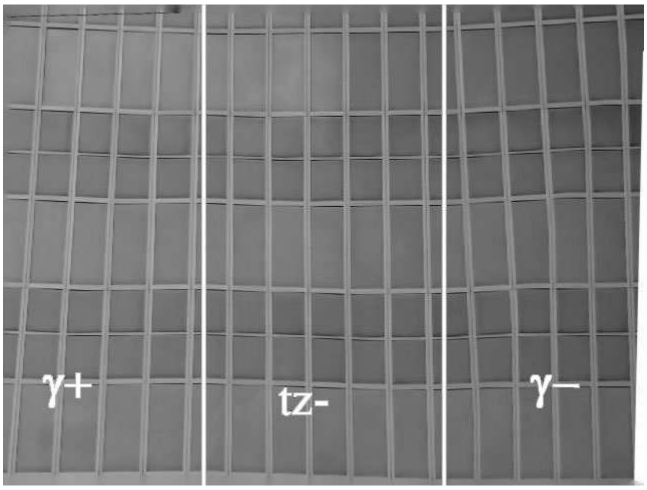


Figure 2: Example of a vertical settlement

rotation. The rotations must be of the same origin, rotating along the same axis. If that is the case, then the system will notify the user that a settlement has occured. Needless to say, to detect a settlement the system needs to perform additional consistency checks, such as the compatibility of the rotations with respect to the negative translation along the z-axis. For example, if there is only a negative translation along the z-axis present (without the x and y), then the adjacent columns need to have a rotation along the z-axis(γ parameter needs to be posive or negative). On the other hand, a z/x translation (diagonal movement) will have γ/α rotation present. Any other combination will result in a warning.

The system will also locate cracks in the wall, and notify the user. A crack occurs, if there is a column of ROIs with all parameters set to zero, but with deformations present at the adjacent columns. Similar to the settlement problem, the system has to make sure that the adjacent columns have the same type of deformation present on each side.

Another possible deformation is a diagonal settlement, that is, one half of the object (split diagonally) is moving to the left, or right. This is the only case, where translation along the x-axis is allowed. Such a settlement can happen if there is something wrong with the soil near the corner of the building. In such a case, the upper part of the building will move slightly left, or right.

Once the system has performed consistency checks, and located possible settlements, or cracks, it will do a final check on the parameter values. If there has been a considerate change in the magnitude of the translation/rotation parameters, the system will alert the user about a potential hazard which could

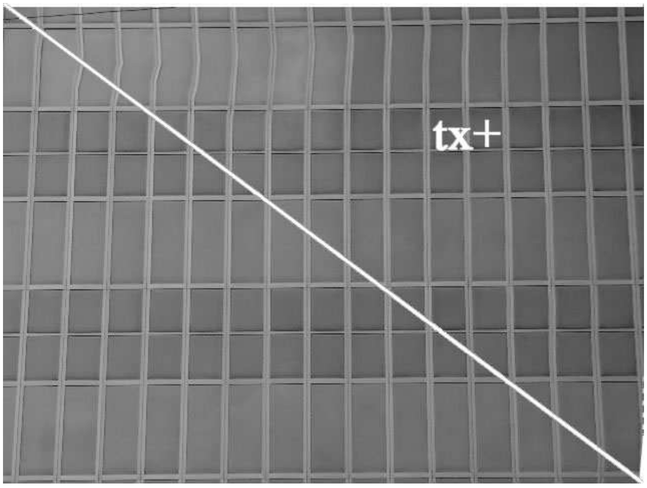


Figure 3: Example of a diagonal settlement

happen in the near future. An example of such a rule is shown below:

```
(defrule check-tz-magnitude

(adjacent (id1 ?id1) (id2 ?id2))
(or (and (or (ROI (id ?id1) (tz "sg"))
 (ROI (id ?id2) (tz "g")))
     (or (ROI (id ?id1) (tz "k"))
         (ROI (id ?id2) (tz "sk"))))
(and (or (ROI (id ?id1) (tz "nsg"))
 (ROI (id ?id2) (tz "ng")))
     (or (ROI (id ?id1) (tz "nk"))
 (ROI (id ?id2) (tz "nsk")))))
=>
(printout t "Magnitude of translation along the
 z-axis changed drastically from " ?id1 " to " ?id2 crlf)

)
```

5 Experiments

The experiments were carried out using a benchmark of 50 images [3]. Deformations have been generated artificially, due to the lack of real data. Once the ROIs have been calculated, together with their appropriate translation and rotation parameters, the KBS was tested with a series of images. 10

out of 10 settlements have been detected successfuly, together with 5 cracks, and 3 different diagonal settlements. The system has not been tested in real life, mainly due to the fact that it is hard to keep track of a building with potential deformations.

6 Conclusion and Future work

A simple KBS for detecting hazardous events, and deformations has been developed. The system can detect settlements, and cracks, and will notify the user about potential risks. The KBS can however operate only on two epochs, and has not been extended to work on time series. If more epochs are included then the system will give more accurate results, and would be able to give better predictions. In addition to this, the system although able to detect deformations does not give a lot of feedback to the user on why the deformation is happening. To do this, additional information is required, such as the material which is being used, information about the soil underneath the building, etc. The system could be further improved by using the Finite Element Method (FEM), which can use these information to give a more accurate prediction about potential hazards.

References

[1] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Workshop, Interlaken*, pages 281–305, 1987.

[2] C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[3] `http://info.tuwien.ac.at/ingeo/research/FWF_Projekt_CogniMess/FdbV2006/`.

What business to make with AI-techniques for EG-problems?

Experiences and thoughts of a company that believes and doubts and believes and . . .

Klaus Chmelina

Geodata ZT GmbH
HütteldorferstraSSe 85
A-1150 Vienna, Austria
email: klaus.chmelina@geodata.at

Abstract

The lecture first presents the Geodata group as an Austrian company mainly busy in the field of tunnel surveying and monitoring, an exciting and highly specialised working field of engineering geodesy. To introduce this field briefly and the current state of the art, some recent projects and used systems are shown. In a next chapter, Geodata's R&D-activities are described, of which a considerable number deals with the development and implementation of AI-methods as they are recognized as upcoming, future business possibility. Finally, it is tried to frankly tell the experiences and lessons learned with/in AI-research projects, especially those that have to do with a very special kind of beings, the so called "experts". The experiences are hoped to motivate and maybe also amuse researchers that, who knows, somewhen might come into the situation of having to sell their development to the real business world.

Keywords: Tunnel Surveying, Geotechnical Monitoring, Decision Support System.

1 Introduction

Geodata Group

The Geodata group (www.geodata.com) is an Austrian SME (Small to Medium Enterprise) found in 1987 as a spin-off of the University of Leoben

(Styria). Since the beginning the company has continuously grown and today has about 150 employees working in several national (Leoben = headoffice, Graz, Vienna, Wels) and foreign branches (Germany, Croatia, Hungary, Greece, Chile).

Beside other business fields like conventional software and measuring instrument development, services for GIS, cadastre and technical surveying, etc. the Geodata group provides AI-solutions (in the sense of All Inclusive). These solutions constitute geodetic and geotechnical systems and services for all needed tasks in underground construction and subsurface mining projects. Typical projects are road and railway tunnels, subway tunnels, hydro power and utility tunnels, investigation galleries or shafts.

In this field, we can offer all what is needed:

- geodetic and geotechnical experts performing all necessary measuring works and installations on-site,
- geodetic and geotechnical measuring equipment and systems for all required machine guidance, mapping, monitoring and surveying operations on-site and
- geodetic and geotechnical software applications for all needed data management, data processing, visualization, analysis and further services like alarming and reporting.

Some Recent Projects

To understand our typical working environment and scope of works, some recent projects are shown below (fig. 1 - 4).

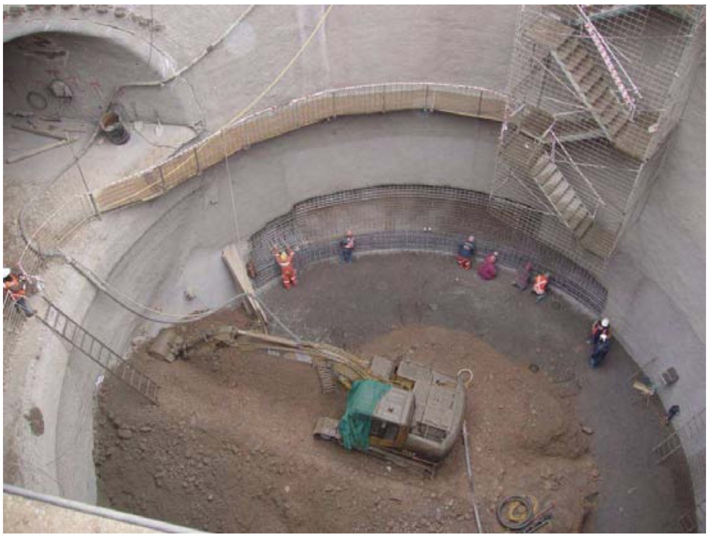


Figure 1: Shafts and tunnels are built for the **Metro Santiago (Chile)** applying the New Austrian Tunneling Method (NATM).

The shaft and tunnel survey include the continuously performed setting out (e.g. with motor laser systems), network control (e.g. gyro measurements, plumbing), as-built documentation (e.g. profile control with 3d-laserscanners) and 3d-displacement monitoring (e.g. with robotic totalstations) during the construction phase.

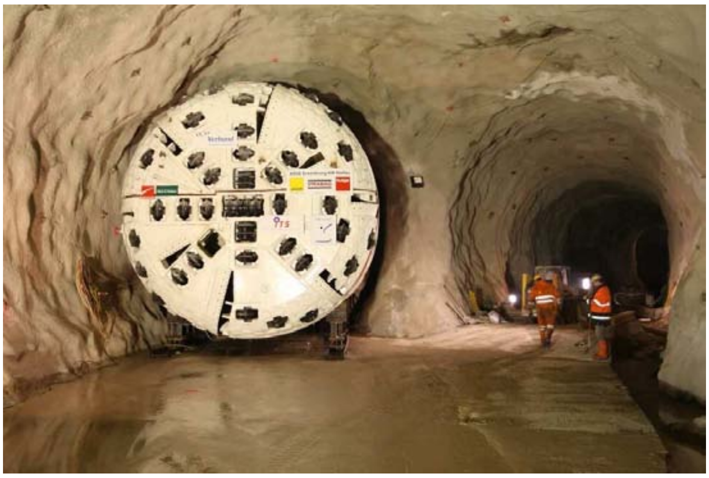


Figure 2: A 5.6 km headrace tunnel extending the existing **hydro power station Hieflau (Austria)** is excavated by a double-shield tunnel boring machine (TBM).

In the project a TBM guidance system has been supplied, installed and serviced during TBM-excavation. Additionally, the complete heading survey and monitoring has been carried out.

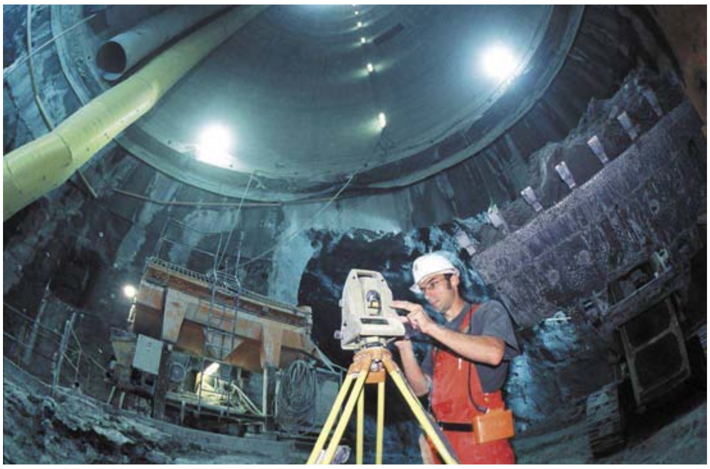


Figure 3: Huge caverns and tunnels have been built for the **CERN LHC Project UX15 (France, Switzerland)**.

A complex geotechnical monitoring network consisting of several hundred different automatic sensors has been installed and maintained during the construction period; various geodetic deformation measurements were carried out manually.

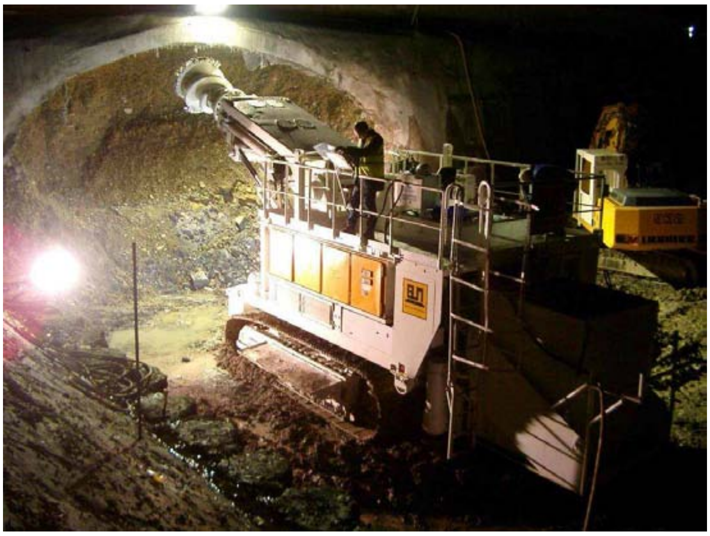


Figure 4: A tunnel is excavated by a roadheader machine at the **Achraintunnel project (Austria)**.

A roadheader guidance system using robotic total-stations has been installed allowing the machine driver to operate his machine even without having sight to the tunnel face.

Some Selected Systems

To reflect the current state of the art of tunnel surveying, some of the systems routinely applied on todayŠs tunnel sites are presented in the figures below (fig. 5 - 7). They illustrate that a tunnel under construction already has become a high-tech world of remote-controlled machines.

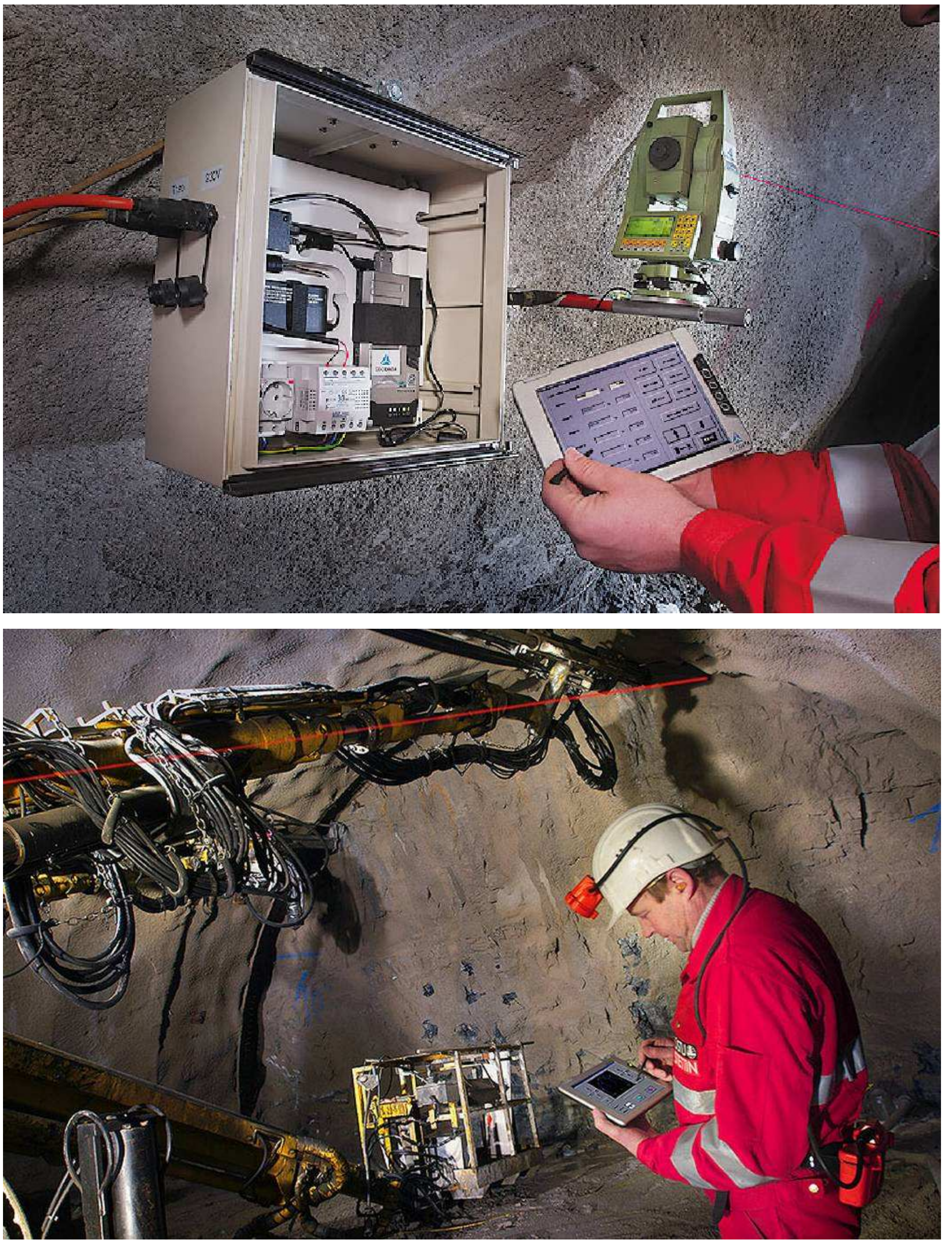


Figure 5: A boring jumbo (bottom) is guided by use of a motor laser system (up) that is remote-controlled by a tunnel worker using a WLAN-touchscreen (bottom). The system serves for carrying out all setting out tasks needed close at the tunnel face. A surveyor almost can not be found there anymore.

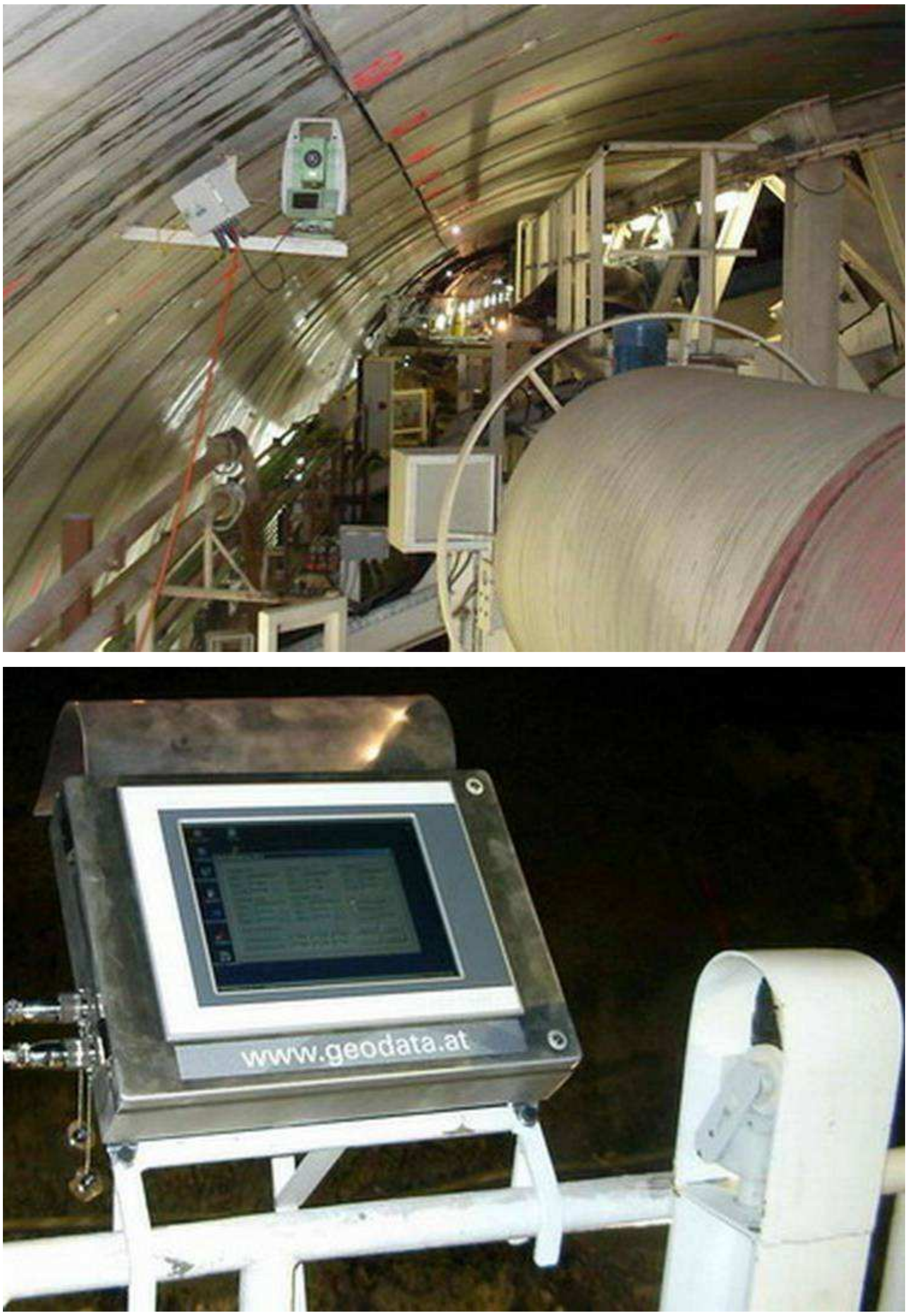


Figure 6: A tunnel boring machine is steered by use of a TBM guidance system that is continuously tracking the cutter head position and orientation (up) and transferring the data to the control computer (bottom).

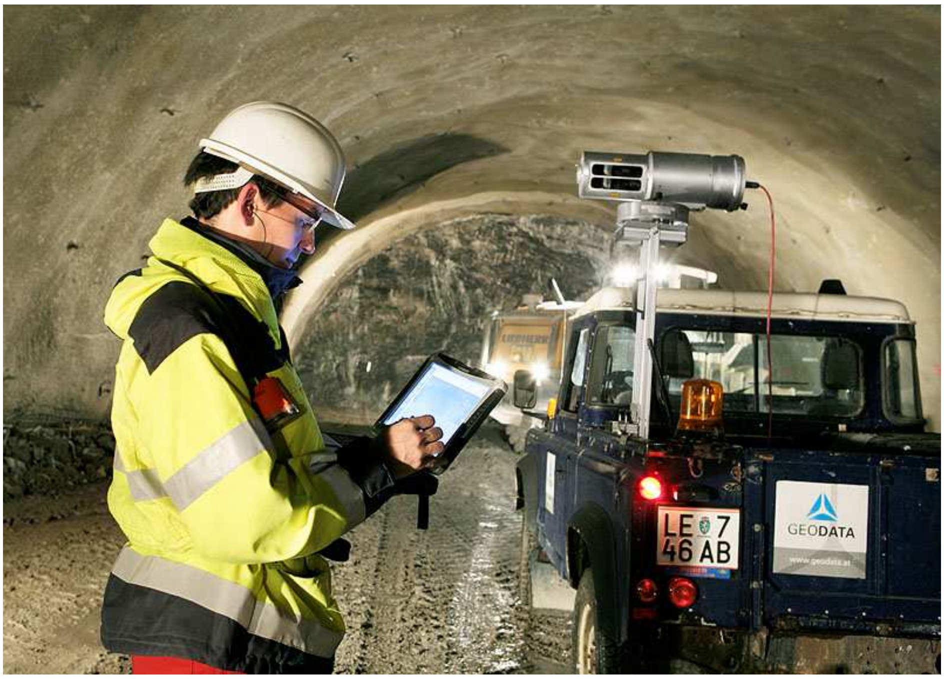


Figure 7: A pick-up based 3d-laserscanner system scans and images the tunnel wall and face for checking the geometry against design, for geological document-ation and for tunnel lining inspection.

The system is remote-controlled by a tunnel worker and tuned up to perform all measurements fully automatical and as quick as possible (< 3 min. per station). All scan data is georeferenced and processed immediately.

2 AI Research Activities

Seeing above EG-systems, it is not hard to imagine that tunneling will be more and more remote-controlled, automized and robotized in future. The next generation of machines and robots (Figure 8 up) and their control units (Figure 8 bottom) is already under development. New tools will soon be available performing operations such as drilling, excavating, shotcreting, concreting, inspection and repair, etc. in an impressive, partly autonomous way. And for these operations they will apply **AI-methods** (but now definitely in the sense of Artificial Intelligence).

The European Construction Technology Platform (ECTP) envisages in its Strategic Research Agenda (SRA) that, already in 20 years, tunnels will be constructed without workers anymore needed underground and, besides, many times faster, cheaper and safer than possible today.

It is further told in this document that a significant role to achieve these ambitious goals is given to AI-technologies, namely expert systems and knowledge-based systems. They will control all relevant underground construction processes, including machine and robot operations. They are given the key role to achieve an efficient, economic and safe future European tunnel construction and that's why almost every current research project on European level makes use of AI-terminology in its proposal. So everyone wants to make expert systems now – either honestly or just to get funding from the European Commission. The term AI is used inflationary in underground research projects today.

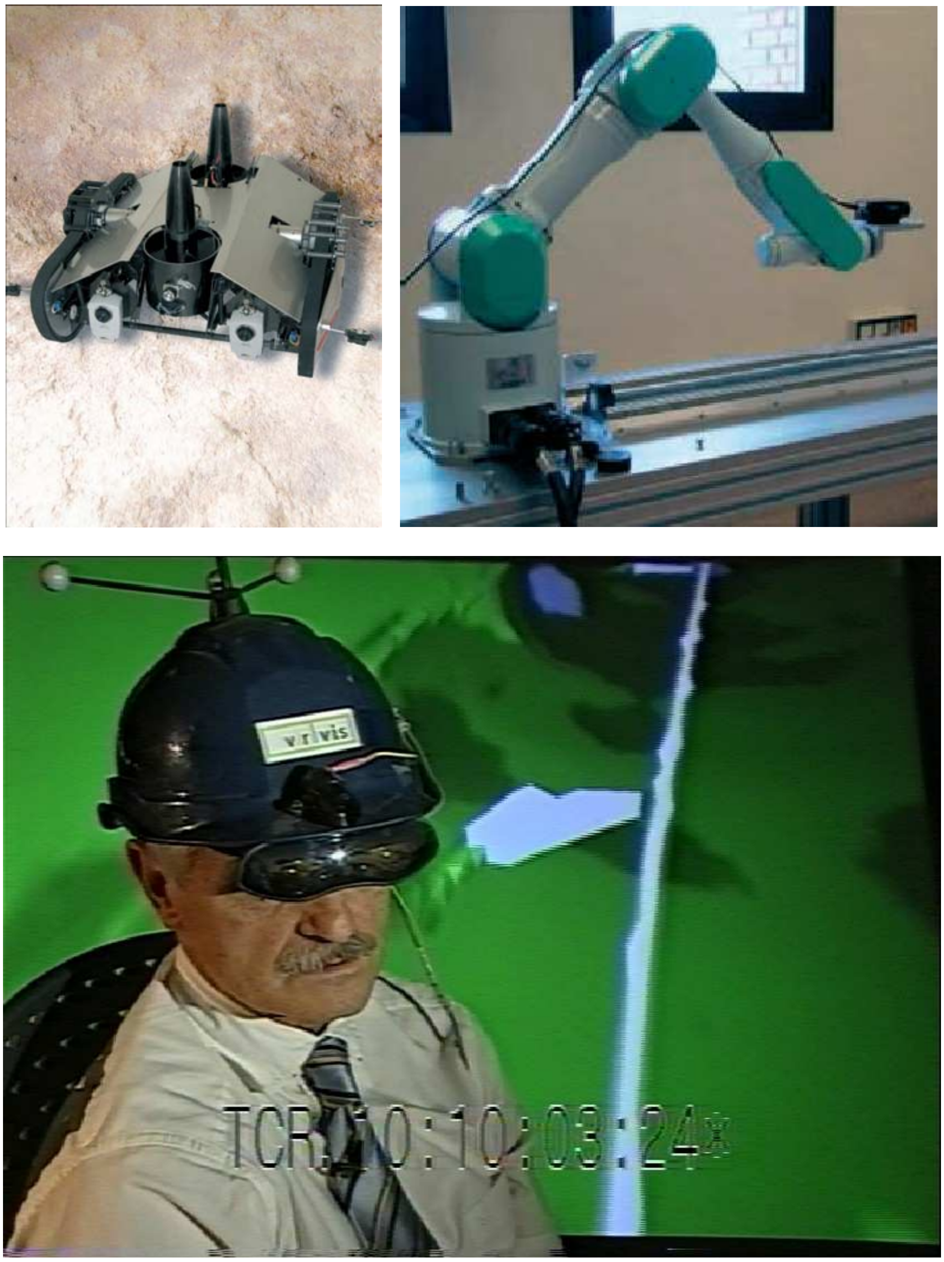
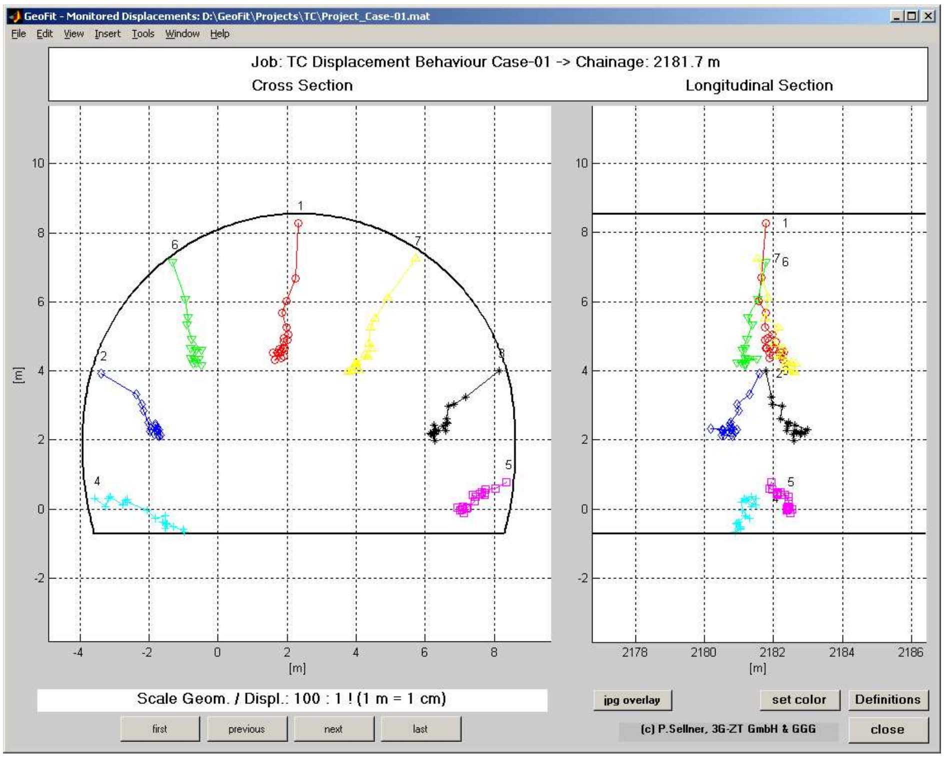


Figure 8: Selection of next generation tunnel robots; left: wall-climbing inspection robot DRAGARITA; right: tunnel crack repair robot IRIS; bottom: Head mounted display for Virtual Reality-assisted robot control.

As we can not lay back, we are also partner of some AI-research projects. The most relevant is the project TUNCONSTRUCT (www.tunconstruct.org) where the following AI-systems are developed:

A decision support system for cyclic tunneling

This expert system allows to evaluate the 3d-displacements periodically measured in a tunnel under construction. The system therefore aggregates various, heterogeneous data sets from geology, construction, support, etc. together with the 3d-displacements to a so called displacement behaviour case, see Figure 9. The case is compared with cases stored in an MS SQL server database (= knowledge based library). The degree of similarity of two cases is determined by use of a graphical method (Bresenham's line algorithm) that finds the case from the database that (graphically) best-matches the encountered case on site and presents it to the user (= a geotechnical expert). By this way the geotechnical interpretation and decision-finding whether the case is normal or abnormal is supported. In the database historical cases with full information on their detailed geotechnical interpretation and recommendations of what to do are stored. The development of the knowledge-based library and the case-matching method is done in a close cooperation between geotechnical experts, geodetic experts and software developers.



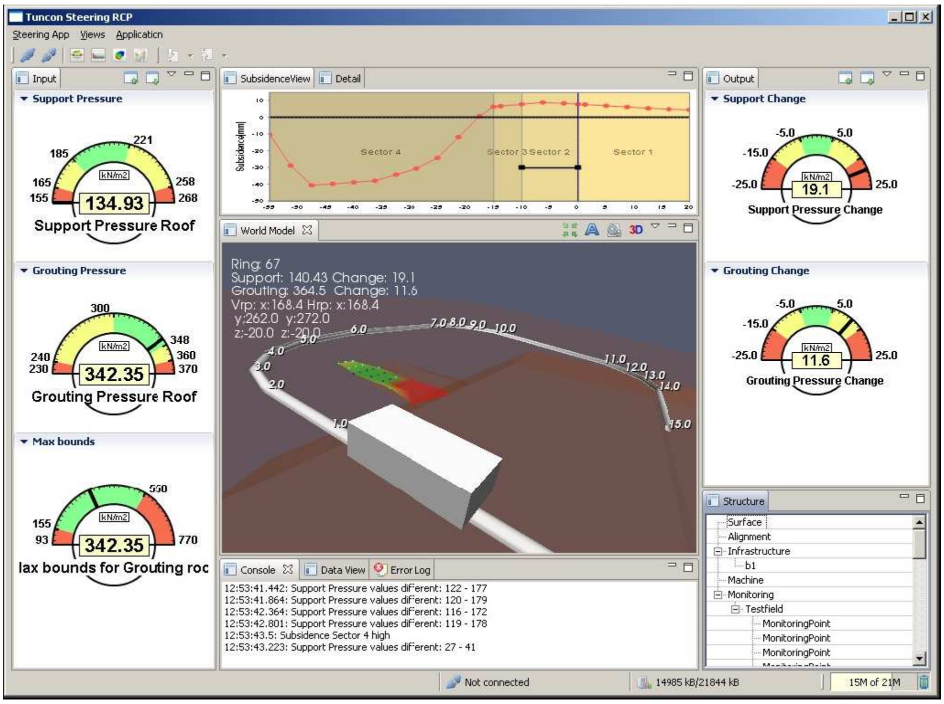
GeoFit - Monitored Displacements: D:\GeoFit\Projects\TC\Project_Case-01.mat

Job: TC Displacement Behaviour Case-01 -> Chainage: 2181.7 m

Scale Geom. / Displ.: 100 : 1 ! (1 m = 1 cm)

(c) P.Sellner, 3G-ZT GmbH & GGG

Figure 9: Graphical representation of the displacements of a single displacement behaviour case stored in the knowledge-based library.



```
12:53:41.442: Support Pressure values diFerent: 122 - 177
12:53:41.864: Support Pressure values diFerent: 120 - 179
12:53:42.364: Support Pressure values diFerent: 116 - 172
12:53:42.801: Support Pressure values diFerent: 119 - 178
12:53:43.5: Subsidence Sector 4 high
12:53:43.223: Support Pressure values diFerent: 27 - 41
```

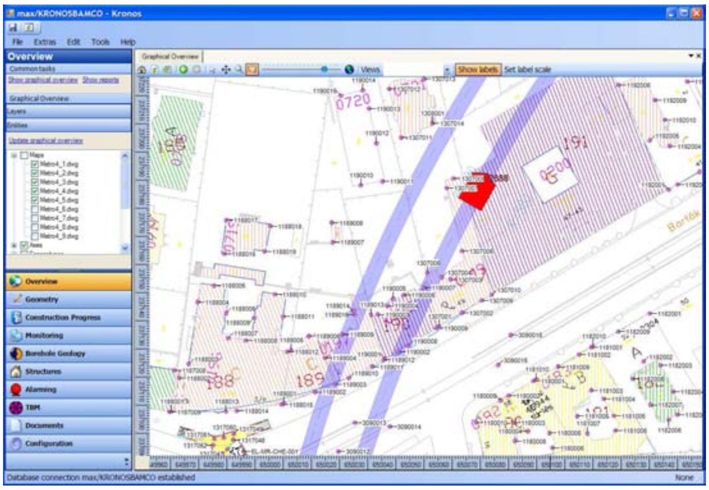
Figure 10: GUI of steering software showing machine data (left side) and recommendations for the TBM driver (right side).

An automated TBM steering software

This expert system is being developed to help a TBM crew to optimally steer the TBM in order to achieve a safe excavation and to avoid unreasonable settlements. The system permanently collects and reasons over machine data (pressures, power consumption, forces), settlements and guidance data (TBM-position and orientation) and concludes instructions (at this stage: recommendations for the driver) for changing specific machine control parameters (e.g. support pressure, grouting pressure), see Figure 10. The system is based on a fuzzy-based backward chaining rule engine wherein the formulation of the up to now implemented 35 fuzzy rules has been accomplished by tunneling experts that benefit from their long lasting experience in the realisation of large tunneling projects. For the development of the system, the Java Rule Engine “Jess” has been used in conjunction with the Java Fuzzy framework “FuzzyJ”.

A Data quality check service

The objective of this development is to automatically check the tons of geodetic data daily produced by our many company surveyors working on construction sites all over the world. It assures a company data quality standard and the meeting of project-specific requirements (e.g. monitoring intervals). Besides, it helps to avoid severe mistakes due to lack of knowledge, stress, fatigue, etc. The expert system retrieves a calculation protocol that is automatically produced by our locally running geodetic data processing software and transferred via ftp-file upload into pre-configured folders of the central server of the Geodata headoffice.



Database connection max/KRONOSBAMCO established

Figure 11: GUI of the information system showing project plan with surveyed objects like monitoring points, buildings, a TBM, etc. The “Alarming” button (left side) leads to the alarming center where the alarming rules can be configured.

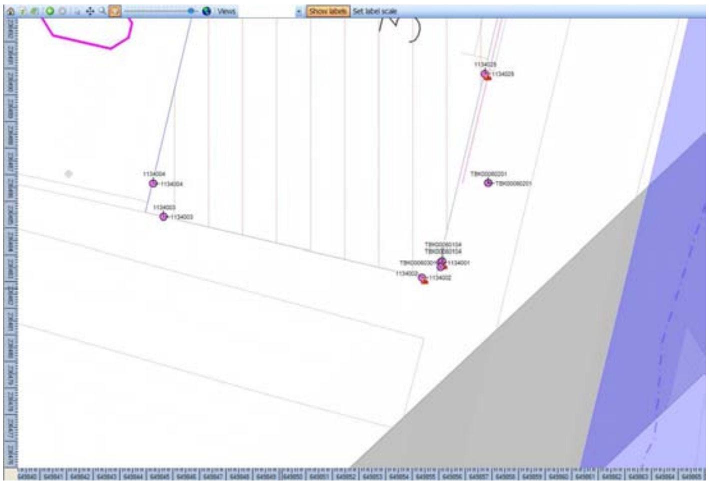


Figure 12: Detail of project plan showing two surveyed monitoring points indicating an alarm; the points have not been measured according to project requirements.

There, a Windows service continuously checks for the existence of new protocols to be imported into the system’s MS SQL Server database where heuristic rules become active and automatically check the protocol content and produce and return a corresponding message to the data producer/sender via e-mail without any server side user interaction. The data quality checking software is an integrated module of an information system enabling the company data quality manager to access all related data quality information in detail. He can manage and display project data and, in particular, all the data quality alarms.

3 Experiences and Lessons Learned

In the year 2000 we have started our first research steps in the AIEG-domain. Since this time manifold experiences have been made in practically all phases of AI-system development. We carried out extensive knowledge acquisition projects making expert interviews, expert observations in their application field, anonymous opinion polls and literature study. Knowledge has been collected, investigated, classified and evaluated by means of more or less objective statistical methods as well as by empirical methods. Different forms of knowledge representations and system designs have been tried and finally some system prototypes have been developed and tested. Of course, we have also eagerly watched the results reported by others in their diploma works, dissertations, etc. and the developments of our competitors. Coming to the most challenging and interesting parts for us (= implementation and business), we must admit that we have not had big success yet. It seems that in our innovative but nevertheless conservative application field, time has not yet come for AI-solutions. However, from our experiences so far the following wisdom has grown (which not necessarily is transferable to other application fields):

Watch the emotional side

Before it comes to start any AI-project, it is advisable to pay attention to fears and other negative emotions of relevant experts. This is especially important for AI-projects as they need the cooperation of experts by nature. Unfortunately, experts often are personally affected by a potential later system implementation (e.g. they can fear to loose control, influence, competence, independence). So their emotions play an important role in making the development a success and, last of all, they will influence or even decide on the later acceptance and implementation of the system. This is why we can hardly trust investigations like feasibility studies where the acceptance, suitability, applicability, etc. of an AI-system is somehow technically assessed and where we only listen to reason. Unfortunately, the validity of such studies is rather poor as the more decisive factor lies in the expert's emotions that are non-verbal communicated, if communicated at all. And if we don't have the experts on our side, we will not be able to implement anything. In our Geodata AI-projects we have not found experts that definitely refused cooperation or rejected ideas as non-applicable, silly or useless; they all were kind and smart. But on the other hand they could hardly be motivated to cooperate over a longer period of time. And enthusiasm had not been found at all. Finally, they all, in most cases, ended up in a wait-and-see attitude. Such kind of reserved attitude is already a sign of resistance, although not perceived consciously. And among experts this attitude is infectious. To cope with this problem, a good project preparation is recommended where a soft-skilled person (the later knowledge engineer?) could try to find out in advance and without too much engagement what the experts really feel (not only think and say) about the idea and also the consequences of its future implementation. Hard stuff to achieve this – and dangerous!

Don't go like a bull at a gate

In our first attempts to develop AI-systems, we focused on our own ideas of how the AI-solution for a problem could be rather than on ideas from outside. At least for making the concept, we really thought to be the experts ourselves. Later, as time had come to find, ask and involve the needed real experts, we used to present the then almost ready "that's-the-way-we-want-to-do-it" concept. This always caused the same annoying reaction that can be better called non-feedback as the experts didn't know what to answer. The reason has not been a bad concept, not at all. The reason has been that ready-made concepts of a certain complexity that can not be overlooked immediately (and this means within a few seconds) are too risky and difficult to be psychologically treated/accepted – and then there is emotional rejection. The permanent desire of a busy expert is to reduce complexity of his work to keep life simple and not to suddenly deal with new things that are complex.

This effect is accompanied by the phenomenon that experts (humans in general) offer a strong resistance to changes. They prefer to continue going the old way instead of following any new one, even if there is proof that the new one is better.

So we recommend to first discuss the problems of an expert and not the solution for it when meeting the first times. Confront experts carefully with new ideas and let them see only one sheet of paper at the very first meeting, maximum.

Establish project identification

If your AI-project then has started, establish and maintain identification of the project team members. It is known to be the best thing to happen in a project and is especially good for AI-projects. It happens automatically if you consider something your own baby and this works if everyone is allowed to be creative and to contribute from the very beginning. In this context, we can say that it never worked well when regarding and involving experts as mere knowledge providers. This kind of involvement even is poison for identification and their contributions will usually be low-level.

Our experience is that, if already the idea is delivered by the expert, the rest of the job runs smoothly and the expert even does not care of details anymore. He keeps his child in his heart, forever, and stays in good mood which is positive for the project. Establishing and maintaining team-spirit, project identification, motivation, communication, etc. is essential in AI-projects – but a big effort. It requires a professional project manager that especially is competent on the soft skill side.

We have heard that cooking, singing and dancing are the latest trends to support project identification – we haven't tried it out yet but consider it seriously.

Make an optimal team

To be successful in developing an AI-system, an optimal project team must be formed. In our company, we use to define the following roles during the design and development phase:

- an overwhelmingly soft-skilled project manager,
- a knowledge engineer (system design, knowledge acquisition, ...),
- a leading domain expert (invited and motivated to contribute to everything),
- further experts (knowledge providers, consultants, advisors) and
- (software) developers.

It is recognized that a good team consists of approx. seven persons. Seven is a holy number (7 wonders of the world, creation of world took 7 days, Rome has 7 hills, 7 mortal sins, 7 cardinal virtues, Snow White and the 7 dwarfs, etc.). In fact, an AI-project requires intensive communication which gets too complicated if too many persons are on the table. If you try to manage a higher number of persons by dividing them into smaller groups that concentrate on smaller subtasks you will have an integration problem of subtask-results at the end; an immanent problem of all so called “integrated projects” which only in very seldom cases can be solved satisfyingly. On the other hand, if you just form a tiny team, of let’s say 3 persons, you run the risk of developing something too tiny, exclusive and narrow; you somehow fail a critical mass.

Teaming up also means to select the right persons for the roles. Here, if there is a choice at all, it another time is not only the technical qualification to turn the scales. We recommend again to watch the soft skills. One such soft skill is the social ability to stay in the role initially assigned and not to make the mistake of playing an other ones role. In this respect, a popular but critical mistake is to slip into the expert role. This makes troubles for sure.

Dealing with many experts in a team is a specific problem. When it comes to intensive work, we prefer to deal with only one that we call the leading domain expert. He is selected by his particular ability and possibility to provide what is needed in the project and not so much by his reputation, position or similar parameters. The other experts are invited to contribute as consultants and advisors. It is quite normal that two experts have different opinions, but it makes troubles if they start to dispute on that in a project meeting. In an AI-project you should be very consensus-oriented.

4 Conclusion

In the presented application field of EG "tunnel surveying and monitoring", some AI-solutions can already be found. In most cases they constitute expert systems that are internally exploited and not (yet) available on the market. This means that they are used by the companies that developed them to improve some of their own services. So far, a business in the way of selling AI-solutions (e.g. licencing AI-software applications, selling knowledge bases) in a considerable number is not yet happening. The presentation informed on three exemplary AI-systems that reason over EG-data and are either still in development stage (chapter 3.1 and 3.2) or already installed and running in the application field (chapter 3.3). They show that AI-methods have returned as a new trend and are considered to play an important role in the future. They will be applied for a variety of challenging, new tasks in data analysis, data interpretation, process control and machine control where conventional (software) solutions will not provide the needed flexibility. The design and development of an AI-solution as an academic exercise is one chapter. But when it comes to real-world implementation where there must be dealt with experts, users, clients, etc. the real difficulties begin. Their management requires particular skills that, up to now, are not teached (or much better: trained) in any EG university lectures. If geodesists want to be successful in these fields in future, their syllabuses must be revised. The geodetic instruments and software systems used today are well established and the changing of their development paradigma from conventional to AI-style will not happen quickly. But we see that they suffer from a lack of "intelligence" in many ways which will make them inappropriate to cope with future challenges. And this will be the chance to dope them with Artificial Intelligence. What business to make with AI-techniques for EG-problems remains an exciting but open question. Congratulation to the idea of organising this first AIEG workshop. It is hoped to find continuation and further friends in future.

Case-Based Deformation Assessment
A Concept

Martin Lehmann & Alexander Reiterer

Research Group of Engineering Geodesy
Institute of Geodesy and Geophysics
Vienna University of Technology
email: martin.lehmann@tuwien.ac.at

Abstract

This paper presents a new deformation analysis system based on techniques originally developed in the area of Artificial Intelligence which shall be used for the task of deformation assessment and deformation interpretation. The presented procedure is based on data captured by image-based measurement systems as well as on terrestrial laser scanning devices. In this article we focus on specific key components and novel techniques that are under development, and briefly report on the current stage of the whole system.

Keywords: Deformation Analysis, Case-Based Reasoning, Knowledge-Based Systems.

1 Introduction

The number of objects in the field of civil engineering, mainly inner-city buildings, increases continuously and results in an increasing demand for highly accurate 3D representations and/or monitoring of these objects.

Deformation measurement enables the early detection of a damage, failure or an injury to the safe operation of an object in order to be able to react appropriately and in time. Several factors, like changes of ground water level, tidal phenomena or tectonic phenomena, can be the reason for deformation.

In the past well-trained "measurement experts" are required to operate such a deformation measurement system, mainly in the analysis and interpretation part. In order to make such systems easy to use even for non-experts, it can be extended by an appropriate decision system which supports the operator.

In this paper the concept of a deformation assessment component of such a system will be presented. The main task of this component is to transfer the information of a deformation analysis in an usable form for an automatic deformation interpretation. Therefor different approaches from the field of artificial intelligence will be used and joined to a case-based reasoning system.

2 Data acquisition

To perform a deformation measurement as mentioned, a new kind of Terrestrial-Laser-Scanner (TLS)- and image-based measurement system, developed at the *Institute of Geodesy and Geophysics* of *Vienna University of Technology* (research project: "Multi-Sensor Deformation Measurement System Supported by Knowledge-Based and Cognitive Vision Techniques"), is used (Reiterer et al., 2008). This system is currently in development stage and will be used for measuring, analyzing and interpreting deformations for the task of quality control in civil engineering.

For detection of object regions, which are interested and suitable for the subsequent point detection, object segmentation is needed. In most cases identifying predefined feature elements can help generate such a structure. Therefore the most important step for object structuring is to formulate a set of feature elements that represent the object in an adequate form. Our system design is focused, in this prototype stage, on monitoring of building facades, which are mainly represented by windows and doors. Therefore we can use these feature elements for structuring our objects.

For a robust object segmentation procedure it is recommended to use all available measurements. Therefore this subsystem is based both on image-based (IATS) and TLS data. This approach results in three different object segmentation procedures, which perform object structuring: by Histogram Median Filtering, by learning, and on the basis of captured TLS point clouds.

Different from that last method, the first two methods operate on images. While these procedures are currently geared towards facades, they will in future stages be extended to and/or complemented with segmentation procedures for other object types such as rock falls. Furthermore, the choice of a suitable method depending on object and scene conditions using a knowledge-based approach is envisaged. To this end, the realization of a more comprehensive knowledge-based object information system for supporting the structuring process is planned. More details about all three structuring methods can be found in (Reiterer et al., 2008).

The simplified measurement and analysis procedure, including the deformation assessment component is shown in Figure 1.

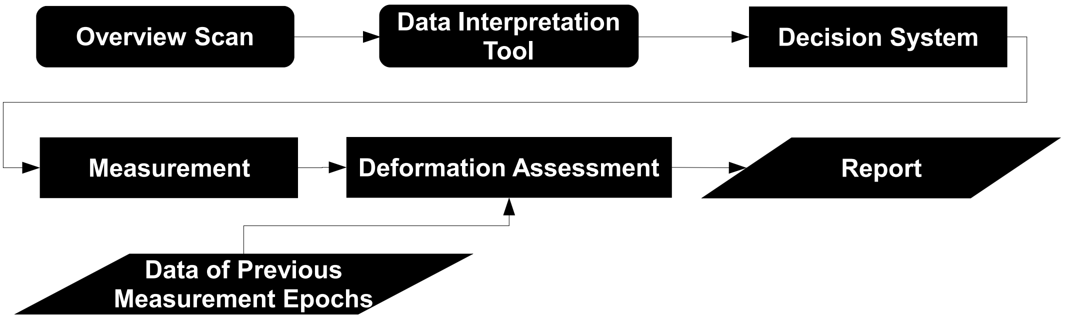


Figure 1: Simplified system architecture, (Lehmann et al., 2007).

Although this deformation assessment is mainly developed for the mentioned TLS- and image-based measurement system – an extension to other data acquisition systems is envisaged.

Using the detected object structure and the assigned ROIs, individual interest points (IPs) need to be detected. For the task of high precision point detection both sensor units can be used (IATS and TLS). In the following we will focus our description on the point detection by means of the IATS.

The internal camera of the IATS captures images covering the extracted ROIs (the internal camera is used instead of the WA-camera due to the larger image scale and in consequence the higher accuracy of image point detection). As a first step image pre-processing can be performed to transfer the image into a form which is better suited for the subsequent automated image point detection (this part is based on knowledge-based decision-making and has already been published – more details can be found in (Reiterer et al., 2008)).

The required conditions for such measurement systems are:

- ability to measure unsignalized points,
- ability to detect same points in different time epochs,
- ability to analyze the measurement data.

3 Deformation Assessment

In the past a deformation measurement system was completely operated by a measurement expert. Even though the processes of measuring the discrete points and analyzing a motion of the points are almost automated, the interpretation of the resulting data is a complex task even for an expert. The reasons for a deformation are mostly manifold and the acquisition and analysis of the deformation data requires information of different fields of knowledge (e.g. geophysics, geology, civil engineering).

A possible solution for such a deformation assessment would be the bundling of the required knowledge into a knowledge-based system. But as mentioned before the measurement is performed by an image-based measurement system. On that score we can only use information about the surface of the object.

The deformation assessment presented in the paper at hand is based on the idea of supporting the expert by former deformation cases. The case base includes the reasons for the former deformations.

The procedure of deformation assessment contains the following steps:

- generate deformation tiles,
- forming a deformation case,
- comparing cases.

In the following the seperate steps will described.

3.1 Generate deformation tiles

In the first stage previously measured points in the predefined/detected regions of interest (ROI) (see Figure 2) are analyzed, summarized and generalized to so called “deformation tiles”.

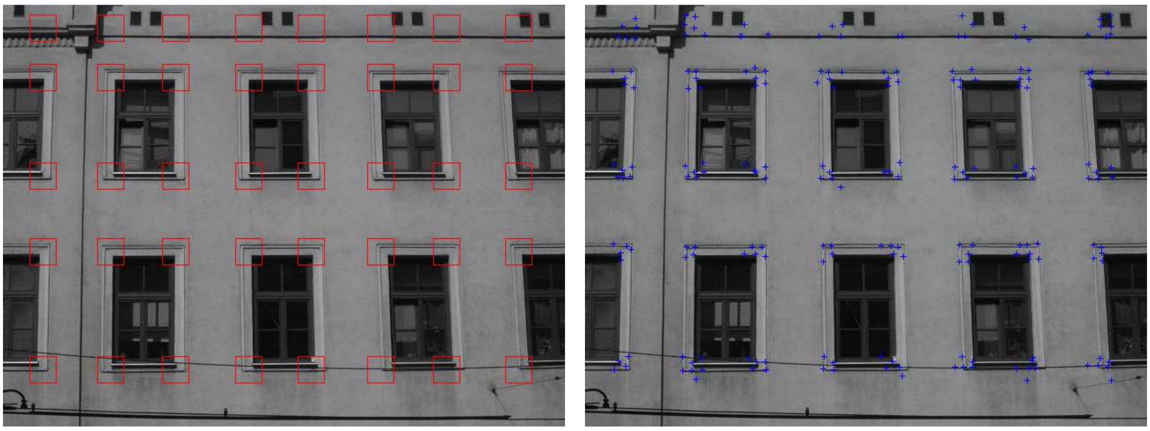


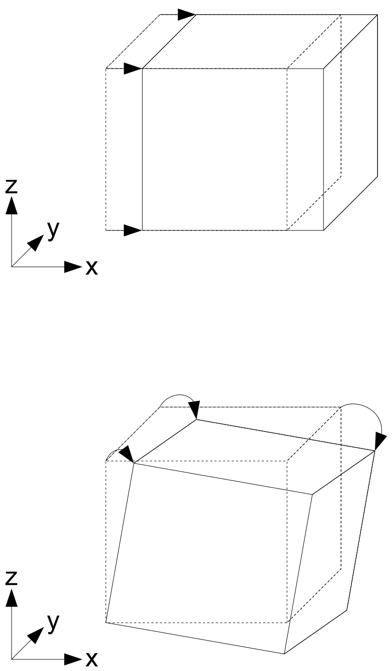
Figure 2: Defined regions of interest (left) and points measured inside these regions (right).

With the points measured in at least two epochs a classical deformation analysis is performed. The result of this analysis is the information about the motion of single points including a statistical secured statement.

The concept of the deformation assessment follows a local-to-global information integration strategy. Thatswhy a set of parameters describing the motion of this regions of interest is calculated, by using a similarity transformation with data of all points inside each ROI.

Afterwards the resulting parameter set of the ROI is fuzzified by a proper membership function. This reduces the complexity/variety of the parameter sets for the later process.

A simplified description for two prototypical motions of a ROI is shown in Figure 3. The first example shows a translation along the x-axis. The value ”1” in the ”zero”-row of each column except t_x means that the corresponding variable (α, β, etc.) has no significant value. The value ”1” in the medium-row of the t_x-column means that t_x has medium value, which is here then the overall fuzzy-value of the translation. The parameter in the second example (small rotation around the y-axis) can be interpreted in the same way. Currently we are using only fuzzy values of 0 and 1 – in a future step a more detailed fuzzyfication is envisaged.



		deformation parameters					
		α	β	γ	t_x	t_y	t_z
fuzzy values	very huge negative	0	0	0	0	0	0
	huge negative	0	0	0	0	0	0
	medium negative	0	0	0	0	0	0
	small negative	0	0	0	0	0	0
	very small negative	0	0	0	0	0	0
	zero	1	1	1	0	1	1
	very small	0	0	0	0	0	0
	small	0	0	0	0	0	0
	medium	0	0	0	1	0	0
	huge	0	0	0	0	0	0
	very huge	0	0	0	0	0	0

		deformation parameters					
		α	β	γ	t_x	t_y	t_z
fuzzy values	very huge negative	0	0	0	0	0	0
	huge negative	0	0	0	0	0	0
	medium negative	0	0	0	0	0	0
	small negative	0	0	0	0	0	0
	very small negative	0	0	0	0	0	0
	zero	1	0	1	1	1	1
	very small	0	0	0	0	0	0
	small	0	1	0	0	0	0
	medium	0	0	0	0	0	0
	huge	0	0	0	0	0	0
	very huge	0	0	0	0	0	0

Figure 3: Example of fuzzified parameter sets.

3.2 Forming a deformation case

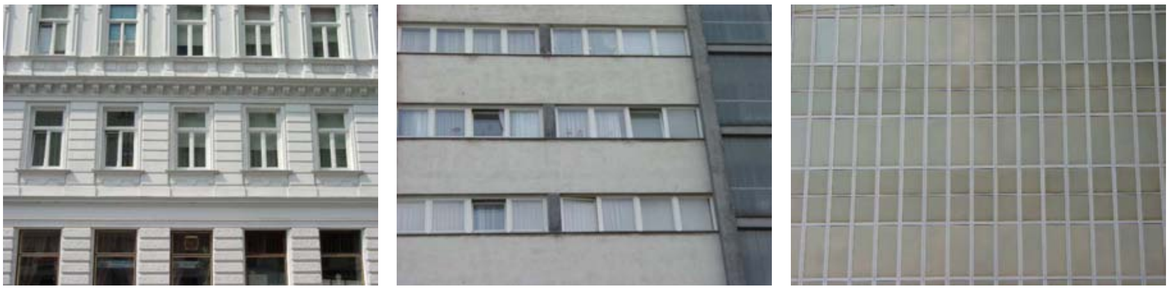
The next stage is to prepare the data for in the case-based reasoning system. A case includes more than only the calculated and fuzzified parameters.

A deformation case can be structured as follows:

- type of facade,
- count of deformation tiles,
- number of different deformation tile classes,
- distribution of deformation tiles,
- deformation pattern (summarized classes),
- …

The type of facade is the result of a classification of the measured object. Classification can be done in the measurement process and can be used for adjusting the parameters for generating the regions of interest and the parameters of point detection. This classification is processed by an artificial neural network (ANN) and is an assignment to one of the following classes:

- old building,
- new building,
- steel-glass facade.



(a) old building (b) new building (c) steel-glass facade

Figure 4: Examples for types of facades.

The count of deformation tiles is correlated to the type of the facade. A steel-glass facade will offer a lot more regions of interest than a facade of the other classes. Different methods of structuring a facade were shown in Chapter 2.

The count of different deformation tile classes can give an information about how the facade is influenced by the deformation (e.g. all tiles have the same class – a rigid body motion of the object was detected). It is a matter of fact, that such a method can be used for detecting (inner) deformations.

The distribution of the deformation tiles can be seen as a measure of quality of the deformation measurement. Objects/Cases with regular distributed ROIs are more comparable than objects/cases with an irregular distribution and will get a higher weighting.

One of the most important steps is to summarize the single ROI descriptions to a higher abstraction level. In the best case we get one kind of description for the whole object – the so-called “deformation pattern”.

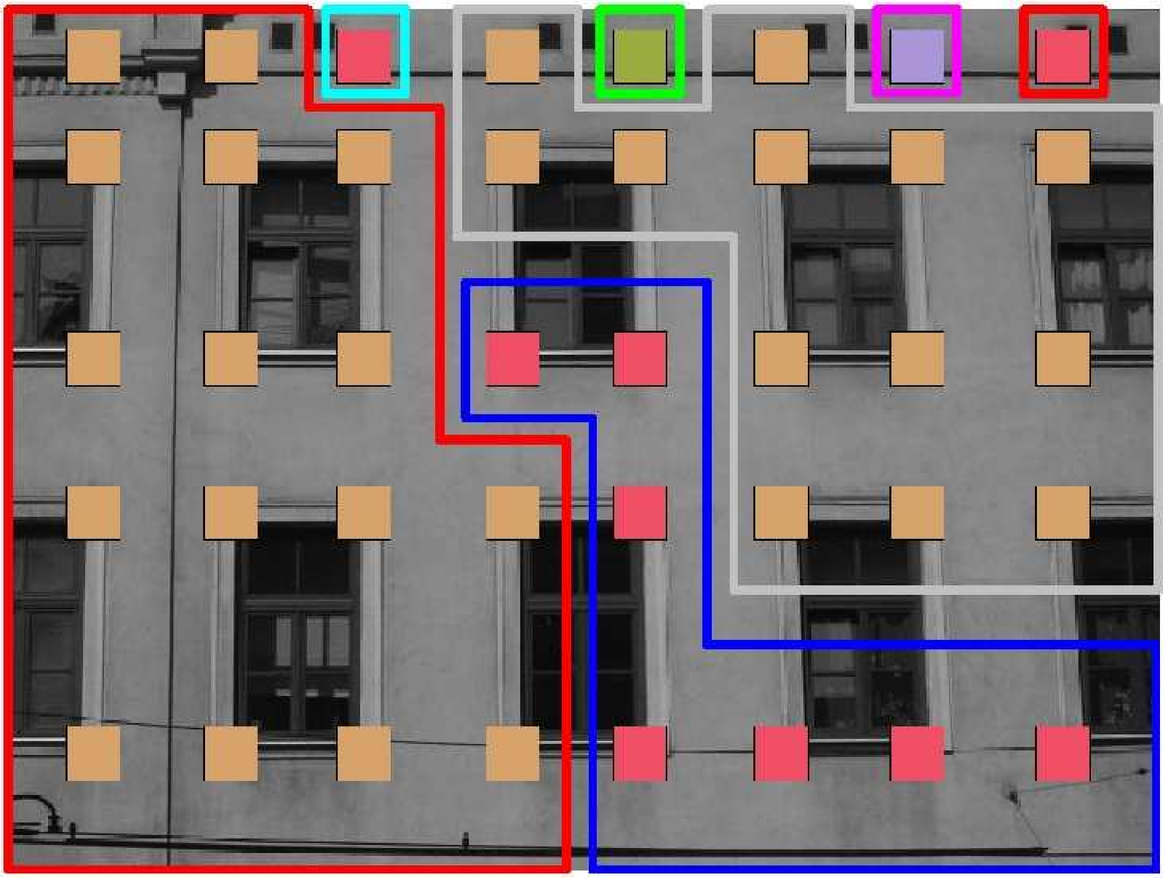


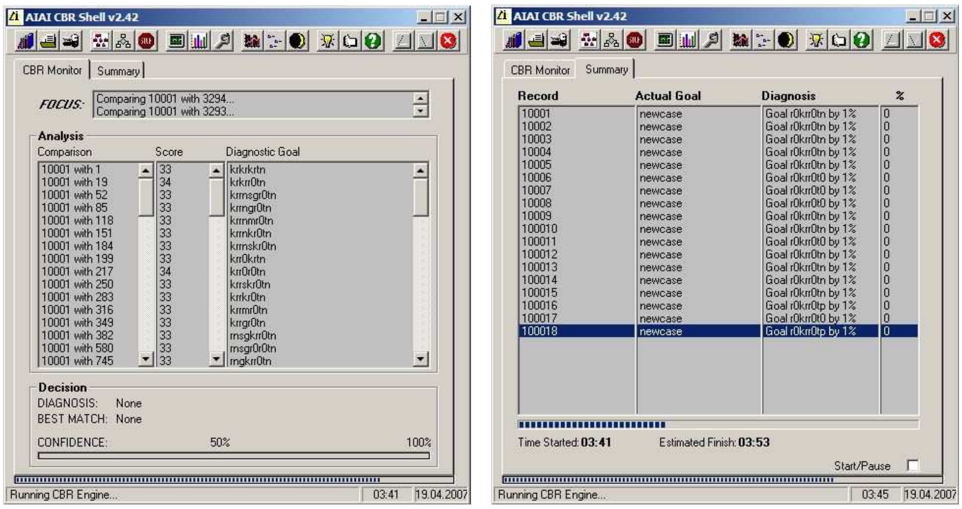
Figure 5: Visualization of deformation pattern.

3.3 Comparing cases

In the third stage a case-based reasoning system compares the resulting “deformation case” with cases held in the case base. These cases are predefined prototypes or result of former deformation measurements.

For first tests the CBR Shell from the *Artificial Intelligence Applications Institute of the School of Informatics of University of Edinburgh* (see Figure 6) was used (AIAI CBR Shell, 2008).

The result of this matching is a measure of similarity between the actual case and the cases in the case base. These similarities are the basis for



FOCUS: Comparing 10001 with 3294...
Comparing 10001 with 3293...

Analysis

Comparison	Score	Diagnostic Goal
10001 with 1	33	krkrkrtn
10001 with 19	34	krkrr0tn
10001 with 52	33	krrnsgr0tn
10001 with 85	33	krrngr0tn
10001 with 118	33	krrnmr0tn
10001 with 151	33	krrnkr0tn
10001 with 184	33	krrnskr0tn
10001 with 199	33	krr0krtn
10001 with 217	34	krr0r0tn
10001 with 250	33	krrskr0tn
10001 with 283	33	krrkr0tn
10001 with 316	33	krrmr0tn
10001 with 349	33	krrgr0tn
10001 with 382	33	rnsgkrr0tn
10001 with 580	33	rnsgr0r0tn
10001 with 745	33	rngkrr0tn

Decision

DIAGNOSIS: None
BEST MATCH: None
CONFIDENCE: 50% 100%

Running CBR Engine... 03:41 19.04.2007

Record	Actual Goal	Diagnosis	%
10001	newcase	Goal r0krr0tn by 1%	0
10002	newcase	Goal r0krr0tn by 1%	0
10003	newcase	Goal r0krr0tn by 1%	0
10004	newcase	Goal r0krr0tn by 1%	0
10005	newcase	Goal r0krr0tn by 1%	0
10006	newcase	Goal r0krr0t0 by 1%	0
10007	newcase	Goal r0krr0t0 by 1%	0
10008	newcase	Goal r0krr0t0 by 1%	0
10009	newcase	Goal r0krr0tn by 1%	0
100010	newcase	Goal r0krr0tn by 1%	0
100011	newcase	Goal r0krr0t0 by 1%	0
100012	newcase	Goal r0krr0tn by 1%	0
100013	newcase	Goal r0krr0tn by 1%	0
100014	newcase	Goal r0krr0tn by 1%	0
100015	newcase	Goal r0krr0tn by 1%	0
100016	newcase	Goal r0krr0tp by 1%	0
100017	newcase	Goal r0krr0t0 by 1%	0
100018	newcase	Goal r0krr0tp by 1%	0

Time Started: **03:41** Estimated Finish: **03:53**

Start/Pause □

Running CBR Engine... 03:45 19.04.2007

Figure 6: Example of the used CBR Shell.

the later deformation interpretation, which can be made by an expert or a knowledge-based system. This last step is the content of actual research work at the *Institute of Gedoesy and Geophysics of Vienna University of Technology*.

4 Outlook

The paper at hand presents a prototypical case-based deformation assessment system. The measurement system, the tool for deformation analysis and the deformation assessment part are still under development. The work for the next years will be the refinement of these subsystems – the development of the deformation interpretation tool is the main part of a new research project. It is planed to include knowledge of different origins, e.g. civil engineering knowledge, geological knowledge, etc.

References

http://www.aiai.ed.ac.uk/project/cbr/, 2008.

Lehmann, M., Reiterer, A., Kahmen, H.: Deformation Classification in High Density Point Clouds. Optical 3-D Measurement Techniques VIII, Volume I (2007), Zurich, 2007.

Reiterer, A., Lehmann, M.: A 3D Optical Deformation Measurement System Supported by Knowledge-Based and Learning Techniques. Journal of Applied Geodesy (in review).

Multi-Sensor System for Direct Geo-Referencing Tasks Based on Terrestrial Laser Scanning

Jens-André Paffenholz

Ingenieurgeodäsie und geodätische Auswertemethoden
Geodätisches Institut
Leibniz Universität Hannover
email: paffenholz@gih.uni-hannover.de

Abstract

This paper deals with sensor fusion as well as sensor integration aspects for terrestrial laser scanning (TLS). This means to build up a multi-sensor system with the main component TLS and additional sensors like, e.g., GNSS equipment and inclinometer. Essential for a multi-sensor system is the time synchronization of the different sensors. Therefore a unique time reference has to be introduced to the multi-sensor system. For TLS applications it is straightforward to use the internal laser clock as temporal time reference. A more suitable possibility and also a way to get an absolute time reference is to introduce the time from an external clock such as, e.g., a GNSS receiver.

In this paper a multi-sensor system based on a phase-based terrestrial laser scanner is presented. First, three different possibilities for introducing the time reference into the multi-sensor system are shown: (i) use of an external process computer which handles as input a trigger signal from the TLS as well as the pulse per second (PPS) pulse from the GNSS receiver, (ii) use of the event marker input of the GNSS receiver to register the TLS trigger signal without any additional hardware, and (iii) use of the laser scanner itself to register the PPS pulse also without any additional hardware. Second, one selected TLS based multi-sensor system is shown. This system consists of a phase-based laser scanner and two GNSS antennas. Besides details about the integration of the sensors in this application the associated method for direct geo-referencing of static terrestrial laser scans is presented. This adapted sensor-driven approach deals with the direct estimation of the sensor position and orientation with GNSS antennas installed on the laser scanner.

Afterwards the link to artificial intelligence is given by a concept of interpret-

ing the presented TLS based multi-sensor system as an intelligent agent. The design of a sensor network is outlined which will be built up by this multi-sensor system and typical geodetic sensors like tachymeters.
Also a brief overview about other TLS based multi-sensor systems especially in the context of kinematic TLS as well as selected applications is given. One thinkable application for kinematic TLS is the high-precise as-built documentation of objects with a broad range of geometric complexity. Another suitable application is the integration of, e.g., inclinometer, in the preceding TLS based multi-sensor system for deformation monitoring tasks.

Keywords: Terrestrial Laser Scanning, Multi-Sensor System, Geo-Referencing, Sensor Network.

An adapted support vector machine for velocity field interpolation at the Baota landslide

Björn Riedel & Michael Heinert

Institut für Geodäsie und Photogrammetrie
Technische Universität Braunschweig
e-mail: b.riedel@tu-bs.de

Abstract

The possibility of landslides will increase dramatically in the vicinity of the Yangtze river due to soil water changes caused by water level rise in the hinterland of the Three Gorges dam. One of these endangered areas can be found at Baota landslide.
Several investigations have been carried out here by Chinese institutions: GPS-measurements, classical geodetic observation methods and geotechnical monitorings. Remote sensing contributons were done by a European research group. But, it is the question how to compare, or better, how to combine the data of the different sensors.

Most of the interpolation algorithms do not use external information, like from topography, geology or vegetation in the computation of such velocity fields. Often this external information exits in a higher geometrical density than the sensor sites and – much more important – these external data represent the processes acting in a landslide area. Accordingly this knowledge must not be neglected actually to support the interpolation.

Some tests with synthetic data showed the high potential of this group of algorithms, and enabled to adapt them to the problem, what means: find the right feature space with the optimal hyperplane. Finally a well adjusted ANOVA-kernel delivered the best results for our purposes at Baota landslide.

Keywords: System Analysis, Statistical Learning Theory, Support Vector Machines, SAR Interferometry, ASTER data, Landslides

1 Introduction

Landslides are one of the most dangerous natural hazards in the world, causing high annual death tolls (Sidle and Ochiai, 2006). On average, landslides annually kill twice as many people as earthquakes and result in high annual damage costs. In the years 2003 to 2006 the European Union (EU) funded a multidisciplinary international project called OASYS (Integrated optimization of landslide alert systems). The scope of OASYS was to set up an integrated workflow for landslide hazard management. This system should lead the practitioner from data acquisition to suggestions of risk management measures. The emphasis of the project was the development of observation methods that allow:

- detection of potential landslides on large scale
- an efficient and continuous observation of critical areas
- a knowledge-based derivation of real time information about actual risks in order to support an alert system (Kahmen et al., 2007).

In the framework of this EU project OASYS, we processed data and compared results from test sites in several countries (Riedel and Walther, 2008). These data sets consisted of inhomogeneous observations of different sensors, like remote sensing data, GPS, total station, inclinometers etc. . Following the above mentioned items it was obvious that we need an approach to come from discret selective observations in space and time to a field of movement for the critical areas. For this purpose one normally uses standard interpolation algorithms, e.g. Kriging, Radial Basis functions or Regression methods. The initial question was whether a learning algorithm could be able to interpolate any velocity field.

2 Support Vector Machines – a brief overview

While we are mostly interested in modelling or let's say in some kind of regression of our data, the SVM is initially designed for pattern recognition purposes. So, to understand this algorithm we unfortunately have to make a detour through the basics of pattern recognition. Therefore we have to focus especially on the thoughts coming from the statistical learning theory (Vapnik, 1998).

A pattern recognition is basically the task to separate labelled patterns into two classes. The SVM is a linear machine (Haykin, 1999, p. 318), that means that the two classes of patterns should be separated linearly. But, as already mentioned (Heinert, 2008a), often the patterns can't be shattered linearly

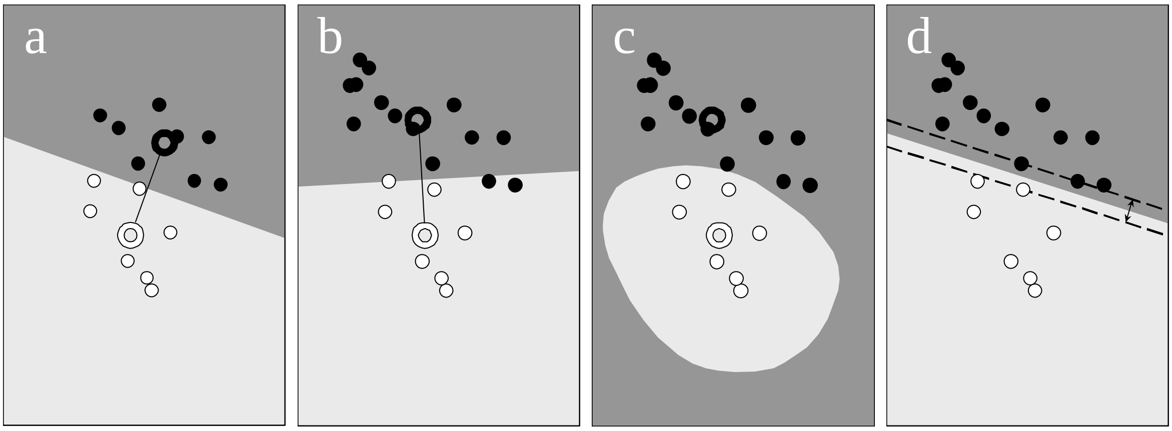


Figure 1: Classification with a linear 2D-algorithm (a,b), with a hyperplane represent by a Gaussian kernel (c) and with a SVM using a dot product kernel (d). The dots are the different patterns of two classes, the rings are the class centers c_+ and c_-.

within their own data space $\mathcal{X} \subset \mathbb{R}^N$, $N \in \mathbb{N}$. One of the best examples is the XOR-problem (Haykin, 1999, pp. 175 & pp. 259). The patterns are arranged here, that they are not linearly separable by any geometrical solution within the 2D data space. To separate them linearly we have to leave the data space and transform the patterns into an at least 3D feature space. Therefore a pattern $x \in \mathcal{X}$ can be transformed by the map

$$\mathbf{\Phi} : \mathcal{X} \rightarrow \mathcal{H} \text{ that } x \mapsto \mathbf{x} = \mathbf{\Phi}(x) \tag{1}$$

to a vector $\mathbf{x}$ into the Kernel-Hilbert-space $\mathcal{H}$. A hyperplane can now separate the two classes. The next idea – the so-called *kernel-trick* – is that kernel functions $\mathcal{K}(x, x')$ like for example the *canonicle dot product* $\langle \mathbf{x}, \mathbf{x}' \rangle = \langle \mathbf{\Phi}(x), \mathbf{\Phi}(x') \rangle$ are a representation of the space of higher order within the given data space (Mercer's theorem).

2.1 2D-classificator for linearely separable patterns

Before entering the world of n-dimensional hyperspaces with their hyperplanes, let's start with a pattern recognition tool within the 2D data space. A simple method for shattering the m_+ patterns x_i labelled with $y_i = +1$ and m_- patterns x_j labeled with $y_j = -1$ would be to place a "1D-hyperplane" in the middle between the two class centres c_+ and c_- (Fig. 1b). The vector connecting the two class centres would be complanary to the normal vector of the 1D-hyperplane (Schölkopf and Smola, 2001, pp. 4). Now it's possible to compute the label

$$y = \text{sgn}\left(\frac{1}{m_+}\sum_{\{i|y_i=+1\}}\mathcal{K}(x,x_i) - \frac{1}{m_-}\sum_{\{i|y_i=-1\}}\mathcal{K}(x,x_i) + b\right) \tag{2}$$

of any other pattern x with the offset

$$b = \frac{1}{2}\left(\frac{1}{m_-^2}\sum_{\{(i,j)|y_i=y_j=-1\}}\mathcal{K}(x_i,x_j) - \frac{1}{m_+^2}\sum_{\{(i,j)|y_i=y_j=+1\}}\mathcal{K}(x_i,x_j)\right).$$

Within the demonstrated algorithm any additional pattern x will shift the centre of its class (Fig. 1b). Accordingly the hyperplane will be shifted as well. This algorithm is able to produce quite good classification results, especially with other kernel functions like e.g. the Gaussian one (Fig. 1c)

$$\mathcal{K}_G(x,x') = e^{\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)}. \tag{3}$$

2.2 SVM for linearely separable patterns

A SVM shatters the classes by an *optimal hyperplane* (Haykin, 1999, p. 320). This approach uses instead of the instable class means the closest patterns to the *margin of separation*. Only these very patterns are able to determine the optimal hyperplane. Any additional patterns in a distance to the margin don't have any impact on the position of the hyperplane any more (Fig. 1d).

A few support vectors $\mathbf{x}^s$ should determine the position and direction of the optimal hyperplane

$$\mathbf{w}_o^{\mathsf{T}}\mathbf{x}^s \mp b_o = \mp 1 \ \forall \ y^{(s)} = \mp 1 \tag{4}$$

with the optimal weight vector

$$\mathbf{w}_o = \sum_{i=1}^{N}\alpha_{o,i}y_i\mathbf{x}_i. \tag{5}$$

The weights can be computed by using the optimal *Lagrange* multipliers $\alpha_{o,i}$. It can be shown that the minimization of the cost function

$$\mathbf{\Phi}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w} \tag{6}$$

leads to the solution function

$$Q(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{(i,j)=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^{\mathsf{T}}\mathbf{x}_j \tag{7}$$

that has to be maximized under the constraints

① $\sum_{i=1}^{N} \alpha_i y_i = 0,$

② $\alpha_i \geq 0 \;\forall\; i = 1 \ldots N.$

This function has now to be maximized (Haykin, 1999, pp. 322). The optimal bias of the hyperplane is given through the avarage of

$$b_o = 1 - \mathbf{w}_o^{\mathsf{T}} \mathbf{x}^{(s)} \;\forall\; y = 1. \tag{8}$$

2.3 SVM for linearely nonseparable patterns

If the data are *a priori* linearely nonseparable, we have to allow a few misclassifications. The classification along the hyperplane can be extended by a range around the hyperplane in which misclassifications are generally allowed. This will be arranged using the so-called *slack variables* $0 \leq \xi_i \leq 1$, that classifier is given by

$$y_i(\mathbf{w}^{\mathsf{T}}\mathbf{x} + b) \geq 1 - \xi_i \;\forall\; i = 1 \ldots N. \tag{9}$$

The related cost function

$$\mathbf{\Phi}(\mathbf{w}, \xi) = \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w} + C\sum_{i=1}^{N} \xi_i \tag{10}$$

referres to the new variables with a second term in which these variables contribute to empirical risk. The parameter C controlls the classifier: a higher C will increase the classifier's complexity. This parameter should be adapted analytically. Therefore the user can observe the number of the support vectors and the model's VC-dimension. The solution function remains the same, beside a revised constraint ② as to be $0 \leq \alpha_i \leq C \;\forall\; i = 1 \ldots N.$

2.4 SVM with optimal kernel

We rewrite the solution function with the *Langrange* multipliers given in eq. (7):

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{(i,j)=1}^{N} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle. \tag{11}$$

The included canonicle dot product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ acts as a kernel function. Accordingly, we rewrite eq. (7) once more as

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{(i,j)=1}^{N} \alpha_i \alpha_j y_i y_j \mathcal{K}(x_i, x_j) \tag{12}$$

with the more general kernel function $\mathcal{K}(x_i, x_j)$. Several kernels like the polynomial, the neural, the Gaussian or the ANOVA kernel have already be tested successfully.

2.5 SVM for nonlinear regression

The last important step is to leave the pattern recognition and to transfer this knowledge to the regression. To start we rewrite eq. (10)

$$\mathbf{\Phi}(\mathbf{w}, \xi, \xi^*) = \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w} + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) \tag{13}$$

using two non-negative sets of slack variables. They are necessary to desribe the asymmetrical robust ε-intensive loss function L_ε (Haykin, 1999, Chap. 6.7). The robustness of this loss function is already introduced in the onset of this algorithm to deal with the sensitivity of the least-squares estimator to the presence of outliers. The minimization of the redefined cost function in eq. (13) with respect to

$$\mathbf{w}^{\mathsf{T}}\mathbf{x} + b \left\{ \begin{array}{l} \geq y_i - \varepsilon - \xi_i^* \\ \leq y_i + \varepsilon + \xi_i \end{array} \right\} \ \forall \ i = 1 \ldots N \tag{14}$$

with

$$\xi_i, \xi_i^* \geq 0 \ \forall \ i = 1 \ldots N$$

can equivalentely be solved by the maximation of the solution function

$$Q(\alpha_i, \alpha_i^*) = \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*) - \varepsilon \sum_{i=1}^{N} (\alpha_i + \alpha_i^*) - \frac{1}{2} \sum_{(i,j)=1}^{N} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\mathcal{K}(x_i, x_j) \tag{15}$$

under the constraints

① $\sum_{i=1}^{N} \alpha_i = \sum_{i=1}^{N} \alpha_i^*$,

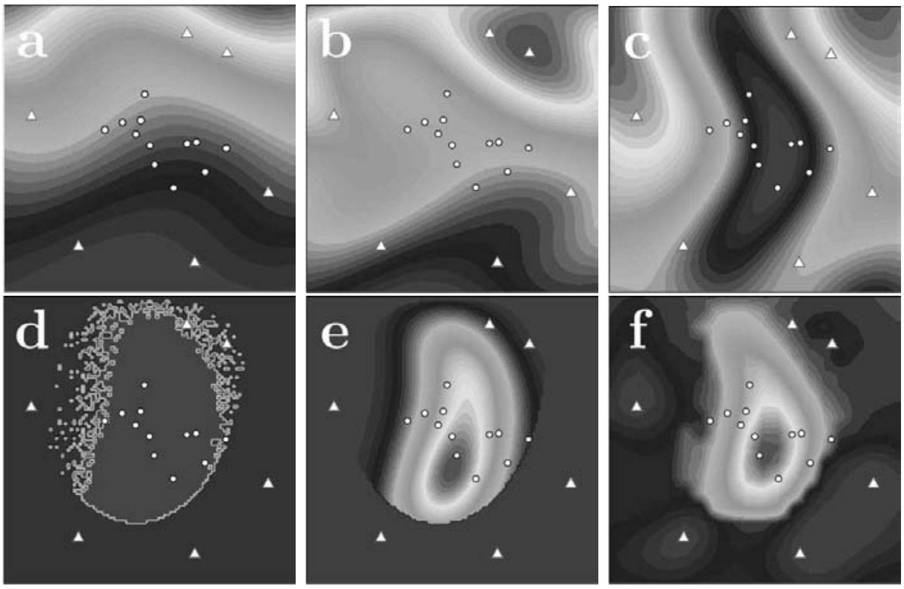


Figure 2: The SVM concept testet by using synthetic data: three topographical layers with height (a), slope (b) and curvature (c) connected with a fourth layer (d) containing the information of land slide evidences like e.g. destroyed vegetation or fissures. The synthetic output is given as a absolute velocity field (e) that could be modelled (f) by only using the velocities at the reference (triangles) and monitoring sites (circles).

② $0 \leq \alpha_i, \alpha_i^* \leq C \;\; \forall \;\; i = 1 \ldots N.$

During the modelling the parameters ε, respectively $+\xi$ and $-\xi$, as well as C have to be tuned simultaniously to reduce the number of support vectors and the VC-dimension.

The theory behind this algorithm has been widely described in both the statistical background (Vapnik, 1998) and the possibly adaptations (Haykin, 1999; Schölkopf and Smola, 2001). We used for our purposes the programme *mySVM* (Rüping, 1999, 2000).

3 Support Vector Machines – tested with synthetic data

For a non-linear interpolation we need input data that controll the velocity which should be estimated. Several authors already presented lists of input data, suitable to model land slides. Exemplarely we would like to present that one which seemed us to be most similar to our ideas (Fernández-Steeger and Czurda, 2001):

- ○ slope direction,
- ○ change of the topographic gradient,
- ○ slope shape,
- ○ brooks, rivers, moats,
- ○ lithology,
- ○ vegetation,
- ○ topographic height,
- ○ slope length,
- ○ slope shape,
- ○ distance to the water courses,
- ○ distance to rocks,
- ○ security measures.

In our first computational approach we reduced this list to four pseudo-layers, as to be the topographic height (Fig. 2a), the local topographic descent along the axis of the steepest descent in general (Fig. 2b) and the change of the topographic gradient across this axis (Fig. 2c). Most important seemed us to be the layer of land slide evidences (Fig. 2d). This layer should content all observable natural signs of landslides like destroyed vegetation, recent fissures, tilted trees and so on.

Generally the ANOVA kernel

$$\mathcal{K}_D(x,x') = \sum_{1\le i_1<\ldots<i_D\le N} \prod_{d=1}^{D} \mathcal{K}^{i_d}(x_{i_d}, x'_{i_d}) \tag{16}$$

is said to be very suitable especially for multidimensional SV regression problems
(Schölkopf and Smola, 2001, p. 412). In the SVM programme *mySVM* (Rüping, 2000) this kernel is defined with

$$D = 1,$$
$$\mathcal{K}^{i_d} = \mathcal{K}_G^{i_d},$$
$$\gamma = \sigma^{-2}.$$

These definitions simplify the implemented kernel to

$$\mathcal{K}_D(x,x') = \sum_{i=1}^{N} \mathcal{K}^d(x_i, x'_i). \tag{17}$$

During the tests *mySVM* delivered the implemented ANOVA-kernel among all tested kernels the best approaches to the synthetic velocity field. This test result corresponds with the theoretical statement about the kernel's ability.

4 The application of SVM to Baota landslide

One test site of the OASYS project was the Baota landslide in China (Fig. 3). The site is near the city of Yunyang, located in the middle of the Three Gorges Reservoir, 223 km upstream of Three Gorges Dam, on the left

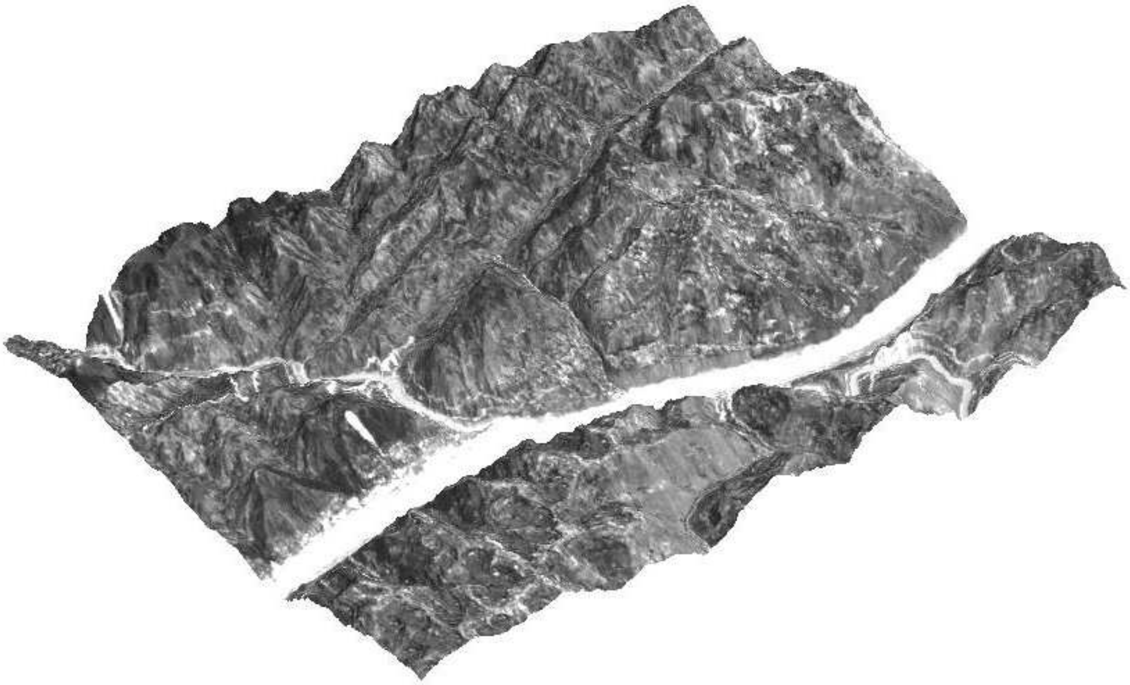


Figure 3: This overlay of the digital elevation model with a false colour ASTER scene shows the Baota landslide. It is situated on the right hand side from the river junction in the central part on the left Yangtze bank. The heights are three times exagerated to get an impression on the topograhical situation in this area.

bank of the Yangtze river. The Baota landslide event occurred on the 18th July 1982, triggered from a hundred year frequency rainfall and flooding. During this event 2.3 million m^3 of rock mass slid into the Yangtze (Cui, 2000). 4000 people live on the landslide, which is still moving. The area of investigation has a length of 1900 m, a mean width of 1000 m and 450 m height difference. The mean surface gradient is 15 degrees and the thickness of landslide deposits is 70 m (Cui, 2003).

In 1997 a GPS and terrestrial network was established which consisted of 5 stable control points and 12 monitoring points on monuments in the moving landslide. This network was measured three times in 1997 using Rogue 8000 receivers for the GPS measurements and EDM from the type Wild DI 2002. The longest time intervall from January to November 1997 is suitable to compute movement rates. Accordingly, there can be founsd between 10-25 mm in the middle of the slope and 5 to 11 mm for the accumulation area within these 11 months (Zhang and Jiang, 2003).

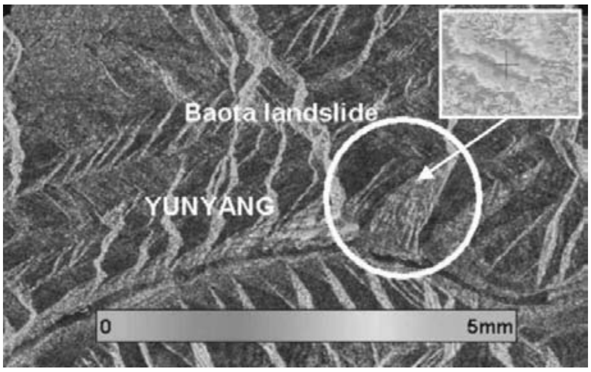


Figure 4: The resulting fringe pattern of the InSAR is overlayed on the amplitude radar image from the 2-pass processing. Movement rates are visible for the landslide and for the city of Yunyang. The inlet figure shows the unwrapped phase for the middle part of the landslide in detail with a movement rate of 14 mm per 6 months.

4.1 Remote sensing investigation

We followed two different remote sensing approaches. We analysed on the one hand for the recognition of surface movements the capability of Synthetic Aperture Radar Interferometry (InSAR) and on the other hand ASTER data to derive a detailed elevation model and geomorphological informations.

InSAR data processing for this test site was done by a 2-pass approach (Riedel and Walther, 2008). The processing chain was applied to the image pair of frame 2979 with the scenes 23424 (date of acquisition 07/01/96) of ERS 1 and 06256 (01/07/96) of ERS 2. The perpendicular baseline length is 23 m derived from precise orbits. Further processing was accompanied by manual corrections to the standard filter parameters, because two vegetation growth periods and the steepness of the terrain in this area complicated the InSAR processing. The resulting interferogram for Baota landslide can be seen in figure 4. The changes of colour in the interferogram, represented by sequences of colour cycles, show the change of earth surface for a time difference of approximately 6 months (07/01.-01/07/1996). One colour cycle, called fringe, is scaled to 5 mm. The displayed displacements in the line of sight of the sensor achieves a maximum value of 14 mm for the middle part of the landslide and 3-5 mm for the accumulation zone at the foot of the slope. Movement rates for the city of Yunyang were also derived. The comparison of our 1996 2-pass InSAR results with the 1997 geodetical derived observations shows a good coincidence under the assumption of linear movement trend for the slope.

The Terra spacecraft launched in 1999 has a platform with five different sensors. One of these five sensors is the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER). It consists of three separate instrument subsystems, each operating in different spectral region, using a separate optical system. These subsystems are the visible and near infrared (VNIR), short wave infrared and thermal infrared with ground resolutions between 15 and 90 m. The VNIR subsystems allows additionally the derivation of elevation models and the combination of all 14 differents spectral bands allows the derivation of vegetation indices and various geological informations (Lillesand and Kiefer, 2000). For the Baota investigation we used an ASTER data set from 31/08/2002 and processed it with standard remote sensing tools.

4.2 The Baota SVM model

As input data sets for our SVM models we used the results from our ASTER remote sensing investigations. The topography is represented by the ASTER digital elevation model with 30 m ground resolution and the different curvature parameters, e.g. minimum, maxmium, plane and profile curvature were calculated out of this digital elevation model. The slope layer consists of the inclination angle for every grid cell. Land – water and land use separation was done by different classification algorithms applied on different spectral band combinations. Each of this data sets for Baota landslide served as one input layer into the SVM modelling and are visualised in figure 5. The input layer for the flow velocity of the slope was realized by the 1997 GPS observations. All these matrix type layers were transformed into vectors for the SVM processing. According to the tests with the synthetic data we used finally the ANOVA kernel.

4.3 The results of the Baota landslide modelling

The objective of the whole SVM modelling approach is to get a reliable information about the landslide behaviour with the use of all available information. The transition from discrete observations of velocity (GPS) to a modelled velocity field for the whole region of interest will be a big step forward in the description of landslide behaviour. Therefore it is of great importance to validate the modelling results with an independent observation data set. InSAR results or in general remote sensing evaluations deliver us areal information about our area of investigation and will give us the required independent validation data set.

The results from the SVM calculations with the ANOVA kernel are displayed in figure 6. It is clear visible in all three modelled velocity fields that

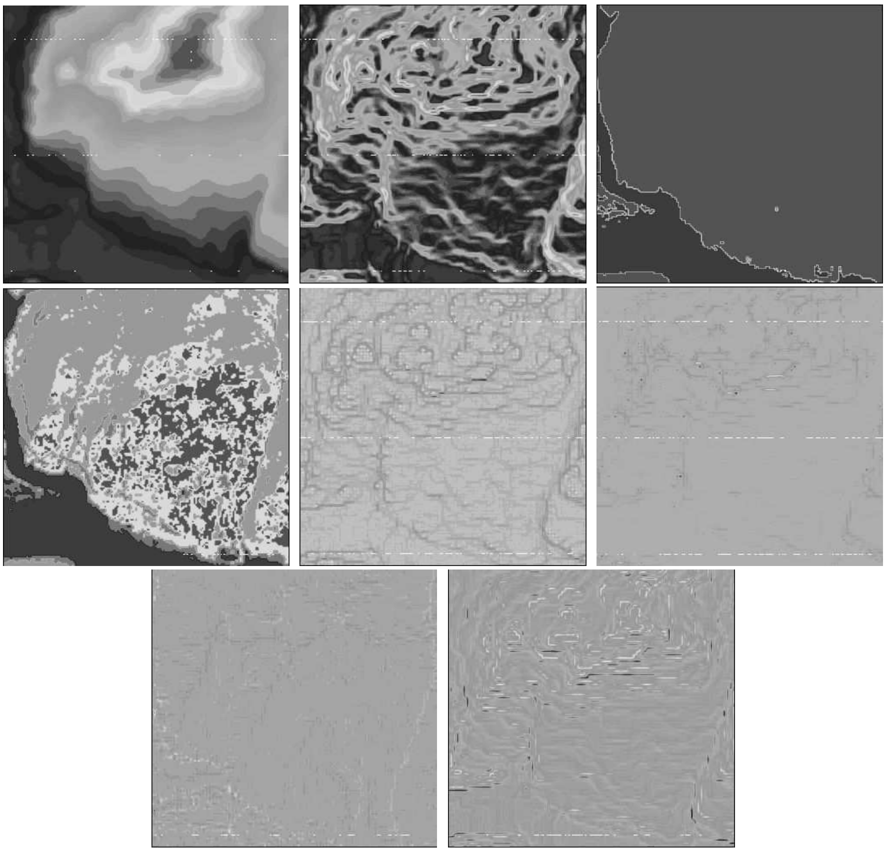


Figure 5: The input layers derived from satellite data: topography, slope, land – water separation, classified surface, minimum curvature, maximum curvature, plane curvature, profile curvature.

there is stable velocity structure on the slope, independent from the boundary conditions. A comparison with InSAR results from figure 4 shows the same pattern of velocity for the whole slope, especially for the central part. The numerical comparisons between modelled and InSAR velocities show high coincidence.

We carried out three different model calculations. The first was limited to the topographical shape of the landslide. This limitation was introduced by a layer containing this shape. Due to the fact that this is quite weak limitation it came out that there are still movement signals outside the limitation, especially on neighbouring slope in the West. The second limitation – including just this very slope – did not yield any significant signals outside anymore. Finally, we modelled the region without any limiting layers. Here we got movements distributed all over the region. It is very impressing that the movement borders came out without any limitations.

This tells us that the first model was not able to suppress the signals outside the limitation, because they were even significant as the those inside the

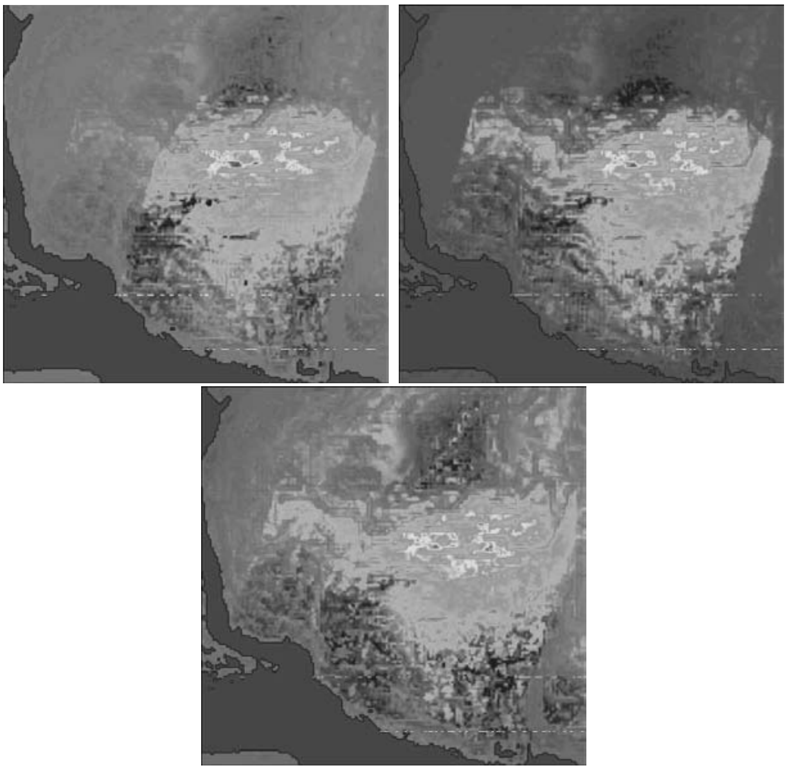


Figure 6: Modelled velocity fields: (a) with a limitation on Baota landslide, (b) with a limitation including the neighboured slope and (c) and unlimited modelling.

limited field. The second model contained all the significant signals, that the insignificant movements outside could easily be suppressed. We may state that we should believe the unlimited model. This fact has a expresses a strong conclusion: even surfaces that are far away of the test field in which single GPS sites deliver velocity information it is possible to derive a reliable velocity field. The chosen algorithm transforms a extrapolation in the data space into an interpolation within the feature space. Accordingly, an extension of the model in space only depends on the input data quality and consistence.

For the slope investigated here, we can come to specific conclusions: the landslide shows quite typical motions. On top of the hill we can find an island of rest. The highest velocities occur in the upper and middle part of the landslide (white). Near the river within the accumulation zone the velocities are slowing down to zero (black). But the most important result is the moving neighboured slope. This slope is not monitored yet. None former landslide is presently known here. So, nobody expects the danger of a natural hazard here.

Acknowledgements

We are most grateful to ir. Jeroen Meeus and ir. Willem Wagemann (formerly University of Ghent), who introduced us into the basics of the statistical learning theory. Further we would like to thank Dr. Stefan Rüping for his downloadable programme *mySVM*. The OASYS project (EVG1-2001-00061) was supported by the European Commission under the Fifth RTD Framework Program.

References

Cui, Z. Q.: Failure Mechanism and Prediction ideology for natural slopes of the Three Gorges Area of Changjiang. Bureau of Geotechnique of Changjiang Water Resources Commission (CWCR), Wuhan, 2000.

Cui, Z. Q.: Brief introduction of BAOTA Landslide - One of the Examination Spots of OASYS. Bureau of Investigation & Survey, Wuhan, 2003.

Fernández-Steeger, T. M. and Czurda, K.: Erkennung von Rutschungsgebieten mit Neuronalen Netzen. In: Czurda et al.: Geotechnik, Sonderband zur 13. Nat. Tagung f. Ingenieurgeologie Karlsruhe, 2001; 61-66; Essen (VGE).

Haykin, S.: Neural Networks – A Comprehensive Foundation. 2nd edition, Prentice Hall, Upper Saddle River NJ, 1999.

Heinert, M.: Artificial neural networks – how to open the black boxes? In this issue.

Heinert, M.: Systemanalyse der seismisch bedingten Kinematik Islands. PhD thesis, Brunswick (Germany), in press.

Kahmen, H., Eichhorn, A. and Haberler-Weber, M.: A Multi-Scale Monitoring Concept for Landslide Disaster Mitigation. Dynamic Planet Monitoring and Understanding a Dynamic Planet with Geodetic and Oceanographic Tools. in Tregoning, P. and Rizos, C. (Eds.) IAG Symposium, Cairns, Australia, 22-26 August, 2005 Series: International Association of Geodesy Symposia; Volume 130, Springer, 2007.

Lillesand, T. and R. Kiefer: Remote sensing and image interpretation. Wiley and son, New York, 2000.

Riedel, B. and A. Walther: InSAR processing of C-band data for the recognition of landslides. Advances in Geosciences, Vol 14, 189-194, 2008.

Rüping, St.: Zeitreihenanalyse für Warenwirtschaftssysteme unter Berücksichtigung asymmetrischer Kostenfunktionen. `http://www.stefan-rueping.de/publications/rueping-99-a.pdf`.

Rüping, St.: mySVM – Manual. `http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/mysvm-manual.pdf`.

Schölkopf, B. and Smola, A. J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning). MIT Press, 2001.

Sidle, R. C. and Ochiai, H.: Landslides- Processes, Prediciton and Land Use. AGU Books Board, Washington, 2006.

Vapnik, V. N.: Statistical Learning Theory. in Haykin, S. (Ed.), Adaptive and Learning Systems for Signal Processing, Communications and Control, John Wiley & Sons, New York-Chichester-Weinheim- Brisbane-Singapore-Toronto, 1998.

Zhang, J. and Jiang, B.: GPS landslide monitoring of Yunyang Baota. Report of University Wuhan, 2003.

Appendix

The example shown in Fig. 2 can easily be re-computed. Therefore we present the equations that are necessary to set up this synthetic example:
❶ **layer:** topography

$$T(x,y) = \left(1 + y + \cos\left(\frac{\pi}{5}\left(x + \cos\left(\frac{\pi y}{5}\right)\right)\right)\right)^2, \tag{18}$$

❷ **layer:** slope

$$\partial T(x,y) = 2\left(1 + y + \cos\left(k(x,y)\right)\right) \cdot \left(1 + \frac{\pi^2}{25} \cdot \sin\left(k(x,y)\right) \cdot \sin\left(\frac{y\pi}{5}\right)\right) \tag{19}$$

with

$$k(x,y) = \frac{\pi}{5}\left(x + \cos\left(\frac{y\pi}{5}\right)\right),$$

❸ **layer:** soil water dependent on curvature

$$\partial^2 T(x,y) = \frac{\left(\frac{9}{5} - \left(\frac{2\pi}{25}\sin^2\left(k(x,y)\right) + \frac{\cos(k(x,y))}{5}\left(\frac{-2}{5} + y + \cos\left(\frac{-2}{5}k(x,y)\right)\right)\right)\right)}{100 + T(x,y)}, \tag{20}$$

❹ **synthetic output:** velocity field

$$z(x,y) = \partial T(x,y)\partial^2 T(x,y) \cdot \left\{ \begin{array}{ll} \left\{ \begin{array}{ll} z_0(x,y) & \forall z_0(x,y) \geq \frac{1}{50} \\ 0 & \forall z_0(x,y) < \frac{1}{50} \end{array} \right\} & \forall g(x,y) \geq \frac{1}{80} \\ 0 & \forall g(x,y) < \frac{1}{80} \end{array} \right. \tag{21}$$

with

$$g(x,y) = e^{-\left(\frac{(x-c_x)^2}{s_x^2} + \frac{(y-c_y)^2}{s_y^2}\right)} \quad c_x = 4.5, c_y = 8, s_x = \sqrt{3}, s_y = \sqrt{8}$$

and

$$z_0(x,y) = \left(1 - \frac{y}{b_y}\right) e^{-\frac{(x-\mathfrak{x})^2}{b_x^2}} \quad \mathfrak{x} = 5, b_x = 4, b_y = 10 \quad \forall x,y = 0, 0.1 \ldots 10,$$

❺ **layer:** evidences

$$E(x,y) = \left\{ \begin{array}{c} 1 \\ 0 \end{array} \right\} \forall \arcsin\left(\frac{z(x,y)}{100} \cdot \frac{180}{\pi} \cdot 250 + \text{rnd}(5)\right) \left\{ \begin{array}{c} \geq 5 \\ < 5 \end{array} \right., \tag{22}$$

❻ positions of the reference stations

$$x_{R,i} = \left[\begin{array}{c} 5 + r\cos(\alpha_i) - r\sin(\alpha_i) \\ 5 + r\cos(\alpha_i) + r\sin(\alpha_i) \end{array} \right] \quad \alpha_i = \text{rnd}(2\pi), r = 3 \ \forall i = 1 \ldots 6, \tag{23}$$

❼ positions of the monitoring stations

$$x_{R,j} = \left[\begin{array}{c} 5 + r\cos(\alpha_j) - r\sin(\alpha_j) \\ 5 + r\cos(\alpha_j) + r\sin(\alpha_j) \end{array} \right] \quad \alpha_j = \text{rnd}(2\pi), r = \text{rnd}(2.5) \ \forall j = 1 \ldots 12. \tag{24}$$

At the positions of the observated stations – independent whether they are defined as to be reference or monitoring stations – the output of a pattern is equal to z. At any grid points, which are determined by the user, the velocity can be estimated now on the base of the inputs ($T(x,y)$, $\partial T(x,y)$, $\partial^2 T(x,y)$, $E(x,y)$).

In the last years Artificial Intelligence (AI) has become an essential technique for solving complex problems in Engineering Geodesy. Current application using AI methodologies are geodetic data analysis, deformation analysis, navigation, deformation network adjustment, optimization of complex measurement procedure, etc.

AIEG will transfer and promote the application of Artificial Intelligence in Engineering Geodesy and will bring together researcher from informatics and geodesy. The workshop is a platform for experts to expound and share their successes in applying AI in their respective fields of engineering geodesy.

The first workshop of AIEG was held in December 2008 in Vienna and was orgaised by the IAG Working Group 4.2.3 “Application of Artificial Intelligence in Engineering Geodesy”.

Editors:

Alexander Reiterer
Institute of Geodesy and Geophysics
Research Group Engineering Geodesy
Vienna University of Technology

Uwe Egly
Institute of Information Systems
Knowledge-Based Systems Group
Vienna University of Technology

ISBN: 978-3-9501492-4-1