Approaches to Invariant Image Object Recognition

Diplomarbeit im Fach Informatik

Lehrstuhl für Informatik VI
Mathematisch-Naturwissenschaftliche Fakultät
Rheinisch-Westfälische Technische Hochschule Aachen
Prof. Dr.-Ing. H. Ney

vorgelegt von:

Cand. Inform. Daniel Martin Keysers
Matrikelnummer 205 354

Gutachter:

Prof. Dr.-Ing. H. Ney
Prof. Dr. W. Oberschelp

Betreuer:

Dipl.-Inform. J. Dahmen

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Alle Textauszüge und Grafiken, die sinngemäß oder wörtlich aus veröffentlichten Schriften entnommen wurden, sind durch Referenzen gekennzeichnet.

Aachen, im Juni 2000

Daniel Martin Keysers

Acknowledgement

The present work originated from the interest in its topic at the Chair of Computer Science VI of the RWTH Aachen University of Technology, where I have been a member of the image recognition group since April 1999. I would like to thank Prof. Dr.-Ing. Hermann Ney for the interesting subject he issued for this work and for the chance to attend the conferences "Bildverarbeitung in der Medizin / Image Processing in Medicine 2000", Munich and "Recherche d'Informations Assistée par Ordinateur / Content-Based Multimedia Information Access 2000", Paris.

This work had not been possible without the many helpful suggestions, discussions and assistances it received from various of people. I would especially like to thank Prof. Dr.-Ing. Hermann Ney and Dipl.-Inform. Jörg Dahmen for the fruitful conversations and their interest in the work. Their ideas and suggestions were a great help.

I would like to thank Prof. Dr. Walter Oberschelp as well, who kindly accepted to attend the work and brought to my attention the interesting subject of the holographic classifier.

Furthermore, I would like to thank the other student members of the image recognition group at the Chair of Computer Science VI, who were always there to discuss things and give helpful comments and tips, namely Alexander Crämer, Mark Oliver Güld, Ralf Perrey and Thomas Theiner.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Arthur listened for a short while, but being unable to understand the vast majority of what Ford was saying he began to let his mind wander, trailing his fingers along the edge of an incomprehensible computer bank, he reached out and pressed an invitingly large red button on a nearby panel. The panel lit up with the words "Please do not press this button again."

[1]

Pattern recognition is a research field with a large number of application areas and it receives a lot of scientific interest. It is concerned with the design and the investigation of systems that automatically detect patterns of predefined classes in their input. The fundamental aim of research in pattern recognition is the performance of the classifier, measured by the error rate, defined as the ratio of misclassifications to the total number of patterns seen in an evaluation.

This work is concerned with the more specific case of the general pattern recognition problem, where the input consists of digital images. In *image recognition* the main problem is to identify the *objects* present in a given image. This task, which seems ridiculously easy to a human perceptor, is a very difficult one to teach a digital computer. For a computer, a digital image consists of an array of pixel values, which has no associated meaning in itself. This work focuses on the recognition of objects in images, where the position of an object is roughly known, although most of the methods can be applied to determining both content and position. Furthermore, the emphasis is placed on *appearance based* pattern recognition, which refers to the paradigm of considering the whole image as input to the classifier.

Dealing with image object recognition, in almost all cases one is interested in designing classifiers that tolerate certain transformations of the input patterns, that is, one wants to achieve *invariant* recognition of the image content. This is because the transformations do not affect the class membership of the represented objects (a rotated image of a car is still an image of a car). Invariance is an important aspect in image object recognition, since images are seldomly normalized, that is, brought to a canonical form when presented to the classifier. Although this does not present a difficulty to humans, who have the ability to recognize objects almost independently of their position and scale, it is a very hard task for an automatic classifier.

Most of the time, the transformations to be tolerated result from exterior transformations of the depicted objects relative to the imaging system and are known a priori. Such a priori knowledge about the classification task is generally called *domain knowledge*. For example, in the case of radiographs the position of the object is not invariable, but is subject to rotation and translation. The transformations of the input space would then be chosen from the group of linear, affine or projective transformations, representing the exterior object transformation. In other cases, for example in recognition of handwritten characters, the transformations of the images are due to other reasons, like different styles in handwriting and different pens used. The resulting transformations may then be approximated by the affine group augmented with a line-thickness transformation.

Most classification algorithms are based on the paradigm of supervised learning, where the classifier is provided a set of labeled training samples from the different classes to be distinguished. Therefore, one may be interested – especially in cases where no domain knowledge about the transformation invariance is available – in possibilities to deduce the transformations (or at least the invariance restrictions) from the training set.

There is a variety of approaches known to achieve invariance in image object recognition (see e.g. [101]). Some of these will be introduced here. However, a strong emphasis is placed on a method called *tangent distance* [89], which is an effective means to compensate small (affine) transformations in distance based classifiers. The following description of the main idea is taken from [100]:

"The key idea is that, when subject to spatial transformations, images describe manifolds in a high dimensional space, and an invariant metric should measure the distance between those manifolds instead of the distance between other properties of (or features extracted from) the images themselves. Because these manifolds are complex, minimizing the distance between them is a difficult optimization problem which can, nevertheless, be made tractable by considering the minimization of the distance between the tangents to the manifolds – the tangent distance (TD) – instead of that between the manifolds themselves."

Tangent distance has been used in a variety of classifiers, including neural networks and memory based techniques like nearest neighbor algorithms (NN) [87]. The experiments carried out for this work focussed on kernel density (KD) based classifiers, which are also memory based, and obtained excellent results [51]. A number of solutions have been proposed for efficient implementation of such algorithms, e.g. usage of hierarchical confidence refinement [88] or models for representing large subsets of the prototypes [38], therefore efficiency is not the main topic to be considered here.

Besides tangent distance, which is able to account for global transformations in the image like affine transformations, a method to compensate small *local* transformations is presented. This method, which yields a distance measure tolerant with respect to local distortions is called *image distortion model* (IDM) and is based on the following considerations. If, due to noise or artifacts irrelevant to classification, only a few pixels in two images have different values, this introduces possibly large distance components in the overall distance between the images. This can be compensated by specifying a region in the matching image for each picture element in which it is allowed to detect a best matching pixel.

The relationship between tangent distance and the image distortion model is considered in this work and a possible generalization is presented. Furthermore, the theoretical background of tangent distance is presented in a probabilistic framework. It can be shown that the tangent distance measure can be inferred from a probabilistic formulation of known intra-class variance. In connection with tangent distance, certain structures of covariance matrices are found and these can in turn be related to structures resulting from neighborhood systems in the images.

Classifiers implementing the above methods were evaluated on databases of different domain, coming from optical character recognition (OCR) and medical imaging. In the experiments carried out for this work an excellent error rate of 2.2% was obtained on the original US Postal Service handwritten digits recognition task. It was achieved using a kernel density based Bayesian classifier that incorporates tangent distance, virtual data and classifier combination and is the best result published on this specific recognition task so far.

Image object recognition has a strong connection to *image retrieval*, where the task is to retrieve a "matching" image from a (possibly large) database. If the desired similarity measure is based on the objects that are present in the image, which is the most common case, the connection is immediately evident. The best match can then be determined after the objects present have been recognized. The methods presented here may be used for such an indexing, although the described task seems to be a lot harder.

The work is organized as follows. Chapter 2 provides a short introduction to the basic notions of statistical pattern recognition and the classifier architectures used for the experiments, then a short summary of the goals of this work is given in Chapter 3. Chapters 4, 5 and 7 contain the main part of this thesis. Chapter 4 is concerned with approaches to invariant image object recognition, focusing on tangent distance, and introducing the image distortion model. The subsequent Chapter 5 presents the probabilistic theory that describes tangent distance and related approaches and the description of structured covariance matrices in relation to the image distortion model and Markov random fields. Chapter 6 describes the databases used and the state of the art in the field. Chapter 7 contains the results of experiments carried out and relates them to the previous descriptions. After a conclusion and perspective are given in Chapter 8, the appendix contains further experiments, some notes on implementation and an additional proof.

Parts of this work are accepted for publication in [51, 50, 20, 21] and have played a role in [23]:

- D. Keysers, J. Dahmen, T. Theiner, and H. Ney. Experiments with an Extended Tangent Distance. In *Proceedings 15th International Conference on Pattern Recognition*, Barcelona, Spain, September 2000. Accepted for publication.

- D. Keysers, J. Dahmen, and H. Ney. A Probabilistic View on Tangent Distance. In *22. DAGM Symposium Mustererkennung 2000*, Springer, Kiel, Germany, September 2000. Accepted for publication.

- J. Dahmen, D. Keysers, M. O. Güld, and H. Ney. Invariant Image Object Recognition using Mixture Densities. In *Proceedings 15th International Conference on Pattern Recognition*, Barcelona, Spain, September 2000. Accepted for publication.

- J. Dahmen, D. Keysers, M. Pitz, and H. Ney. Structured Covariance Matrices for Statistical Image Object Recognition. In *22. DAGM Symposium Mustererkennung 2000*, Springer, Kiel, Germany, September 2000. Accepted for publication.

- J. Dahmen, T. Theiner, D. Keysers, H. Ney, T. Lehmann, and B. Wein. Classification of Radiographs in the 'Image Retrieval in Medical Applications' System (IRMA). In *Proceedings of the 6th International RIAO Conference on Content-Based Multimedia Information Access, Paris, France*, pages 551–566, April 2000.

Chapter 2

Statistical Pattern Recognition

"Yes, by introducing some random element that can be shaped by that pattern."
"Like how?"
"Like by pulling Scrabble letters out of a bag blindfolded."

[2]

This chapter introduces the basic concepts of classification used in this work. Of course this chapter does not aim for a complete coverage of the subject of statistical pattern recognition. An in-depth introduction can be found for example in [27, 32]. For the basics presented here it is assumed that the reader has basic knowledge about (statistical) pattern recognition, e.g. from a lecture on the subject.

Recognition problems can be coarsely divided in problems with well-defined classes and more complex ones. For this work only the first type is considered, which includes questions like "Which digit is present in this image?" or "Which of the six defined regions of the human body does this radiography belong to?" The second type of problem may contain questions as "What can be seen in this image?" or "Is there a tumor present in this radiography?", and their rather complex nature is not subject of this thesis.

The "art" of pattern recognition is sometimes also called *machine learning* since the designed system is supposed to learn to automatically classify the given patterns. The subject of this chapter is termed *statistical* pattern recognition, because patterns to be classified usually are results of some sort of measurement and therefore subject to stochastical processes as e.g. noise. This in turn should be taken into account when the data is modeled, leading therefore to statistical models. The following sections are based in many parts on [73].

2.1 Pattern Recognition

Consider a number of *classes* to be distinguished given as $k = 1, \dots, K$. From an observed signal s and the extracted feature vector $x \in \mathrm{I\!R}^D$ the corresponding class shall be determined. To do so, a decision function $r : \mathrm{I\!R}^D \to \{1, \dots, K\}$ is needed, which is usually based on a discriminant function $g(x, k)$ by

$$r : x \longmapsto \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}} \{g(x, k)\} \tag{2.1}$$

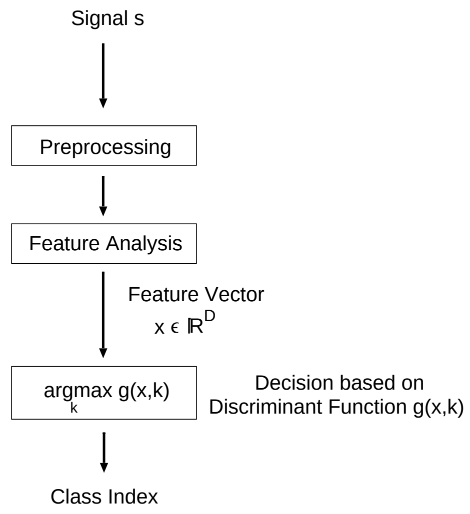

Figure 2.1: Typical Structure of a Recognition System

Figure 2.1 illustrates the basic structure of a classifier, which includes the feature extraction step $s \mapsto x$, which is regarded as given here. The discriminant function can be modeled in a wide variety of ways, including e.g. polynomial functions or artificial neural nets (ANN). The criterion for the discriminant function usually is

$$\begin{array}{ll} g(x,k) \longmapsto 1 & \text{for the "right" class} \\ g(x,k) \longmapsto 0 & \text{for the "false" class} \end{array} \tag{2.2}$$

which in general can only be approximated. In the statistical approach one considers the *a priori probability* density functions for the classes $p(k)$ and the *class conditional probability* density functions $p(x|k)$ for a feature vector given a class. From these the *a posteriori probability* density function $p(k|x)$ can be determined using Bayes' rule

$$p(k|x) = \frac{p(x|k)p(k)}{p(x)} = \frac{p(x|k)p(k)}{\sum_{k'=1}^{K} p(x|k')p(k')} \tag{2.3}$$

The a priori density is usually modeled by relative frequencies or in the case of digit recognition it is often set to $p(k) = \frac{1}{K}$. To determine the class for a given x the statistical approach uses *Bayes' decision rule*:

$$\begin{aligned} r(x) &= \underset{k}{\operatorname{argmax}} \{p(k|x)\} \\ &= \underset{k}{\operatorname{argmax}} \left\{ \frac{p(x,k)}{p(x)} \right\} \\ &= \underset{k}{\operatorname{argmax}} \{p(x,k)\} \\ &= \underset{k}{\operatorname{argmax}} \{p(k)p(x|k)\} \end{aligned} \tag{2.4}$$

that is, $g(x,k) = p(k|x)$ or equivalently (that is, leading to the same decision) $g(x,k) = p(k)p(x|k)$ or $g(x,k) = \log[p(k)p(x|k)]$. One can show that Bayes' rule is optimal for known distributions with respect to the expected error rate (for a proof see e.g. [27, 73]). Note that this implies the assumption of a cost function assigning cost one to a misclassification and cost zero to a correct classification.

Now, since the true distributions are usually unknown, the arising problems include finding suitable models for $p(k)$ and $p(x|k)$ respectively $g(x,k)$ and finding suitable criteria and algorithms to determine (respectively estimate) the free parameters in the models during the training phase. In pattern recognition one is often confronted with (and only this case is considered here) the case of *supervised learning*, that is construction of a classification procedure from a set of data for which the true classes are known. That means one is given a set of pairs (x_n, k_n), $n = 1, \ldots, N$ where x_n is a feature vector belonging to class k_n and is asked to determine (learn, estimate) the parameters for the classifier from this set. Usually, the criterion for the performance of the developed classifier is the *empirical error rate* (ER) which is given by the ratio of classification errors made on a test data set to the number of tests performed.

Maximum Likelihood Estimation

One widely used method to determine parameters from a set of given data is *maximum likelihood estimation*. Consider a density function $p(x|c, \vartheta_k)$ that depends on a parameter set ϑ_k, which in turn depends on the modeled class k. For each class N_k training vectors $x_{1k}, \ldots, x_{nk}, \ldots, x_{N_k k}$ are given. The *likelihood function* is then given by

$$\vartheta_k \longmapsto \prod_{n=1}^{N_k} p(x_{nk}|k, \vartheta_k) \tag{2.5}$$

respectively the *log-likelihood function* is

$$\vartheta_k \longmapsto \sum_{n=1}^{N_k} \log p(x_{nk}|k, \vartheta_k) \tag{2.6}$$

Then the *maximum likelihood estimator* $\hat{\vartheta}_k$ is defined by

$$\begin{aligned} \hat{\vartheta}_k &:= \operatorname*{argmax}_{\vartheta_k} \left\{ \prod_{n=1}^{N_k} p(x_{nk}|k, \vartheta_k) \right\} \\ &= \operatorname*{argmax}_{\vartheta_k} \left\{ \sum_{n=1}^{N_k} \log p(x_{nk}|k, \vartheta_k) \right\} \end{aligned} \tag{2.7}$$

The term *discriminative training* is used for approaches that take the a posteriori probability as a criterion for the training phase, for example

$$\vartheta \longmapsto \prod_{n=1}^{N} p(k_n|x_n, \vartheta) \tag{2.8}$$

respectively the logarithm

$$\vartheta \longmapsto \sum_{n=1}^{N} \log p(k_n|x_n, \vartheta) \tag{2.9}$$

These methods are discriminative, because they take into account the relation between the classes.

Relation to Distance Based Classifiers

Since distance based classifiers play an important role in this work, the connection to the statistical point of view is considered here (see also Chapter 5). Consider a Gaussian distribution (also called *normal distribution*)

$$p(x|k) = \mathcal{N}(x|\mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)\right) \tag{2.10}$$

where $|\cdot|$ denotes the determinant of a matrix, and consider the discriminant function $g(x,k) = \log[p(k)p(x|k)]$. If the terms constant in k are dropped, one arrives at

$$g(x,k) = -\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\Sigma_k| + \log p(k) \tag{2.11}$$

Ignoring the term $-\frac{1}{2}\log|\Sigma_k| + \log p(k)$ and defining

$$g(x,k) = -d_k(x,\mu_k) \tag{2.12}$$

with the so called *Mahalanobis distance*

$$d_k(x,\mu_k) = (x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k) \tag{2.13}$$

the decision rule finally becomes

$$r(x) = \operatorname*{argmin}_k \{d_k(x,\mu_k)\} \tag{2.14}$$

which is called *nearest neighbor* (NN) decision rule or nearest prototype / center / mean, respectively minimum distance rule. It can be shown that in the fictitious case of an infinite amount of training data the error rate for the NN classifier is at most twice the (optimal) Bayes error rate. The resulting classifier type is a special case of the k-nearest neighbor algorithm[1] for the choice $k=1$. In k-nearest neighbor classification the classes of the k closest prototypes to the observation x are considered and the decision of the classifier is based on different voting schemes, where each of the k prototypes has one vote (of possibly different weight).

If $\Sigma_k = \sigma^2 I$ with identity matrix I is assumed, the Mahalanobis distance becomes a (weighted) squared Euclidean distance which is a special case of the squared l_p norms[2] for $p=2$, where

$$\begin{aligned} d_{l_p}(x,\mu_k) &= ||x-\mu_k||_p^2 \\ &= \left[\sum_{d=1}^{D} |x_d - \mu_{kd}|^p\right]^{2/p} \end{aligned} \tag{2.15}$$

For $p=1$ this yields the squared city block distance and for $p \to \infty$ the squared maximum distance.

Training Set Size

For most applications, the size of the training set used has a strong influence on classification results. It seems obvious that a classifier, in particular one based on statistics, should perform better with increasing number of training samples. This is especially true for high-dimensional feature spaces (which is sometimes called the "curse of dimensionality"), which is related to the "emptiness" of high dimensional space. It is a general problem that only limited data is available for training. Having access to infinite training data and resources even a trivial algorithm would

[1] It should bec clear from the context, whether 'k' is meant to be the class number or the number of prototypes in the nearest neighbor classifier.

[2] In most contexts the norms themselves and not the squared norms are considered, but in the following it is easier to directly use the squared norms instead. Note that these usually do not meet the distance measure criterion $d(a,b) + d(b,c) \geq d(a,c)$ (triangle inequality).

perform optimally. One approach to alleviate this problem is to use a priori or domain knowledge for regularization, for example represented by tangent vectors, which will be introduced in detail in Section 4.2 together with the concept of tangent distance. On this aspect, SIMARD et al. comment that "using tangent distance or tangent propagation is like having a much larger database" [87]. In compliance with this is the empirical result stated by VAPNIK "As the number of examples increases [from 7291] to 60,000 the advantage of *a priori* knowledge decreased." [97, p. 159] (referring to the USPS and NIST databases, see Chapter 6).

With respect to the impact of the training set size in optical character recognition, one can find in [92] the statement "For every tenfold increase in database size the error rate is cut by half or more though the performance seems to be leveling off slightly for the larger database sizes." And furthermore "there is good reason to believe that performance will continue to improve as the training database grows even larger. In some ways, this is an obvious result. If the database is large enough it will eventually saturate the space of all possible bitmaps and the system could only fall short of perfect performance due to errors or noise in the training database." From this the authors deduce that "researchers might better spend their time collecting data than writing code."

Overview of Algorithms

The choice of the model or classifier to use is in general somewhat arbitrary, but an empirical analysis [93] shows that the accuracy of different algorithms depends on the data characteristics. For example k-NN performance decreases as the relative number of feature variables to the training cases increases. One can even show, that for each regularity that a given machine can learn there exists another regularity for that the machine does the opposite, that is it generalizes worse than a random classifier. This statement is sometimes referred to as "no free lunch".

In the following sections two statistical methods will be considered in more detail, namely Gaussian mixture densities and kernel densities. The statistical pattern recognition approach is one of the three main approaches besides the empirical based nonlinear approach for discriminant functions using *artificial neural nets* and the *support vector machine* approach based on statistical learning theory. This distinction between approaches is somewhat arbitrary, since e.g. a support vector machine can be seen in the context of statistical pattern recognition. Furthermore, it can be shown, that with respect to the squared error the global optimum of an ANN is reached if the discriminant function equals the a posteriori probability density function [73]. There also exists a variety of methods based on rules or decision trees. For an introduction to ANN see e.g. [41]. Interesting extensions of ANNs to achieve invariance with respect to given transformations called *tangent propagation* can be found in [87, 91].

A way to formalize learning a classification function from examples is *statistical learning theory* [97, 98, 99]. One central point of the analysis of learning algorithms is the so called VC dimension (Vapnik-Chervonenkis dimension, equal to the maximum number h of vectors from two classes which can be separated in all 2^h possible ways using (discriminant) functions of this set), which is related to such notions as generalization ability, minimum description length and overfitting [99]. One basic result is that there exists a tradeoff between the quality of approximation and the complexity of the approximating function. From statistical learning theory the support vector machine evolved, which uses optimal separating hyperplanes in high-dimensional feature spaces. It effectively transform patterns into high-dimensional space, constructs a hyperplane for separation of classes and thus allows algorithmic control of the VC-dimension. One empirical finding is that only few training examples are effectively used in constructing the hyperplanes, which are called *support vectors* and are characteristic for the data. Classification can then be done by comparison with the support vectors. SVMs can also be equipped with transformation invariance, central topic of this work, which leads to so called invariant support vector machines [81].

Support vector machines, methods like k-NN and kernel densities are usually considered *memory based* techniques, because (a subset of) the training samples is memorized and compared to the observation during the classification process. In contrast to this, methods like ANNs are regarded as *learned function* techniques, since the training data are here used to determine the free parameters of a discriminant function. One can argue that this distinction is arbitrary, because the memorized examples can be considered parameters of a complex function.

As a drawback of memory based methods it is sometimes seen that they are time consuming since the test pattern has to be compared to all stored references. To this problem a number of solutions have been proposed, including hierarchies of distances or models for representing large subsets of prototypes [87, 38, 58]. One can also use methods such as partial distance calculation, hierarchical structuring of the training vectors or related approaches. Furthermore, as computers grow faster, this steadily becomes less of a drawback MOORE's law about exponential growth in computational resources is supposed to become true for the next couple of generations of computers).

2.2 Gaussian Mixture Densities

One effective method to describe the conditional probability density is to assume that the data is distributed according to a linear mixture of multivariate Gaussian distributions, thus allowing multimodal distributions. This assumption does not impose any restriction on the modeling power, since the resulting *Gaussian mixture density* (GMD) can still approximate any density function with arbitrary precision.

First consider a unimodal Gaussian distribution

$$\begin{aligned} p(x|k) &= \mathcal{N}(x|\mu_k, \Sigma_k) \\ &= \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)\right) \end{aligned} \tag{2.16}$$

with the according maximum likelihood estimates

$$\hat{\mu_k} = \frac{1}{N_k} \sum_{n=1}^{N_k} x_{nk} \tag{2.17}$$

$$\hat{\Sigma_k} = \frac{1}{N_k} \sum_{n=1}^{N_k} (x_{nk}-\mu_k)(x_{nk}-\mu_k)^T \tag{2.18}$$

$$\tag{2.19}$$

Since in the experiments the setting $\Sigma_k = \sigma_k^2 I$ was used, here the maximum likelihood estimator for σ_k^2 is given (as one easily verifies by differentiating the log-likelihood)

$$\hat{\sigma_k^2} = \frac{1}{DN_k} \sum_{n=1}^{N_k} (x_{nk}-\mu_k)^T (x_{nk}-\mu_k) = \frac{1}{D}\mathrm{trace}(\hat{\Sigma_k}) \tag{2.20}$$

This means that the estimator equals the arithmetic mean of the diagonal elements of the empirical covariance matrix. Now, a Gaussian mixture is a linear combination of Gaussians

$$p(x|k) = \sum_{i=1}^{I_k} c_{ki} \cdot \mathcal{N}(x|\mu_{ki}, \Sigma_{ki}), \quad c_{ki} > 0, \quad \sum_{i=1}^{I_k} c_{ki} = 1 \tag{2.21}$$

with mixture weights c_{ki} and component densities $\mathcal{N}(x|\mu_{ki}, \Sigma_{ki})$. The maximum likelihood estimators for the parameters cannot be determined explicitly any more, but there exists an iterative algorithm which can be used for this purpose called EM-algorithm (Expectation-Maximization) [24, 73, 17, 18]. A classifier using GMD is also called *radial basis function* classifier (RBF) and produces the same type of decision rules as a support vector machine with Gaussian kernel [98]. For a short introduction to image object recognition using GMD see [18]. The use of GMD based classifiers has proven to be effective for image object recognition in various settings [17, 18, 19], and is a widely used method in speech recognition.

2.3 Kernel Densities

The description of the class conditional probability density function by *kernel densities* (KD) (also called parzen windows or parzen densities) can be seen as extreme case of GMD where each reference serves as a center of its "own" (usually, but not necessarily normal) distribution. That is, each training sample x_n defines a single density (e.g. Gaussian $\mathcal{N}(x|x_n, \Sigma_{x_n})$ with covariance matrix Σ_{x_n}), that is the sample itself is interpreted as mean vector. Although in general Σ_{x_n} may depend on the sample x_n, it is usually chosen to be equal for all considered x_n. The method belongs to the class of so called *nonparametric* procedures (as for example the nearest neighbor method) that can be used without assuming that the form of the underlying density is known [27, p. 85]. Since all the training patterns are kept and compared to the observation, this method is also closely related to the (k-)NN technique. A good informal description in the context of digit recognition can be found in [43]: "For instance, kernel density estimation [...] is a popular nonparametric modeling technique. For this, the probability density for a particular digit is the weighted sum of a collection of kernel functions. The functions all have the same shape, but each is centered on one of the patterns in that class in the training set. Each kernel function typically integrates to one and the weights in the sum are usually $1/M$, where M is the number of the patterns in the training set, so the overall kernel density estimate is correctly normalized. Having built ten such models, one for each digit class, the class to which the a new image belongs is inferred by evaluating the density under each of the models at the location of the new image, and reporting the one that is highest. If the kernel functions are radially symmetric, monotonically decreasing, and have unbounded extent (e.g. a Gaussian), then relative density estimation becomes identical to nearest neighbor classification as the width parameter of the kernel goes to zero." Since each training sample defines its own density center the covariance matrix must be chosen by other methods than maximum likelihood, because ML estimation leads to zero variances in this case. To this problem FUKUNAGA writes "The neighborhoods should take the same ellipsoidal shape as the underlying distribution." [32, p. 267]

Starting with a kernel function $\varphi_k(x)$ that is itself a probability density function usually centered around zero (possibly depending on the class k) the kernel density approximation of the class conditional probability density function is

$$p(x|k) = \frac{1}{N_k} \sum_{n=1}^{N_k} \varphi_k(x - x_{nk}) \tag{2.22}$$

and using a Gaussian kernel this becomes

$$p(x|k) \quad = \quad \frac{1}{N_k} \sum_{n=1}^{N_k} \mathcal{N}(x|x_{nk}, \Sigma_{x_{nk}})$$

$$= \frac{1}{N_k} \sum_{n=1}^{N_k} \frac{1}{\sqrt{(2\pi)^D |\Sigma_{x_{nk}}|}} \exp\left(-\frac{1}{2}(x - x_{nk})^T \Sigma_{x_{nk}}^{-1} (x - x_{nk})\right) \tag{2.23}$$

Inserting this into Bayes' rule together with the ML-estimation $p(k) = \frac{N_k}{N}$ yields the decision rule

$$\begin{aligned} r(x) &= \underset{k}{\operatorname{argmax}} \{p(k)p(x|k)\} \\ &= \underset{k}{\operatorname{argmax}} \left\{ \frac{N_k}{N} \frac{1}{N_k} \sum_{n=1}^{N_k} \frac{1}{\sqrt{(2\pi)^D |\Sigma_{x_{nk}}|}} \exp\left(-\frac{1}{2}(x - x_{nk})^T \Sigma_{x_{nk}}^{-1} (x - x_{nk})\right) \right\} \\ &= \underset{k}{\operatorname{argmax}} \left\{ \frac{1}{\sqrt{|\Sigma_{x_{nk}}|}} \sum_{n=1}^{N_k} \exp\left(-\frac{1}{2} d_{x_{nk}}(x)\right) \right\} \end{aligned} \tag{2.24}$$

where $d_{x_{nk}}(x)$ represents the Mahalanobis distance of x to x_{nk}. Now the KD based classifier can be used with different other distance measures. For example the squared Euclidean distance could be used or distance measures that are invariant with respect to some transformation as e.g. tangent distance which will be introduced in Chapter 4. Consider for example the setting of $\Sigma_{x_{nk}} = \sigma_k^2 I$, which was used in the experiments with Euclidean distance. Then the decision rule becomes

$$r(x) = \underset{k}{\operatorname{argmax}} \left\{ \frac{1}{\sigma_k^D} \exp\left(-\frac{1}{2\sigma_k^2} ||x - x_{nk}||^2\right) \right\} \tag{2.25}$$

To compensate for the fact that variances are usually underestimated using the limited amount of training data, one can multiply the variances σ_k by a constant factor greater than one.

Because of the exponential decay with increasing distance only the reference patterns closest to the test pattern result in a significant contribution to the sum. The experiments with digit recognition showed that using more than the ten closest matches does usually not change classification results. Note that this can be interpreted as a probabilistic justification for the use of k-NN based classifiers. For these it is generally sufficient to compute the distance for the 100 closest references, which can be efficiently determined using Euclidean distance, thus justifying the hierarchical filtering approach presented in [88].

To avoid numerical instabilities with exponentiation when implementing the kernel density based classifier one may choose the following method. First, all distances needed are calculated and the minimum distance $d_{\min}$ is determined. Then the probabilities may be calculated by

$$p(x|k) = \exp\left(-\frac{1}{2} d_{\min}\right) \frac{1}{\sqrt{|\Sigma_{x_{nk}}|}} \sum_{n=1}^{N_k} \exp\left(-\frac{1}{2}(d_{x_{nk}}(x) - d_{\min})\right) \tag{2.26}$$

where the leading factor may be dropped for classification purposes, since it does not depend on the class. This method assures that the exponential terms in the sum stay in ranges that are numerically more stable (at least the term with minimum distance has the value one).

2.4 Feature Reduction

A typical problem for statistical Bayesian classifiers based on Gaussian mixture densities or kernel densities is the estimation of covariance matrices. In case of the USPS task (see Chapter 6), with feature vectors $x \in \mathbb{R}^{256}$, a single covariance matrix requires (due to symmetry) the estimation of $256 \cdot (256 + 1)/2 = 32.896$ parameters. Given only 7.291 training samples, this is infeasible. A common approach to overcome this difficulty is the use of variance pooling

- *class specific variance pooling* :
  estimate only a single Σ_k for each class k, i.e. $\Sigma_{ki} = \Sigma_k \;\forall\; i = 1, ..., I_k$
- *global variance pooling* :
  estimate only a single Σ, i.e. $\Sigma_{ki} = \Sigma \;\forall\; k = 1, ..., K$ and $\forall\; i = 1, ..., I_k$

in combination with diagonal covariance matrices, i.e. variance vectors.

Another way to overcome the difficulties with the estimation of covariance matrices is the use of *feature reduction.* Employing feature reduction the aim is to capture the essential information of the high dimensional feature vector in a smaller number of features, usually by means of a linear transformation of the feature space, but nonlinear methods are also used [40]. In the following sections two methods frequently used are presented.

Karhunen-Loève Transformation, Principal Components Analysis

The *Karhunen-Loève Transformation* (KLT) or *Principal Components Analysis* (PCA) is a linear transformation aimed at minimizing the representation error. After calculating the (empirical) covariance matrix Σ, it is diagonalized using an eigenvector decomposition with eigenvectors $v_1, \ldots, v_D$ and corresponding eigenvalues $\lambda_1, \ldots, \lambda_D$ sorted in decreasing order, i.e. $\lambda_d \geq \lambda_{d+1}, d = 1, \ldots, D-1$. This decomposition can be achieved e.g. using a *singular value decomposition* (SVD) [79]. Then Σ can be written as

$$\begin{aligned}
\Sigma &= \sum_{i=1}^{D} \lambda_i v_i v_i^T \\
&= [v_1 \cdots v_D] \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_D \end{pmatrix} [v_1 \cdots v_D]^T \\
&= [v_1 \cdots v_D] \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_D \end{pmatrix}^{\frac{1}{2}} \left([v_1 \cdots v_D] \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_D \end{pmatrix}^{\frac{1}{2}} \right)^T \\
&= \Sigma^{\frac{1}{2}} \left(\Sigma^{\frac{1}{2}} \right)^T
\end{aligned} \tag{2.27}$$

where the last steps are given in order to help the considerations of Section 5.1.2, where $\Sigma^{-\frac{1}{2}}$ is used, being the inverse of $\Sigma^{\frac{1}{2}}$. ($\Sigma^{-\frac{1}{2}}$ is also the transformation matrix of the *whitening transformation.* After application of the whitening transformation the covariance matrix in the transformed space is equal to the identity matrix and the distribution is called 'white' (compare [32, pp. 26ff]).) The eigenvectors $v_1, \ldots, v_d$ (for some d fixed or to determine) corresponding to the largest eigenvalues are also referred to as *principal components.* Now the KLT or PCA consist in representing each vector by its projection onto the principal components, which is a linear transformation $x \in \mathbb{R}^D \mapsto \hat{x} \in \mathbb{R}^d$ with the matrix representation of the transformation being $[v_1 \cdots v_d]$, which has the property that the expected error $E\{||x - \hat{x}||^2\}$ is minimal for all linear transformations to d dimensions. Note that PCA discards the directions of small variance. One now hopes that the transformation captures the most relevant part of the information contained in the vectors x. This point of view of information based on magnitude of variance and minimal reconstruction error may not be suitable for classification purposes, since it does not take into account the class information and there are various examples for this fact [82, p. 116].

Linear Discriminant Analysis

The *linear discriminant analysis* (LDA), also called Fisher's LDA takes into account the class information in feature reduction [27, pp. 118ff]. It tries to simultaneously maximize the distances between the class centers μ_k and to keep the distances within one class constant. This can be achieved using within-class and between-class scatter matrices, leading to a generalized eigenvalue problem. Another method leading to the same result is to employ a whitening transformation and then (using the fact that the within-class scatter matrix is the identity matrix then) use the subspace spanned by the vectors $\mu_k - \mu$ where μ is the total mean vector. This subspace method is numerically more robust in some cases [17]. The dimension of the obtained subspace is at most $K-1$, which might be too small for some applications. A method to circumvent this problem is to create so called 'pseudoclasses' using for example the EM-algorithm and then use LDA within the new problem with $K' > K$ classes yielding at most a $K'-1$-dimensional feature space. The LDA has the advantage over the PCA that it takes into account the available class information and aims at maximizing the separability of the classes, which is usually wanted in pattern recognition.

2.5 Holographic Classifiers

In this section a classification algorithm is presented which is algorithmically based on artificial neural nets, but deriving its motivation from the phenomenon of optical holography [33], therefore called *holographic classifier*. The method to be described here implements an associative memory and is therefore also called *holographic associative memory*. The approach was introduced by KHAN and presented in [55, 52, 56, 53, 54], especially in the context of content based image retrieval. A discussion of the method can be found in [48].

At first sight the connection between associative memory[3] and pattern recognition might not be apparent, but any associative memory relies inherently on a specific distance measure that determines the closeness of an input pattern to the samples presented during training and therefore the resulting output. An associative memory – also called content addressable memory – is equipped with a learning algorithm which transforms a set of given stimulus-response pairs into a certain joint representation and a decoding algorithm which determines the response to a given query stimulus according to the inherent distance measure on the stimuli. If the training data for a classification problem now is considered as a set of pairs of stimuli (feature vectors) and responses (class labels), an associative memory performs a classification task. It remains to say that there might be different targets for the two viewpoints. In pattern recognition the aim is to reduce the classification error rate, while for an associative memory this may not be the most important aspect.

Holography has been used in hardware realization (see e.g. [80]) as a memory medium and is subject of current research, because it allows high density distributed information storage. The physical model of holography can be described mathematically in various ways (which will not be considered here), leading to a possible description of a discrete hologram. This in turn can be used to model the process of holography in software. In the holographic paradigm the associative memory is bimodal and is represented as a complex hologram – that is a complex matrix – which allows modulation of assertion / attention / confidence using the amplitude of the complex domain. Based on the physical model one can derive the description of calculations necessary to simulate

[3]The topic of associative memories has been addressed by KOHONEN [60, 61, 62].

holography in software. For that it is necessary to transform the feature vectors to a complex representation. (KHAN suggests the use of *multidimensional* complex numbers but leaves open the way to handle those.) The desired function should map the assertive value onto the magnitude and the feature value onto the phase, such that each feature vector is transformed to a vector of complex numbers. This is done for stimulus and response, that is feature vector and class label in this context. Great care must be taken in choosing the mapping function, since its characteristics together with the data characteristics determine the performance of the algorithm. This data dependency and lack of rules for the choice may be seen as a severe drawback of the method. The desired goal in the mapping of features is to reach a high symmetry in the transformed data, meaning that on the average the sum of all complex representations should be close to zero.

For each training vector x_n and corresponding class k_n let x'_n and k'_n denote the complex representations. The representation used for holographic classification may in general have a dimensionality different from the original vectors, since a minimum dimensionality is desired. This is due to effects in the hologram that occur, when the load, defined as the number of stored patterns divided by the feature vector length, exceeds a certain threshold. If the holographic paradigm is used for classification purposes, it seems reasonable, that this load threshold can be higher than for associative memory, since it is not desired to distinguish elements belonging to one class. A higher dimensionality can be achieved by using outer products for the feature vectors or binary representation for the class labels. After transfer to the complex domain for each training vector the correlation matrix

$$h_n = \overline{x'_n}^T \cdot k'_n \tag{2.28}$$

is defined, where the bar denotes the complex conjugate. Adding up these yields the hologram

$$h = \sum_{n=1}^{N} h_n = \sum_{n=1}^{N} \overline{x'_n}^T \cdot k'_n \tag{2.29}$$

The hologram itself is usually modified in training using discriminative training procedures similar to backpropagation for artificial neural networks.

For holographic reproduction (which is the first step of classification if the method is used for that aim) the observation x is transformed to the according complex representation x' and then multiplied with the hologram, yielding the complex representation of the answer

$$k' = \frac{1}{c} x' \cdot h \tag{2.30}$$

with c being a normalization factor equal to the sum of magnitudes of the elements of the complex representation x' (of dimension J). The magnitude of k' now is an indicator for the confidence in the given answer. For use in classification a second step needs to be performed, which is finding the class number k best matching the complex representation k', using the inverse of the mapping function used to transform the class labels to the complex domain.

The classification procedure inherently relies on a specific distance measure for similarity, which can be written as (without proof given here)

$$1 - d(x', \mu') = \frac{1}{c} \sum_{j=1}^{J} |x'_j| \, |\mu'_j| \left(\cos(\mathrm{phase}(x'_j) - \mathrm{phase}(\mu'_j)) + i \sin(\mathrm{phase}(x'_j) - \mathrm{phase}(\mu'_j))\right) \tag{2.31}$$

For a "unary" representation of the class label, with

$$k'_{ni} = \begin{cases} 1 & i = k \\ -1 & i \neq k \end{cases}, \quad i = 1, \ldots, K \tag{2.32}$$

the discriminant function can then be explicitly given as

$$g(x,k) = \sum_{n=1}^{N_k} 1 - d(x', x'_{nk}) - \sum_{k^*=1, k^* \neq k}^{K} \sum_{n=1}^{N_k^*} 1 - d(x', x'_{nk^*}) \tag{2.33}$$

Assuming symmetrical distribution of the values represented by the hologram one can show that the discriminant function is a function of the weighted sum of cosines of the phase difference of the complex pattern representations (the sine components cancel out on the average under the given assumption). Looking at the corresponding distance function $1 - \cos x$ in comparison with the underlying distance function of the Euclidean function x^2, the basic effect is, that large differences in feature values do not contribute quadratically more to the total distance than lesser differences, as is the case for the square function. The two feature distances are depicted in Figure 2.2. A

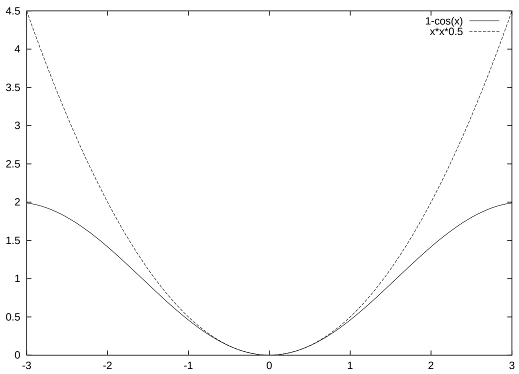

Figure 2.2: Comparison of cosine and square feature distance

very similar effect can be achieved by thresholding the individual feature distances at a certain level, which was used on the IRMA database (compare Chapter 7). The effect of changing the contribution function is task dependent, though.

The holographic method combines naturally with the usage of the Fourier transformation for feature extraction, since the Fourier transform is a complex representation of the signal, where magnitudes correspond to importance of a certain frequency in the image, but the phase information contains highly relevant information in the case of image processing [65, p. 140]. If the Fourier transform is used, there exists a connection to symmetric phase-only matched filtering (SPOMF, see e.g. [15]), where the emphasis is also on the *phase* of the transformed image. The basic differences are, that in SPOMF the magnitudes are completely disregarded, while backtransformation is done using the inverse Fourier transform rather than a sum of cosines measure.

Some of the advantages for the method of holographic classification are that with a binary representation of class labels a logarithmic reduction in complexity for large number of classes compared to ANN can be obtained and that translation invariant recognition can be achieved without great cost by using the fast Fourier transformation and the convolution theorem. Among the disadvantages one can find that it is seemingly very difficult to model the mapping of real features to complex ones with respect to the output symmetry, which is connected to the specific type of variability present in the data. Furthermore the method has not been thoroughly investigated in software and it is unclear whether the associative paradigm inherent in the method is suitable for pattern recognition.

Chapter 3

Goal of this Work

He smiled with a curious kind of manic joy as he flipped again through the mysteriously re-instated entry on the planet Earth. He had a major piece of unfinished business that he would now be able to attend to, and was terribly pleased that life had suddenly furnished him with a serious goal to achieve.

[4]

This section gives a short overview of the aim of this work. It originated from the interest in the subject of invariant image object recognition, especially the use of *tangent distance*, at the Chair of Computer Science VI (i6), and it was desired to perform a deeper investigation on this subject. Thus, the goals of this work are:

- The description of current research in the field of invariant image object recognition and invariant distance measures, including
  - the evaluation of existing publications related to the subject and
  - the development of possible extensions or new models.
- The theoretical study of topics related to invariant image object recognition, including
  - the examination and description of the tangent distance model within a probabilistic framework and
  - the investigation of statistical properties of classifiers for image object recognition and the relevance of domain knowledge for them.
- The experimental investigation of invariance models in image object recognition, as well as the proposed extensions, including
  - the implementation of algorithms apt to achieve invariance in image object recognition and their incorporation into a statistical classifier,
  - the investigation of the properties of tangent distance and other invariant distance measures with respect to alternative approaches for invariance and with respect to different tasks and
  - the evaluation of the implemented classifiers with emphasis placed on the empirical error rate and the comparison of the achieved results to those of other state of the art classifiers.

This work describes the results obtained and the experience gained in the course of research and implementation in the field of invariant image object recognition.

Chapter 4

Invariant Image Object Recognition

> *"Yes," said Deep Thought. "Life, the Universe, and Everything. There is an answer. But," he added, "I'll have to think about it."*
>
> [1]

After some basic methods for pattern recognition have been introduced in Chapter 2, this chapter is concerned with various methods one can apply to achieve invariance of the classification process with respect to certain transformations. That is, one may be interested in the design of a classifier that does not change its output when the pattern to be classified changes under some transformation. Since this work concentrates on images as patterns, typical transformations of the patterns include affine or projective transformations, although some of the methods presented can be applied to arbitrary transformations as well. The reason for the importance of invariance is that in many cases there exists domain knowledge about invariant transformations that do not affect class-membership, so it is desired for the classifier to eliminate irrelevant variabilities, but to identify meaningful differences. One example for the importance of invariance in image recognition is depicted in Figure 4.1. Here an observation pattern is shown, which contains the object of a handwritten digit '7'. If it is compared with the two references on the right side, a classifier based on Euclidean distance would find that it is closer to the first reference, showing an image of the digit '9', because the sum of squared grayvalue differences is smaller than the one for the second, 'correct' reference. If the classifier used a distance measure invariant to line thickness of drawings, it would find that the 'correct' image is actually more similar to the observation and therefore

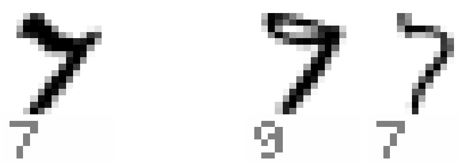

Figure 4.1: Pattern to be classified (left), two prototypes. According to Euclidean distance the pattern to be classified is closer to the first prototype. A distance measure invariant to line thickness should find that the second, correct prototype is closer. (Compare [87])

correctly classify the given pattern. Note that if a sufficiently large set of training data is available, it would probably contain also versions of the digit '7' with modified line thickness, such that the advantage of invariance would be reduced.

In this work an image x is considered a real valued function on a discrete image grid consisting of pixel locations from $\mathcal{I} \times \mathcal{J} = \{1,\ldots,I\} \times \{1,\ldots J\}$, that is $x \in \mathbb{R}^{I \times J}$. On the other hand an image can be considered a simple feature vector with one dimensional indices and dimension $D = I \cdot J$. The graylevel value of an individual pixel at pixel position (i,j) will generally be denoted with x_{ij}. The modeling of images (or filters etc.) is in many cases done in the continuous domain, since the discrete plane is difficult to handle. In that case, an image is considered a function $x : \mathbb{R}^2 \to \mathbb{R}$ and one can consider the discrete version as the result of sampling the continuous function. The considerations presented are mainly based on the paradigm of *appearance based pattern recognition*, that is the regarded features are equal to the sequence of pixel values. Other approachess include the extraction of local or global features, e.g. color, shape or texture.

There is an inherent connection between invariant recognition and image *registration*. The term registration refers to the mapping of images with the same or nearly identical content onto each other, such that the important structures are in the same image position. This is an important paradigm for example in medical imaging, when images have to be compared that were produced at different points in time. Usually, registration assumes images of the same content. But when a powerful registration algorithm is at hand, it can be applied to invariant recognition, by using it for normalization or determining the mapping function, hypothesizing each class in question and comparing the results. On the other hand, when an invariant classification algorithm is known that can return the transformation that connects two given patterns (which is the case for tangent distance), the registration problem is solved as a by-product.

4.1 Invariant Classification

This section aims at giving an overview of the different methods for invariant classification, before some of the methods are regarded in more detail. There exists a variety of techniques for solving the problem of invariant pattern recognition [101]: "Such techniques include integral transforms, construction of algebraic moments and the use of structured neural networks. In all cases we assume that the nature of the invariance group is known a priori." The last statement is quite essential in most approaches. In contrast to the restriction to domain knowledge, a method to estimate the derivatives of transformation from the given data is presented in Section 5.1. One approach not mentioned here is the use of invariant distance measures, which play an important role for this work. WOOD furthermore states [101]: "Since we have prior knowledge of the classification problem, we should be able to improve the generalization ability of any given pattern classifier by incorporating this knowledge into the classification system." Moreover, the author introduces a distinction between invariance and tolerance, which will not be considered here, since in practice complete invariance is often not obtainable or even not desired: "in practice only an approximate invariance (which we might call transformation tolerance) may be obtainable. This may arise through computational inaccuracies combined with the continuous nature of some transformation groups."[101] A theoretical statement of invariance can be given as follows [101]: Consider patterns as functions on some set, e.g. in image recognition $x : (i,j) \in I \times J \mapsto x_{ij}$, furthermore there exists a classification function which maps patterns onto class numbers, e.g. $r : x \mapsto 1,\ldots,K$ and a transformation group $\mathcal{G}$ which acts on the set the pattern is defined on and therefore on the pattern space, e.g. $g \in \mathcal{G} : (gx)_{ij} = x_{g^{-1}(i,j)}$, and does not affect class membership. Thus the desired classification function should be invariant under the action of the group, that is $r(gx) = r(x) \forall g \in \mathcal{G}$. That is, the patterns with the same invariant content form an equivalence class with respect to a group operation describing the geometric transform [14]. In practice, in some cases one may want to restrict the actions of the group, e.g. in digit recognition in order to distinguish between the digits '6' and '9'. This is sometimes referred to as *6-9-problem*. Other properties of interest in invariant classification include discriminability, computational complexity, ease and speed of training, generalization ability, flexibility and the possibility of transformation retrieval. Note that discriminability is an important aspect here, as for instance a mapping of any feature vector to a constant value yields a perfectly invariant mapping, which of course is useless for classification. In some cases one may want to distinguish between global and local invariances, depending on the context and the given data, but this distinction can be reduced to the assumption of different transformations which are present. One trivial solution to the problem of invariance in pattern recognition is employing brute force. In this context this means to compare all the possible transformations of the patterns and extract the optimal coincidence.

In the following some different methods to deal with known invariances are presented. The distinction between the approaches is somewhat arbitrary, for example one can regard normalization as a process of invariant feature extraction (normalized images are of course invariant with respect to the chosen transformations) or one can define an invariant distance measure as the distance of the normalized images. A further (equally arbitrary) distinction can be made concerning the time step the invariant process takes place, since normalization and feature extraction usually are performed before the actual classification process, whereas invariant distance measures and classifier combination are methods used in later steps of the classification procedure.

4.1.1 Normalization

With the term *normalization* one usually refers to the construction of a canonical representation for each pattern with respect to the regarded transformations. These representations can then be compared without the influence of the differences of the transformations. One drawback of such methods is that they may be very sensitive to noise and artifacts in the patterns.

For example one may use the following normalization procedure in order to achieve invariance with respect to rotation, translation and scale for images (sometimes referred to as RST-invariance) [35, 101]:

- compute the center of gravity and translate the origin to that point (translation-invariance)
- normalize for average radius (scale-invariance)
- rotate such that direction of maximum variance coincides with x-axis (rotation-invariance)

Fourier Spectrum Normalization

One method that was developed during this work involves the computation of the Fourier transform. Since the amplitude of the frequency spectrum of the Fourier transform is invariant under translation, it is frequently used for the extraction of invariant features. If the Fourier transform is used in this way, usually the Phase information is neglected. Now the idea is to use the phase information for translation *normalization*, the straightforward solution being to transform

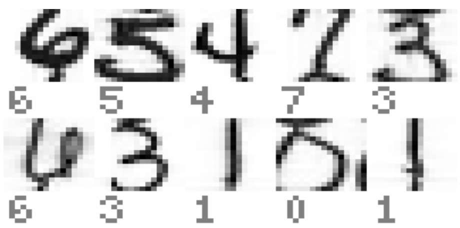

Figure 4.2: Result of normalization setting the first coefficients in the frequency domain to phase zero

the image in such a way that the first coefficients in each dimension (corresponding to the lowest frequencies) have phase zero. This corresponds to two degrees of freedom equivalent to the x/y translation offsets, which are lost. This procedure, carried out in the frequency domain, has effects on the remaining phases as well, which can be easily inferred from the definition of the Fourier transform. The obtained result is that for consistency the phase differences at a particular point in the frequency domain are given by the sum of the index of the point in each dimension multiplied by the phase difference for the 'first' coefficients. That is, let $\varphi_1 = \text{phase}(X(1,0))$ and $\varphi_2 = \text{phase}(X(0,1))$, then the transformation is given by the assignment of new phase information according to

$$\text{phase}(X(i,j)) \leftarrow \text{phase}(X(i,j)) - (i \cdot \varphi_1 + j \cdot \varphi_2) \tag{4.1}$$

Improvements to this straightforward solution are probably possible, since the changes in phase depend on only two of the $I \cdot J$ phases. Figure 4.2 shows the result of the applied normalization after application of the inverse Fourier transform. The results are not very convincing, but one advantage is, that most of the phase information can be kept using this method. No further experiments have been performed yet and it is still open how the phase information could be easily used for classification (note that the phase information is inherently 'wrap-around').

4.1.2 Invariant Features

If one wants to obtain a classification procedure that is invariant with respect to certain transformations, another approach is to calculate a set of features from the pattern, which is not affected by these transformations but still contains all information relevant for classification. This ideal view of *invariant features* can be expressed as [101]: "A complete system of invariants must be able to distinguish with arbitrary precision between any two vectors not in the same orbit under $\mathcal{G}$; i.e. the system must possess perfect discriminability." But in practice it is the case that a "complete set of continuous invariants under a given representation of a given group does not always exist." [101] On the other hand complete invariance is not always wanted, for example for digits, as complete invariance with respect to rotation would lead to the mentioned ambiguity between the digits '6' and '9'. Yet, invariant features may be very useful for other data as for example images of red blood cells [19]. The process is described in [83] as "extraction of suitable features in signal space prior to classification. These features should represent the patterns in S [the signal space] uniquely up to redundant information; i.e. only patterns differing in superfluous parts should have the same feature vector. Although this is a sound theoretical concept, no general strategy for feature extraction is known. Sometimes it is even difficult to characterize the superfluous part of the information in S. In many cases, however, it is possible to trace back this redundancy to the action of a group G on S." Yet, it must be considered that an invariant feature space does not exist for all kinds of transformation. In [83] the "nonexistence of such a space for the dilations and any group containing the dilations as a subgroup" is proven. This can be illustrated by looking at the scaling transformation. If features are required to be invariant with respect to scaling, all images should lead to the same features as a single point.

A number of performance aspects for invariant features is presented in [14], which include completeness (ability to discriminate between all possible images), robustness (tolerate deterministic and stochastic errors), continuity (clustering, metric) and computational complexity.

Features based on the Fourier Transform

Looking at the nature of invariant features that are extracted from images, one can distinguish two main classes, those based on algebraic invariants (considered farther below) and "invariants which are computable by integral transformations. Such transformations are generally based on the Fourier transform (FT) and its variants"[101]. The continuous one dimensional Fourier transform of a signal $f(t)$ and the corresponding inverse Fourier transform is defined by [49]

$$\begin{aligned} F(\omega) &= \int_{-\infty}^{+\infty} f(t)\exp(-i\omega t)dt \\ &= \int_{-\infty}^{+\infty} f(t)(\cos(-\omega t) + i\sin(-\omega t))dt \\ f(t) &= \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)\exp(i\omega t)d\omega \end{aligned} \tag{4.2}$$

and the *discrete Fourier transform* (DFT) of a one dimensional signal and its inverse can be defined by

$$\begin{aligned} F(k) &= \sum_{m=0}^{M-1} f(m)\exp\left(\frac{-i2\pi mk}{M}\right) \qquad k = 0, 1, \ldots, M-1 \\ f(m) &= \frac{1}{M}\sum_{k=0}^{M-1} F(k)\exp\left(\frac{i2\pi km}{M}\right) \qquad m = 0, 1, \ldots, M-1 \end{aligned} \tag{4.3}$$

The Fourier transform is an important and well known tool in many areas, which is partly due to the existence of an efficient algorithm for the calculation of the DFT if the pattern size is an integer power of two in all dimensions, called *fast Fourier transform* (FFT). The FFT in combination with the convolution theorem also allows to efficiently calculate discrete convolutions. These aspects of the FT shall not be considered here but for the extraction of invariant feature another property of the FT is important, which is the invariance of the squared magnitude of the FT spectrum (also called power spectrum) under translation of the pattern [65, 79, 74]. This is connected to the fact that a translation of the pattern corresponds to a phase shift in the Fourier domain, which does not affect the magnitude. Using this invariance property of the power spectrum one can obtain a set of features invariant under translation by using the power spectrum of a given pattern as feature vector. Doing this, one must be aware of the fact that by ignoring the phase of the spectrum a lot of information is lost, which might be important for classification. This is reflected in the fact that the power spectrum of a real valued image is symmetric and therefore the resulting feature vector has effectively only half the number of dimensions as the original vector.

If the FT is applied to higher dimensional object, as for example a two dimensional image, the equations and properties extend analogously. What should be mentioned is the fact that the spectrum of the FT is rotation *variant*, i.e. a rotation of the image is reflected in a rotation of the Fourier spectrum, while it is inversely variant with respect to scaling.

Features based on the Fourier-Mellin transform

If more than just translation-invariance is desired, this can be achieved with variants of the Fourier transform, e.g. the *Mellin transform*. This a Fourier transform evaluated over an exponential scale, which is invariant under the scaling transformation [77, 101]. If aspects of the Fourier and Mellin transform are combined in two steps together with a transformation to polar coordinates of an image (resulting in a circular Fourier, radial Mellin transform), one can achieve invariance with respect to rotation, scaling and translation simultaneously. The resulting transform is called *Fourier-Mellin transform* and can be calculated in the following way [101]:

(1) Calculate the power spectrum of the Fourier transform of the two-dimensional input. This is invariant under translation.

(2) Convert the power spectrum to polar coordinates. This converts rotations to translations.

(3) Perform a complex-log mapping. This converts scalings to translations.

(4) Calculate another two-dimensional Fourier transform power spectrum. This will be rotation-, scale- and translation-invariant.

The resulting features are now RST-invariant, but a lot of information is lost due to usage of only magnitudes in steps (1) and (4).

Local Features and Fourier Transform

The FT can also be employed to obtain locally rotation- and scale-invariant features in combination with the Gabor transform or the Wavelet transform. Consider for example the Gabor transform for the extraction of local features. The Gabor transform [49, 76, 65] is a so called windowed FT or short-time FT (here the one dimensional case)

$$G_f(\omega,\tau) = \int_{-\infty}^{+\infty} f(t) g_\alpha(t-\tau) \exp(-i\omega t) dt \tag{4.4}$$

with a Gaussian window of the form

$$g_\alpha(t) = \frac{1}{2\sqrt{\pi\alpha}} \exp\left(-\frac{t^2}{4\alpha}\right) \tag{4.5}$$

which is especially used for texture classification [7, 9]. It can also be used to extract additional features for image classification, for example the gradient can be considered a special case of a Gabor transform for low frequency, which was helpful in classification of chair images [18]. Now if the answers of a set of two dimensional Gabor filters for different angles and different frequencies are arranged on a grid, the DFT can be used to extract local features, which are invariant under rotation and scaling [31].

Fourier Descriptors, Complete Feature Spaces and Monomials

Another application of the FT is the extraction of *Fourier descriptors* for binary images. They can be obtained by parameterizing the object boundary and analyzing the Fourier transform of the resulting boundary function [14]. These Fourier descriptors are invariant with respect to translation and rotation and can be enhanced for affine invariance. The Fourier descriptors for shape can be generalized to grayscale objects (given a separation from the background) by not only parameterizing the object boundary but also the grayvalue distribution. BURKHARDT et al. state that the performance of affine invariant gray level Fourier descriptors is superior to that of affine invariant moments, "because they are less sensitive to noise in real applications"[14].

The authors furthermore derive some results about the existence of polynomial invariant *complete feature spaces*, which allow to distinguish different patterns, but yield the same features for patterns that are transformed. They prove that there exists a complete feature space, if two conditions are fulfilled:

1. The representations of the transformation group are completely reducible (and therefore the set of invariants is finitely generated).
2. The orbits of the transformation group are closed in the Zariski topology (and therefore separating polynomial invariants exist).

For a lack of space (and time) the complete elaboration of the notions is not feasible in this presentation. But it might be interesting to observe, that from the theorem it follows, that for any *finite* group a complete feature space exists.

Furthermore, the authors give constructive results about invariants for finite transformation groups and show that a basis of invariants can be given using *group averages*. A group average $\tilde{f}$ of a polynomial f is defined by

$$\tilde{f}(x) := \sum_{g \in G} f(g(x)). \tag{4.6}$$

A basis can then be constructed by calculating all *monomials* $x_0^{b_0} x_1^{b_1} \cdots x_N^{b_N}$ with the sum of the exponents less then or equal to the group order. The calculation of a basis from group averages becomes impractical for large dimensions of the signal space, but using certain mappings it is possible to obtain high separability with only a few group averages of monomials [14].

Features based on Moments

Algebraic invariants, or moment invariants, are obtained by taking quotients and powers of moments. A moment is a weighted sum of the pattern x_{ij} over the whole input field, with weights equal to some polynomial in i, j [101]. The geometrical moments or regular moments in two dimensions are defined by

$$m_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p \, y^q \, f(x,y) \, dx \, dy \tag{4.7}$$

and analogously for discrete functions as digital images by

$$m_{p,q} = \sum_{i=1}^{I} \sum_{j=1}^{J} i^p \, j^q \, f(i,j) \tag{4.8}$$

Some interpretations of these moments are for example that $m_{0,0}$ corresponds to the surface of the object and $m_{1,0}/m_{0,0}$ corresponds to the first coordinate of center of gravity. To be invariant with respect to translation, one can use the central moments and the centralized image:

$$\mu_{p,q} = \sum_{i=1}^{I} \sum_{j=1}^{J} \left(i - \frac{m_{1,0}}{m_{0,0}}\right)^p \left(j - \frac{m_{0,1}}{m_{0,0}}\right)^q f(i,j) \tag{4.9}$$

Furthermore, to be invariant with respect to scaling, the normalized moments can be regarded:

$$\eta_{p,q} = \mu_{p,q} / \mu_{0,0}^{\frac{p+q}{2}+1}, \quad p+q = 2, 3 \ldots \tag{4.10}$$

Hu proposes 7 (polynomial) combinations of these basic moments as invariant features, which are translation-, scale- and also rotation-invariant [46]. These invariant features seem to work well only on binarized patterns in absence of distortion and/or noise, which is reflected in extremely low recognition rates for example on the USPS digit recognition task [77]. This is consistent with the statement that "Regular moments are highly noise-sensitive." [101]. Another form of moments based on pairwise orthogonal Zernike polynomials are the *Zernike moments*, which are rotation invariant and even RST-invariant if normalized and "outperform other kinds of moments" [101].

Other Invariant Features

Among the remaining approaches to extract invariant features, one should mention the use of *cooccurrence matrices*, which describe the distribution of pairs of grayvalues occurring at pixel positions which are separated by a certain displacement. These are translation-invariant and can be extended to rotation invariance if matrices for displacements of the same lengths are combined. Furthermore all *histogram based features* are naturally invariant with respect to rotation and translation. Based on this fact, RT-invariant histogram based features are presented in [85]. They are computed using a nonlinear, invariant integration method consisting of integrating a nonlinear function with local support (hence local invariant features) over all considered transformations. Then a histogram of these features is used for classification. Furthermore, in [86] a technique for fast calculation of these features using a Monte-Carlo-Method is presented. In [12] the usage of invariant moments of contour lines as features for object recognition in digital radiographs of the IRMA database is proposed and proves more successful than Fourier coefficients or invariant (elastic) signatures.

4.1.3 Invariant Distance Measures

While normalization and the extraction of invariant features aim at the elimination of the considered transformations before the actual classification process, invariance can also be incorporated directly into the classifier. This can be done by using *invariant distance measures*. An invariant distance measure would ideally have the property that the distance between two patterns is always equal to the minimum distance between the 'best matching' transformed instances of those patterns. Since the orbits that arise from regarding the set of all possible transformations of a pattern form a *manifold* in pattern space, this ideal invariant distance is called manifold distance. A definition of a manifold is given for example in [42]: A manifold is a locally Euclidean space together with a differential structure. "One can think about a manifold as a way to piece together "bent" pieces of $\mathbb{R}^n$. Thus the manifold has the same local properties as $\mathbb{R}^n$, but may have different global properties. One can also think about a manifold as a generalization of surfaces in

$\mathbb{R}^n$." The main problem with the notion of a manifold distance is that it is in most cases a very hard problem to determine the minimum distance, because the manifolds are difficult to handle. A few approaches to this problem of modeling the manifolds are discussed in Section 4.2.2, one of which consists in following the surface of the manifold in small steps, a method reminding of the Euler-Cauchy method for handling differential equations.

Since probability density functions are often based on a distance function, one can use invariant distance measures to define transformation invariant probability distributions. On the other hand one can show (see Section 5.1) that starting from a distribution invariant with respect to some transformation an invariant distance measure can be derived. The two concepts may therefore be regarded as equivalent.

The most common distance measure encountered is (squared) Euclidean distance, which is also inherent in the normal distribution (with the identity matrix as covariance matrix). For images the squared Euclidean distance is defined by:

$$d(x, \mu) = ||x - \mu||^2 = \sum_{i=1}^{I} \sum_{j=1}^{J} ||x_{ij} - \mu_{ij}||^2 \tag{4.11}$$

There are many other distance- (respectively similarity-) measures used in pattern recognition, like the dot product of two vectors $x^T \cdot \mu = \sum_{d=1}^{D} x_d \mu_d$, which is used as a similarity measure and is related to the angle θ between to vectors

$$\theta = \arccos \frac{x^T \cdot \mu}{||x|| \; ||\mu||} \;\Leftrightarrow\; \cos\theta = \frac{x^T \cdot \mu}{||x|| \; ||\mu||} \tag{4.12}$$

where the cosine of the angle is also called normalized dot product. A connection to the Euclidean distance is given by the relation

$$||x - \mu||^2 = ||x||^2 - 2x^T \cdot \mu + ||\mu||^2 \tag{4.13}$$

which can be simplified if the two vectors are normalized to $||x|| = ||\mu|| = 1$ to $||x - \mu||^2 = 2(1 - x^T \cdot \mu)$. This relation is also helpful for pattern matching in larger images, that is, if the best fitting match x (a part of a larger image) to a reference μ is desired. In that case the Euclidean distance can be decomposed into a term independent of the position ($||\mu||^2$), a term easily calculated for each position of the smaller template in the image ($||x||^2$, only the sum of squares of the border needs to be considered when stepping through the image) and a convolution ($x^T \cdot \mu$) which can be efficiently calculated using the FFT.

These distance measures are not invariant with respect to variations in the images like affine transformations, in fact they are very sensitive to such distortions. In the context of image object recognition Simard et al. introduced a new locally invariant distance measure called tangent distance [89]: "Memory-based classification algorithms such as radial basis functions or K-nearest neighbors typically rely on simple distances (Euclidean, dot product...), which are not particularly meaningful on pattern vectors. More complex, better suited distance measures are often expensive and rather ad-hoc (elastic matching, deformable templates). We propose a new distance measure which (a) can be made locally invariant to any set of transformations of the input and (b) can be computed efficiently." Since tangent distance is one of the main topics of this work, it will be regarded in more detail in Section 4.2. There are also other methods as elastic matching methods, which try to fit an elastic model to the observation and determine the distance as a function of the necessary deformation and the remaining differences between deformed model and observation. The elastic matching models are in turn related to methods based on dynamic programming such as warping and Levenshtein-Moore distance, which will be considered in Section 4.4. For an empirical comparison of different distance measures see also Sections 7.1.9 and 7.2.5.

In connection with invariant distance measures one should also mention invariant discriminant functions, like invariant ANNs, based for example in the tangent prop(agation) algorithm [90, 91]. Invariances from a priori knowledge is here incorporated by "a scheme that minimizes the derivative of the classifier outputs with respect to distortion operators of our choosing"[91]. That is, the network directly learns the effect of the regarded transformations using directional derivatives as regularizers (the output derivative should be zero for changes due to these transformations). Another method to achieve invariance in such implementations of ANN for discriminant functions is to enforce weight sharing between the connections in the ANN [101]. Similar methods have also been used in autoassociative multilayer perceptrons [84]. Finally, a method to obtain transformation tolerance in an ANN is to present each input pattern in a number of positions in its invariance group orbit to the network during training. This is a method also applicable to other classification algorithms and will be regarded in the following section.

4.1.4 Extended Data and Classifier Combination

A simple way to incorporate invariance into a classifier is the explicit generation of transformed data to be used, which may be called *virtual data*. The brute force method already mentioned would be to produce all possible transformations in order to achieve complete invariance, but this is not feasible in most practical settings. Therefore, one usually restricts the multiplication of the data to a few variants of the transformations. This can be done for the training data, which is quite a common approach, but the method is also efficient if used for the test samples. The two methods are considered in the following.

Multiplication of the Training Data

Using the domain knowledge about transformations of the patterns that do not affect class membership, it is easy to generate *virtual* training data from given training data. One only needs to apply the transformation to the patterns using different parameters and thus obtains new data (which keeps its class labels) since the class does not change under the used transformation. Note that this approach is different from adding more samples to the training data, because the data is generated from existing samples automatically. This is reflected in the statement "Distortion models can be used to increase the effective size of a data set without actually taking more data." [8] For example, the domain knowledge about invariance with respect to image shifts in optical character recognition can be used to implicitly enrich the training set with shifted copies of the given training data. In the experiments for this work, displacements of one pixel in eight directions were used, leading to an increase in the effective training set size by the factor nine. Besides the incorporation of invariance into the classifier this has the additional advantage that the estimation of parameters becomes more reliable due to the increase in examples. With respect to this subject Vapnik states: "[...] when one has a relatively small amount of training examples, the effect of using a priori information can be even more significant than the effect of using a learning machine with a good generalization ability. [...] this is not true when the number of training examples is large. However, in all cases to achieve the best performances, one must take into account the available a priori information." [98] Two possible drawbacks of this method are that the user must choose the magnitude of the transformation parameters and the number of instances to be generated beforehand and that the generated data is highly correlated [91].

The approach can be used with most classification approaches and has led to good results in optical character recognition (for a comparison see Chapter 6) It has been used for invariance in neural nets (LeCun on NIST 600,000 training examples, 0.9% ER [98]), in support vector machines (in that context only new support vectors need to be constructed, 0.8% ER on NIST data [98]) and for boosting neural networks: "Using models of characters (the same that was used for constructing the tangent distance) and 60,000 examples of training data, H. Drucker, R. Schapire, and P. Simard generated more than 1,000,000 examples which they used to train three LeNet 4 neural networks, combined in the special "boosting scheme" (Drucker, Schapire, and Simard, 1993) which achieved a 0.7% error rate." [97, p. 159], the citation refers to [26].

Multiplication of the Test Data

As it is possible to use the knowledge about invariance for the training data by applying both tangent distance and explicit shifts, this should be the case for the test data as well. Here the interpretation is not as straightforward as for the training data case, but inspired by methods for combining classifiers [57] one can arrive at the following solution called *virtual test sample method* (VTS):

When classifying a given pattern, transformed versions of the pattern are generated (using the a priori knowledge about the data) and independently classified by the same classifier. The overall decision is then obtained by combining the individual results using the sum rule ("the sum rule and its derivatives consistently outperform other classifier combination schemes" [57]), i.e.

$$p(x|k) = \sum_{\alpha} p(x, \alpha|k) \tag{4.14}$$

where α denotes the used transformation parameters. Note that in the case of VTS, the motivation for the sum rule differs from that proposed by Kittler. To justify the sum rule in the case of using multiple classifiers to classify a single test pattern, he assumed that the posterior probabilities computed by the respective classifiers do not differ much from the prior probabilities. In contrast to this, using multiple test patterns and a single classifier, the sum rule simply follows from the fact that the transformations considered are mutually exclusive, if we assume that the respective prior probabilities are equal (e.g. the prior probability for a right shift should be the same as for a left shift, which seems reasonable). More detailed discussions of this method can be found in [20, 17]. One advantage of the VTS method is that one is able to use classifier combination rules and their benefits without having to create multiple classifiers. Instead, one simply creates virtual test samples. Thus, classifying a pattern has the same computational complexity as compared to using any other classifier combination scheme, yet the (computationally expensive) training phase remains unaffected. VTS thus leads naturally to a certain invariance of the resulting classifier to the transformations regarded in multiplying the test data.

4.2 Tangent Distance

This section is concerned with a more detailed discussion of one particular invariant distance measure "using important a priori information about invariants of handwritten digits incorporated into a special measure of distance between two vectors, the so-called tangent distance." [98]

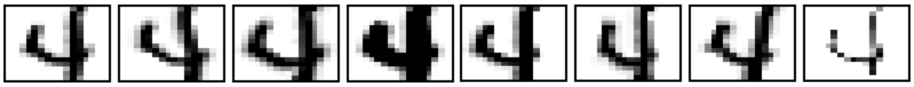

Figure 4.3: Examples for tangent approximation using Eq. (4.18)

Two major advantages of this measure are its general purpose applicability and its computational simplicity.

4.2.1 Overview of Tangent Distance

In 1993, SIMARD et al. proposed an invariant distance measure called *tangent distance* (TD), which proved to be especially effective in the domain of OCR [89]. The authors observed that reasonably small transformations of certain image objects do not affect class membership. Simple distance measures like the Euclidean distance do not account for this, instead they are very sensitive to affine transformations like scaling, translation, rotation, shearing or axis deformation. When an image is transformed (e.g. scaled and rotated) by a transformation $t(x,\alpha)$ which depends on L parameters $\alpha \in \mathbb{R}^L$ (e.g. the scaling factor and rotation angle), the set of all transformed patterns

$$M_x = \{t(x,\alpha) : \alpha \in \mathbb{R}^L\} \subset \mathbb{R}^{I\times J} \tag{4.15}$$

is a manifold of at most dimension L in pattern space. The distance between two patterns can now be defined as the minimum (squared) distance between their respective manifolds, being truly invariant with respect to the L regarded transformations:

$$d_{\text{Manifold}}(x,\mu) = \min_{\alpha_x,\alpha_\mu \in \mathbb{R}^L} \{||t(x,\alpha_x) - t(\mu,\alpha_\mu)||^2\} \tag{4.16}$$

Unfortunately, computation of this *manifold distance* is a hard non-linear optimization problem and the manifolds concerned generally do not have an analytic expression, since a "simple image translation corresponds to a highly non-linear transform in the high-dimensional pixel space" [87]. Therefore, small transformations of the pattern x are approximated by a tangent subspace $\hat{M}_x$ to the manifold M_x at the point x. This subspace is obtained by adding to x a linear combination of the vectors $x_l, l = 1,\ldots,L$ that span the tangent subspace and are the partial derivatives of $t(x,\alpha)$ with respect to α_l. These so called *tangent vectors* $x_l = \frac{\partial t(x,\alpha)}{\partial \alpha_l}$ are also called Lie derivatives of the transformations. Using the first order Taylor series approximation, i.e. the Taylor expansion of $t(\cdot,\cdot)$ around $\alpha = 0$

$$t(x,\alpha) = x + \sum_{l=1}^{L} \alpha_l \frac{\partial t(x,\alpha)}{\partial \alpha_l} + O(\alpha^2) \approx x + \sum_{l=1}^{L} \alpha_l x_l \tag{4.17}$$

one obtains a first order approximation of the Manifold M_x, which has the considerable advantage of being a linear function in α. It is therefore analytically and computationally easy to handle.

$$\hat{M}_x = \{x + \sum_{l=1}^{L} \alpha_l \cdot x_l \ : \alpha \in \mathbb{R}^L\} \subset \mathbb{R}^{I\times J} \tag{4.18}$$

The tangent vectors x_l can be computed using finite differences between the original image x and a reasonably small transformation of x [89]. The computation of the tangent vectors is considered in more detail below. Example images that were computed using (4.18) are shown in Fig. 4.3 (with the original image on the left). The description of the transformation by the tangent approximation

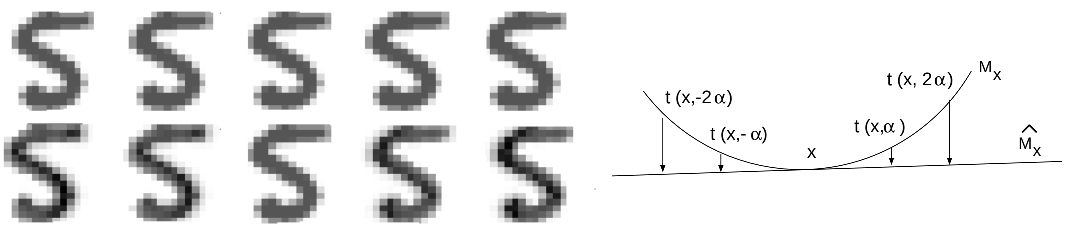

Figure 4.4: Images obtained by shifting a digit and by finding the closest point in the tangent space, original image in the middle. The upper row shows the shifted images with the closest tangent approximation in the lower row. Schematic illustration on the right. The transformation t is a horizontal shift here and α corresponds to the displacement of one pixel

is locally invariant, but not globally invariant. This may be a disadvantage in some cases, but it also can be an advantage, since global invariance is not desired in many cases. For example one does not want to model complete rotational invariance in digit recognition or it is not desired to compare all images at a scale of one pixel.

Some examples for the linear approximation are given in Figure 4.4, which shows images of the digit '5' obtained by shifting the original image and finding the closest corresponding image in the tangent subspace for translation. On the right a schematic illustration is given. One can see that the approximated image corresponds well to the shifted image for shifts with a displacement of one pixel (second and fourth column), but the linear tangent subspace cannot describe well larger shifts (see outer columns, the images are almost identical to the ones obtained for one pixel shifts).

Now, it is possible to define a tangent distance using the approximations on the side of the observation, on the side of the reference or both. The *single sided* (SS) (squared) tangent distance with tangent approximation on the side of the observation $d_{\mathrm{SS},x}(x, \mu)$ is defined as

$$d_{\mathrm{SS},x}(x,\mu) = \min_{\alpha \in \mathbb{R}^L} \{ ||x + \sum_{l=1}^{L} \alpha_l \cdot x_l - \mu||^2 \} \tag{4.19}$$

and analogously the single sided TD with tangents on the reference side $d_{\mathrm{SS},\mu}(x, \mu)$ is defined as

$$d_{\mathrm{SS},\mu}(x,\mu) = \min_{\alpha \in \mathbb{R}^L} \{ ||x - (\mu + \sum_{l=1}^{L} \alpha_l \cdot \mu_l)||^2 \} \tag{4.20}$$

Finally, the *double sided* (DS) tangent distance $d_{\mathrm{DS}}(x, \mu)$ using both linear subspaces is defined by

$$d_{\mathrm{DS}}(x,\mu) = \min_{\alpha_x, \alpha_\mu \in \mathbb{R}^L} \{ ||(x + \sum_{l=1}^{L} \alpha_{xl} \cdot x_l - (\mu + \sum_{l=1}^{L} \alpha_{\mu l} \cdot \mu_l)||^2 \} \tag{4.21}$$

The resulting distances are illustrated in Figure 4.5. It shows the linear subspaces M_x and M_μ as well as the projections of reference and observation in the opposite subspace. The four regarded distances corresponding to the use of the tangent subspace on either side are also shown. Care should be taken in extrapolating this figure to higher dimensional spaces, since the 'probability' that lines (respectively hyperplanes) intersect in a higher dimensional space is much smaller. (Compare also page 91.)

The minimization can be easily solved, since the problem is linear. It amounts to solving a linear least squares problem or to computing an orthogonal basis of the tangent subspace and then

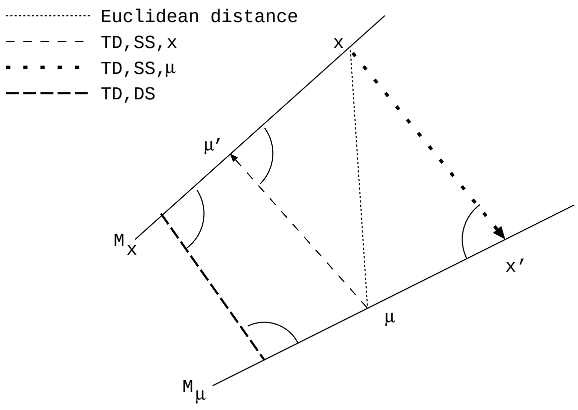

Figure 4.5: Schematic illustration of the points of interest in double sided tangent distance

calculating the minimizing α by scalar products of the vector $x - \mu$ and the base vectors (compare also Section 5.1). These two methods have about the same computational complexity. A more efficient way to calculate the single sided distance (considered in the references here) is possible, if many comparisons are made as for example in a nearest neighbor classifier. In that case, an orthonormal basis of the tangent subspace is precalculated and the distance can be determined as follows with about half the computation time:

$$d(x, \mu) = ||x - \mu||^2 - \sum_{l=1}^{L} [(x - \mu)^T \cdot \mu_l]^2 \tag{4.22}$$

Note that this method can only be used, if no subsequent steps are performed that use the closest point x' in the tangent space, because the coefficients α are not determined. If x' is needed (e.g. for further computation of the image distortion model presented below), one can compute the projection and then calculate Euclidean distance.

Determining the Tangent Vectors

Tangent distance was introduced for the use within an OCR system and the application of seven transformations was proposed. These transformations consist of six accounting for affine variations of the image and one that models a line thickness deformation. Although these transformations proved especially useful for the recognition of handwritten digits, tangent distance can in principal be used with any transformation that allows to compute its derivative in a certain point effectively. In the following the derivation of the seven basic tangents is described, following the descriptions in [98].

Consider the affine transformations of the image grid given by

$$\begin{pmatrix} i' \\ j' \end{pmatrix} = \begin{pmatrix} 1+\alpha_1 & \alpha_2 \\ \alpha_3 & 1+\alpha_4 \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} + \begin{pmatrix} \alpha_5 \\ \alpha_6 \end{pmatrix} \tag{4.23}$$

Then one can determine the derivatives $x_1, \ldots, x_6$ with respect to the six parameters and the heuristic tangent x_7 for line thickness (which corresponds to the squared gradient) as presented in the following:

horizontal translation:

$$\alpha_l = 0,\ l = 1,2,3,4,6 \qquad i' = i + \alpha_5 \qquad j' = j$$

$$x_1(i,j) = \lim_{\alpha_5 \to 0} \frac{x(i+\alpha_5, j) - x(i,j)}{\alpha_5} \tag{4.24}$$

vertical translation:

$$\alpha_l = 0,\ l = 1,...,5 \qquad i' = i \qquad j' = j + \alpha_6$$

$$x_2(i,j) = \lim_{\alpha_6 \to 0} \frac{x(i, j+\alpha_6) - x(i,j)}{\alpha_6} \tag{4.25}$$

rotation:

$$\alpha_l = 0,\ l = 1,4,5,6 \qquad \alpha_2 = -\alpha_3 \qquad i' = i + \alpha_2 j \qquad j' = j - \alpha_2 i$$

$$\begin{aligned} x_3(i,j) &= \lim_{\alpha_2 \to 0} \frac{x(i+\alpha_2 j, j-\alpha_2 i) - x(i,j)}{\alpha_2} \\ &= \lim_{\alpha_2 \to 0} \frac{x(i+\alpha_2 j, j-\alpha_2 i) - x(i, j-\alpha_2 i)}{\alpha_2} + \lim_{\alpha_2 \to 0} \frac{x(i, j-\alpha_2 i) - x(i,j)}{\alpha_2} \\ &= j x_1(i,j) - i x_2(i,j) \end{aligned} \tag{4.26}$$

scaling:

$$\alpha_l = 0,\ l = 2,3,5,6 \qquad \alpha_1 = \alpha_4 \qquad i' = i + \alpha_1 i \qquad j' = j + \alpha_1 j$$

$$x_4(i,j) = i x_1(i,j) + j x_2(i,j) \tag{4.27}$$

axis deformation:

$$\alpha_l = 0,\ l = 1,4,5,6 \qquad \alpha_2 = \alpha_3 \qquad i' = i + \alpha_3 j \qquad j' = j + \alpha_3 i$$

$$x_5(i,j) = j\, x_1(i,j) + i\, x_2(i,j) \tag{4.28}$$

diagonal deformation:

$$\alpha_l = 0,\ l = 2,3,5,6 \qquad \alpha_1 = -\alpha_4 \qquad i' = i + \alpha_4 i \qquad j' = j - \alpha_4 j$$

$$x_6(i,j) = i\, x_1(i,j) - j\, x_2(i,j) \tag{4.29}$$

line thickness deformation:

$$x_7(i,j) = (x_1(i,j))^2 + (x_2(i,j))^2 \tag{4.30}$$

Note that the above equations do not exactly describe the named transformations, e.g. the parameters for a rotation of angle ϕ are given by

$$\begin{pmatrix} 1+\alpha_1 & \alpha_2 \\ \alpha_3 & 1+\alpha_4 \end{pmatrix} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \tag{4.31}$$

so the setting given in (4.27) is correct for the limiting case of small angles of rotation and therefore the derivatives coincide. Nevertheless the transformations from (4.24) to (4.29) span the whole group of affine transformations. Equivalently one could use the six transformations resulting from setting five parameters to zero and varying the remaining one as a basis. In that case it is hard to find mnemonic names for the resulting canonical basis transformations, though. The resulting four transformations for the linear component then have the form

$$\begin{aligned} x_3(i,j) &= j\, x_1(i,j) \\ x_4(i,j) &= j\, x_2(i,j) \\ x_5(i,j) &= i\, x_1(i,j) \\ x_6(i,j) &= i\, x_2(i,j) \end{aligned} \tag{4.32}$$

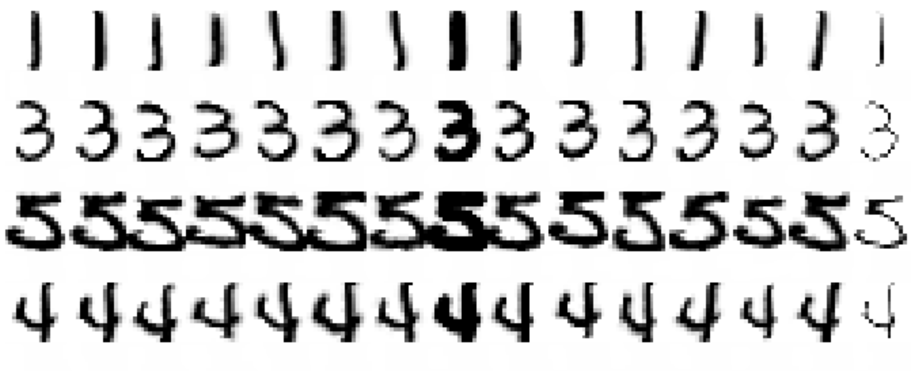

Figure 4.6: Images obtained via tangent approximation of the basic 7 transformations. First column: Original image, column 2–8: positive tangent direction, column 9–15 negative tangent direction

Figure 4.6 shows images obtained via tangent approximation of the basic transformations. The original image is shown on the left of each row, followed by seven images for positive tangent tangent direction and seven for negative tangent direction. The tangents are applied in the order horizontal translation, vertical translation, diagonal deformation, axis deformation, scaling, rotation and line thickness. It can be observed that the modeled variation is high and the approximation is visually correct for the chosen parameters.

In the following, a different way to derive the tangents is presented, which allows easier considerations of other transformations of the image grid. For the model, a pattern is considered a continuous function $x : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. Now a coordinate transformation of the plane $t : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \times \mathbb{R}$ is considered, e.g. as before affine (with parameters at zero representing identity):

$$t(i,j) = (t_1(i,j), t_2(i,j)) = ((\alpha_1 + 1)i + \alpha_2 j + \alpha_5, \alpha_3 i + (\alpha_4 + 1)j + \alpha_6) \tag{4.33}$$

Now for each image grid point the partial derivative with respect to the transformation parameter is sought using the chain rule:

$$\frac{\partial x}{\partial \alpha} = \frac{\partial x}{\partial t}\frac{\partial t}{\partial \alpha} \tag{4.34}$$

where $\frac{\partial x}{\partial t}$ is composed of the local x- and y-gradient (since $t_{\alpha=0} = id$), that is

$$\frac{\partial x}{\partial t}(i,j) = (\frac{\partial x}{\partial i}(i,j), \frac{\partial x}{\partial j}(i,j)) \tag{4.35}$$

In the following two examples for the application of this method are given for affine transformations.

- x-translation ($\alpha_l = 0,\ l = 1,2,3,4,6$)

$$\frac{\partial t}{\partial \alpha_5}|_{\alpha_5=0}(i,j) = (1,0) \tag{4.36}$$

$$\frac{\partial x}{\partial \alpha_5}(i,j)|_{\alpha_5=0} = (\frac{\partial x}{\partial i}(i,j), \frac{\partial x}{\partial j}(i,j))(1,0)^T = \frac{\partial x}{\partial i}(i,j) \tag{4.37}$$

That is, the derivative is the x-gradient.

- scaling ($\alpha_1 = \alpha_4 = \alpha$, other parameters 0)

$$\frac{\partial t}{\partial \alpha}(i,j)|_{\alpha=0} = (i,j) \tag{4.38}$$

$$\frac{\partial x}{\partial \alpha_5}(i,j)|_{\alpha=0} = (\frac{\partial x}{\partial i}(i,j), \frac{\partial x}{\partial j}(i,j))(i,j)^T = i\frac{\partial x}{\partial i}(i,j) + j\frac{\partial x}{\partial j}(i,j) \tag{4.39}$$

  Conforming the previous result

This derivation makes it easier to extend the approach to other than affine transformations, e.g. projective transformations with

$$t(i,j) = (t_1(i,j), t_2(i,j)) = \left(\frac{(\alpha_1+1)i + \alpha_2 j + \alpha_5}{\alpha_7 i + \alpha_8 j + 1}, \frac{\alpha_3 i + (\alpha_4+1)j + \alpha_6}{\alpha_7 i + \alpha_8 j + 1}\right) \tag{4.40}$$

The resulting tangents are the same for the first six parameters, and for the remaining two one obtains the result

$$x_7(i,j) = -i^2\frac{\partial x}{\partial i}(i,j) - ij\frac{\partial x}{\partial j}(i,j) = -i^2 x_1(i,j) - ij x_2(i,j) \tag{4.41}$$

and

$$x_8(i,j) = -ij\frac{\partial x}{\partial i}(i,j) - j^2\frac{\partial x}{\partial j}(i,j) = -ij x_1(i,j) - j^2 x_2(i,j) \tag{4.42}$$

Finally, it should be mentioned that there exists a third way to derive the tangents, using the first order Taylor approximation at the parameter values for the identity transformation, and then differentiating with respect to the transformation parameters, which was presented in [37] and yields quite similar results to the previous ones.

As an application of the tangent method one may want to consider the following image illumination model involving a multiplicative and an additive brightness parameter, α_1 and α_2 respectively, where each image pixel is subject to the transformation

$$t(x, \alpha_1, \alpha_2) = \alpha_1 x + \alpha_2 \tag{4.43}$$

Now the differentiation is straightforward, since no transformation of the image grid is present, but the pixel values are transformed directly. The derivation of t with respect to α_1 yields the image vector x as a brightness tangent vector and the differentiation with respect to α_2 yields a constant brightness tangent vector. These results are so easily obtained, because here the manifold resulting from applying the transformation is linear itself. This means that the result is not an approximation, but the exact representation of the orbit. Note that, if this illumination model is applied in double sided tangent distance, all patterns will have zero distance, because the null vector is always element of the tangent subspace.

Calculating the Tangent Vectors

To determine the derivatives in horizontal and vertical direction at the individual pixel locations, one must choose a method to cope with the discrete nature of digital images (since the derivative is a continuous concept). For that one has three possibilities:

(1) use finite differences

(2) convolve with a smooth kernel function (yielding a differentiable function), then differentiate; equivalently differentiate the kernel first, then convolve

(3) smooth the image, then use finite differences

where basically (3) describes a method identical to (2). In (2) and (3) the scale of the used kernel function, for example a Gaussian kernel, controls the locality of the tangents. If a large kernel is used, the approximation will fit better to larger transformations, while a smaller kernel models local transformations better. The discrete counterpart of a Gaussian kernel is the binomial kernel. The 2D binomial kernel can be obtained by calculating the outer product of a row from Pascal's triangle (a vector of binomial coefficients) with itself, e.g. the 3×3 binomial kernel is given by

$$\frac{1}{4}(1,2,1)\,\frac{1}{4}(1,2,1)^T = \frac{1}{16}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \tag{4.44}$$

There are several possibilities to calculate the discrete derivative. One may resort to local differences which yields the following filter mask (for differentiating along the horizontal axis)

$$\begin{array}{|c|c|}\hline -1 & 1 \\ \hline \end{array} \tag{4.45}$$

But this rather calculates the derivative at a position between two pixels than at a certain pixel location. Therefore one may consider a parabola fitted to the values of three consecutive pixels and take its derivative at the center position. This leads to the filter mask

$$\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline \end{array} \tag{4.46}$$

The *Sobel operator* combines differentiation with a smoothing kernel. Its four directional variants are given by [65, p. 213]:

$$S_i = \frac{1}{4}\begin{array}{|ccc|}\hline 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \\ \hline \end{array} \qquad S_j = \frac{1}{4}\begin{array}{|ccc|}\hline 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \\ \hline \end{array}$$

$$S_/ = \frac{1}{4}\begin{array}{|ccc|}\hline 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \\ \hline \end{array} \qquad S_\backslash = \frac{1}{4}\begin{array}{|ccc|}\hline -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \\ \hline \end{array} \tag{4.47}$$

For tangent calculation, only the horizontal and vertical filters are needed, but it might be a useful extension to also use the diagonal operators, when the direction modeled at a certain pixel position is diagonal.

In the experiments carried out for this work, different templates starting from the Sobel operator were evaluated for tangent calculation and best results were obtained by the template shown in Fig. 4.7. (The template for calculation of the tangent for shift in horizontal direction is depicted. The vertical template is the result of a 90° rotation) This 'modified Sobel operator' performed slightly better (about 0.2% absolute improvement in error rate) than the basic Sobel operator on the IRMA corpus and on the USPS corpus when Tangents were used on the observation side. For the reference side no improvements were obtained.

Figure 4.8 shows the result of tangent vector calculation using the methods introduced for three examples from the USPS database (see Chapter 6). The tangents are presented in the order horizontal translation, vertical translation, diagonal deformation, axis deformation, scaling, rotation and line thickness. Bright pixels represent increase in grayvalue, while darker pixel stand for a

	-0.15	0	0.15	
-0.08	-0.62	0	0.62	0.08
	-0.15	0	0.15	

Figure 4.7: Template used for tangent calculation

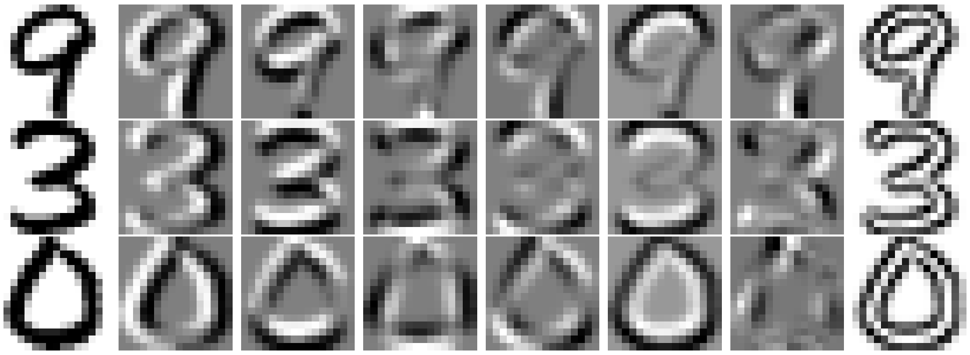

Figure 4.8: Tangent vectors for USPS data. The first column shows the original images, followed by the tangents for horizontal translation, vertical translation, diagonal deformation, axis deformation, scaling, rotation and line thickness.

decrease (except for the line thickness tangent, that only consists of non negative values; here dark regions correspond to large values). It can be observed, that the line thickness tangent corresponds to a gradient image, as expected and that the remaining tangents seem to model the desired transformations well.

4.2.2 Extensions to Tangent Distance and Further Considerations

This Section contains some considerations with respect to tangent distance, including presented extensions to tangent distance, connections to the intrinsic dimensionality, considerations about approaches to model the transformations more closely and the hierarchical filtering approach.

Extensions to Tangent Distance

Some extensions to the basic tangent distance methods have been proposed. Simard et al. proposed to use tangent distance within a nearest neighbor classifier and achieved excellent results with this approach [89]. Before this, it had already been applied in a different variant to neural networks [91]. Other natural extensions (which are actually not extensions to tangent distance itself) are to incorporate tangent distance into a kernel density or Gaussian mixture density based classifier and to combine it with other methods to achieve invariance, like data multiplication or other invariance models [20, 51, 23]. Partly, these approaches have been tested in the experiments for this work. In [100] the authors furthermore proposed a "multiresolution tangent distance, which exhibits significantly higher invariance to image transformations" applied to larger invariances. Two other approaches called 'tangent centroid' and 'tangent subspace' have been presented in [38], which have the aim to represent subsets of the training data and are examined more closely in Section 7.1.2 and inherently connected to the considerations of Section 5.1.

Intrinsic Dimensionality

There exists an important connection between the concept of the manifold in the context of tangent distance and the concept of *intrinsic dimensionality* as presented by Fukunaga in [32, pp. 280ff]. The following quotation expresses this connection: "Whenever we are confronted with high-dimensional data sets, it is usually advantageous for us to discover or impose some structure on the data. Therefore, we might assume that the generation of the data is governed by a certain number of underlying parameters. The minimum number of parameters required to account for the observed properties of the data, n_e, is called the *intrinsic* or *effective dimensionality* of the data sets, or, equivalently, the data generating process. [...] The geometric interpretation is that the entire data set lies on a topological hypersurface of n_e-dimension." The author goes on to state that a measure of the dimensionality is the number of dominant eigenvectors of the covariance matrix and that these form the effective subspace, but that this approach is only suitable for linear surfaces. For nonlinear surfaces the intrinsic dimensionality can be determined locally, similar to the local linearization of a nonlinear function. Therefore it is also called *local dimensionality.* This is closely connected to the considerations presented in Section 5.1, where methods to estimate the directions of variation are derived based on dominant eigenvectors of the (local) covariance matrix.

Modeling the Manifolds

Several methods have been proposed to model the manifolds that arise, when a pattern is subject to some transformation, more closely than it is done by the linear tangent subspace. One straightforward method is to approximate the orbit of a pattern using an iterative procedure based on *Newton's method* [34, pp. 1138ff.]. The extension of tangent distance with an iterative Newton-type approximation was proposed in [89] and successfully used for face-recognition [100]. The algorithm for the calculation of the distance between two patterns x and μ consists in alternating the following steps until a suitable convergence criterion is reached (considered here for single sided tangent distance in μ):

(1) calculate $\hat{M}_\mu$ and x', the projection of x with the corresponding parameters α

(2) apply the (nonlinear) transformations (e.g. scaling and rotation) to μ using the parameters α and continue with this transformed version of μ

One drawback of this method is, that in step (2) the exact transformation of the pattern needs to be calculated. If this shall be avoided, one can use a similar algorithm inspired by the Euler-Cauchy method used in the context of differential equations. In contrast to the Newton procedure it does not require the calculation of the actual transformation but uses the tangent approximation instead. That is, step (2) in the above algorithm is replaced by

(2) apply the linear approximation of the transformations to μ using the parameters α and continue with this transformed version of μ, that is, replace μ with x'

This algorithm was used in the experiments carried out for this work. It conceptually consists of iteratively calculating the closest point in the tangent subspace, "moving" into the corresponding

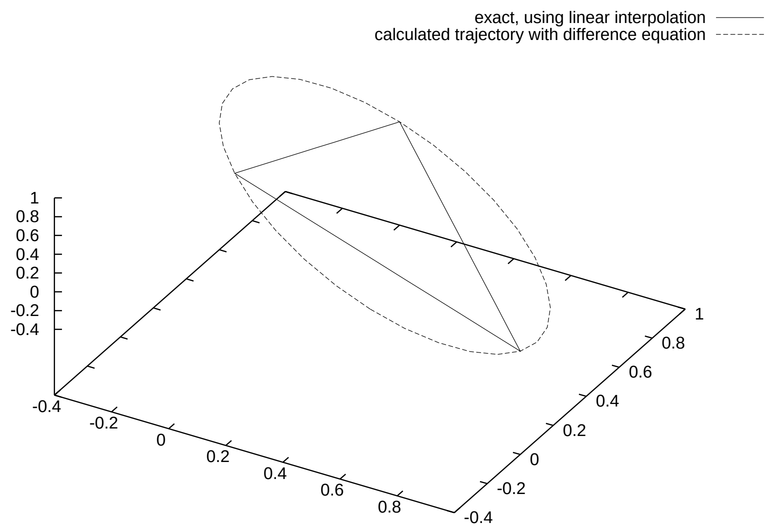

Figure 4.9: Low-dimensional example of translation manifold

direction and recalculating the tangents until convergence. One drawback of this version is that the manifold is not modeled as closely as with the Newton-type algorithm. Yet, one can use smaller movements in step (2) of the algorithm and thus model the manifold with arbitrary precision. For that, the assignment $\mu \leftarrow \mu + (x' - \mu)$ is generalized to $\mu \leftarrow \mu + \gamma(x' - \mu)$ for some *displacement fraction* $\gamma < 1$, where the precision increases for $\gamma \to 0$ as well as the number of necessary iterations until convergence.

An interesting question that arises in this context is, whether the manifolds can be calculated explicitly. In the case of the affine transformations there seems to be a way to derive an exact representation. Since the tangent to the manifold is a linear function of the pattern itself at any point (which is not true for line thickness), the manifold should be the solution to a differential equation of the form

$$\frac{\partial x}{\partial \alpha}(\alpha) = A\, x(\alpha), \quad x(0) = x_0 \tag{4.48}$$

which can be solved by standard methods [28, 11, 34]. At this point one encounters the problem, that the image is not continuous, but the differential equation models a continuous transformation. In the case of images, since they are inherently discrete signals when represented in a computer, this leads to an increasing blur in the images when infinitesimal steps are taken along a direction that is correct only for discrete steps. The solution to this problem is to resort to linear *difference* equations. These model the manifold according to the relation

$$x(\alpha + 1) = x(\alpha) + A\, x(\alpha) = (I + A)\, x(\alpha), \quad x(0) = x_0 \tag{4.49}$$

Yet, the problems arising with this model seem still complicated. For example it seems necessary to use wrap-around in the pattern in order to keep the matrix A non-defective, which is necessary for the determination of all eigenvalues and eigenvectors, and the treatment of larger system seems quite a difficult problem. For example, the authors of [79] state “We consider the problem of finding all eigenvectors of a nonsymmetric matrix as lying beyond the scope of this book.” A low dimensional example of dimension three has been successfully treated in the course of this work and is depicted in Figure 4.9. The pattern $(0, 1, 0)^T$ was used together with a cyclic shift as transformation. The solution to the arising linear difference equation can then be obtained easily, but it remained open, which methods could be used for higher dimensional problems. The figure depicts the obtained manifold, correctly describing the transformation, together with a linear interpolation of the discrete transformation steps.

HASTIE & SIMARD state in [37] that "Deriving the manifold exactly is impossible, given a digitized image, and would be impractical anyway." The considerations just presented seem to imply that the first part of this citation is not true, although the second part seems to be a correct statement. The remaining question now is, whether a direct computation of the exact manifold distance would lead to better results than the tangent distance, and if so, under which circumstances. For example, looking at the remaining test errors (see Figure 7.1) of the USPS database, it seems very unlikely that on this particular database a better method can be developed easily.

While the previously described methods are based on a *single* pattern, from which a description of the manifold is derived, some methods have been proposed for description of the manifold from a set of patterns. For example HINTON et al. use a blended linear approximation to the manifold fitted with an EM based algorithm [43]. This method can be viewed as a mixture density implementation of the approaches proposed in Section 5.1. A similar approach is taken by BREGLER & OMOHUNDRO in [13], interpolating between specified images with "manifold learning" by "inducing a smooth nonlinear constraint manifold from a set of examples from the manifold", while linear interpolation just averages the two pictures. The underlying principle of the approach is basic, i.e. a "mixture model of local linear patches" is fit to the data by clustering, PCA and EM. The final step of interpolation is then achieved by (different methods) of projection into the manifold.

Hierarchical Filtering

Since tangent distance is computationally more expensive than Euclidean distance one can use Euclidean distance as a "prefilter" [89, 88]. This method of *hierarchical filtering* is a special approach for distance based classifiers where different distance measures with different reliability and computational costs are available. It consists of first computing the less costly distance (e.g. the Euclidean) and sorting out the most unlikely samples. In a second step the distances for the remaining samples are recomputed using the more expensive distance measure (e.g. tangent distance), yielding better estimates of the respective distances. Generalizing this, "In the case of images, another time-saving idea is to compute tangent distance on progressively smaller sets of progressively higher resolution images."[89] For example, in the experiments performed with prefiltering on the USPS database with about 7000 training samples, it was observed that a Euclidean prefilter which extracts 100 samples before calculation of tangent distance already was sufficient in the sense that a larger set did not change classification results.

4.3 Image Distortion Model

Tangent distance compensates for small global changes, since the tangent vectors are applied to the image as a whole, but it is sensitive to local image transformations e.g. caused by noise. Therefore the following *image distortion model* (IDM) is proposed [51, 23]. When calculating the distance between two images x and μ, local deformations are allowed, i.e. the 'best fitting' pixel in the

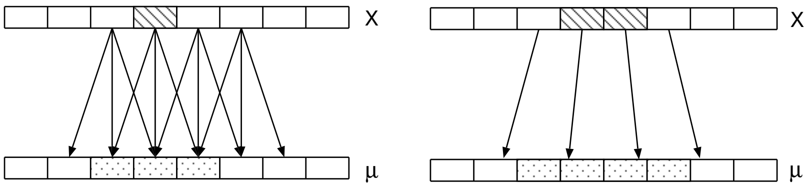

Figure 4.10: 1D comparison of Image Distortion Model and Tangent Model (Scaling)

reference image within a certain neighborhood R_{ij} is regarded instead of computing the squared error between x_{ij} and μ_{ij}. Fig. 4.10 shows a 1D example for the IDM (left) where individual pixel displacements are independent, in comparison to TD (right), where displacements are coupled, forming an affine transformation (here scaling). The resulting distance is

$$d_{\text{IDM}}(x,\mu) = \sum_{i=1}^{I}\sum_{j=1}^{J} \min_{(i',j')\in R_{ij}} \{\|x_{ij} - \mu_{i'j'}\|^2 + C_{iji'j'}\} \tag{4.50}$$

The cost function $C \geq 0$ represents the cost for deforming a pixel x_{ij} in the input image to a pixel $\mu_{i'j'}$ in the reference image[1] and is introduced to compensate for the fact that in an unrestricted distortion model (i.e. with $C \equiv 0$) wanted as well as unwanted transformations can be modeled. With growing neighborhood R the admissible transformations may violate the assumption that they respect class-membership. In fact, the distortion distance between almost any two images can be reduced to a value near zero by increasing R, leading to a significant decrease in classification performance. In the experiments, an appropriate choice of R led to a significant improvement of radiograph classification, even when the cost function was disregarded. To determine the cost function C, two methods may be proposed [23]:

- Choose $C_{iji'j'}$ empirically, e.g. by using a weighted Euclidean distance between pixels (i, j) and (i', j') (see Equation (4.53)). This way, small local transformations are preferred to (most probably unwanted) long-range pixel transformations.
- Learn $C_{iji'j'}$ by using training samples and a maximum likelihood approach. That is, apply meaningful transformations in training and choose $C(i, i', j, j')$ using relative frequencies of possible transformations; the more often a transformation was performed in training, the lower its cost.

The region size is most commonly taken as a square region R of pixels, where its size is best described by the 'radius' of the square r. For higher flexibility it may be desired to model fractional region sizes as well as integer sizes, which can be achieved using linear interpolation. These possibilities are illustrated in Figure 4.11 [95]. Figures 4.12 and 4.13 visualize the effect of the image distortion model for two pairs of images of digits [95]. In the top rows two images of different classes are shown, while in the bottom rows one can see two images of the same class. In the top rows, in terms of Equation (4.50), x is the image of the '7', while μ is an image of the '5'. The image shown for the different parameter settings is composed of the pixel values best matching each pixel in the observation image x. In Figure 4.12 one can see that for region radius zero, which corresponds to Euclidean distance, the best fitting pixel is exactly the one at the corresponding pixel position, since no distortion is allowed. With increasing region size, more pixels of the '7' can

[1] The IDM distance can of course be used exchanging reference and observation in the equations, leading to an explanation of the reference by the observation, which is usually not wanted.

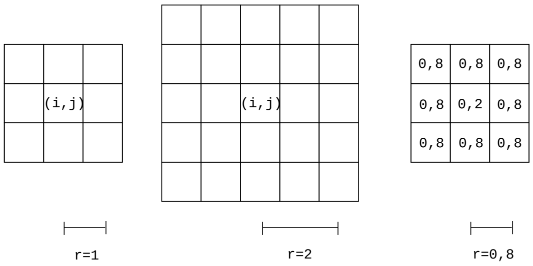

Figure 4.11: Examples for integer and fractional values for the region radius in the IDM

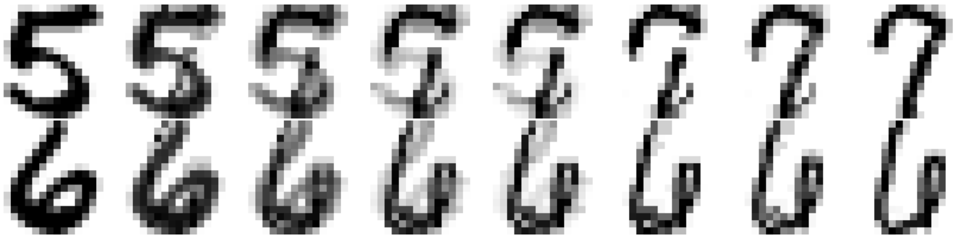

Figure 4.12: Increasing radius of neighborhood at cost 0 (radius from left to right 0.0, 0.2, 0.5, 0.8, 0.9, 1.0, 1.5, 2.0)

be explained by the reference and the image resembles more to the reference. One can also see the effect of linear interpolation in the images corresponding to fractional region radii. In Figure 4.13 one can see the effect of the cost function C, which restricts the used deformation effectively.

The IDM is a natural approach and the idea can be found in various settings in the literature (where the following paragraphs state some). Nevertheless it is an effective means to compensate for small local image variations and the intuitiveness of the model may be seen as an advantage.

The approach can be regarded in relation to the one presented in [47], which also mentions the aim of extending a linear (eigenspace) method: "View-based recognition methods, such as those using eigenspace techniques, have been successful for a number of recognition tasks. Such approaches, however, are somewhat limited in their ability to recognize objects that are partially hidden from view or occur against cluttered backgrounds." The authors present a technique based on the generalized Hausdorff measure applied to binary (edge) images. The classical Hausdorff measure for two sets P, Q (for binary images the sets of points with value 1) is introduced as

$$h(P,Q) = \max_{p \in P} \min_{q \in Q} ||p - q|| \tag{4.51}$$

The generalized Hausdorff measure is then defined by replacing the maximum operation by the f-th quantile, yielding $h_f(\cdot,\cdot)$. The authors then define (as the used distance measure) the Hausdorff fraction, being the largest f for which $h_f \leq d$, for some fixed neighborhood size d. It can be computed by dilating Q with a radius of d to Q^d and then computing the fraction of points in P for which the corresponding point in Q^d exists. "when the dilation is zero, the Hausdorff fraction is simply a normalized binary correlation." Now, the correspondence to the IDM can be seen in the following quotation: "The improvement over binary correlation is to be expected because the Hausdorff fraction handles small perturbations in the locations of image features (whereas, for binary correlation, either feature points are directly superimposed or they do not match)." The IDM can be seen as a generalization of this method to graylevel images. One can regard the IDM as dilating every pixel in Q with a radius $r = d$, such that each pixel afterwards is assigned a *set*

Figure 4.13: Increasing cost factor for Euclidean cost at constant neighborhood size 1.0 (weight factor from left to right $\gamma = 0.0,\ 1.0,\ 2.0,\ 3.0,\ 4.0$)

of grayvalues. Then the basic IDM distance measure is composed of the sum of squared distances for each pixel in the image P to the closest one in the set of values for the corresponding pixel in Q^d.

Another distance measure similar to the IDM distance was used in [92] by the name of "pixel distance metric": "For mismatched pixels between the test and training images, it takes into account the distance to the nearest pixel of the same color" This can be compared to the cost function C in the IDM, but here only exact matches respectively binary images are considered.

IDM and Gradient Magnitude

There is an interesting connection between the image distortion model and the local gradient magnitude in the image. If one reduces the information about the local region R_{ij} to the gradient vector (magnitude plus direction), which may be a justified assumption if the image is sufficiently smooth, one can explicitly solve the minimization of Equation (4.50) over the (continuously modeled) region, given the cost term. Using this model with a weighted Euclidean cost function, the term $\mu_{i'j'}$ referring to a pixel of the reference in Equation (4.50) is replaced by

$$\mu_{i'j'} = \mu_{ij} + \Delta i\ \frac{\partial \mu}{\partial i} + \Delta j\ \frac{\partial \mu}{\partial j}, \quad \text{with} \quad \Delta i = i' - i,\ \ \Delta j = j' - j \tag{4.52}$$

the partial derivatives being approximated by the local gradient. Furthermore the cost function is replaced by

$$C_{iji'j'} = \gamma(||i' - i||^2 + ||j' - j||^2) = \gamma(\Delta i^2 + \Delta j^2) \tag{4.53}$$

Now the size of the used region can be adjusted with the value of γ, so for the considerations unbounded regions can be assumed. The minimization therefore can be found by the following calculations:

$$\begin{aligned}
&\min_{(i',j') \in R_{ij}} \{||x_{ij} - \mu_{i'j'}||^2 + C_{iji'j'}\} \\
&\quad = \min_{\Delta i, \Delta j \in \mathbb{R}} \{||x_{ij} - (\mu_{ij} + \Delta i \frac{\partial \mu}{\partial i} + \Delta j \frac{\partial \mu}{\partial j})||^2 + \gamma(\Delta i^2 + \Delta j^2)\} \\
&\quad = \min_{\Delta \in \mathbb{R}^2} \{||x_{ij} - (\mu_{ij} + \Delta^T p)||^2 + \gamma||\Delta||^2\} \\
&\qquad\qquad [\text{with } \ \Delta = (\Delta_i, \Delta_j)^T,\ p = (\frac{\partial \mu}{\partial i}, \frac{\partial \mu}{\partial j})^T, \text{ and dropping the indices}] \\
&\quad = \min_{\Delta \in \mathbb{R}^2} \{||(x - \mu) - \Delta^T p)||^2 + \gamma||\Delta||^2\} \\
&\quad = \min_{\Delta \in \mathbb{R}^2} \{(x - \mu)^2 - 2(x - \mu)\Delta^T p + ||\Delta||^2 ||p||^2) + \gamma||\Delta||^2\}
\end{aligned}$$

$$
\begin{aligned}
&= \min_{\Delta\in\mathbb{R}^2}\{(||p||^2+\gamma)(\frac{(x-\mu)^2}{||p||^2+\gamma} - \Delta^T\frac{2(x-\mu)p}{||p||^2+\gamma} + ||\Delta||^2)\} \\
&= \min_{\Delta\in\mathbb{R}^2}\{(||p||^2+\gamma)(\frac{(x-\mu)^2}{||p||^2+\gamma} - (\frac{(x-\mu)p}{||p||^2+\gamma})^2 + (\frac{(x-\mu)p}{||p||^2+\gamma} - ||\Delta||)^2)\} \\
&= (||p||^2+\gamma)(\frac{(x-\mu)^2}{||p||^2+\gamma} - (\frac{(x-\mu)p}{||p||^2+\gamma})^2) \\
&= (x-\mu)^2(1-\frac{||p||^2}{||p||^2+\gamma}) \\
&= (x-\mu)^2\frac{\gamma}{||p||^2+\gamma} \qquad (4.54)
\end{aligned}
$$

In this setting, usage of the IDM now amounts to multiplication of the local Euclidean distance with a factor based on the local gradient magnitude and the minimization over the region can be omitted.

$$d_{\text{IDM-grad}}(x,\mu) = \sum_{i,j} ||x_{ij} - \mu_{ij}||^2 \frac{\gamma}{||p_{ij}||^2+\gamma} \qquad (4.55)$$

where p_{ij} denotes the local gradient vector. This variation of the IDM amounts to a weighted Euclidean distance, where large distances are less costly in the proximity of a large gradient, which seems sensible since in that area a small distortion can lead to a large change in grayvalue. On the other hand one can view this variation as a variant of the line thickness tangent, where the thickness is allowed to vary independently in the different parts of the image, while the tangent requires uniform variation throughout the image.

4.4 Levenshtein-Moore Distance and Warping Models

There is a variety of algorithms for image matching that deform two images to be compared in order to obtain a good match between them. This *warping* of images usually is constrained by costs for large distortions or by requiring a certain continuity of the warp. “Elastically deformable templates [...] have been shown to model nonnormalised images of characters well [...]. Unfortunately they are also computationally too expensive for normal use.” [43] Deformable models have been used in a variety of settings, e.g. using deformable splines for digit recognition [16], applying combined hidden Markov models to continuous writing recognition [6], using physically motivated mesh deformation for face recognition [70] or using piecewise continuous mappings for face recognition and other data [30]. One class of these warping approaches is based on allowing almost any transformation of the plane, but imposing certain cost and restriction terms to local or global deformation parameters. The globally optimal warp is then determined using dynamic programming [67, 96]. One such approach will be considered in more detail now.

In 1979, Moore presented an algorithm for finding the distance between two finite areas using dynamic programming [72]. In the one-dimensional case the described distance is also known as *Levenshtein-distance* or *edit-distance*. It is defined “as the minimum cost of changing one sequence into the other by the substitution, deletion, and insertion of elements in either sequence.” [72] This principle can be applied to two dimensional areas yielding a distance measure based on the minimum cost of changing one area into the other using the same operation on the pixel level. One of the advantages of the proposed distance over e.g. Euclidean distance is, that it can be applied without changes to the matching of images of different size. Moore applied the algorithm to areas containing symbolic values (as characters are in a string), and thus images were always regarded as binarized. A description of the algorithm is best started with the one-dimensional case. Here, the distance $d(a,b)$ between two sequences $a = a_1a_2\ldots a_i$ and $b = b_1b_2\ldots b_j$ is defined recursively as the minimum of

- the distance between $a_1a_2\ldots a_{i-1}$ and $b_1b_2\ldots b_j$ plus the cost for deletion/insertion of a_i
- the distance between $a_1a_2\ldots a_i$ and $b_1b_2\ldots b_{j-1}$ plus the cost for insertion/deletion of b_j
- the distance between $a_1a_2\ldots a_{i-1}$ and $b_1b_2\ldots b_{j-1}$ plus the cost for substitution of a_i by b_j

More formally:

$$\begin{aligned} d(a_1a_2\ldots a_i, b_1b_2\ldots b_j) \quad = \quad & \min\{d(a_1a_2\ldots a_{i-1}, b_1b_2\ldots b_j) + d(a_i,\epsilon), \\ & \quad d(a_1a_2\ldots a_i, b_1b_2\ldots b_{j-1}) + d(\epsilon, b_j), \\ & \quad d(a_1a_2\ldots a_{i-1}, b_1b_2\ldots b_{j-1}) + d(a_i, b_j)\} \end{aligned} \tag{4.56}$$

and

$$\begin{aligned} d(\epsilon,\epsilon) &= 0 \\ d(a_1a_2\ldots a_i,\epsilon) &= \sum_{i'=1}^{i} d(a_{i'},\epsilon) \\ d(\epsilon, b_1b_2\ldots b_j) &= \sum_{j'=1}^{j} d(\epsilon, b_{j'}) \end{aligned} \tag{4.57}$$

with ϵ denoting the empty sequence. This recursion can be solved efficiently using dynamic programming in time $O(i \cdot j)$ [72]. Now MOORE proposes a similar algorithm for two dimensional sequences (i.e. lattices or images). The number of different terms that occur in the minimization then is 15 (in general $2^{2n} - 1$, with the number n of dimensions of the lattice as one can verify by a combinatorial argument). Since the resulting recursion formula is very complex, only a more informal descriptions is given here (for the recursion formula see [72]). In order to allow a comparison with the one dimensional case, first an informal description of the above recursion is given. Informally, in each matching step between the sequences a and b (here performed backwards, as in the recursion) one can

1. discard the last symbol of a
2. discard the last symbol of b
3. match the last symbol of a with the last symbol of b

while regarding the introduced cost in each step. Now for the two dimensional case in each matching step between the images a and b

$$a = \boxed{\begin{matrix} a_{11} & \cdots & a_{1J} \\ \vdots & \ddots & \vdots \\ a_{I1} & \cdots & a_{IJ} \end{matrix}} \qquad b = \boxed{\begin{matrix} b_{11} & \cdots & b_{1L} \\ \vdots & \ddots & \vdots \\ b_{K1} & \cdots & b_{KL} \end{matrix}} \tag{4.58}$$

one can

1. discard the right column of a
2. discard the right column of b

3. discard the lower row of a
4. discard the lower row of b
5. discard the right column of a and the lower row of b
6. discard the lower row of a and the right column of b
7. match the right columns of a and b
8. match the lower rows of a and b
9. match the right columns of a and b, discard the lower row of a
10. match the right columns of a and b, discard the lower row of b
11. match the lower rows of a and b, discard the right column of a
12. match the lower rows of a and b, discard the right column of b
13. discard the lower row and the right column of a
14. discard the lower row and the right column of b
15. match the lower rows and the right columns of a and b

where the matching of rows and columns can be performed using the one-dimensional algorithm. Again, dynamic programming can be applied to calculate the distance without using recursion.

Since the original algorithm was proposed for binarized images, a straightforward extension is to also take into account images with grayvalues. In that case, one can use the (weighted) squared difference in grayvalue as cost for a substitution of two pixels. This leads to a new degree of freedom, because the distance component introduced by an insertion or deletion must be adjusted in relation to the changed interval for the substitution cost. Other possible extensions to the basic model include to adjust the cost of insertions or deletions in relation to the grayvalues of the surrounding pixels, which has a connection to the image distortion model. Possibly, the cost for insertion and deletion of pixels at the image border could be weighted, such that images presenting shifted versions of the same object could be matched at lower total cost. Furthermore, in the cases, where a row and a column of the same image are regarded simultaneously, one could regard the two sequences as joined and perform matching 'around the corner'. The arising (non-trivial) question that remains in this context is, which transformations can be modeled with this approach and how it is related to other warping models.

4.5 A Generalization

The approaches of tangent distance and image distortion model towards an invariant distance measure are quite different. Yet, both can be seen from a common point of view, which is the resulting vector field of pixel displacements due to the modeled transformation. A one dimensional example for this connection is shown in Figure 4.10 and a different viewpoint shown in Figure 4.14. The figure shows typical examples for the resulting pixel displacement vector field in the two models. The difference between the two approaches taken is that in the IDM a free displacement within the considered region is possible for each pixel (the minimization of the distance is done for each pixel

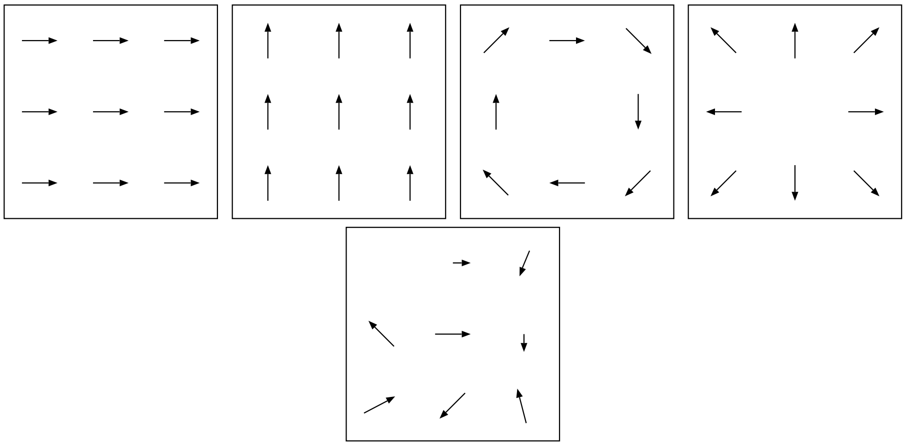

Figure 4.14: Transformations in the tangent model and the image distortion model for size 3×3 images. Top row: x-shift, y-shift, rotation, scaling in the tangent model. Bottom: distortion model. Typical examples of resulting pixel displacement vector fields.

independently), while for tangent distance an interdependence between pixel shifts exists and the minimization is calculated over all possible combinations of allowed transformations.

Trying to relate the approaches of tangent distance and image distortion model (with a connection to the warping methods) it becomes clear, that one can be expressed in terms of the other. Expressing the IDM in terms of tangent distance is difficult, when a non-zero cost function is involved (it requires additional restrictions on the values permitted for the transformation parameters α). On the other hand generalizing the IDM leads to an expression also covering tangent distance:

$$d_{C,\mathcal{F}}(x,\mu) = \min_{f \in \mathcal{F}} \{C(f) + \sum_{i,j} \|x_{ij} - \mu_{f(i,j)}\|\} \tag{4.59}$$

where $\mathcal{F} \subset (\mathbb{R} \times \mathbb{R})^{I \times J}$ is a class of functions assigning to each pixel its (interpolated) counterpart and $C : \mathcal{F} \to \mathbb{R}^{\geq 0}$ a cost function for these assignment functions. For the IDM one has

$$\mathcal{F}_{\text{IDM}} = \{f : f(i,j) \in R_{ij}\},\ C_{\text{IDM}}(f) = \sum_{i,j} C_{ijf(i,j)} \tag{4.60}$$

while for the manifold distance of affine transformations C and $\mathcal{F}$ have the following representation:

$$\mathcal{F}_{\text{TD}} = \{f \ : \ f \text{ affine}\}, \quad C_{\text{TD}}(f) = 0 \tag{4.61}$$

This general expression is an intuitive representation of a distance being invariant to arbitrary functions f of some class $\mathcal{F}$ and a superset of TD and IDM. Computing (4.59) may be very hard or impossible with some classes and cost functions, but TD and IDM are two examples with known solutions. (Strictly speaking, TD only models an approximation of Equation (4.61).) Some questions arising are e.g. which other cases are interesting in the setting of invariant pattern recognition and if one can learn the functions efficiently from training examples. For instance, a model that extends the IDM naturally is to introduce a dependency between the displacements of pixels in a neighborhood, such that displacements in the same direction are cheaper than displacements in opposite directions. This leads to more complex minimization problems, which may be still efficiently solved using dynamic programming, if the number of possible displacements is small. These are related to the warping approaches described in the previous Section. Note that it is difficult to embed the XYI image warping approach presented in [70] into the model (4.59) as the implicit XYI cost function depends on the intensity values.

Chapter 5

Theoretical Considerations

> *"An SEP," he said, "is something that we can't see, or don't see, or our brain doesn't let us see, because we think that it's somebody else's problem. That's what SEP means. Somebody Else's Problem. The brain just edits it out, it's like a blind spot. If you look at it directly you won't see it unless you know precisely what it is. Your only hope is to catch it by surprise out of the corner of your eye."*
>
> [3]

This chapter contains some theoretical considerations regarding the topic of this work. Section 5.1 takes a closer look at invariant distance measures like tangent distance from a probabilistic point of view [50], while Section 5.2 deals with structured covariance matrices [21] and their relation to the IDM.

5.1 A Probabilistic View on Tangent Distance

Tangent distance and related approaches are usually seen in the context of distance based classifiers, as the name 'tangent distance' already suggests. In many cases the focus on distances can be related to the focus on probability densities (which is central to statistical pattern recognition) via the exponential function:

$$p_\mu(x) = e^{-\frac{1}{2}d(x,\mu)} \;\Leftrightarrow\; -2\log p_\mu(x) = d(x,\mu) \tag{5.1}$$

For example Equation (5.1) states the relation between Euclidean distance and a Gaussian distribution with the identity matrix as covariance matrix. This section tries to find a relation between tangent distance and the according distribution. (See also page 21.)

For related work the following two publications should be mentioned, while others are cited throughout this chapter. In [64] Laaksonen considers a probabilistic view on subspace methods. Yet the author does not derive the distribution in general, but only the distribution of the distances from the subspace, which has the form of a gamma distribution. In [68] Meinicke & Ritter present a statistical framework for local PCA learning, but do not relate the method to domain knowledge about class-specific variance in the data.

The approaches can be divided into certain categories, that is if the variation is modeled on the side of the references respectively on the side of the observation vectors. On the other hand one may also distinguish between known derivatives of variation and cases where this information is not available. Each of the following sections deals with one of these possibilities before an attempt is made to combine the results. In the following "direction of variation" is considered synonymous to "derivative of variation", since the derivation with respect to the variation leads to the tangent vectors as first order approximation, which can be regarded as pointing in the corresponding direction.

5.1.1 Known Derivatives of Variation in the References

First consider the case where for each reference vector μ it is known that it may be subject to certain (small) variations that do not change the class it belongs to. That is, there is some a priori knowledge about the reference. For example the class a picture of a digit belongs to generally does not change when a slight affine transformation is applied. This means that variations in the directions of the tangent vectors μ_l with respect to the transformations $l = 1, \ldots, L$ should also represented by μ. Let α_l be the amount of variation in direction μ_l, the vector $\alpha = (\alpha_1, \ldots, \alpha_L)^T$ and consider a Gaussian distribution of the references with covariance matrix Σ. A first order approximation of the transformed reference can then be expressed as $\mu + \sum_l \alpha_l \mu_l$ and the corresponding density function for given α can be written as follows:

$$
\begin{aligned}
p(x|\mu,\alpha,\Sigma) &= \mathcal{N}(x|\mu + \sum\nolimits_l \alpha_l\mu_l, \Sigma) \\
&= \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}(\mu + \sum\nolimits_l \alpha_l\mu_l - x)^T\Sigma^{-1}(\mu + \sum\nolimits_l \alpha_l\mu_l - x)\right) \qquad (5.2)
\end{aligned}
$$

Now assuming a probability density function $p(\alpha)$ for α modeling the variability leads to the following expression for $p(x|\mu) = p(x|\mu, \Sigma)$ (for some parameter Σ, considered constant here):

$$
\begin{aligned}
p(x|\mu) &= \int p(x,\alpha|\mu)\, d\alpha \\
&= \int p(\alpha|\mu)\, p(x|\alpha,\mu)\, d\alpha \\
&= \int p(\alpha)\, p(x|\mu_\alpha)\, d\alpha \\
&\quad \text{[with } \alpha \text{ independent of } \mu \text{ and } \mu_\alpha = \mu + \textstyle\sum_l \alpha_l\mu_l] \\
&\approx \max_\alpha \{p(\alpha)p(x|\mu_\alpha)\} \qquad (5.3) \\
&\quad \text{[using maximum approximation]} \\
&= \max_\alpha \left\{\mathcal{N}(\alpha|0,\gamma^2 I)\, \mathcal{N}(x|\mu_\alpha,\Sigma)\right\} \\
&\quad \text{[assuming independent Gaussian distribution of the } \alpha_l \text{ with} \\
&\quad \text{variance } \gamma^2 \text{ and mean 0]} \\
&= \max_\alpha \left\{ \frac{1}{\sqrt{2\pi\gamma^2}^L} \exp\left(-\frac{1}{2\gamma^2}\sum\nolimits_l \alpha_l^2\right) \cdot \right. \\
&\quad \left. \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}(\mu + \sum\nolimits_l \alpha_l\mu_l - x)^T\Sigma^{-1}(\mu + \sum\nolimits_l \alpha_l\mu_l - x)\right)\right\} \qquad (5.4)
\end{aligned}
$$

The use of maximum approximation in Equation (5.3) is not essential. The same results (except for some constant terms) can be obtained without its application, but the calculations are somewhat more complex. They are included in the Appendix A.3.

The assumption of a Gaussian distribution of the α_l can be justified by the central limit theorem as presented in [64, p. 62ff].

Expression (5.4) is maximized when the (double) negative logarithm is minimized, which can now be interpreted as distance between x and μ. This shows the possibility of deriving the invariant distance measure from this probabilistic interpretation of variability. (Constant terms have been dropped as they are of no influence in the maximization.)

$$\begin{aligned}
d(x,\mu) &:= -2\ \log p(x|\mu) \\
&\approx \min_\alpha \left\{ \frac{1}{\gamma^2} \sum\nolimits_l \alpha_l^2 + (\mu + \sum\nolimits_l \alpha_l\mu_l - x)^T \Sigma^{-1} (\mu + \sum\nolimits_l \alpha_l\mu_l - x) \right\} \\
&= \min_\alpha \left\{ \frac{1}{\gamma^2} \sum\nolimits_l \alpha_l^2 + (\mu - x)^T \Sigma^{-1} (\mu - x) + (\mu - x)^T \Sigma^{-1} (\sum\nolimits_l \alpha_l\mu_l) \right. \\
&\qquad \left. + (\sum\nolimits_l \alpha_l\mu_l)^T \Sigma^{-1} (\mu - x) + (\sum\nolimits_l \alpha_l\mu_l)^T \Sigma^{-1} (\sum\nolimits_l \alpha_l\mu_l) \right\}
\end{aligned} \tag{5.5}$$

Assuming orthogonality of the μ_l with respect to Σ^{-1}, that is $\mu_l^T\Sigma^{-1}\mu_{l'} = 0$ for $l \neq l'$ (which can be achieved without altering the spanned subspace using for example a singular value decomposition), it follows that $(\sum_l \alpha_l\mu_l)^T\Sigma^{-1}(\sum_l \alpha_l\mu_l) = \sum_l \alpha_l^2\mu_l^T\Sigma^{-1}\mu_l$. Furthermore the third and fourth term of the above sum are identical and the second term is independent of α. Therefore the expression reduces to

$$\begin{aligned}
d(x,\mu) &\approx (\mu - x)^T \Sigma^{-1} (\mu - x) \\
&\quad + \min_\alpha \left\{ \sum\nolimits_l \alpha_l^2 \left(\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l \right) + 2(\mu - x)^T \Sigma^{-1} (\sum\nolimits_l \alpha_l\mu_l) \right\} \\
&= (\mu - x)^T \Sigma^{-1} (\mu - x) \\
&\quad + \min_\alpha \left\{ \sum\nolimits_l \left(\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l \right) \left(\alpha_l + \frac{(\mu - x)^T\Sigma^{-1}\mu_l}{\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l} \right)^2 - \sum\nolimits_l \frac{((\mu - x)^T\Sigma^{-1}\mu_l)^2}{\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l} \right\} \\
&= (\mu - x)^T \Sigma^{-1} (\mu - x) \\
&\quad \underbrace{+ \min_\alpha \left\{ \sum\nolimits_l \left(\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l \right) \left(\alpha_l + \frac{(\mu - x)^T\Sigma^{-1}\mu_l}{\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l} \right)^2 \right\}}_{=0^1} - \sum\nolimits_l \frac{((\mu - x)^T\Sigma^{-1}\mu_l)^2}{\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l} \\
&= (\mu - x)^T \Sigma^{-1} (\mu - x) - \sum\nolimits_l \frac{((\mu - x)^T\Sigma^{-1}\mu_l)^2}{\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l}
\end{aligned} \tag{5.6}$$

At the boundaries of the considered range for γ which is $[0;\infty)$ this yields Mahalanobis distance for $\gamma \to 0$ and tangent distance with tangents μ_l for $\gamma \to \infty$. This is not a necessary condition for TD, but e.g. in the domain of OCR experiments showed, that no gain could be obtained by restricting the value of γ (Compare page 132). The authors of [87] call the inverse of γ "spring constant", based on a model with physical intuition. They describe the minimization process as similar to the energy minimization taking place in a physical system with movable points and springs along the ranges between the projection point and the observation point and between the projection point and the reference point. They also state that "Contrary to intuition, there is no danger of sliding too far in high dimensional space, because tangent vectors are always roughly orthogonal and they [i.e. the points in the tangent space] could only slide far if they were parallel." This means that in high dimensional spaces the minimizing value for α is usually small.

[1] This term is a minimization over a sum of quadratic terms of the form $\ldots \cdot (\alpha + \ldots)^2$, which is always zero.

Using the relation

$$x^T(A^{-1}+bb^T)x = x^TA^{-1}x + x^Tbb^Tx = x^TA^{-1}x + (b^Tx)^2 \tag{5.7}$$

(5.6) can be rewritten as

$$d(x,\mu) \approx (\mu-x)^T(\Sigma^{-1} - \sum_l \frac{(\mu_l^T\Sigma^{-1})^T(\mu_l^T\Sigma^{-1})}{\frac{1}{\gamma^2}+\mu_l^T\Sigma^{-1}\mu_l})(\mu-x) \tag{5.8}$$

$$\overset{\gamma\to\infty}{=} (\mu-x)^T\underbrace{(\Sigma^{-1} - \sum_l \frac{(\mu_l^T\Sigma^{-1})^T(\mu_l^T\Sigma^{-1})}{\mu_l^T\Sigma^{-1}\mu_l})}_{*}(\mu-x) \tag{5.9}$$

This modification in the distance can be regarded as assuming 'infinite' variance in the directions of the μ_l, since the inverse of the above matrix $(*)$ can be interpreted as covariance matrix. This is proven by showing that the product of the following matrices equals the identity matrix:

$$\begin{aligned}
&\left(\Sigma^{-1} - \lambda\sum_l \frac{(\mu_l^T\Sigma^{-1})^T(\mu_l^T\Sigma^{-1})}{\mu_l^T\Sigma^{-1}\mu_l}\right)\left(\Sigma + \kappa\sum_l \frac{\mu_l\mu_l^T}{\mu_l^T\Sigma^{-1}\mu_l}\right)\\
=\;& I - \lambda\sum_l \frac{(\mu_l^T\Sigma^{-1})^T(\mu_l^T\Sigma^{-1})}{\mu_l^T\Sigma^{-1}\mu_l}\Sigma + \kappa\sum_l \Sigma^{-1}\frac{\mu_l\mu_l^T}{\mu_l^T\Sigma^{-1}\mu_l}\\
&-\lambda\kappa\sum_l\sum_{l'} \frac{(\mu_l^T\Sigma^{-1})^T\;\overbrace{(\mu_l^T\Sigma^{-1})\;\cdot\;\mu_{l'}}^{=0\text{ for }l\neq l'}\;\mu_{l'}^T}{\mu_l^T\Sigma^{-1}\mu_l\;\cdot\;\mu_{l'}^T\Sigma^{-1}\mu_{l'}}\\
=\;& I - \lambda\sum_l \frac{\Sigma^{-1}\mu_l\mu_l^T}{\mu_l^T\Sigma^{-1}\mu_l} + \kappa\sum_l \frac{\Sigma^{-1}\mu_l\mu_l^T}{\mu_l^T\Sigma^{-1}\mu_l} - \lambda\kappa\sum_l \frac{\Sigma^{-1}\mu_l\mu_l^T(\mu_l^T\Sigma^{-1}\mu_l)}{(\mu_l^T\Sigma^{-1}\mu_l)^2}\\
=\;& I - (\lambda-\kappa+\lambda\kappa)\sum_l \frac{\Sigma^{-1}\mu_l\mu_l^T}{\mu_l^T\Sigma^{-1}\mu_l}
\end{aligned} \tag{5.10}$$

The latter becomes the identity matrix I if $\lambda-\kappa+\lambda\kappa = 0$ or $\kappa = \frac{\lambda}{1-\lambda}$. Thus, as λ approaches 1 as in TD, κ goes to infinity, so that one can write (being aware of the fact that the inverse does not exist in $\mathbb{R}^{D\times D}$):

$$\lim_{\kappa\to\infty}\left(\Sigma + \kappa\sum_l \frac{\mu_l\mu_l^T}{\mu_l^T\Sigma^{-1}\mu_l}\right) = \left(\Sigma^{-1} - \sum_l \frac{(\mu_l^T\Sigma^{-1})^T(\mu_l^T\Sigma^{-1})}{\mu_l^T\Sigma^{-1}\mu_l}\right)^{-1} \tag{5.11}$$

and one can write (again only for notational convenience, as Σ' does not exist)

$$p(x|\mu) = \mathcal{N}(x|\mu,\Sigma') \quad \text{with} \quad \Sigma' = \lim_{\kappa\to\infty}\left(\Sigma + \kappa\sum_l \frac{\mu_l\mu_l^T}{\mu_l^T\Sigma^{-1}\mu_l}\right) \tag{5.12}$$

As $\lambda = 1$ is the setting for tangent distance Equation (5.12) shows that tangent distance is the limiting case of a Gaussian distribution with variance approaching infinity ($\kappa\to\infty$) in the direction of the tangents. That is the distribution can be considered as a degenerate case of the normal distribution. Alternatively, it can be regarded as a normal distribution in the reduced vector space that results from projection along the directions of the μ_l, that is in the vector space dual to the one spanned by the μ_l. Such a model is generally called a *linear model*, which brings about some normalization problems for the case where $\gamma\to\infty$. HINTON et al. state that such a model, e.g. resulting from a PCA, "is not properly normalizable", yet very useful, and refer to factor analysis as a resort [43].

The problem with normalization can be circumvented by either looking at the distribution in the space originating from projection along the subspace or by viewing the dimensions of the subspace as equipped with codebook exponents approaching zero (for an explanation of codebook exponents see [75]). Another approach is to "use a convex combination of the orthogonal distance [i.e. TD] and the point-to-point centroid distance [i.e. the distance in the spanned subspace], whereby the resulting individual level surfaces are hyperellipsoids and hence of finite extent" [68]. Yet another method, closely related to the one last mentioned, is to model not only the *distance from feature space* (DFFS) which is equivalent to TD, but also the *distance in feature space* (DIFS), which is the orthogonal residual [70].

The method described here theoretically is well known for practical applications, for example in [71] it is said that "eigenvector decomposition has been shown to be an effective tool for solving problems which use high-dimensional representations of phenomena which are intrinsically low-dimensional." and the authors describe their method by the following: "Our learning method estimates the complete probability distribution of the object's appearance using an eigenvector decomposition of the image space. The desired target density is decomposed into two components: the density in the principal subspace (containing the traditionally-defined principal components) and its orthogonal complement (which is usually discarded in standard PCA)." "The reconstruction error (or residual) of the eigenspace decomposition [...] is an effective indicator of similarity."[71]

For the special case of $\Sigma = I$ the authors of [37] derived a seemingly similar result in the context of tangent subspace estimation. They define a projection operator onto the tangent subspace using the tangent vectors and their norm, then describe the usage of a metric defined by a positive semi-definite matrix consisting of the projection operator subtracted from the identity matrix.

5.1.2 Estimating Derivatives of Variation in the References

In some cases there may not exist a-priori information about the *directions* of variation of the data to be modeled, but it is known that there exists class specific variability in the data. That is, there is knowledge about the existence of variability in some classes, but one is not aware of the kind of variability. In this case the goal is to estimate the derivatives of variation for each class in order to be able to use the methods described in the previous section.

Given data $x_1, \ldots, x_N$, a reference μ and a covariance matrix Σ one can apply a maximum likelihood (ML) approach to estimate the directions μ_l, assuming knowledge of the number of dimensions L to be sought for. Maximizing the likelihood

$$\prod_n p(x_n|\mu) \tag{5.13}$$

is equivalent to minimizing the (double) negative log-likelihood (constant terms have been dropped)

$$\sum_n d(x_n, \mu) = \sum_n (\mu - x_n)^T \Sigma^{-1} (\mu - x_n) - \sum_l \frac{((\mu - x_n)^T \Sigma^{-1} \mu_l)^2}{\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l} \tag{5.14}$$

This in turn is equivalent to the maximization with respect to the μ_l of

$$\begin{aligned} \sum_n \sum_l \frac{((\mu - x_n)^T \Sigma^{-1} \mu_l)^2}{\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l} &= \sum_l \sum_n \frac{\mu_l^T \Sigma^{-1} (\mu - x_n)(\mu - x_n)^T \Sigma^{-1} \mu_l}{\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l} \\ &= \sum_l \frac{\mu_l^T \Sigma^{-1} S \Sigma^{-1} \mu_l}{\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l} \end{aligned} \tag{5.15}$$

with $S = \sum_n (\mu - x_n)(\mu - x_n)^T$ being the sample covariance matrix of the data. This is maximized when the vectors $(\Sigma^{-\frac{1}{2}})^T \mu_l$ correspond to the L eigenvectors with the largest eigenvalues of the matrix $(\Sigma^{-\frac{1}{2}})^T S \Sigma^{-\frac{1}{2}}$, its principal components.[2] For a proof one only needs to consider the constraint that the vectors $(\Sigma^{-\frac{1}{2}})^T \mu_l$ are orthonormalized and the problem is similar to finding the principal components for a given covariance matrix, leading to an eigenvalue equation (see e.g. [65, p. 297]).

For example, assuming $\Sigma = \sigma^2 I$ (as for example in a nearest neighbor setting with Euclidean distance) this implies using the directions of largest variance of the data. In a more general case one might consider using the global covariance matrix for Σ and the class specific covariance matrix for S. This is equivalent to performing a global whitening transformation for a transformation of parameter space and then employing the L eigenvectors with the largest eigenvalues of the class specific empirical covariance matrix as tangent vectors. Results for this method are presented in Chapter 7. These considerations lead to algorithms similar to those presented in [38] and [68].

In this maximum likelihood setting one obtains no satisfactory solution for the case $\Sigma = S$ (because S is already the estimate for Σ resulting in maximum likelihood). In this case the expression to be maximized reduces to

$$\sum_l \frac{\mu_l^T \Sigma^{-1} \mu_l}{\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l} \tag{5.16}$$

and for $\gamma \to \infty$ the term is a constant and therefore not helpful for finding the 'best' μ_l. For other values of γ a further transformation of the expression to

$$\sum_l 1 - \frac{1}{\gamma^2} \frac{1}{1 + \gamma^2 \mu_l^T \Sigma^{-1} \mu_l} \tag{5.17}$$

shows that the only information obtained is that the term $\mu_l^T \Sigma^{-1} \mu_l$ should be maximized, which only states that the length of the vectors μ_l should grow infinitely but does not include information about the direction. If unit (or constant) lengths are assumed, one obtains the directions of smallest variance as directions for the tangent vectors, because the product contains Σ^{-1}. This may not be very helpful for practical applications, but it makes sense as a result of maximum likelihood considerations, because it minimizes the reconstruction error, that is, most information (in the meaning of variance) is retained.

The usage of the (local) principal components as directions of increased variance has been mentioned in the context of (local) subspace classifiers before, but it is usually not derived from domain knowledge. For example in [68] the largest principal components are preserved (although not increased, as in the approach stated here) while other directions are assumed to be directions resulting from noise, but no theoretical justification for that approach is given. In [44] a mixture of (local) linear models is regarded, where the directions are estimated like in PCA, but no justification for this approach is given.

Note that if PCA is used for feature reduction the approach is usually contrary to the one proposed here, since the directions of largest variance are preserved. In this context one must be careful to distinguish between class specific principal components and global principal components.

[2] Here $\Sigma^{-\frac{1}{2}}$ is defined as the matrix for which $\Sigma^{-\frac{1}{2}} \Sigma (\Sigma^{-\frac{1}{2}})^T = I$ holds, which exists, if Σ is a non singular covariance matrix. This is also the transformation matrix of the whitening transformation (see [32, pp. 28ff] and page 27).

Local Estimation of the Derivatives of Variation

The following considerations deal with estimation of the directions of variation for a certain vector x_n of the training set locally. One method accomplishing this is to find a subset X_n of the training data belonging to the same class and then use the set of vectors $\{x' - x_n \,|\, x' \in X_n\}$ (or an orthonormalized equivalent set that spans the same subspace) as a set of tangents for that pattern. The set X_n can be chosen in different ways, two straightforward solutions being (a) all patterns within a certain distance, but this has the drawback that the cardinality of X_n is not fixed. Or (b) one fixes the cardinality $|X_n| = L$ and uses the L closest vectors.

This method is known as local subspace classifier (LSC)[63], which "fills the gap between the subspace and prototype principles of classification." [3] It can be extended using the previously described ideas employing the eigenvectors with largest eigenvalues of the matrix

$$\Xi_n := \sum_{x' \in X_n} (x' - x_n)(x' - x_n)^T \tag{5.18}$$

If the first $|X_n|$ eigenvectors are used (and $|X_n| \leq \operatorname{rank} \Xi_n$) this approach is identical to the first one. On the other hand this approach is more flexible (which may be an advantage or a disadvantage). For more flexibility one might introduce coefficients for the outer products in the sum, for example depending on the distance of x' to x_n. Note that Ξ_n can be regarded as a local covariance matrix, if x_n is considered the mean of this local distribution.

Extending the approach to discriminative training can be done for example by using a local LDA instead of the PCA. This path has been presented in [39]. The authors state there that their approach of a "discriminant adaptive nearest neighbor metric" (DANN) based on a local LDA, could be generalized using invariant distance measures like tangent distance.

In [68] this idea has been pursued for a mixture of subspace-constrained Gaussians. The authors state that "To overcome the limitations of a globally linear model, local PCA's can provide an effective means to deal with non-linear structures in multivariate data." They propose not to fix the local dimension L, but estimate it locally according to a global resolution parameter.

5.1.3 Known Derivatives of Variation in the Observation during Recognition

As well as the reference vector μ can be subject to transformations that do not affect class-membership, this can be the case for the observation vectors x. Similar to the first case one can now consider for a given x all variations $x_\alpha = x + \sum_l \alpha_l x_l$ with the same notational conventions as in Section 5.1.1. The corresponding density function for given α considering an underlying Gaussian distribution can then be written as

$$\begin{aligned} p(x|\mu, \alpha, \Sigma) &= \mathcal{N}(x_\alpha|\mu, \Sigma) \\ &= \frac{1}{\sqrt{(2\pi)^D |\Sigma|}} \exp\left(-\frac{1}{2}(\mu - (x + \sum_l \alpha_l x_l))^T \Sigma^{-1} (\mu - (x + \sum_l \alpha_l x_l))\right) \end{aligned} \tag{5.19}$$

and with the same considerations as before

[3] "In the LSC process, the nearest prototypes to the input vector in all the classes are sought. A local subspace – or more precisely a *linear manifold* – is then spanned by these prototype vectors in each class. The classification is based on the minimum distance from the input vector to these subspaces." [63]

$$\begin{aligned} p(x|\mu) &= \int p(x,\alpha|\mu)\, d\alpha \\ &= \int p(\alpha|\mu)\, p(x|\alpha,\mu)\, d\alpha \\ &= \int p(\alpha)\, p(x_\alpha|\mu)\, d\alpha \\ &\approx \max_\alpha \left\{ \mathcal{N}(\alpha|0,\gamma^2 I)\, \mathcal{N}(x_\alpha|\mu,\Sigma) \right\} \end{aligned} \tag{5.20}$$

Since the only difference in the calculations is the replacement of the term '$+\sum_l \alpha_l \mu_l$' by '$-\sum_l \alpha_l x_l$', one can perform exactly the same calculations, substituting μ_l with $-x_l$ and one obtains (as the negation cancels out in all places)

$$\begin{aligned} d(x,\mu) &\approx (\mu - x)^T \Sigma^{-1} (\mu - x) - \sum\nolimits_l \frac{((\mu - x)^T \Sigma^{-1} x_l)^2}{(\frac{1}{\gamma^2} + x_l^T \Sigma^{-1} x_l)} \\ &= (\mu - x)^T (\Sigma^{-1} - \sum\nolimits_l \frac{(x_l^T \Sigma^{-1})^T (x_l^T \Sigma^{-1})}{\frac{1}{\gamma^2} + x_l^T \Sigma^{-1} x_l})(\mu - x) \end{aligned} \tag{5.21}$$

The resulting form of the distribution cannot be expressed as a (degenerate) Gaussian here as the matrix depends on the value of x. It seems reasonable, that the problems with normalization can be regarded in the same way as before, with projection not linear, but along a curve in pattern space. Yet, this insight may not be helpful for practical purposes, because the curve space and the manifold resulting from projection along it are possibly very difficult to handle.

5.1.4 Known Derivatives of Variation in the Observation during Training

One can also look at the a-priori knowledge about the data from another point of view, namely during estimation of parameters for a distribution, for example when training a Gaussian (mixture) density for recognition. In that case, one might be interested in using the additional knowledge only during the training respectively the estimation procedure. It is not modeled in the distribution then, but rather used for a more reliable estimation of parameters.

Consider a Gaussian distribution $\mathcal{N}(x|\mu,\Sigma)$ with parameters μ and Σ to be estimated and training data $x_1, \ldots, x_N \in \mathbb{R}^D$. Furthermore, one has knowledge about the variability of the data such that for each x_n the tangents $x_{n1}, \ldots, x_{nL} \in \mathbb{R}^D$ are known and thus for a vector $\alpha \in \mathbb{R}^L$ one obtains as before $x_{n\alpha} = x_n + \sum_l \alpha_l x_{nl}$. If we introduce a matrix $T_n \in \mathbb{R}^{D\times L}$ which consists of the L tangent vectors this can be written as $x_{n\alpha} = x_n + T_n \alpha$ for ease of notation. One can now modify the maximum likelihood estimates for the parameters

$$\mu = \frac{1}{N} \sum_n x_n \,, \quad \Sigma = \frac{1}{N} \sum_n (x_n - \mu)(x_n - \mu)^T \tag{5.22}$$

by distributing the weight 1 of each training vector x_n over "infinitely many" variations $x_{n\alpha}$ with weight $p(\alpha)$, here $p(\alpha) = \mathcal{N}(\alpha|0,\Sigma_\alpha)$.

This has no effect on the new means μ_T:

$$\mu_T = \int \frac{1}{N} \sum_n p(\alpha) x_{n\alpha}\, d\alpha$$

$$
\begin{aligned}
&= \frac{1}{N} \sum_n \int p(\alpha)(x_n + T_n\alpha)\, d\alpha \\
&= \frac{1}{N} \sum_n \int p(\alpha) x_n \, d\alpha + \int p(\alpha) T_n \alpha \, d\alpha && (5.23) \\
&= \frac{1}{N} \sum_n x_n = \mu && (5.24)
\end{aligned}
$$

where in the last step it was used that x_n and T_n are independent of α and the expected value of α is zero, which implies that the expected value of the linear function $T_n\alpha$ is also zero (and the expected value of x_n is x_n). Thus, the second term in (5.23) vanishes.

Using similar calculations one can show that on the other hand the new covariance matrix Σ_T changes:

$$
\begin{aligned}
\Sigma_T &= \int \frac{1}{N} p(\alpha) \sum_n (x_{n\alpha} - \mu)(x_{n\alpha} - \mu)^T \, d\alpha \\
&= \frac{1}{N} \int p(\alpha) \sum_n (x_n + T_n\alpha - \mu)(x_n + T_n\alpha - \mu)^T \, d\alpha \\
&= \frac{1}{N} \sum_n \int p(\alpha) \left[(x_n - \mu)(x_n - \mu)^T + (x_n - \mu)(T_n\alpha)^T \right. \\
&\qquad \left. + (T_n\alpha)(x_n - \mu)^T + (T_n\alpha)(T_n\alpha)^T \right] \, d\alpha \\
&= \frac{1}{N} \left[N \cdot \Sigma + \int p(\alpha) \sum_n (T_n \alpha\alpha^T T_n^T) \, \mathrm{d}\alpha \right] \\
&= \frac{1}{N} \left[N \cdot \Sigma + \sum_n T_n \left(\int p(\alpha)\alpha\alpha^T \, \mathrm{d}\alpha \right) T_n^T \right] \\
&= \Sigma + \frac{1}{N} \sum_n T_n \Sigma_\alpha T_n^T && (5.25)
\end{aligned}
$$

If independence and equal variance of the components of α is assumed, that is $\Sigma_\alpha = \sigma_\alpha^2 I$, this can be rewritten as

$$
\Sigma_T = \Sigma + \sigma_\alpha^2 \sum_l \frac{1}{N} \sum_n x_{nl} x_{nl}^T \qquad (5.26)
$$

which resembles equation (5.12) with the μ_l replaced by an average of the x_{nl}. But the two expressions are not exactly of the same structure, because the average is taken over the outer product of the vectors and not the outer product of the average is used, which is an important difference.

After these calculations had been carried out, two publications came to my attention, in which a similar version of (5.25) had been published: A result for support vector machines had been presented in [81], where the resulting matrix had been called *tangent covariance matrix* and almost the same approach as presented here, had lead HASTIE & SIMARD to their result in [37], but they did not consider it in the general setting. The latter publication reports no improvements in classification accuracy, though, which is contrary to the findings of the image recognition group at the Chair of Computer Science VI, who could improve results for Gaussian mixture densities using this variance description.

The estimation of parameters changes in a fundamental way, if it is assumed that tangent distance will also be used during recognition. This has consequences for the references as well as the covariance matrix. As HASTIE et al. pointed out in [38] it is possible to compute the references as models which "minimize the average tangent distance from a subset of the training images". In the experiments carried out for this work these models did not lead to a better recognition performance (compare Chapter 7).

On the other hand if tangent distance is used for the observation vectors and one applies maximum likelihood estimation for Σ the result is that the variance in the direction of the x_{nl} is reduced. This is due to the fact that this variance is already accounted for by the tangent vectors, that is the variance to be explained diminishes in those directions. It is not clear whether this should be seen as a positive or as a negative effect. Some more considerations with respect to this topic are presented in Chapter 7.

If the resulting probabilistic models are interpreted as generative models for images, the obtained results are similar to those of HINTON et al. [44], who infer them from a variant of the neural net inspired tangent prop algorithm [87].

5.1.5 Estimating Derivatives of Variation in the Observation

If no information about the derivatives of transformation is available for the observation vectors, they may be estimated from patterns of the same class, which are close to the regarded one. This can be done in the same way as described in Section 5.1.2 for the training patterns, but the method may not be useful for the recognition process, because these directions need to be calculated once for each class that is hypothesized. Furthermore this method cannot be used in a nearest neighbor classifier, since this leads to zero distance for all classes, if used in the straightforward manner. This can be explained by the following argument. If the closest references to the observation are taken into account to calculate the directions of variation, these vectors point exactly towards the used references. Then employing these directions as tangent vectors implies zero distance component for the direction and thus zero overall distance since it is the only component.

5.1.6 Combining the Approaches

It is possible to combine the different approaches presented, e.g. combining (5.9) and (5.21) yields double-sided TD. This may be combined with (5.26) giving

$$d(x,\mu) = (\mu - x)^T \left(\Sigma_T^{-1} - \sum_{l=1}^{2L} \frac{(u_l^T \Sigma_T^{-1})^T (u_l^T \Sigma_T^{-1})}{u_l^T \Sigma_T^{-1} u_l} \right) (\mu - x) \tag{5.27}$$

With $\{u_1, \dots u_{2L}\}$ being a set of vectors spanning the same subspace as the set $\{x_1, \dots x_L, \mu_1, \dots \mu_L\}$ with the condition $u_l^T \Sigma_T^{-1} u_{l'} = 0$ for $l \neq l'$. Since the x_l and the μ_l play essentially the same role here, and this is in turn the same as for the differences $x' - x_n$ from the previous Section (Equation (5.18)), one might construct an even more general case, in which the first principal components of the matrix

$$\sum_{x' \in U(x_n)} \beta_1(||x' - x_n||) \cdot (x' - x_n)(x' - x_n)^T + \sum_l \beta_2 \sum_{n'} x_{n'l} x_{n'l}^T + \beta_3 x_{nl} x_{nl}^T + \beta_4 \mu_l \mu_l^T \tag{5.28}$$

are used as tangent vectors for the calculation of the distance $d(x_n, \mu)$. Different settings of the coefficients $\beta_1(\cdot), \beta_2, \beta_3, \beta_4$ allow to reproduce each special case considered before, thus arriving at a valid generalization.

5.2 Structured Covariance Matrices

Tangent distance leads to a certain structure in the covariance matrix and its inverse. This section deals with this and other approaches, especially those based on pixel neighborhoods, that also result in a typical structure of the (inverse) covariance matrix.

As the previous Section showed, the tangent distance for tangents on the side of the references can be computed using (an approximation of) the structured covariance matrix Σ' (5.12):

$$\Sigma' = \lim_{\kappa\to\infty} \left(\Sigma + \kappa \sum\nolimits_l \frac{\mu_l \mu_l^T}{\mu_l^T \Sigma^{-1} \mu_l} \right) \tag{5.29}$$

where Σ is the empirical covariance matrix of the data. The structure here consists of directions with infinite variance, although the matrix Σ' cannot be used explicitly (as it does not exist for $\kappa \to \infty$), yet calculating single-sided tangent distance is equivalent to using Σ'.

The tangent structure is inherently linked with a structure in the inverse covariance matrix (which actually appeared first in the considerations of the probabilistic description of tangent distance). This is given by Equation (5.11) and consists of a zero distance component in the directions of the tangent vectors, caused by the term subtracted from Σ^{-1}. The additive structure in the covariance matrix is thus reflected in a negative structure in the inverse covariance matrix.

Similar considerations about influences on the covariance structure are of course possible for double sided tangent distance or the local subspace classifier, which is a special case of single sided tangent distance. HINTON et al. [44] consider this in the context of local PCA and describe "PCA as a way of fiercely constraining a full covariance Gaussian but nevertheless leaving it free to model important correlations." And in [43] one finds: "Note that FA [Factor Analysis] is just a particular way of limiting the number of parameters that define the covariance matrix used to model data."

5.2.1 Structures based on Pixel Neighborhoods

It is interesting to see that structures in covariance matrices are also used for other reasons, especially parameter reduction during model estimation. One drawback using Bayesian classifiers based on Gaussian mixture densities or kernel densities is the fact that the number of model parameters for such a classifier is extremely high, requiring a very large amount of training data (which is not always available) for reliable parameter estimation. A common approach to overcome this difficulty is the use of diagonal instead of full covariance matrices, i.e. the use of variance vectors. Note that the use of a diagonal covariance matrix can be interpreted as a very simple approach to structuring covariance matrices, where a rather harsh approximation of a full covariance matrix is used in order to reduce the number of free model parameters.

Special structures in covariance matrices for image distributions can be obtained by assuming that the grayvalue of a certain pixel only depends on the grayvalues of the neighboring pixels. This is an assumption quite frequently found in image analysis and looking at empirical covariance matrices, this seems to be the case in many datasets. For example in [81] the authors describe for their experiments in the context of support vector machines "Local correlations in the images were assumed to be more reliable than long-range correlations."

Using full covariance matrices for object recognition implies that any two pixels within an image are correlated. On the other hand, using diagonal covariance matrices, it is assumed that there is no correlation between different pixels at all. Both such approaches are somewhat extreme: the

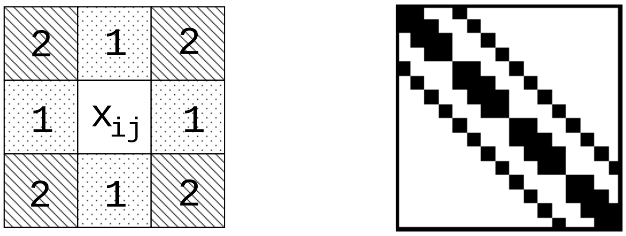

Figure 5.1: Neighborhoods N_1 (1) and N_2 (1,2) (left). Resulting band structure of the inverse covariance matrix Σ^{-1} for N_1 and 4×4 pixels sized images (right). Black pixels represent non-zero entries in Σ^{-1}.

first suffers from a large amount of parameters, whereas the latter may be an unrealistic model in some applications. As a compromise, one could use a full covariance matrix with the restriction that the grayvalue of a given pixel only depends on its neighbors. Thus, the number of non-zero entries in the respective inverse covariance matrix can be significantly reduced.

Regarding the neighborhoods N_1 and N_2 as shown in Figure 5.1 and assuming that the grayvalue of a pixel x_{ij} only depends on its neighboring pixels, the respective *inverse* covariance matrix Σ^{-1} has a band structure, where the number of bands increases as the regarded neighborhood grows (four bands for N_1, eight for N_2). This can be shown using Markov random field theory. [10] One can show that a certain neighborhood structure in a Markov random field (MRF) implies that all elements of the inverse covariance matrix, which pertain to pixels not belonging to a *clique*, i.e. not mutual neighbors, are zero. This follows from the equivalence between MRF and Gibbs random fields (GRF) (Hammersley-Clifford theorem): Given a neighborhood system N, a random field is a MRF if and only if its joint distribution is a Gibbs distribution with respect to the cliques of N [65, pp. 180ff]. Informally stated, the probability density function for the realization of pixels that are not mutual neighbors are stochastically independent. Therefore, the contribution of the second order term for such pixels to the value of the joint probability density function is zero.

Note that the above correspondence is only true, if only cliques of size two are allowed. The number of dependencies of pixels, that is the clique size, finds a direct match in the dimensionality of the covariance description. For clique sizes of maximum one (no dependencies between neighboring pixels, empty neighborhoods) a first order variance vector is sufficient, for cliques with a maximum of two members one needs elements of the second order covariance matrix. This consideration could be continued for larger clique sizes leading to higher order covariance structures as models.

Thus, any entry of Σ^{-1} that does not lie on the diagonal or the bands is zero. Note that some entries on the first band are zero, too (cp. Figure 5.1). This is due to the fact that wrap-around is not considered here, e.g. a pixel at the left border of an image is not a neighbor of the corresponding pixel at the right border.

Considering this, a maximum-likelihood estimation of Σ (i.e. maximization of $\prod_n p(x_n)$ with respect to Σ, given the training observations x_n, $n = 1, ..., N$) yields the interesting result, that one can only give estimations for those entries in Σ that lie on the diagonal or the bands. Thus, one knows each entry in Σ that is not known in Σ^{-1} (where one has knowledge about the occurrences of zeros) and vice versa. Hence, an estimation for Σ^{-1} (under the constraint that only neighboring pixel depend on each other) can be found by solving

$$\Sigma \cdot \Sigma^{-1} = I \tag{5.30}$$

with known elements in both matrices. This is a (very large) bilinear equation system (that is, the highest order terms are of the form $const \cdot xy$ where x and y are unknowns) with the same number of unknowns as equations. This implies that in the general case there is a unique solution. It is not a trivial task to find that solution, though, when the system consists of $n(n+1)/2$ equations (with the same number of unknowns), with n being the number of pixels in the image. For example for the USPS database that means a bilinear system of 32896 equations needs to be solved.

5.2.2 Relation to Tangent Distance

At the beginning of this Section one relationship between structures in the covariance matrix and tangent distance has already been mentioned. Now it is interesting to find the connection between the modification in the covariance structure introduced by pixel neighborhoods and tangent distance respectively invariance. This can be traced by the following considerations.

Consider an existing neighborhood structure N with a set $\mathcal{C}$ of cliques of the form $C = \{c_1, c_2\}$ with pixels c_1, c_2 being mutual neighbors. Since the maximum clique size is two, only pairs are considered here with cliques of size one denoted by $c_1 = c_2$. Consider furthermore that on particular data set this structure led to an estimation of the inverse covariance matrix Σ^{-1} with

$$\Sigma^{-1}_{ij} = 0, \quad \text{for} \quad (i,j) \notin \mathcal{C} \tag{5.31}$$

and possibly nonzero entries in all other positions. Moreover, the estimated covariance matrix Σ has the known entries

$$\Sigma_{ij} = S_{ij}, \quad \text{for} \quad (i,j) \in \mathcal{C} \tag{5.32}$$

and unknown entries elsewhere.

Now one can have a look at the changes that occur, when a clique $\{c_1, c_2\}, c_1 \neq c_2$ is introduced into $\mathcal{C}$. The changes introduced in Σ will occur mainly at the position (c_1, c_2) and (c_2, c_1), where the previous entry is replaced by $S_{c_1 c_2}$, yielding a new estimate Σ'. If the other changes introduced are neglected (they are actually zero if the introduced clique is the last possible one) one can write for the difference $\Delta\Sigma = \Sigma' - \Sigma$:

$$\Delta\Sigma_{ij} \approx \begin{cases} 0 & \text{, for } \{i,j\} \neq \{c_1, c_2\} \\ s & \text{, for } \{i,j\} = \{c_1, c_2\} \end{cases} \tag{5.33}$$

with $s > 0$. Now $\Delta\Sigma$ has the following eigendecomposition:

$$\Delta\Sigma = s v_1 v_1^T - s v_2 v_2^T \tag{5.34}$$

with the two vectors v_1 and v_2 given by $v_{1c_1} = v_{1c_2} = 1/\sqrt{2}$ and $v_{2c_1} = -v_{2c_2} = 1/\sqrt{2}$ and all other vector components equal to zero. Using the interpretation of Equation (5.11) this can be viewed as a tangent with weight $\kappa = s$ instead of $\kappa = \infty$ (or $\lambda = 1/(1+s)$ instead of $\lambda = 1$) in direction of v_1, which represents exactly the introduced clique and another tangent with *negative* weight $\kappa < 0$ in direction of v_2, which represents a divergent behavior of the two clique pixels.

Informally this can be described as follows: Adding the additional clique $\{c_1, c_2\}$ to the neighborhood structure has the effect, that deviations of the pixel values from the reference values at positions c_1 and c_2 in the *same* direction (e.g. greater grayvalues than the references for both pixels) result in smaller distance than before, while deviations in *opposite* directions result in increased distance. This can be interpreted as increased invariance of the distance measure with respect to *locally consistent* changes in greyvalue.

To conclude this section the connection to the (discrete)cosine transformation (DCT) is discussed. The DCT diagonalizes a covariance matrix, i.e. the covariance matrix is diagonal in the transformed pattern space, if the requirement is met, that the covariance matrix has a band structure similar to a Toeplitz matrix [74]. This is the case, if the covariance between two pixel position depends only on their relative (wrap-around) distance (in the feature vector, which usually resembles the image structure except for the image borders). The restriction imposed by the structuring described in this section is of a different nature. Here the assumption is, that the Gibbs potential of the grayvalue of a pixel only depends on neighboring pixels, but this connection may be different throughout the image. This can lead to a different structure, although a certain connection between the two approaches is, that the covariance of close pixels is usually greater than that of pixels far from each other in the image.

Chapter 6

Databases and State of the Art

He would never have discovered it if he hadn't been busy engineering a mental block himself. He came across a whole slew of smooth and plausible denial procedures and diversionary subroutines exactly where he had been planning to install his own. The computer denied all knowledge of them, of course, then blankly refused to accept that there was anything even to deny knowledge of, and was generally so convincing that even Ford almost found himself thinking he must have made a mistake.

[5]

This chapter contains an overview of the recognition problems considered for this work. First, the data which is to be classified is described and secondly the results obtained by other research groups are presented.

6.1 Databases

During the experiments for this work a variety of databases were used. In the following, these databases are presented in order to give an idea of the classification tasks the algorithms were designed for. The first two databases contain images of handwritten digits, the third one consists of digitized radiographs of the human body.

Digit recognition (a subproblem to optical character recognition, OCR) has at the same time a great practical importance – one can think of automatic processing of mail envelopes or bank transfers – and serves as an evaluation task since the problem is well defined and common databases are in widespread use. "A digitized handwritten numeral can be represented as a binary or grayscale image. An important pattern recognition task that has received much attention lately is to automatically determine the digit, given the image." [37]

In the medical context the field of image recognition is of some importance because in medical systems a broad variety of images is present, such as radiographs, ultrasound images, computer tomography images and so on.

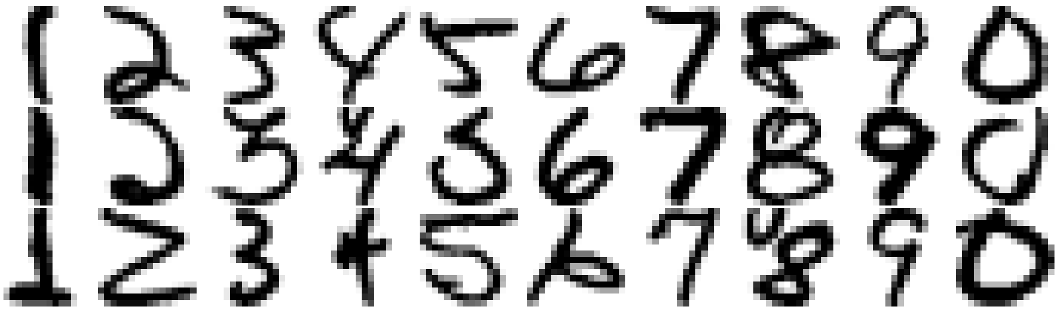

Figure 6.1: Some examples images taken from the USPS test set

6.1.1 US Postal Service Handwritten Digit Database

The well known United States Postal Service Handwritten Digit Database (USPS) consists of handwritten, isolated and normalized images of handwritten digits coming from US mail envelopes. The images are quantized to 256 grayscales[1] and their size is 16×16 pixels (pixel = picture element). The database contains a separate training and test set, where the training set includes 7291 images and the test set consists of 2007 samples. The database is available via ftp through `ftp://ftp.kyb.tuebingen.mpg.de/pub/bs/data`.

Figure 6.1 shows some example images for each of the ten classes taken from the USPS corpus. Despite of the normalization there is still a large variability in the data, which the classifier needs to take into account. Furthermore one can see artifacts due to the fact that the images are segmented from an area containing more writing, for example in the image of an '8' in the last row.

The USPS test set is known as a hard recognition task which can be inferred from the human error rate on the data of 2.5% measured by SIMARD et al. [89]. Figure 7.1 shows the errors together with the correct class label which the best classifier developed during the experiments for this work makes. From that figure the number given for human performance seems comprehensible.

One disadvantage of the corpus is, that there exists no development test set, which leads to effects known as 'training on the testing data' for each of the research groups performing experiments. This refers to effects of evaluating the method over and over on the same data until the best performance for the method seems to be reached. Ideally a development test set would be used to determine the best parameters for the classifiers and the results would be obtained from one run on the test set itself. Nevertheless a comparison of 'best performing' algorithms may lead to valid conclusions. In [37] the authors compare the performance of different algorithms on the USPS database and comment the subject with the following: "Although there is an official test set of data to be used to evaluate different methods, it can be overused. For example, a group may attempt tens or hundreds of different configurations, but only report the results of the best. These caveats hold for any technique with tunable parameters, but are especially pertinent for neural networks which have many."

On the other hand a definite advantage of the USPS task is the availability of many recognition results reported by international research groups, allowing a meaningful comparison of results. Some results for different algorithms are listed in Section 6.2 in Table 6.1.

[1]For the experiments the 256 levels were projected linearly into the range [0.0; 2.0].

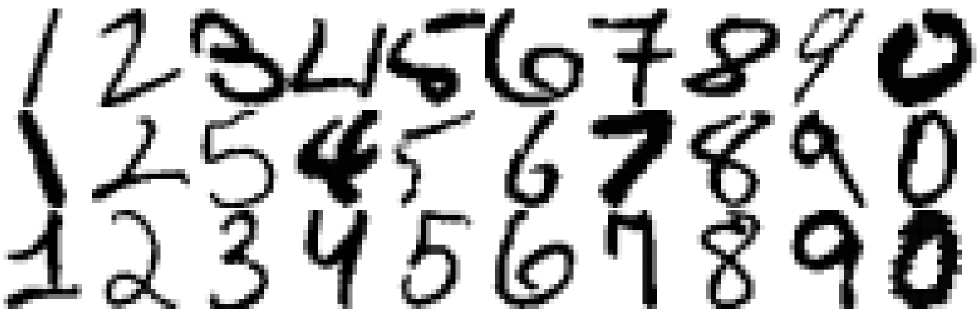

Figure 6.2: Example images taken from the NIST database

6.1.2 NIST Handwritten Digit Database

The (modified[2]) National Institute of Standards and Technology handwritten digit database is very similar to the USPS database in its structure. The main differences are that the images are not normalized and that the corpus is much larger. It contains 60000 images in the training set and 10000 patterns in the test set of size 20×20 pixels with 256 graylevels. It is available from the www through `http://www.research.att.com/~yann/ocr/mnist/`. Some examples from the NIST corpus are shown in Figure 6.2, which illustrate the effects of normalization if compared to Figure 6.1.

The task is generally considered easier than the USPS task for two reasons. On the one hand the human error rate is only 0.2%, although it has not been determined for the whole test set [89]. Secondly the (almost ten times) larger training set allows machine learning algorithms to generalize better. With respect to the connection between training set size and classification performance it is said in [92] that increasing the training set size by a factor of ten about cuts the error rate by half. Looking at Table 6.1 this may also be true for the USPS and NIST databases.

The same arguments for the USPS concerning the absence of a development test set and the availability of research results from other groups also hold true for the NIST database.

6.1.3 IRMA Radiograph Image Database

The IRMA radiograph database contains medical image data from the IRMA project (Image Retrieval in Medical Applications [66]) of the RWTH Aachen, which belong to the six classes abdomen, breast, chest, limbs, skull and spine. The images come from daily routine, are anonymized and secondary digital, that is they have been scanned from conventional film-based radiographs. All images were scanned using 256 gray levels, with the image sizes ranging from about 200 × 200 pixels (e.g. a radiograph of a single finger) to 2000 × 2000 pixels (e.g. a chest radiograph). The anonymized images reflect the distribution of images in the Department of Diagnostic Radiology and were labelled with the six classes by an expert. The corpus consists of 110 abdomen, 706 limbs, 103 breast, 110 skull, 410 chest and 178 spine radiographs, summing up to a total of 1617 images. Furthermore, a smaller set of 332 images that are not labelled exists for testing purposes. Figure 6.3 shows example images from the database which clearly show the different classes. The database contains a wide variation of images, which is shown in Figure 6.4 for the class 'chest',

[2]There also exists a larger database of which this one is a subset. Therefore this database is sometimes also referenced as MNIST database.

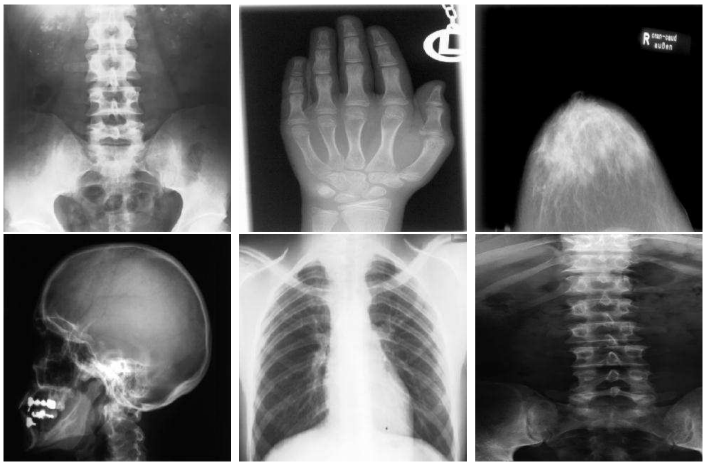

Figure 6.3: Example radiographs taken from the IRMA database, scaled to a common, square size. Left to right: abdomen, limbs, breast, skull, chest and spine.

giving an idea of the high variability. The original images are of varying sizes up to about 2000 pixels in width but were scaled to a common size of 32×32 for classification purposes. The rescaling did not produce significant decrease in recognition rate [23].

Although each image is originally labelled with an 8-digit IRMA category code, in the categorization step one concentrates on the six anatomic regions. Nevertheless, radiograph classification is a hard problem, since on the one hand, the qualities of radiographs vary considerably and there is a great within-category variance (as caused by different doses of X-rays, varying orientations, images with and without pathologies, changing scribor position etc.). On the other hand, there is a strong visual similarity between many images of the classes abdomen and spine (compare Figure 6.3).

Because there are only 1617 images available, a leaving-one-out approach was adopted for cross validation, thus the database served as training and development test set, classifying each image while using the remaining 1616 as training set. After parameter adjustment the classifier was evaluated on a new set of 332 additional radiographs. So the final result does not suffer from training on the testing data, although most results given were obtained on the first set used for evaluation of different approaches and parameter settings.

One drawback of this database is that so far only few results for comparison exist. A few results from other members of the IRMA research group exist as well as a 1-NN baseline result and some results from experiments based on cooccurrence matrices.

To describe the context in which this classification task belongs, in the following a short description of the IRMA system is given, following [23].

An Overview of the IRMA system

From the medical point of view there exist three major applications for automated content based image retrieval [66]:

(1) automatic retrieval of relevant images for follow-up studies within a picture archiving system,

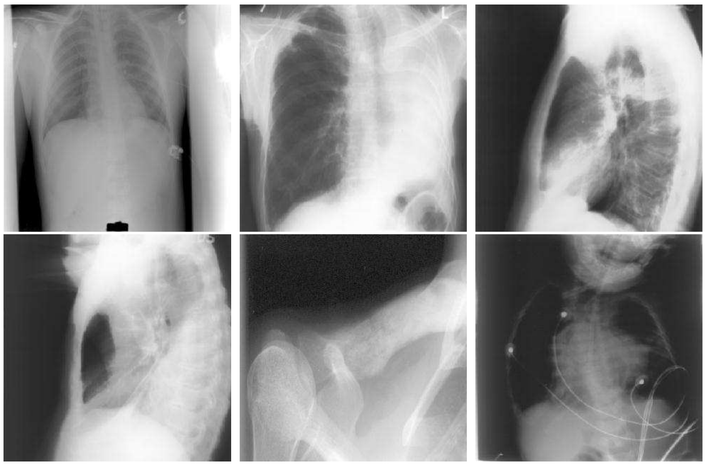

Figure 6.4: Variations within the class ‘chest’

(2) searching for representative images of known diseases and

(3) scientific and educational studies on X-ray patterns.

In contrast to common approaches to image retrieval, the IRMA concept is based on a strict logical and algorithmic separation of the following steps to enable complex image content understanding:

- image-categorization (based on global features)
- image-registration (in geometry and contrast)
- feature extraction (based on local features)
- feature selection (category and query dependent)
- indexing (multiscale blob-representation)
- identification (incorporate a-priori knowledge)
- retrieval (on blob-level)

To enable complex queries for medical purpose, the information retrieval system must be familiar with the class of a given image prior to query processing, as this information is of great interest for the following IRMA steps. For example, searching a pulmonal tumor in a skull radiograph is senseless (as - by definition - a pulmonal tumor is always located in the lungs), and ultrasound images need different processing than radiographs (as the characteristics of an ultrasound image greatly differ from those of a radiograph). Thus, if a radiologist is searching the image database for all radiographs showing a pulmonal tumor, the IRMA system only processes radiographs which are classified as ‘chest’ (or have a posterior probability for ‘chest’ that is higher than a user-defined threshold). On all pictures fulfilling this constraints, the (probably computational more expensive) search for tumors is done, for instance by using statistical classifiers such as proposed in [17]. The categorization step therefore not only reduces the computational complexity needed to answer an IRMA query, it will also most probably reduce the ‘false-alarm’-rate of the system, improving its precision.

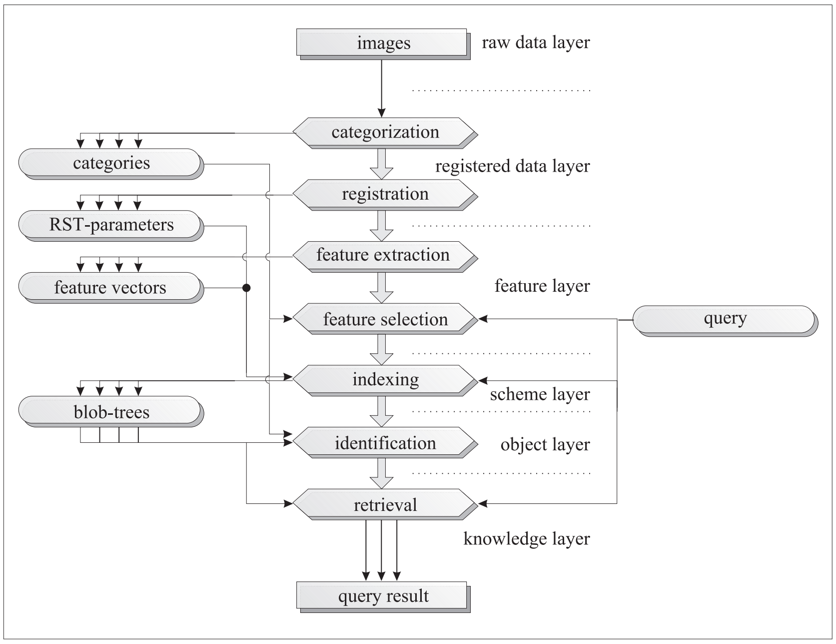

Figure 6.5: The IRMA architecture

Three major classes are defined: image modality (physical), anatomic region (anatomical) and image orientation (technical). In a first step, six anatomic regions are distinguished: (1) abdomen, (2) limbs, (3) breast, (4) skull, (5) chest and (6) spine. These instances build subclasses resulting in hierarchically structured IRMA-categories. While modern DICOM imaging devices provide information required for image classification, automatic content based classification is required for fast archiving of images acquired by film-based modalities such as radiographs. Once the class of a given image has been determined using global features, subsequent IRMA processing steps can use this information to extract problem specific features needed to answer complex queries. As classification is not necessarily unique (a chest radiograph might be labelled 'chest' and 'spine' at the same time), this step is called 'categorization' within the IRMA system. Thus, each image can be linked to several categories and the likelihood for each of these is also stored in the IRMA database. Therefore, classifiers used for categorization should be rather sensitive than specific.

After categorization, the image is registered to a prototype which has been previously defined by an expert or by a statistical data analysis [17, 22]. In the following feature extraction step it is distinguished between so called 'category-free' features (which are suitable for all categories, i.e. a gradient image) and 'category-specific' features, (i.e. segmentation of the ribs in a chest radiograph). In the feature selection step, appropriate features for a given query are chosen. One possibility to do this is performing a linear discriminant analysis (LDA) [27, pp. 114-123], which proved to be very efficient in first experiments [22]. In the indexing step, a compact representation of the given query image and the features extracted is created. Based on each set of feature images, the query image is segmented into relevant regions. Region representation (at multiple scales) will then be done via blobs. This hierarchical multiscale approach will allow the user to retrieve from entire images as well as from regions of interest. The blob-identification step might be useful for queries concerning details defined within organs or other objects in an image. In the final retrieval step, the query is processed via suitable distance measures defined on the entire image or on blob-level respectively.

6.2 State of the Art

In this section state of the art results for the described databases are presented.

Optical Character Recognition

Reported results for the OCR databases are summarized in Table 6.1. The table shows the human performance and the results of the 1-NN classifier as a basis for comparison. ("Nearest neighbor classifiers are extremely simple and always worth trying as a benchmark with any classification task." [37]) The LDA error rate seems high in comparison, but only nine features were used for classification and the authors of [37] report the same error rate for LDA based features.

Best results reported so far on the USPS corpus were obtained with an extended training set augmented with about 2,400 machine printed digits, using a nearest neighbor classifier implementing TD and a boosted neural network. In contrast to this approach in the experiments for this work the effective size of the training set is increased by data multiplication but no new data is added to the training set. In the experiments carried out for this work no better results than 3.3% error rate with the original training set were obtained employing a 1-NN classifier with TD (affine transformations and line thickness). Using a bagged kernel density based classifier and virtual training and testing data (by shifting the images one pixel into eight directions), where different test results were combined using the sum rule, it was possible to reduce the error rate further to 2.2%, showing the effectivity of the TD approach [51].

Table 6.1: Results for OCR databases

Method		Error rate [%]	
		USPS	NIST
Human Performance (SIMARD'93, [89])		2.5	0.2
Linear Classifier (for comparison [8])		-	8.4
Neural Net (LeNet1, LECUN'90, [8])		4.2	1.7
Neural Net (LeNet4, LECUN'95, [8])		-	1.1
Neural Net (LeNet5, LECUN, [98])		-	0.9
Invariant Support Vectors (SCHÖLKOPF'98, [81])		3.0	0.8
Support Vectors (CORTES'95, [87])		-	1.1
Tangent Distance (SIMARD'93, [89])		*2.5	1.1
Boosting (DRUCKER'93, [26])		*2.6	0.7
Local PCA, GMD (MEINICKE'93, [68])		-	1.6
i6:	MD[35]	3.4	1.7
	Invariant Moments, MD[77]	4.0	-
This work:	1-NN, Euclidean distance	5.6	3.5
	1-NN, Euclidean dist., 9D LDA reduced features	10.7	–
	Holographic Classifier	6.0	–
	TD, 1-NN classifier	3.3	1.9
	TD, extensions	2.4	1.0
	TD, extensions, bagging [51]	2.2	–

* training set extended with 2.400 machine printed digits

To somewhat circumvent the 'dangers' of training on the testing data, the parameters which were optimized on the USPS corpus were tried on the NIST corpus for the results given. This shows that no overfitting to the special problem of the USPS database occurred, but the algorithm generalizes considerably well.

One drawback of the USPS database is the relatively small test set size, which makes the error rates statistically less significant [8]: "As our test error rates moved in the range of 3% (60 errors), we were uncomfortable with the large statistical uncertainty caused by the small sample size." This is not the case for the five times larger test set of the NIST database, which makes a more thorough evaluation possible. Nevertheless the USPS corpus is an excellent means for developing a classifier, especially because of the small size.

This work is to a certain degree not concerned with algorithmic resources, so they are not presented in this comparison. For a comparison of different classifiers with respect to time and memory requirement see e.g. [8].

Radiograph Images

For the IRMA database only few results are available since it is not in widespread use but originated from the project at the RWTH Aachen. Table 6.2 shows the results available so far.

The unfavorable results for the linear discriminant analysis (LDA) may be explained by the fact that the estimation of the necessary covariance matrices determines $32^2 \cdot (32^2 - 1)/2 = 523776$ values from only about 1600 training samples. If test and training data are the same the LDA features achieve an error rate of below 1%. This may be surprising at first, but considering the enormous number of degrees of freedom (alreay the number of features is in the same order of magnitude as the number of samples) it seems sensible that the LDA can separate the classes almost perfectly when all data is known. On the other hand this underlines the need for a large amount of training data.

Table 6.2: Results for the IRMA database

Method		ER [%]
1-NN		18.2
Kernel densities (KD)		16.4
Cooccurrence Matrices		29.0
Active shapes (Bredno 2000 [12])		51.1
i6:	KD, thresholding	14.2
	+ Tangent distance	12.9
	+ Image distortion model	10.3
	+ Aspect ratio	8.6
This work:	+ Optimization	8.2
	1-NN, LDA, 5 features	53.2

BREDNO et al. applied an active shape approach to the categorization problem [12]. For form based image retrieval they extracted the outline of the shapes using balloon models and extracted invariant signatures from the outlines for classification. Using a 1-NN classifier the best error rate for leaving-one-out of 51.1% was achieved using invariant moments. On the 496 images where the outline detection was subjectively successful the error rate achieved was 34.9%.

KOHNEN et al. used edge detection in combination with a principal component analysis of training data shapes for invariant classification of extracted forms [59]. The optimization over possible transformation parameters is a computational expensive step in their model and is achieved by applying a simulated annealing procedure. So far two active shape models incorporating domain knowledge have been implemented for the forms “hand” and “vertebra”, which are separated well, but no results for the complete database are available.

Chapter 7

Experimental Results

"Forty-two," said Deep Thought, with infinite majesty and calm.

[1]

This chapter contains the description of the various results that were obtained in the experiments carried out for this work using the databases described in Chapter 6. The chapter is divided into two parts concerning optical character recognition and radiograph categorization, followed by a a short comparison of the two tasks. The main emphasis lies on the different aspects of tangent distance and other invariant distance measures considered, but some further results are also presented.

7.1 Optical Character Recognition

The following section deals with the results for the databases presented in Section 6.1.1 (USPS) and 6.1.2 (NIST). Most of the experiments were conducted on on the USPS database due to its smaller size, which renders it more suitable for testing different approaches. The NIST database was only used as a verification corpus here, in that the best performing classifier for USPS was also tested on this larger data set. First, an overview of the achieved results is given, with some emphasis on data multiplication and classifier combination, then different approaches are presented in more detail.

Table 7.1: Summary of basic results, error rate on USPS [%]

Method			Data multiplication			
			1-1	9-1	1-9	9-9
Baseline, $\Sigma = \sigma^2 I$	1-NN		5.6	4.6	4.7	4.3
	KD		5.5	4.5	4.5	4.2
TD, KD	a priori tangents	SS, μ	3.7	–	–	–
		SS, x	3.3	3.0	2.9	2.8
		DS, μ, x	3.0	2.5	2.6	2.4
	estimated tangents, 7 dim.	SS, μ	5.0	–	–	–

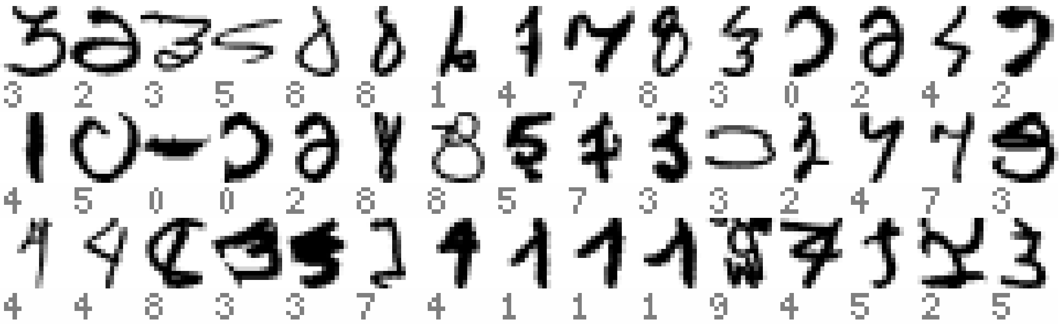

Figure 7.1: USPS errors with class labels for the best result with 2.2% error rate

Table 7.1 summarizes the main results of experiments with the USPS database concerning tangent distance. The notation 'a-b' indicates the increased number of training samples by factor 'a' and increased number of test samples by factor 'b' using data multiplication with image shifts in eight directions. The term 'a priori tangents' refers to the tangents calculated using the derivatives with respect to the affine transformation group and line thickness as proposed by SIMARD and described in Section 4.2, while 'estimated tangents' refers to the estimation of tangents from the covariance matrix Σ as described in Section 5.1.2 and usage of the same tangent directions for all references (here).

Regardless of the chosen distance measure multiplying training and test data consistently improved classification results. The experiments showed that it is advisable to compute the tangents for the test data when computing the single sided tangent distance on this corpus (1-NN performance: 3.4% for observation side vs. 3.8% for reference side, KD performance 3.3% vs. 3.7%). 1-NN was chosen here as baseline result as according to [87], $k = 1$ was the best choice for k-NN on USPS. The usage of the proposed Euler-Cauchy distance measure (see page 52) did not improve the classification results here. Multiplying the training data with tangent approximations or thinned versions of the images did not achieve lower error rates, while the usage of different norms $||\cdot||_\gamma$ enhanced results for the basic KD classifier, but not for the best.

The variety of implemented and tested classifiers respectively parameter settings invited the usage of classifier combination [57] with the hope that the different classifiers together could improve the single best result. This hope seemed to be justified, because the sets of images that were misclassified by the different approaches were not strictly included in each other, but various distinct mistakes were made by the classifiers. With respect to this KITTLER stated in [57]: "It had been observed [...], that although one of the designs would yield the best performance, the sets of patterns misclassified by the different classifiers would not necessarily overlap. This suggested that different classifier designs potentially offered complementary information about the patterns to be classified which could be harnessed to improve the performance of the selected classifier."

Using a combination of five classifier results of different settings the best result could thus be improved to 2.2% error rate. Fig. 7.1 shows the remaining errors with their class labels. The remaining errors can be interpreted differently, some of them appear to be label mistakes, some are hard tasks even for a human (or illegible as the first and third image of the second row), and some shapes do not appear in the training data, as it is the case with the three consecutive '1's in the lower row, which were classified as '7's. Errors like the first image illustrate the limitations of distance based classifiers (there exists a training image of class 'five' which is almost identical, shown in the second row from the bottom of Figure 7.2).

The best result of 2.2% error rate was obtained combining the a posteriori probability density outputs $p(k|x)$ of five different parameter settings for the kernel density classifier using the majority vote rule (in this case the sum rule led to 2.3% error rate). The five settings used were the following:

(1) basic KD classifier, single sided TD, 3.3% error rate

(2) 15 times multiplication of the training data with tangent approximations (7 tangents in 2 directions plus original), single sided TD, KD, 3.4% error rate

(3) double sided TD with 9 times multiplication of training and testing data, KD, 2.4% error rate

(4) double sided TD with 9 times multiplication of training and testing data, squared l_3-norm, k-NN with $k = 2$, 2.4% error rate

(5) double sided TD with 9 times multiplication of training and testing data, squared l_3-norm, k-NN with $k = 6$, 2.5% error rate

The combination results compare well to the experiences expressed in [57]: "Mean rule as well as the median rule have the best classification results. Majority vote rule is very close in performance to the mean and median rules."

For comparison the single experiment which obtained the best result of 2.4% on the USPS corpus (3) was repeated on the NIST database and an error rate of 1.0% was obtained. Table 6.1 of Chapter 6 shows the results in comparison to those obtained by other groups, being not the best but well comparable to the state of the art results. Considering that all optimizations for the method were performed for USPS, the NIST error rate of 1.0% is surprisingly low, which shows that the approach generalizes well and the parameters were not overfitted. Since not all experiments were repeated for NIST, bagging was not applied on this database.

Figure 7.2 shows some examples for the basic 1-NN classifier on the USPS database. 1-NN was used as a baseline result for most experiments here and achieves an error rate of 5.6% on the data. The figure demonstrates the "judgment" of the Euclidean distance for the appearance based approach taken. For instance in the examples for correct classification it can be noticed that similar line thickness seems a very strong factor for overall similarity. This is a result confirmed by the investigations on tangent distance, since best improvements could be obtained using the tangent for line thickness. Also, in the correct classifications the best matching references are very similar to the observation images, which underlines the necessity of large training data sets in order to be able to recognize varying input patterns. Another observation may concern the limitations of the appearance based approach for OCR. The matchings found for the incorrectly classified examples are very similar in a pixel to pixel comparison, but for the human viewer other concepts than intensities are more important here, including e.g. stroke directions and line endpoints.

Table 7.2 shows a summary of the results for tangent estimation for the reference side (in double sided tangent distance, a priori tangents were used on the side of the observation data) from the covariance information of the training data, following suggestions from [38] and the theoretical considerations of Chapter 5. It can be concluded that the estimated tangents provide a better means for the description of the density functions than the a priori tangents, if a low number of references per class is used (at least for classification purposes in this context). This is an advantage if the aim is to build a *fast* recognizer, since the computational effort is closely related to the number of references (or densities) used. A similar result is reported for discriminative training of Gaussian

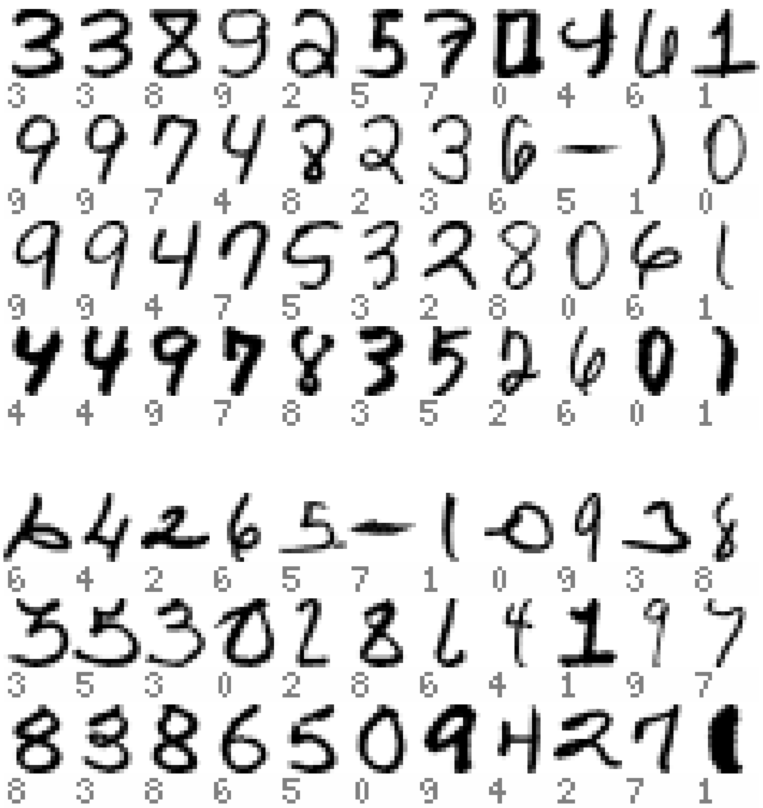

Figure 7.2: Examples for Nearest Neighbor recognition on USPS (with class labels), first image: test pattern, following: best references from each class in order of increasing distance to the test pattern. Top four rows: correct classification. Bottom three rows: incorrect classification.

mixture densities in [22]. If each training sample is used as a reference, the use of the a priori tangents leads to better results than the estimation. A closer look on experiments conforming this is taken in Section 7.1.2. Using the squared relative eigenvalues as weight coefficients for the estimated tangent directions means overestimating the variance proportions quadratically. This implies that large variances are increased, while low variances are decreased relatively. One possible interpretation for the success of this method is that large variances may represent intra-class variation, while low variances represent variance by chance or noise components, which should not be used to represent the class-specific information.

7.1.1 Implementing Tangent Distance

When implementing tangent distance as described by Simard et al. in [89], the first experience was that the way of calculating the tangents is crucial. Taking only finite differences on the original data did only marginally improve classification results. Only when larger templates approximating a suited smoothing prefilter were used, the error rate could be reduced significantly. The used filter is a variant of the Sobel operator [65, p. 213], which can be interpreted as a Gaussian filter kernel combined with differentiation as described in Section 4.2.

Table 7.2: Summary of results for tangent estimation, error rate on USPS [%]

			Subspace dimension						
# Features	# References	Tangent usage	0	7	10	12	14	20	a priori (7)
256	1	SS	18.6	6.8	6.5	5.7	6.2	6.9	11.8
		DS	18.6	6.1	5.6	5.4	5.0	5.7	9.9
	8	DS	12.4	4.6	4.1	4.3	4.7	–	6.5
39[1]	1	SS	12.5	9.9	9.0	9.0	9.2	11.8	–
		SS[2]	12.5	8.9	8.8	8.8	8.7	8.7	–

[1] obtained via LDA using 40 pseudoclasses
[2] vectors weighted with relative squared eigenvalues

The basic results for tangent distance based classification are given in the previous section.[1] In comparison to the results of SIMARD et al. the improvements achieved are based on

- the incorporation of tangent distance into a kernel density based classifier, which proved superior to a k-NN based classifier, for which best results were reported for $k = 1$ on the USPS database and
- the usage of multiplied training and test data, using the virtual test sample method.

Although tangent distance should already compensate shifts of one pixel displacement, the data multiplication approach still led to improvements. This is probably due to the fact that the two approaches model invariance differently (compare Fig. 7.3). While the explicit image shift leads to a movement of the pattern directly along the shift transformation manifold, the tangents model all the regarded transformations and at the same time are a first order approximation of the manifold.

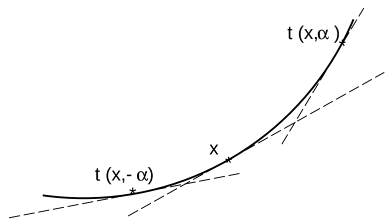

Figure 7.3: Tangents of shifted data in 2D

Since tangent distance has the advantage not to depend on a classifier design, it can be used in a variety of classifiers. Results of experiments with tangent distance in combination with a Gaussian mixture density based classifier were presented by DAHMEN et al. in [17], improving classification performance.

When tangents on the side of the observations as well as on the side of the references are used, there are several methods to circumvent the expensive minimization over the 14 dimensional resulting space spanned by the tangents of both sides (which can be achieved using singular value decomposition or solving the corresponding least squares problem, where the two methods have about the

[1] Note that the value used for σ^2 was 1.0, which yielded best results on the USPS data with graylevels in the range $[0.0; 2.0]$.

Table 7.3: Some results for "line distance"

Method	ER [%] 1-1	9-9
Double sided line distance	3.0	2.8
Double sided tangent distance	3.2	2.4

same operations count). Figure 4.5 shows the points and distances involved. The illustration shows Euclidean distance $||x - \mu||$, single sided tangent distance $||x - x'||$ with tangent t_μ on the side of the reference, single sided tangent distance $||\mu - \mu'||$ with tangent t_x on the side of the observation and double sided tangent distance as indicated. SIMARD et. al. proposed to use the minimum distance between the lines that connect (μ, x') and (x, μ') (called "line distance" here) instead of the overall minimum distance [89]. This requires only a minimization over two dimensions, if the tangents are precomputed and orthonormalized, which is computationally cheaper. (Note that in the case of one-dimensional tangent subspaces, as in the illustration, the two methods are identical, but this is not the case if tangent subspaces have dimensions greater than one.) Table 7.3 shows two results obtained with this method, performing even better than the computationally double sided tangent distance without virtual data, but not in the case of multiplied data.

Looking at the three different distances introduced above, one might think about using a (weighted) combination, but of several experiments with different combinations none improved the performance. One could also use the distance of the projections $||x' - \mu'||$ for classification, but again the experiments were not successful in improving classification.

Furthermore, SIMARD et al. report improvements for a normalization of the tangent distance with respect to the length of the compared vectors when using tangent distance [89]. In the experiments for this work experiments for normalization with respect to Euclidean and squared Euclidean norm were tested, but neither led to improvements in recognition.

7.1.2 Centroid Model and Learned Tangents

In Chapter 5 a method has been derived to estimate the tangent vectors from the training data under the assumption that an underlying low-dimensional variation is present in the data distribution. Some experiments have been carried out in order to verify these theoretical results.

The approach presented in this work is similar to the one presented by HASTIE et. al. in [38], although more importance is placed on the calculation of the references point there. In the experiments for this work the modification of the *reference* with respect to tangent distance did not yield superior results in all cases. The authors describe what they call *centroid* of a set of points as the linear subspace (of a given dimension) that minimizes the average squared norm to the points in that set. If Euclidean distance is used, this yields exactly the subspace spanned by the mean vector and the principal components of the empirical covariance matrix. If instead of the Euclidean distance tangent distance is used, this is no longer true. First consider the case of double sided tangent distance. To determine the minimizing subspace no better algorithm than an iterative method is known here [38]. Conceptually it iterates two phases after calculating the tangents for the given points until convergence:

(1) calculate the tangents for the center

Table 7.4: Single reference results for a priori tangents, error rate on USPS [%]

Distance used in recognition	Center used		
	Mean	Tangent Centroid SS	Tangent Centroid DS
Euclidean	18.6	20.7	19.3
TD, SS, x	14.0	13.3	13.5
TD, DS	9.9	9.5	9.6

(2) calculate the center as point that minimizes the tangent distance given the tangent directions (which amounts to calculating a mean vector in the orthogonal subspace and using the resulting displacement vector in the original space)

Although no guarantee for convergence can be given, the algorithm converges quite successfully. The resulting center is called *tangent centroid*, if the a priori tangents are used. On the other hand, if in step (1) the tangents are determined as *learned* tangents (by calculating the principal components of the data distribution of the given data points with respect to the current center) *and* considering the known tangents of the data in (2), the result is called *tangent subspace* by the authors.[2] Step (1) then amounts to performing a singular value decomposition of the difference vectors in the orthogonal subspace or (equivalently) to calculation of the principal components of the corresponding covariance matrix.

One might also want to regard single sided tangent distance in this context. For this setting there are three different possibilities:

(1) Use a priori tangents on the side of the center.

(2) Use estimated tangents on the side of the center.

(3) Use tangents on the side of the data. (Here a distinction between a priori and estimated tangents is not useful, one can just consider the tangents for the data as 'given'.)

For cases (1) and (2) the center is not affected but is just the arithmetic mean of the data vectors, since the mean does not change under orthogonal projections, which is the effect of tangent distance in this case. For case (3) the mentioned iterative algorithm can be used to determine the center.

As already mentioned in Chapter 5, the approaches which take into account the tangents of the data in the manner described here have the possible disadvantage that directions which coincide with the average tangent directions of the data points receive lower attention. This is because they are disregarded due to the prior tangent distance calculation. With respect to that subject Hastie & Simard write in [37] "Note that the SVD without tangent distance would tend to mix the affine invariances with these digit specific invariances." It is not clear, however, why this should be a disadvantage. To the contrary, it might be considered an advantage if the algorithm is able to determine the best mix of variations needed.

In the following some results from the experiments are given. Table 7.4 shows that the tangent centroids are better suited as references for recognition than the arithmetic mean vector if the a priori tangents are used (here with one reference per class). In this case the description of the

[2] Although this name seems somewhat too general, because it is also used for the subspace created by the a priori tangents here, no confusion should arise, since the meaning should be clear from the context.

Table 7.5: Results for tangent subspace method, error rate on USPS [%], 256 dimensions. Single sided refers to the side of the center.

# References per class	Method		Tangent subspace dimension					A priori tangents
	Calculation[1]	Classification	7	10	12	14	20	(for reference, 7 dim.)
1	Double sided	Double sided	6.4	6.0	5.4	5.7	5.6	9.9
	Single sided (emp. mean)	Single Sided	6.4	6.2	5.5	5.8	5.7	11.8
		Double sided	6.1	5.6	5.4	5.0	5.7	9.9
8	Single sided (emp. mean)	Double sided	4.6	4.1	4.3	4.7	–	6.7
15			4.2	–	–	–	–	6.5

[1] of the tangent subspace

data seems to improve with the usage of the centroid models. Note that the mean vector column also corresponds to the case 'tangent centroid, SS, Center side', as stated above, while the SS tangent centroid in the table refers to the tangents on the side of the data. For a comparison of achieved classification rate one may regard the results for a Gaussian single density, which yields an error rate of 19.5% for the 256-dimensional images respectively 12.8% with prior LDA dimension reduction to 39 dimensions [17]. But as Table 7.5 shows, the estimated tangents perform far better than the a priori tangents with small number of references. Already for one reference the error rate can be improved from 9.9 to 6.1%. (Note that this advantage of the estimated tangents vanishes with increasing number of references; error rates for 7291 references are presented in Tables 7.1 and 7.6. If the number of references is increased while the tangents are not calculated for each reference, but the same tangents are used for all the references of one class, the performance is significantly inferior to the individual tangents, that is for seven estimated tangents the error rate is 5.0% and for twelve tangents it is 4.9%. If one considers that for single references per class and a 14 dimensional tangent subspace already an error rate of 5.0% can be achieved, this is not a significant gain.) Experiments were concentrated on the tangent subspace method in the following, because it performed better in this comparison. It can be seen that the optimum dimensionality for the tangent subspace seems to be about twelve, which is the same result as obtained by the authors of [38]. Furthermore it can be noticed that double sided classification is superior to single sided classification in all observed cases. But on the other hand in calculation of the reference the empirical mean (identical to single sided center calculation) with principal components seems superior to the more complicated double sided tangent subspace calculation using the iterative algorithm. This is a result different from the one presented in [38]. A possible explanation is the lower importance of data point tangent directions in the resulting subspace, which may not be desired. It also shows that the estimated tangent directions have more importance than the estimated means. This setting was tested on a larger number of prototypes, where pseudo classes were constructed using EM training of Gaussian mixture densities [35]. An increasing number of prototypes benefits classification performance as would be expected. (Note that Hastie et. al. in [38, 37] report an error rate of 3.8% for 5 prototypes per class and 4.1% for single references, which are results the experiments described here do not confirm.) For large number of references per class it gets increasingly difficult to estimate the tangent subspace, since during the clustering step some clusters are assigned a low number of data points, such that the number of non zero eigenvalues in the respective covariance matrices decreases. For example with 15 references per class on the average each cluster contains $7291/(15 \cdot 10) \approx 50$ data points, but clusters with far fewer points cannot be easily avoided. Therefore for more references no experiments for greater tangent subspace dimension could be performed.

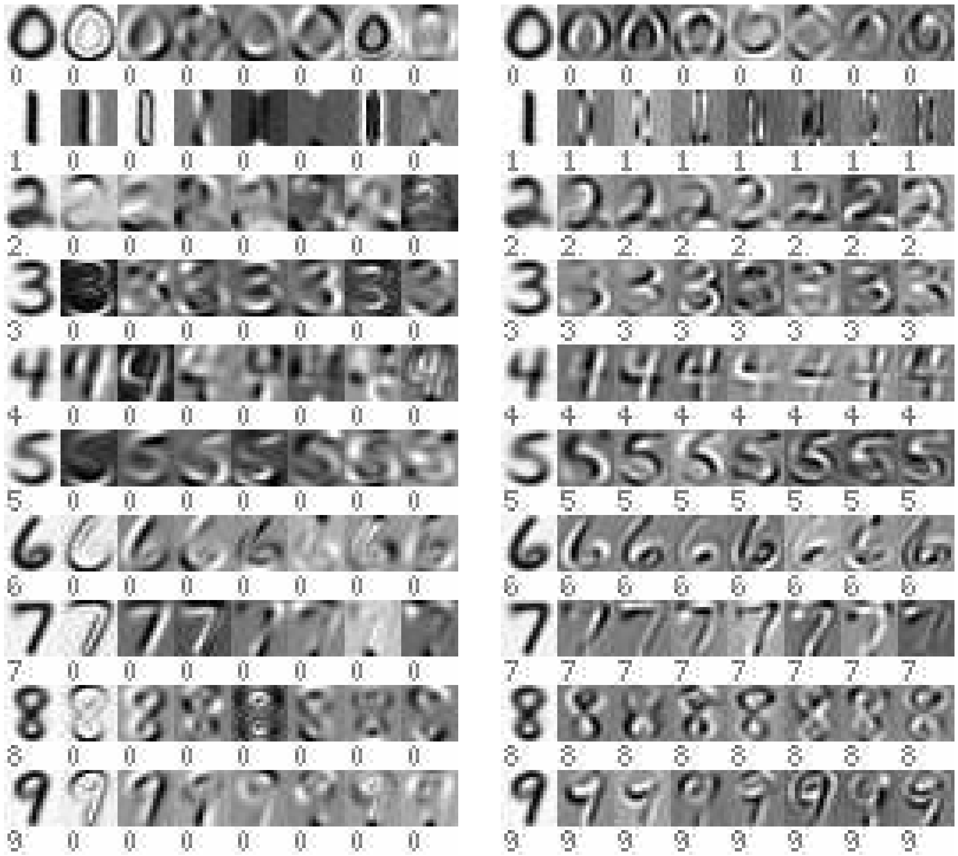

Figure 7.4: Comparison of subspace of a priori tangents (left) and subspace of estimated tangents (right), both orthonormalized for a better comparison and ordered by decreasing eigenvalue. First column: reference vectors.

Figure 7.4 shows the different subspaces for single references. One can see that the estimated subspace contains variations that are not of the geometrical nature as is the case for the a priori tangents. For example the third image in the '2' row (on the right) depicts a tangent vector that modifies the size of the loop in the digit. This is clearly not an affine transformation, but seems a very logical modification to be modeled. This raises the hope of improving tangent distance by adding the estimated tangents, and indeed the estimated tangents outperform the a priori tangents for single references. But as the number of references increases it becomes more difficult to estimate meaningful tangents from less samples per reference.

To circumvent this problem one can resort to the methods introduced in Section 5.1.5. For these local subspace experiments nearest samples from the same class of each reference were taken, then principal components of the local covariance matrix (Eq. (5.18)) were determined and used as tangents. This is only possible for the side of the references, therefore the results should be compared with the a priori tangents on that side (which gave lower performance than the use of the tangents on the observation side in the single sided experiments). Table 7.6 shows the obtained results with that method for a varying size of the local set X_n. The obtained result is best for a set size of about 20 and surprisingly close to the one for a priori tangents, but not as good as for a priori tangents on the side of the observations with 3.3% error rate.

Table 7.6: Some results for the local subspace approach on USPS. Subspace dimension 7.

# samples used for estimation, $\lvert X_n \rvert$	7	10	15	19	20	21	22	25	(a priori)
Error rate [%]	4.3	4.3	4.0	4.0	3.8	3.8	4.0	4.2	3.7

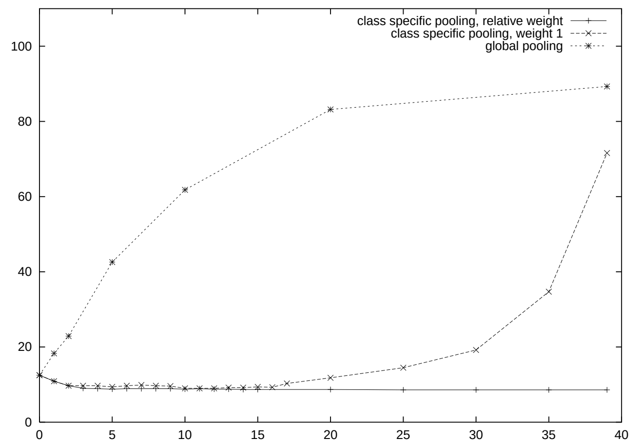

Figure 7.5: Error rate vs. number of dimensions of the tangent subspace, for different settings. USPS, 39 dimensional LDA-reduced features

To obtain results for patterns for which the directions of variation within each class are not known a priori, experiments were carried out after a transformation to a reduced feature space. The patterns were transformed performing an LDA using 40 clusters of the data, yielding 39 features [20, 35]. These features reduce the error rate without tangents and with Euclidean distance from 18.6% to 12.5% for single references. Using the estimated directions of variation this result can be improved to 9.0% for $L = 12$. Employing a weighting of the directions with a function of the eigenvalue lets the error rate drop further to 8.6% and the error rate becomes a monotonously decreasing function of L, because the components with small eigenvalues are practically discarded. This has the advantage that the parameter L needs not be determined explicitly. This dependency is shown in Figure 7.5 for three different variants. As proposed in Section 5.1.2 the setting of $\Sigma = \sigma^2 I$ was chosen and the tangents were estimated as the principal components of the class specific empirical covariance matrix. (The graph labeled with 'global pooling' shows the results for usage of the global covariance matrix. This does not contain the class specific variation information in this case and the error rate increases quickly with the number of eigenvectors used.) If the eigenvectors are equally weighted, the error rate can also be reduced from 12.5% to 8.6% but the right number L of eigenvector needs to be determined, which is not the case for squared relative weight of the eigenvalues here. This corresponds to the following codebook exponents θ_l as a function of the corresponding eigenvalue λ_l:

$$\theta_l = \left(1 - \left(\frac{\lambda_l}{\lambda_{\max}}\right)^2\right) \tag{7.1}$$

Using not the squared but the direct relative eigenvalue the recognition could even be improved to 8.2%, but the monotonicity in the dependency on L was lost. Figure 7.6 illustrates the eigenvalues of the ten class specific covariance matrices for the 39 dimensional features. For reference the performance of the 39 dimensional feature space with all 7291 training patterns should be considered, which is 7.0% error rate if no covariance information is considered.

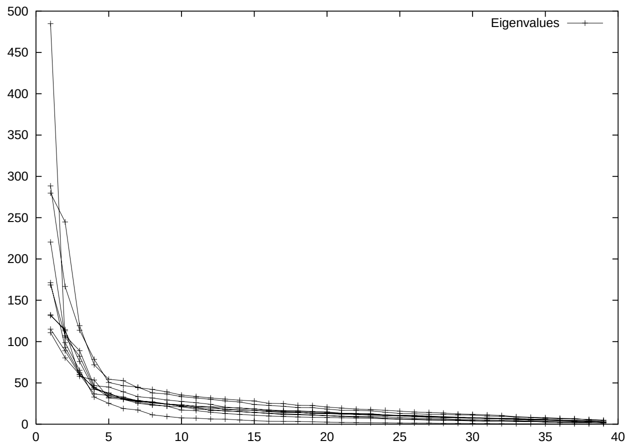

Figure 7.6: Eigenvalues of the class specific covariance matrices for the ten digits. USPS, 39 dimensional LDA-reduced features

On the original images with 256 features, the usage of the weighting of the principal components led to an inferior performance. The best result for squared eigenvalue weighting is here 8.7% error rate and for weighting with the simple eigenvalues it is 6.7%, while for no weighting the error rate can be reduced to 5.5%. The reference value in absence of variational modeling is here 18.6% for one reference per class respectively 5.6% for 7291 references. It seems quite remarkable that the using the variational modeling the error rate for only *one* reference per class can be lower than the error rate for about 700 references per class using just Euclidean distance.

Figure 7.7 shows the estimated tangents for the NIST database, which illustrate the increased variability in the corpus in comparison to the prenormalized USPS images. For example the first tangent vector for the class '1' models the rotational variance in that class, which is not present to that degree in the USPS collection. This explains the extremely high error rates obtained using single references per class, which were 77% for a priori tangents and 60% for estimated tangents. These results suggest that normalization and invariant distance measures can very well be used together to achieve good classification results with small numbers of references.

7.1.3 Comparison of Tangent Vectors

TD is usually applied using the seven transformations proposed in [89] (translations (2), scaling, rotation, axis-deformations (2), and line-thickness) where the first six account for affine variations. Here a number of different tangents were tested including projective transformations, brightness, contrast and different versions of the thickness tangent, but it was not possible to improve the results of the original tangents with USPS. In the following the importance of the different tangents is compared.

The main results are listed in Table 7.7, giving absolute error rate improvements for the different tangents in two settings. In the first part the improvement due to adding the tangent to the remaining six tangents is presented while in the second part the improvement due to adding the specific tangent as the first tangent is shown. For the latter case also the results for the projective tangents are included. The ranking in the two settings is consistent and it can be seen that the combination of all the tangent vectors is the best choice. The combination of these seven tangents

Figure 7.7: Estimated tangents for the NIST database.

may not be the optimal one of all the possible combinations of the various tested tangents. It should be considered, though, that the optimum combination is hard to determine and moreover may not be optimal for other databases at all.

In addition to the tangents listed above, some other approaches were tested. For example splitting of the thickness tangent in two gradient tangents with respect to horizontal and vertical line thickness was implemented with the hope to account for possible independent changes, but the results showed no improvement over the single line thickness deformation. Furthermore a contrast tangent was used, consisting of the relative offset of pixel values with respect to the mean value, but again no improvement could be obtained. These results agree with the statement "Additional transformations have been tried with less success" [87].

In order to confirm the theoretical result (see page 48) stating that the four linear transformations rotation, scaling, axis deformation and diagonal deformation can be expressed in the canonical basis resulting from variation of exactly one of the linear parameters, the resulting four basis tangents were implemented and – not surprisingly – led to identical results.

7.1.4 Euler-Cauchy Approximation

The Euler-Cauchy algorithm was used in the experiments in different parameter settings, the most important ones in this implementation being the iteration number (a preset number of iterations was used as stop criterion here) and the displacement fraction, controlling the relative displacement

Table 7.7: Comparison of tangent vector influence. Improvement is given as absolute difference in error rate with respect to the KD reference.

Tangents used	ER [%] 1-NN	ER [%] KD	Improvement
all 7	3.4	3.3	-
without thickness deformation	4.0	4.0	0.7
without vertical translation	3.9	3.8	0.5
without rotation	3.8	3.6	0.3
without horizontal translation	3.7	3.6	0.3
without diagonal deformation	3.7	3.6	0.3
without scaling	3.6	3.6	0.3
without axis deformation	3.5	3.4	0.1
no tangents	5.6	5.5	-
only thickness deformation	5.0	4.8	0.7
only vertical translation	5.1	5.0	0.5
only horizontal translation	5.4	5.2	0.3
only rotation	5.4	5.3	0.2
only scaling	5.5	5.4	0.1
only diagonal deformation	5.5	5.5	0.0
only axis deformation	5.6	5.5	0.0
only projective Eq.(4.41)	5.4	5.2	0.3
only projective Eq.(4.42)	5.6	5.5	0.0

along the tangent vector. None of the tested versions of the Euler-Cauchy method improved the overall classification rate, although it was hoped that the possibly better modeling of the manifolds could lead to an improvement.

When the algorithm was used with ten iterations and displacement fraction of 0.3 (these settings produced the best results) it was able to correctly classify 57% of the test data which the basic 1-NN classifier failed (if this improvement had been consistent, 2.4% total error rate would have been achieved), but on the other hand introduced additional mistakes in the remaining data, the overall error rate being 3.5%. This result is in agreement with the findings of Simard et al. : "This process did not improve handwritten character recognition, but it yielded impressive results in face recognition." [87]

This result leads to the conclusion that introducing more matching power seems to result in a trade-off between improved matching of correct samples and restricted matching of incorrect samples, since it allows larger variations in the alignment of patterns, both towards correct and incorrect reference images. (Fig. 7.11 in Section 7.1.9 visualizes this effect showing the tolerance of the different distance measures with respect to a horizontal displacement, where one image from each class was randomly selected and the distances to one displaced image were calculated.)

Table 7.8: 8×8 pixels USPS, class-specific covariance matrices for estimation,'N_i-structured' refers to structure according to the Neighborhood in the image of Figure 5.1.

Structuring of Σ		diagonal	N_1-structured	N_2-structured	full	tangent
ER [%]	threshold	5.7	5.5	5.1	4.6	4.6
	interpolation, $\lambda = 0.9$	5.3	5.2	4.8	4.0	4.6

7.1.5 Structured Covariance Matrices

This section contains various results performed to investigate the influence of structured covariance matrices as proposed in Section 5.2. The approach described lead to a large bilinear equation system (Equation (5.30))

$$\Sigma \cdot \Sigma^{-1} = I$$

with known elements in both matrices, that must be solved in order to apply the statistical pattern recognition methods. Without deeper knowledge on the subject of numerical algorithms for that specific task experiments were started using a gradient descent algorithm, which proved not suitable for the task. A Newton algorithm was ruled out, since a derivative matrix of size $32896 \cdot 32896$ would have to be calculated and inverted. Finally one arrived at the Newton-SOR or Gauss-Seidel algorithm which seemed suitable for this problem, seeking help in the basic literature [34, 79, 94]. It was discovered that convergence for large matrices Σ with $256 \cdot 256$ entries could not be achieved in acceptable time, but the algorithm worked well for smaller matrix sizes.[3] Therefore the experiments were carried out using scaled down images of the USPS to sizes of 8×8 and even 4×4 pixels.

The most representative results obtained with an image size of 8×8 pixels are shown in Table 7.8. All results in this section were obtained using a 1-NN classifier. In order to cope with the problems of zero variances, which occur in some diagonal entries of the estimated covariance matrix, two possible methods were tested. First, a minimum threshold for the estimated variance was fixed consistently with the minimum occurring non zero entry, which is denoted by 'threshold' in the table. Secondly, the estimated matrix was linearly interpolated with the identity matrix, that is

$$\Sigma' = \lambda\Sigma + (1 - \lambda)I$$

This amounts to a log-linear interpolation of the probabilities resulting from Euclidean and Mahalanobis distance. As one would have expected, estimation of a band structured covariance matrix reduces the error rate as compared to a diagonal structure. The best results are obtained using a full covariance matrix. This is not surprising, as only a single covariance matrix per class was estimated, using downscaled USPS images. Interestingly, using the tangent distance based structure yields the same results as compared to a full Σ here for the threshold method. In contrast to this, the full covariance matrix yields better results with interpolation, which proved superior in all cases. At the same time, the usage of the covariance structure reduces the computational complexity significantly. Using the original 16×16 pixels sized USPS data, the tangent structure (3.3%) significantly outperforms a full covariance matrix (6.3%), as the number of free parameters in Σ increases by a factor of 16).

[3] Shortly before termination of this work a method described by Pösl in [78] came to my attention, which may be used to speed up the process significantly, but due to a lack of time it could not be determined if and how this may be possible. Furthermore I cannot rule out the existence of much faster, better performing methods than the ones used here.

Table 7.9: Results on USPS size 16×16 with class specific covariance matrices inflated from size 8×8 covariance matrices. 'N_i-struct.' refers to structure according to the Neighborhood in the image of Figure 5.1.

Structuring of Σ		$\Sigma = \sigma^2 I$	$\Sigma = \sigma^2 I$, infl.	diagonal	N_1-struct.	N_2-struct.	full
ER [%]	structure	5.6	5.3	5.6	5.4	5.3	4.5
	+ tangents	3.3	3.8	4.8	4.7	4.3	4.0

One idea to overcome the problem with high-dimensional covariance matrices which lead to inferior classification results was to estimate the respective matrices using the lower dimensional transformed data, then rescale the estimated matrices. This at the same time solves the problem with the convergence of the algorithm used for solving the bilinear equation system, which worked well for size 8×8 but did not converge in acceptable time for size 16×16. For the rescaling, called *inflating* here, one needs to be careful in determining how to rescale the matrices, since the structures in the smaller matrices may be present differently, which depends on the mapping of the two dimensional image to the one dimensional feature vector. Usually it is not enough to just duplicate the entries in the 'inflated' covariance matrix, but they must be distributed to the correct positions. Results for this method are presented in Table 7.9. Note that two different results are given for $\Sigma = \sigma^2 I$, since for a fair comparison the inflation structure needs to be considered. Interestingly this improved classification in the basic case, which may be due to the increased modeling of dependencies between neighboring pixels resulting from inflation. Again it can be seen that an increasing number of parameters in the covariance matrix is useful, if enough data to estimate them is available. It furthermore can be concluded that such a tying of parameters, which is enforced by downsizing and inflating (each estimated parameter in the covariance matrix for the size 8×8 images represents 16 entries of the covariance matrix for size 16×16, but the number of overall pixels is only cut by the factor four) can be used to estimate parameters with higher reliability if only a small number of training samples is available. If the structured covariance matrices are used in combination with tangent distance, it can be observed that unfortunately the positive effects are not additive in this case, but the tangent distance based classifier performs best when the covariance matrix is used as $\Sigma = \sigma^2 I$. This may be due to the fact that the tangents already account for a structuring of the covariance matrix, as presented in Section 5.2.

7.1.6 Image Distortion Model

In this section results for the (unsuccessful) experiments with the image distortion model as presented in Section 4.3 on the USPS corpus are presented. Figure 7.11 (lower right graph) gives an idea why the IDM can probably not be applied successfully to the USPS task, the reason being that almost all images can be mapped well onto each other using this transformation model. In fact, in combination with tangent distance no experiment showed an improvement in classification performance.

If the performance of the image distortion model alone is examined, that is without other variation models as tangent distance, the results must be considered in more detail. The first experiment for a region of radius one, implying a region of size 3×3 allowed for distortion, lead to an increase in error rate of the kernel density classifier from 5.5% to 8.9%. This inferior performance may be due to the large region size in comparison to the image size of 16×16 pixels. Therefore experiments with fractional radii were performed, where the pixel values at positions off the image

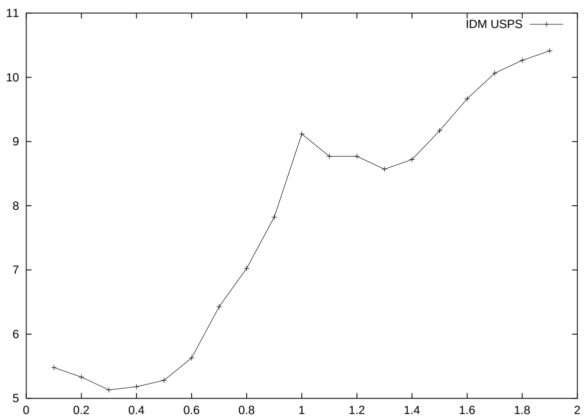

Figure 7.8: IDM error rate [%] on USPS with respect to region radius [pixels]

grid were determined by linear interpolation. Figure 7.8 shows the performance of the resulting 1-NN classifier with interpolated IDM distance on the USPS corpus with respect to the IDM region size given by the radius in pixels. The best obtained error rate was 5.1%, with a corresponding kernel density error rate of 4.9%. Seeing that the *integer* region sizes correspond to local maxima of the error rate here (region size 1 pixel producing significantly higher error rates than size 0) leads to the suspicion of a strong influence of the smoothing effect due to interpolation in the non integer region sizes. Following this presumption leads to the result that with a local smoothing kernel of the form

$$\frac{1}{20}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 12 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

for convolution one obtains the same result of 5.1% respectively 4.9%, which means that the positive effect in this case is only due to the smoothing effect of linear interpolation inherent in non integer region IDM distance. Note that this positive effect of smoothing does not improve classification using tangent distance as other experiments showed. Smoothing is sometimes also seen as 'poor man's approach to invariance', which is affirmed by these results.

The gradient based image distortion model following Equation (4.55) proved somewhat more successful on the USPS database than the basic image distortion model. It did not lead to any improvement on the error rate when used together with tangent distance, but without tangent distance the error rate could be reduced from 5.6% to 4.9% for the 1-NN classifier. The gradient based IDM is closely related to the thickness deformation, which explains, that in combination with tangents no improvements could be obtained. Furthermore, the improvement induced by the gradient based IDM is almost the same as the one resulting from tangent distance with thickness deformation alone, which experimentally underlines the relation. Figure 7.9 shows the error rate as a function of the weight parameter γ from Equation (4.55), where a clear minimum is recognizable and for high value of γ the error rate converges to the values known without application of the IDM.

7.1.7 Levenshtein-Moore Distance

Some experiments were carried out based on the algorithm for two dimensional Levenshtein distance proposed by MOORE [72]. One considerable drawback is the extremely high computational

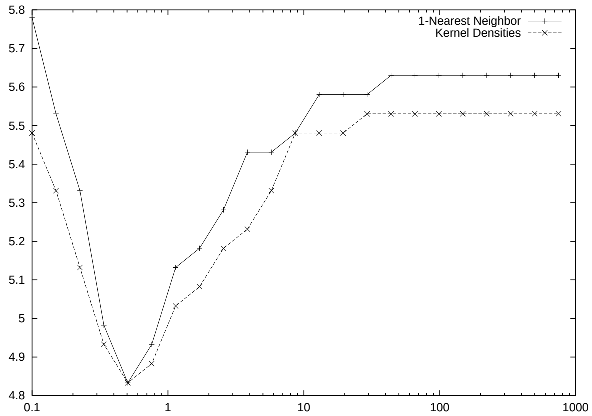

Figure 7.9: Error rate USPS with respect to region factor in gradient based IDM

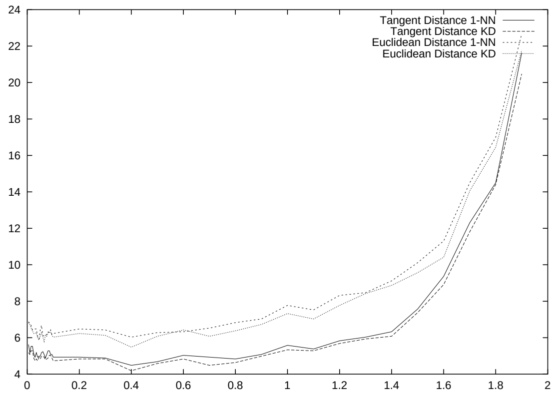

Figure 7.10: Error rate vs. binarization threshold on USPS database

complexity of the algorithm. This made it quite difficult to determine strengths or weaknesses of the algorithm, because only very few experiments could be carried out. Since the original algorithm was developed for images where the cost for substitution is independent of the pixel values, experiments were conducted with binarized images. For a basis of comparison, results with the classifiers discussed so far were produced on the USPS corpus. For that, it is necessary to fix a binarization threshold, which distinguishes between 'black' and 'white' pixels. Figure 7.10 shows the error rates of the basic classifiers with respect to the binarization threshold. Interestingly, the error rate does not grow much when the threshold is moved to 0. Even if only the grayvalues 'white' and 'not white' are distinguished, the error rate is surprisingly low. Best results are obtained for a threshold of 0.4 which is 20% of the maximum value. Note that for the experiments here the images of the USPS database were used with 256 graylevels in the range $[0.0; 2.0]$.

The results obtained with the Levenshtein-Moore distance are contained in Table 7.10. The weights for substitution, insertion and deletion of pixels are all set to the same value here, as proposed in [72]. The error rates presented here for tangent distance are somewhat higher than for the basic classifier results (about 0.1% absolute on the average), because for speedup only the nearest 20 images with respect to the Euclidean distance were used for the computationally more expensive distance computation of the Levenshtein-Moore distance and the same was done for the comparison results with tangent distance (prefiltering, compare page 54). The gain obtained over Euclidean

Table 7.10: Results for binarization and Levenshtein-Moore-Distance on USPS

Binarization	Euclidean dist.		Tangent dist.		Levenshtein dist.		Levenshtein + tangent dist.	
Threshold	1-NN	KD	1-NN	KD	1-NN	KD	1-NN	KD
0.4	6.0	5.5	4.5	4.2	6.0	5.4	5.0	4.8
0.7	6.5	6.1	5.0	4.7	6.3	5.9	5.0	4.7
1.0	7.8	7.3	5.6	5.4	7.2	6.9	5.0	4.7
1.3	8.5	8.4	6.4	6.2	7.6	7.5	5.8	5.6

distance by the application of the Levenshtein-Moore distance is only minimal for low binarization thresholds and grows for higher thresholds. Surprisingly, if it used together with tangent distance, results are impaired for a low threshold but improved for higher thresholds. If one can draw a conclusion from these few experiments, it is that the proposed Levenshtein-Moore distance improves results, but at a high computational cost, rendering it difficult to perform a large number of experiments. What remains still open is the question of suitability for grayscaled images. The extension to this case seems straightforward, but an additional parameter relating the weight of a substitution to the weight of an insertion or deletion is needed.

7.1.8 Holographic Classification

In a few experiments the performance of the holographic classifier described in Section 2.5 was investigated. On artificial training data with pattern length up to 200 consisting of binary vectors drawn independently from a uniform distribution the algorithm performed reasonably well within the operating range of parameters in correctly retrieving the pattern number for the training data. For further investigation, tests on the USPS corpus were performed. Different transfer functions to the complex domain were tested with the goal to achieve the best symmetry in the resulting feature distribution, among them polynomial functions and histogram-based functions, but none of the tested approaches seemed to provide outstanding performance, the obtained symmetry seemed to be the important criterion, which was low in all of the tested methods after adjustment of parameters. The experience was that it seems quite difficult to handle the algorithm, because of the large number of open parameters for which no easy rules exist. Because of the long training time for the hologram the basic validation results and parameters were determined using a subset of the USPS training corpus consisting of the first 1000 patterns of the database. Using the gained parameters the training was repeated on the complete corpus. To enlarge the pattern length (which should have a certain minimum length, ideally greater than 12 times the number of given samples according to Khan) the images as well as the independent components of the outer products of the feature vector with themselves were taken as feature vectors yielding $256 + \frac{256 \cdot 257}{2} = 33152$ features, not reaching the factor twelve (this may be nevertheless justified, since it is not necessary to distinguish all patterns but only the corresponding classes). This large number of features of course has a strong influence on the training time. Table 7.11 shows the obtained results for the holographic classifier on the USPS database. If only the subset of the training set was used for training, the best result is an improvement over the nearest neighbor classifier but still far from the result for tangent distance based classification, which yields an error rate of 7.1% in that case. For the complete database the classifier had an remaining error rate of 1.5% on the training data and did not reach the result for nearest neighbor classification. Since the obtained results are

Table 7.11: Results for holographic classifier on USPS

Number of training samples	Method description	Error rate [%]
1000	binary class coding	13.4
	unary class coding	9.4
	unary class coding, FT features	11.7
	Nearest Neighbor (Reference)	10.2
7291	unary class coding	6.0
	Nearest Neighbor (Reference)	5.6

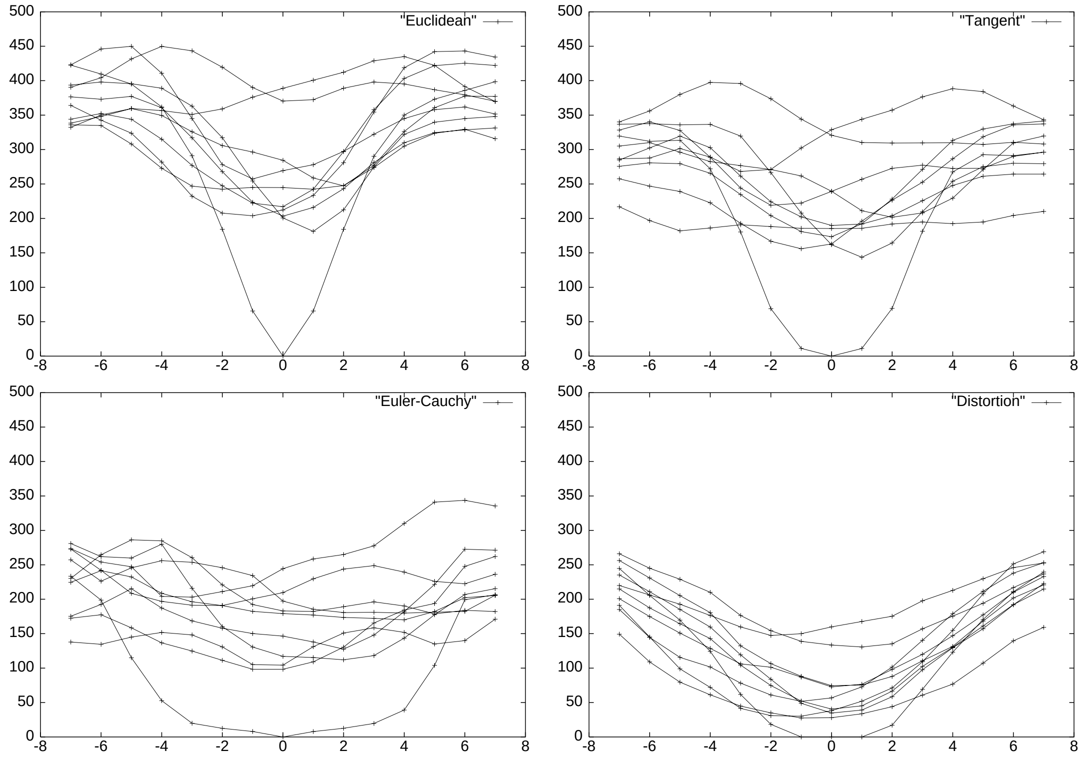

Figure 7.11: Typical distances for different distance measures (USPS). Distance vs. Image shift [pixels], Euclidean distance (top left), tangent distance (top right), Euler-Cauchy distance (bottom left) and IDM distance (bottom right)

not completely out of the range of acceptable error rates, it might be interesting to examine the possible strengths and weaknesses of this classifier more deeply.

Due to the limited time it was not possible to perform experiments with holographic classification on the IRMA corpus, although this corpus might be better suited for the algorithm because of larger feature vector dimension. Furthermore it was not possible to pursue further the usage of Fourier transformed, complex features.

7.1.9 Behavior of Different Distance Measures

Figure 7.11 shows the dependency of different distance measures on image transformations, here a horizontal shift of $+/-$ seven pixels. One image from each class was chosen randomly as reference

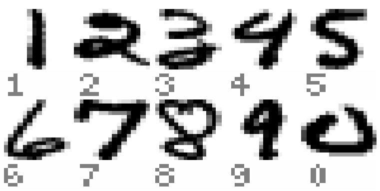

Figure 7.12: Images used for the distance graphs

(the images are shown in Figure 7.12) and one image was shifted to 15 positions (here the digit of class 'five' was used). Then the distances to all the images were computed, thus yielding zero distance for a shift of zero pixels for the identical image. It can be seen that while Euclidean distance would lead to the correct decision in this artificial setting for $+/-$ two pixel shifts, tangent distance degrades more gracefully and tolerates $+/-$ three pixel shifts. The Euler-Cauchy approximation of the manifold is able to keep the distance very small over a wide range of shifts for the identical images, but at the same time the increased matching power leads to smaller distances to the remaining images, too. Since in practice nearly identical images are seldomly encountered, this might explain, why the Euler-Cauchy distance did not lead to better results on the USPS database. Finally the distortion distance measure keeps the distance to the identical image at zero for a shift of one pixel, as is expected for a region radius of one pixel, but at the same time allows so much distortion that almost all images can be mapped well onto the reference image, therefore it is not equipped with sufficient discriminative power, which is reflected in the low recognition rate for the IDM with region size one.

7.2 Radiograph Categorization

This section is concerned with the results obtained during the experiments carried out on the IRMA radiograph database described in Section 6.1.3. On this database a large number of experiments was performed by THEINER [95], which already led to very good results. The experiments included methods of invariant image object recognition like tangent distance[4] and image distortion model, which improved classification significantly. As a summary of the results obtained in the experiments for this work it may be stated that none of the additionally tested methods could improve the previous results obtained by THEINER significantly. Only a thorough parameter optimization and an improved tangent calculation for the image borders led to an improved performance from 8.6% error rate [95] to 8.2%. In the following sections first the previously used methods will be described shortly (following [23, 51]), followed by a description of the experiments and their results.

Figure 7.13 shows examples for the basic 1-NN classifier using Euclidean distance on the IRMA database. With this basic setting an error rate of 18.1% is obtained, which can be significantly reduced using various methods for invariant classification. For an overview of obtained results see Tables 6.2 and 7.12. A leaving one out approach was adopted for all experiments because of the small database, using $N-1$ training samples in each step and one test sample, while using the arithmetic average over the N obtained results as total result.

[4] in cooperation with of the author of this work

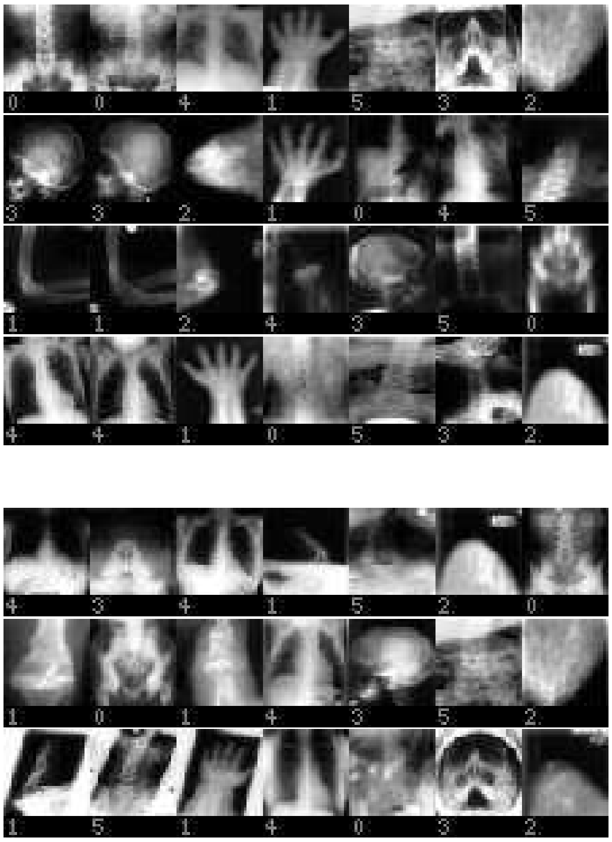

Figure 7.13: Examples for Nearest Neighbor recognition on the IRMA database. First image: test pattern, following: best references from each class in descending order. Top 4 rows: correct classification, lower 3 rows: incorrect classification. (class numbers: 0 = 'abdomen', 1 = 'limbs', 2 = 'breast', 3 = 'skull', 4 = 'chest', 5 = 'spine')

7.2.1 Previous Results

For the experiments, the radiographs were scaled down to a standard size of 32×32 pixels. This can be done without a significant change in classification error rate, but leads to a considerable system speedup. Computing a simple 1-NN on the radiographs with a size of 320×320 pixels yielded a classification error of 18.0%, requiring about 30 CPU seconds on a 500MHz Digital ALPHA CPU to classify a single image. Downscaling the images to the proposed size of 32×32 pixels, an error rate of 18.1% was obtained, requiring about 0.4 CPU seconds.

Having chosen the image size, single-sided tangent distance was used for radiograph classification. This reduced the KD error rate from 16.4% to 14.8%. Then experiments were started with the image distortion model, using the cost function $C_{iji'j'} = 0$. With an error rate of 14.7% the result of the distortion model is slightly better than that obtained with tangent distance. In another experiment it was tested whether the gains of both approaches were additive. Indeed, combining both distance measures (by computing IDM distance of the previously tangent-registered images) reduced the error rate from 14.8% to 12.5%. Figure 7.14 shows the achieved error rates with respect to the size of a square neighborhood R of dimension $(2r+1) \times (2r+1)$. The best result

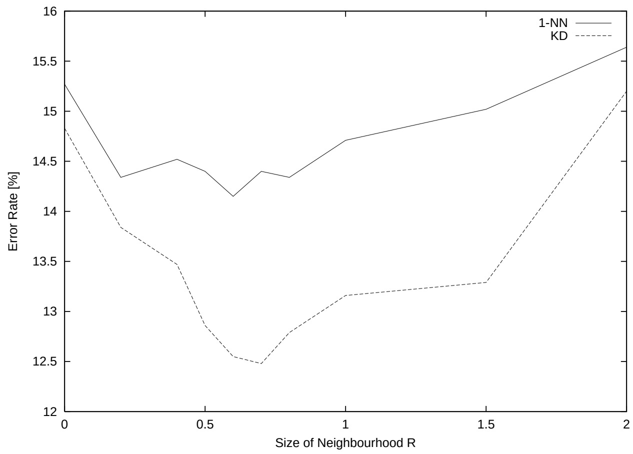

Figure 7.14: Error rates for distorted tangent distance with respect to size of neighborhood Region, without local thresholding

of 12.5% was obtained using $r = 0.7$ (using linear interpolation between pixels).[5] Observing that the maximum contribution of a pixel to any of the used distance measures is $255 \cdot 255 = 65025$ times greater for the maximum difference than for the minimum difference, as the radiographs are 256-grayscale images. Thus, a single pixel may have a significant contribution to the total distance, so that a few distorted pixels (as caused by noise or changing scribor position) can lead to a misclassification. If the maximum contribution of a single squared pixel difference was restricted in the experiments ('local thresholding' see page 110), to a maximum value this effect can be compensated and the error rate could thus be further reduced to 10.3%. Analyzing the remaining errors it was found that many misclassifications could be easily avoided by taking into consideration the original image aspect ratios (by downscaling the images to a standard size this information was lost). To compensate for this an aspect ratio penalty term was introduced, based on the squared difference in aspect ratio between the given image and the reference image. This penalty term reduced the classification error from 10.3% to 8.9%. Then choosing $C_{iji'j'}$ to be a weighted Euclidean distance between pixel positions the error rate was reduced from 8.9% to 8.6% (with the class-specific error rates ranging from 27.3% for 'abdomen' to 3.4% for 'chest'). This result could be improved only marginally by using the Euler-Cauchy distance measure not justifying the additional amount of computation involved.

The use of *cooccurrence matrices* [36] is often considered to be helpful for content based medical image retrieval. However, the experiments on radiograph classification did not support this thesis. In two experiments, global cooccurrence matrices were used for feature analysis within a synergetic classifier [95] and within a kernel density based classifier. In both cases, it was not possible to obtain classification error rates below 29%. Apparently, cooccurrence matrices do not provide discriminative features for radiograph classification. This does not mean that they may not be useful for the following IRMA processing steps, e.g. to detect tumors within a (previously categorized) radiograph. In this case, cooccurrence matrices would be computed from small parts of the image, not from the complete image.

In a domain like medical imaging, the thickness tangent loses its a priori nature and can be replaced by a brightness tangent (here defined as a constant function over (i, j), compare page 49), modeling different doses in x-ray imaging. This is reflected in the corresponding recognition rates

[5]See also page 112.

Table 7.12: Comparison of results for IRMA database

Distance Measure	Error Rate [%]		
	1-NN	KD	KD, threshold
Euclidean	18.0	16.4	14.2
TD	15.3	14.8	12.9
IDM	16.5	14.7	13.2
TD, IDM	14.7	13.2	11.7
TD (brightness), IDM	13.5	12.9	10.3

and shows in comparison to the USPS results that the selection of tangents is task dependent. Table 7.12 shows the results of different distance measures for the IRMA database, 'brightness' indicating that the tangent for line-thickness was replaced by the brightness tangent. The results show, that in this domain thresholding is appropriate and the improvements of TD and IDM are nearly additive. Combination of the two approaches was achieved here by replacing Euclidean distance with distortion distance (4.50) in the last step of distance computation, when the optimum coefficients for the tangents are already known, which can be interpreted as a previous registration of images.

7.2.2 Extended Experiments

In the following the experiments built upon the previously presented results are described, none of which did result in a significant improvement of classification performance.

Since training data multiplication led to considerable improvements on the OCR databases, it was a natural approach to try the same for the IRMA data as well. But unfortunately multiplication of the training data using image shifts (one or two pixels in all possible directions of the 8- respectively the 24-neighborhood) did not improve classification results here. The reason for this negative result may be that image shifts are not the main source of variation in the radiographs contained in the database, while they are a very important factor when digits are to be recognized.

On a different task, consisting of images of chairs, considerable improvements could be obtained using the gradient of the images as additional features [18]. This raised the hope that this sort of additional information per image grid point might improve classification results for the IRMA data, too. But in none of the various experiments using gradient images as additional features, improvements could be achieved. The gradient was computed using the Sobel operator or equivalently the tangents for image shifts, since these are computed using a modified Sobel operator. Even when applied to a basic result without use of tangent distance or IDM the performance was not improved using the gradient information as feature.

A number of experiments were also performed concerning the distribution of the grayvalues in the images. One idea is to enforce the full usage of the available grayvalue interval, possibly accompanied by a spreading which changes the grayvalues such that a certain percentage of the pixel values lies outside the allowed range and then is cut to fit the interval. This method is called *histogram stretching* [65, p. 196]. Let $g_{\max}$ and $g_{\min}$ denote the maximum and minimum grayvalue in the allowed range and let $x_{\max}$ and $x_{\min}$ denote the maximum and minimum grayvalue in the image x, then each pixel of the image is transformed to the new image x' by

$$x'_{ij} = \min\left\{g_{\max},\ \max\left\{g_{\min},\ \frac{(x_{ij}-x_{\min})(g_{\max}-g_{\min})\,\epsilon}{x_{\max}-x_{\min}} + g_{\min} - \frac{1}{2}(\epsilon-1)(g_{\max}-g_{\min})\right\}\right\} \quad (7.2)$$

for some factor ϵ usually chosen around 1.05. The effect of a different spreading function with a low cutoff at both sides was tested, but is neglectable. The factor was tried in the range between 1.0 and 1.1 but did not influence classification results significantly, neither positive nor negatively.

In this context it should be mentioned that the different grayvalue histogram distribution is partly accounted for by the usage of the constant brightness tangent, which can be derived from an additive illumination model (derivation see page 49), and performed better than the tangent for the multiplicative illumination model or a combination of both.

The usage of local thresholding, which induced significant improvements in radiograph categorization is described by a piecewise defined function for the local distance:

$$d(x,\mu) = \sum\nolimits_i d_{\text{local}}(x_i,\mu_i) \quad (7.3)$$

where for squared Euclidean distance the local distance is the squared difference

$$d_{\text{local, Euclidean}}(a,b) = ||a-b||^2 \quad (7.4)$$

which is replaced by a piecewise defined function

$$d_{\text{local, thresholding}}(a,b) = \begin{cases} ||a-b||^2, & \text{for } ||a-b||^2 < t \\ t & \text{, for } ||a-b||^2 \geq t \end{cases} \quad (7.5)$$

for some predefined threshold t (respectively determined by cross-validation). One can now try to find out if there maybe are better suited local distance functions, which for example are smoother around the value t (the above function is not differentiable at $||a-b||^2 = t$). Several other local distance functions, including polynomial and exponential functions were tested in various experiments, but no improvements were obtained. This approach is quite common, for example Vasconcelos et al. write in [100] to the subject of thresholding in image classification that “It is well known, that a few (maybe even one) outliers of high leverage are sufficient to throw mean squared error estimators completely off-track.” and propose – similar to the approach taken here – to substitute the square function by a functional, which “weighs large errors less heavily”, then propose to use a thresholding function for that functional.

The aspect ratio of the radiographies had been used as additional feature, which increased classification performance considerably. The next step in this direction was then performed by also taking into account the variances of the aspect ratios for the different classes. It showed that the variances varied by almost two order of magnitudes, being lowest for class ‘breast’ and highest for class ‘limbs’. The first result was promising, since the 1-NN error rate dropped below 9%, which was the lowest rate obtained, but as in most other experiments the best error rate of 8.2% for the kernel density classifier could not be improved.

A method was implemented to automatically detect multiply labeled images like the one shown in Figure 7.15 by signalling images with (near) zero distance in leaving one out classification. But after considering that the number of such images is fairly small they were left in the database in order to retain comparability of different methods. It is not clear in which way the multiply labeled images affect the error rate, since there are identical images in the same class (leading to correct classification) as well as in different classes (leading to misclassification).

Figure 7.15: Multiply labeled image, part of class ‘skull’ as well as ‘spine’

7.2.3 Tangent Distance

Since tangent distance was already used fully implemented in the previously described experiments, few additional experiments concerning tangent distance itself were carried out, which are described in the following. In this context it might be interesting to mention that the downscaling of images in combination with tangent distance can be compared to the multiresolution approach described in [100]. The connection is that the restricted range of a few pixels which is inherent in tangent distance modeling small transformations can be enlarged by using downscaled versions of the images, possibly – though not here – over a set of scaling factors.

Some experiments were performed concerning the treatment of the tangents at the image border. On the OCR database there exists a canonical extension of the picture, which is to imagine the image continued outside the actual image with the background grayvalue. This allows to calculate a meaningful gradient even on the image border and therefore meaningful tangents. Such a canonical extension does not exist for the radiograph database. Therefore first experiments [95] were done using a predefined border value or an extension with the same grayvalue as was encountered on the image border. It was observed, though, that best performance was obtained by ignoring the value of the tangents for the image border pixels. A further improvement was made during the experiments for this work by assigning a weighting factor to the tangent elements at the image border. It was found that with a weight value of about one third relative to the remaining pixels best results were obtained and tangent performance could be improved by about 0.2% absolute error rate.

There are at least four methods to be considered when thinking about treatment of the image border [65, p. 203]:

(1) disregard border pixels (leads to shrinking respectively information loss)

(2) extrapolate (may lead to extrapolation errors)

(3) disregard parts of the convolution mask outside (may lead to discontinuities)

(4) use wrap-around (only if periodicity assumption is valid)

Method number four must be ruled out in this context while experiments favored method number one over two and three. Best results were obtained by combining methods number one and three in the way described above.

7.2.4 Image Distortion Model

This section deals with the IDM in the context of the IRMA database and some experiments with extensions to it. First consider again the interpolation argument concerning the IDM approach for non integer region sizes. When looking at Figure 7.14 it can be observedthat integer region sizes correspond to local maxima of the error rate, as was pointed out before in connection with the experiments on the USPS database (compare Figure 7.9). This can be attributed to the inherent smoothing effect of the fractional IDM, which is calculated using linear interpolation. The difference in this case to the previously considered one is, that for integer radius 1 the error rate is significantly lower than for radius size 0, which is not true for USPS. This gain must therefore be granted completely to the positive effects of the IDM, while for non integer radii the gain is probably due to the effects of both IDM and smoothing. But the effects of smoothing in this case are significantly lower than they are on USPS, which can be seen from the larger value for the region radius for minimum error rate of 0.7 pixel, where a significant importance is placed on the outside pixel (in comparison to 0.3 for USPS) and from the comparison results obtained by using a smoothing prefilter. In the smoothing experiments carried out the maximum improvement obtained by smoothing was never more than one third of the effect of IDM plus interpolation in contrast to the USPS results, where the total improvement of the IDM could also be obtained by just smoothing.

A number of experiments were performed with the aim to regularize the IDM vector field. By that the array of displacement vectors is meant, which are the result of subtracting the position of the observation image pixel from the one found to minimize the local IDM distance. That is the IDM vector field $v_{x,\mu}$ in a distance computation between x and μ can be written as (compare Equation (4.50))

$$v_{x,\mu}(i,j) = \operatorname*{argmin}_{(i',j') \in R_{ij}} \{\|x_{ij} - \mu_{i'j'}\| + C_{iji'j'}\} \tag{7.6}$$

In the generalized model of Equation (4.59) the vector field is equal to the minimizing displacement function f:

$$v_{x,\mu} = \operatorname*{argmin}_{f \in \mathcal{F}} \{C(f) + \sum_{i,j} \|x_{ij} - \mu_{f(i,j)}\|\} \tag{7.7}$$

Now it seems intuitive to restrict the set of possible displacement fields such that only "meaningful" transformations are allowed. One approach that does this is tangent distance, which only allows affine transformations. In contrast to this the basic IDM allows almost arbitrary transformations, restricted only by the region size (compare Figure 4.14). In order to favor regular displacement fields, different cost terms for irregularity were implemented and tested, but none of them improved the best result obtained before. The proposed methods are based on optical flow [45] and pixel fertility.

The first method tried was inspired by optical flow. Optical flow relies on the assumption that "the apparent velocity of the brightness pattern varies smoothly almost everywhere in the image." This can be applied to the IDM, where two images are compared, which are supposed to be deformations of each other. It was implemented by adding a distance term to the implied distance, representing the deviation from a smooth optical flow. Ideally the overall distance should then be calculated as a global minimum over all possible transformations, but in this work only experiments with a multi-step algorithm were performed, first determining the vector field, then adding the cost term. Since the optical flow constraint is equivalent to the minimization of the second partial derivatives of the image velocities [45], the sum of these derivatives across the vector field was used as cost term. For the computation of the discrete derivative the Laplace operator was used, that is the

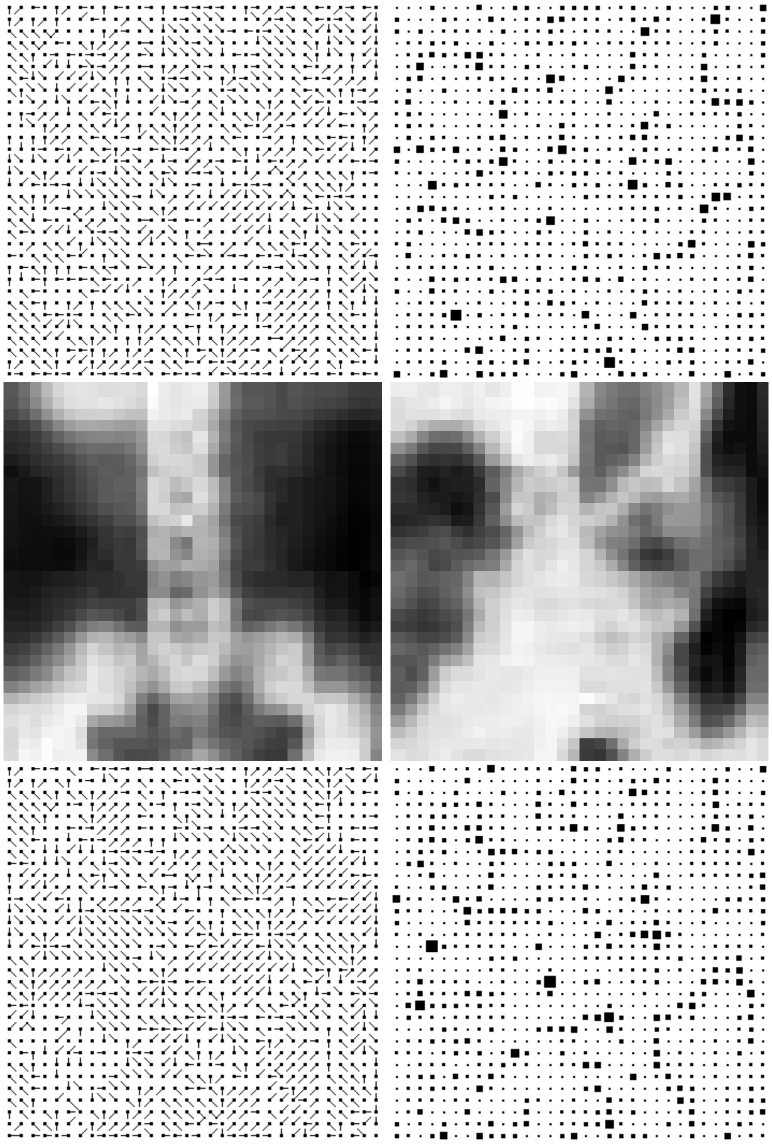

Figure 7.16: Visualization of the IDM displacement vector field and the pixel fertility (represented by box size) for two images of class chest. Upper row: with prior application of tangent registration, lower row: without usage of tangent registration. (Left image used as observation, right image as reference. Each pixel in the observation must be "explained" by the reference in this case.)

displacement fields were convolved with the mask (-1, 2, -1) in both directions and for each of the two vector components. Then the sum of absolute values was used as cost term. Another experiment was carried out using the discrete measure of regularity for optical flow from [42]. The cost term was added to the precomputed distance measure after multiplication with a weighting factor. In none of the experiments any improvement for the best result was achieved.

Another way to achieve a preference of regular IDM vector fields is to determine the pixel fertility for each pixel in the reference and to define a cost term for a deviation from 1. The pixel fertility is defined as the number of times the grayvalue was used to explain a grayvalue in the observation. In the experiments a Euclidean cost term was used with varying weight, but its use did not improve

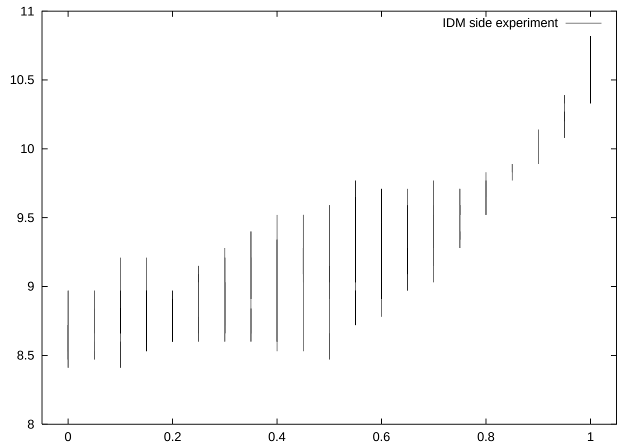

Figure 7.17: KD error rate (bars indicate ranges for different variance factors) vs. weighting of IDM side. 0 refers to explanation of the observation, 1 to explanation of the reference, values in between to linear mixture of distances.

classification performance for the KD classifier. Only small improvements (of about 0.4% absolute) were observed for the 1-NN classifier.

Figure 7.16 shows two visualizations of the IDM vector field and the pixel fertility distribution for region radius 1 pixel. The flow field is very inhomogeneous, which is not surprising for these two randomly selected images of the same class, which differ widely. In the upper row the flow field is shown for previous use of tangent registration, while the lower row shows the same images without prior registration. Since the images are very different in their pixel representations, the fields do not differ greatly, but some differences can be noticed. The average absolute difference from 1 of the pixel fertility is 0.87 for the upper row and 0.90 for the lower row, which might be an indicator, that with application of tangents, the IDM gets more homogeneous.

Another approach is to use the gradient as additional hint on which pixel from the region to use as matching pixel. When the minimization over the region corresponding to a pixel in the image is performed, the gradient difference can be taken into account. This is equivalent to using the gradient as additional feature, which does not improve classification, as mentioned above. But one can use the gradient information only as hint on the best matching pixel and then take the squared grayvalue difference as distance contribution. This is only an additional restriction on the IDM vector field distribution and does not affect the features used in the distance computation. Unfortunately this approach did not improve classification results on the IRMA database, either, at least for the best KD classification result. For the 1-NN classifier, small improvements of about 0.2% absolute could be observed

Figure 7.17 shows the results of experiments for the usage of the image distortion model on the side of the references and mixtures of both differences. The results show, that on this data it is best to keep all pixels in the observation and explain each one with a value from the reference. Linear interpolation of the distances did not improve the one sided approach. One could think about other ways of combining the two approaches, especially in combination with restriction of pixel fertility or with restricting the fraction of pixels to be neglected in each image, involving a minimization including all cost terms simultaneously. These approaches would require a complete restructuring of the used algorithms, but might be interesting to try, although experience with the additions lets a significant improvement seem rather unlikely on this database.

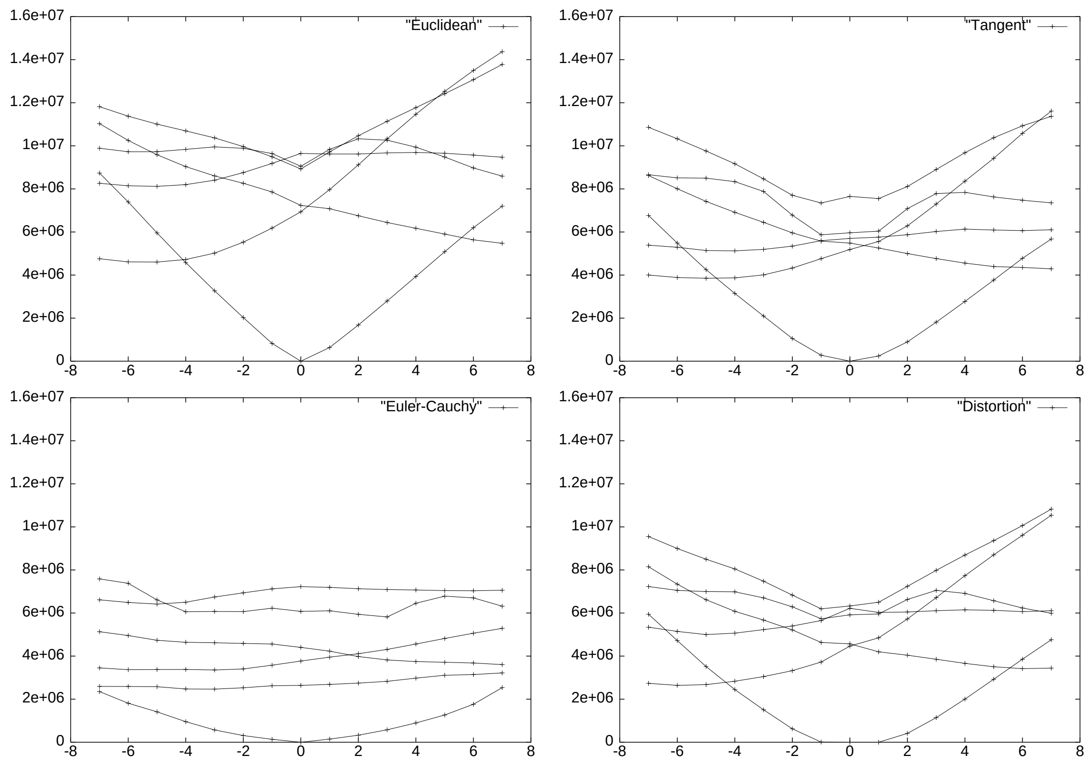

Figure 7.18: Typical distances for different distance measures (IRMA). Distance vs. Image shift [pixels], Euclidean distance (top left), tangent distance (top right), Euler-Cauchy distance (bottom left) and IDM distance (bottom right)

Another possible explanation for the better performance of the IDM when keeping all observation pixels in this context is, that it was used in combination with tangent distance on the side of the observation, that is tangent deformation was used on the side of the observation and IDM deformation was used on the side of the reference.

The gradient based image distortion model (Equation (4.55)) did not improve classification results on the IRMA database, neither with nor without tangent distance. This result maps well with the lack of a priori suitability of the thickness tangent in this domain.

7.2.5 Behavior of Different Distance Measures

Figure 7.18 shows the dependency of different distance measures on image transformations, here a horizontal shift of $+/-$ seven pixels for images from the IRMA database. One image from each of the six classes was randomly chosen and one of these was chosen as reference image (here from the class 'abdomen'). The images are shown in Figure 7.19. Then the distances to all the images were computed, thus yielding zero distance for a shift of zero pixels for the identical image. If compared to the similar graphs for the USPS database on page 105, some differences can be observed. The most prominent difference is, that here tangent distance and distortion distance with region radius one behave very similarly. This corresponds well with the experience, that both invariant distance measures lead to improvements in classification of about the same amount. The Euler-Cauchy approximation of the manifold is able to keep the distance very small over a wide range of shifts for the identical images, but at the same time the increased matching power leads to

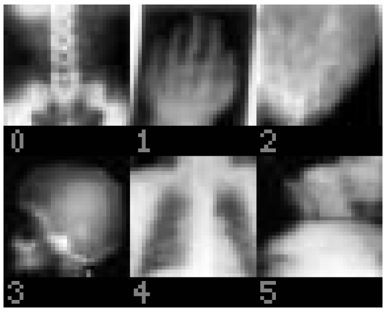

Figure 7.19: Images used for the distance graphs

smaller distances to the remaining images, too, although not as much as for the USPS images. On this data the Euler-Cauchy distance seems to provide better discriminating power than on USPS, and this again is reflected in the fact that it was able to improve classification marginally (by about 0.2% absolute improvement in error rate), but not worth the considerable increase in time consumption. From the diagrams it can not be seen that Euclidean distance performs significantly worse than tangent distance in this case.

7.2.6 Generalization Test

After adjusting all parameters on the database consisting of 1617 images a generalization test was performed on a set of 332 previously unseen radiographs[6]. Using the values determined by the leaving-one-out strategy, the algorithm misclassified 30 out of the 332 new radiographs (corresponding to an error rate of 9.0%) with the training set now consisting of 1617 images. This means that an adequate generalization was achieved, since the error rate on the new set does not deviate much from the leaving-one-out training set error rate of 8.2%

7.3 Task Dependency

During the experiments performed on the two different classification tasks optical character recognition and radiograph categorization it became clear (which is by no means a spectacularly new finding) that the performance of the different distance measures and extensions depends very much on the specific task [51]. The property that there is no model that is globally optimal, but the performance depends on the adequateness for the specific data is sometimes referred to as "no free lunch"-property [69]. Towards this issue in relation to the task of OCR in [63] it is stated that "The performance of the local subspace method is dependent on the nature and density of the data in the Bayesian class border area." In the following some salient task dependencies concerning the two tasks studied here are given.

One of the main differences between the two tasks is that the image distortion model leads to considerable improvements in classification performance on the IRMA database while this is not true for the USPS database. One demonstrative reason for this difference may be that it is easy to "erase" a line in an image representing a handwritten digit completely using the IDM when this line is only one or two pixels wide. If the line distinguishes two numerals from each other, as

[6]Thanks go to Thomas Theiner for implementing the used visualization tool.

for example with the digits '3' and '8' or with '4' and '9', this is enough to lead the classifier off track. This is also reflected in the distance graphs of Figures 7.11 and 7.18, which illustrate the differences in the distance measures.

Another prominent difference in the classification results is, that data multiplication led to a significant improvement in OCR, while no gain could be observed for the radiograph database. One possible explanation for this is, that image shifts may not be the main source of variation in the IRMA corpus.

Among the other differences one should mention the higher importance of smoothing on the OCR task and the better performance of the Euler-Cauchy approximation for radiograph categorization. Furthermore, the lack of a priori explanation for the line thickness transformation on the radiograph corpus matches well with the experiments and while the brightness tangent did not improve results for OCR it did for the radiographs. Another difference, reflected in results for the tangent estimation at the image border, is that for the numeral images there exists a canonical extension of the images with the background graylevel, which is not the case for radiographies.

Finally it should be observed that the usage of tangent distance led to considerable improvements in both tasks, which is probably due to the fact that the considered transformations (affine transformations and line thickness respectively brightness) are an important source of variation in the image data used.

Chapter 8

Conclusion and Perspective

This planet has - or rather had - a problem, which was this: most of the people on it were unhappy for pretty much of the time. Many solutions were suggested for this problem, but most of these were largely concerned with the movements of small green pieces of paper, which is odd because on the whole it wasn't the small green pieces of paper that were unhappy.

[4]

Conclusion

In this work, different methods to achieve *invariance in image object recognition* have been presented, theoretically investigated, and experimentally evaluated. A strong emphasis was placed on the concept of *tangent distance* and related invariant distance measures like the *image distortion model*.

The theoretical results obtained allow insight into the statistical properties important for image object recognition. As a by-product, a novel model for the *description of transformation-manifolds* using linear difference equations was obtained. More importantly, a new *probabilistic interpretation of tangent distance* was presented and it was shown that the tangent distance model can be derived from a statistical model of intra-class variability. Within this model, different possible settings were examined and the corresponding distance measures (as well as a combination of these) were inferred. It was also shown how domain knowledge about variability can be used to allow a more *reliable parameter estimation* in the context of statistical modeling. Furthermore, a novel approach for using *structured covariance matrices* for image object recognition within a statistical classifier based on the concept of pixel neighborhoods was motivated and described. Properties of the resulting distance measure and its relation to tangent distance were investigated. These distance measures may be helpful in the design of classification algorithms if this type of variation is present in the data.

Tangent distance and related approaches are apt to model variability in image data successfully. This was proven by implementing tangent distance and the described image distortion model for use within a statistical pattern recognition system. Tangent distance effectively compensates variation resulting from small global transformations, for example affine, projective, line-thickness or brightness transformations, while the image distortion model can compensate small local transformations of the image. The effect of virtual data creation for the training and the testing phase was observed to be very important for the performance of the implemented classifiers. Using a *kernel density based Bayesian classifier*, excellent results were obtained on the USPS handwritten digits recognition task. The *result of 2.2% error rate* is the best known result so far. For the NIST digit recognition database a state of the art classification performance of 1.0% error rate could be obtained using the implemented classifier. On the IRMA radiograph database the obtained error rate of 8.2% error rate is the best known result, yet only few results for competing methods exist so far. The results obtained with invariant distance measures are better than those obtained using invariant features, at least on the regarded data sets [77]. This supports the thesis that it is better to incorporate things like feature extraction, determination of transformation parameters and classification into a single classification step, instead of regarding them separately.

From the experimental results it can be deduced that

- incorporating *domain knowledge* about invariant transformations into a classifier significantly improves its performance,
- tangent distance provides an effective means to model pattern variance in digit and radiograph recognition,
- the performance of different invariant distance measures and of combinations of these depends on the given task,
- the image distortion model can improve classification considerably for radiograph recognition, but not for optical character recognition,
- it is possible to successfully use *estimated derivatives* of variation for the modeling of pattern distributions,
- the obtained theoretical results are supported by the experimental results,
- the use of *virtual training and test data* is a valuable tool for improving classification performance,
- using Levenshtein-Moore distance improves classification results slightly compared to Euclidean distance, but at the cost of great computational complexity,

An important issue is the *task dependency* of the different approaches. It is well known that there is no model that is globally optimal, but that the performance depends on the adequateness for the specific data. In the experiments it could be observed that the performance of the different distance measures and extensions heavily depends on the particular task. The image distortion model performs very well on the considered radiograph data, but does not lead to improvements for handwritten digit recognition. On the other hand, data multiplication leads to significantly increased performance for character recognition, but does not enhance the classification of radiographs. Yet, tangent distance performs very well on both tasks considered, which is probably due to the appropriate modeling of small affine transformations, which are a source of variations in the three considered databases. A difference observed was that the line thickness transformation is suitable for optical characters, while it looses its a priori nature on x-ray images and is better replaced by a brightness transformation. In the comparison of different distance measures a general observation is that concerning the matching power there exists a tradeoff between improved matching of correct samples and restricted matching of incorrect samples. Increasing the allowed variation always leads to better alignment of patterns, both towards correct and towards incorrect reference images, so the crucial point is to find the 'right' model of variation.

Perspective

At this point of the present work, some questions remain open and many ideas need still to be investigated. One major point is that basically all considerations presented here are based on the maximum likelihood approach, i.e. each class is handled separately in the models. Future work should include the investigation of discriminative training in this context, taking into account the information of competing classes and aiming at optimizing class separability. One such approach that may be combined with local tangent information was presented in [39], where the authors apply a local linear discriminant analysis to obtain a metric that locally takes into account the concurrent classes. Since there are many connections to principal component analysis in this work, discriminative training might lead from PCA to LDA in some applications, possibly improving discrimination between classes. On the other hand, other researchers give arguments in favor of the relative density approach compared to the discriminative approach [43] or state "We are currently exploring [...] discriminative versus non-discriminative learning in a variety of different contexts. Our preliminary experience is that we do not see any improvement, but the jury is still out."[37]

A different application field of the methods described here can be found in the area of automatic *image indexing by object recognition* [17]. This is already a topic of research [35] and methods include combinations of template matching and probability distribution via eigenspace decomposition, with possible speedup by using the fast Fourier transform for the convolutions corresponding to eigenspace composition leading to the local likelihood [71]. In this context, it is an important aspect that segmentation should not be a process separated from recognition in object detection, but that the two should form a whole (which is suggested by experiences from continuous speech recognition). Also, the separation can only lead to more errors, since errors in early steps of a multi-step algorithm usually cannot be corrected in later steps.

The same argument probably holds for the use of regularity constraints in the image distortion model. In this work, the determination of the IDM vector field and its evaluation using optical flow and pixel fertility were treated as separate step in the calculation of the distance. This approach did not yield any performance increase in the experiments carried out on the radiograph database. Also the calculation of tangent distance and image distortion model were treated as separate steps, but here considerable improvements could be obtained. It is very likely that the *joint minimization* of all the distance terms involved – tangent distance, distortion model, regularity constraints and possibly other models – will lead to better results. It remains an interesting task to develop the necessary algorithms for this minimization process.

Some other open questions are

- if the use of estimated derivatives of variation can be successfully applied to other domains, like speech recognition, where the transformations of the patterns are not known a priori,
- if the calculation of the exact manifold representation is feasible and if so, if it leads to improvements in classification,
- if the combination of the local subspace approach for estimated tangents can be successfully combined with the a priori tangents (compare Equation (5.28)) to yield improvements in classification,
- if higher order covariance structures respectively larger clique sizes can be effectively modeled to obtain a better description of pixel dependencies in images (despite the increasing number of free model parameters),

- if the solutions proposed in [78] can be used to efficiently solve the equation systems involved in the structured covariance matrix model,
- if the extension of the Levenshtein-Moore distance to graylevel images can be used to improve classification results and how it performs on tasks other than optical character recognition,
- how the presented approaches can be used to model image sequences, e.g. using statistical methods in video indexing like temporal Gibbs random fields [29].

References

[1] D. Adams. *The Hitchhiker's Guide to the Galaxy.* Pan Books, London, 1979.

[2] D. Adams. *The Restaurant at the End of the Universe.* Pan Books, London, 1980.

[3] D. Adams. *Life, the Universe and Everything.* Pan Books, London, 1982.

[4] D. Adams. *So Long, and Thanks for all the Fish.* Pan Books, London, 1984.

[5] D. Adams. *Mostly Harmless.* Pan Books, London, 1993.

[6] O. Agazzi and S. Kuo. Pseudo Two-Dimensional Hidden Markov Models for Document Recognition. *AT&T Technical Journal*, pages 60–72, September 1993.

[7] R. Azencott, J.-P. Wang, and L. Younes. Texture Classification Using Windowed Fourier Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):148–153, February 1997.

[8] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. Jackel, Y. L. Cun, U. Muller, E. Sackinger, P. Simard, and V. N. Vapnik. Comparison of Classifier Methods: A Case Study in Handwritten Digit Recognition. In *Proceedings of the International Conference on Pattern Recognition, Jerusalem, Israel.* IEEE Computer Society Press, 1994.

[9] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel Texture Analysis Using Localized Spatial Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, January 1990.

[10] U. Brandes. Markov-Felder als Hilfsmittel der Bildverarbeitung. Diploma thesis, RWTH Aachen, Aachen, Germany, 1994.

[11] M. Braun. *Differential Equations and Their Applications.* Springer, New York, 3rd edition, 1983.

[12] J. Bredno, S. Brandt, J. Dahmen, B. Wein, and T. Lehmann. Kategorisierung von Röntgenbildern mit aktiven Konturmodellen. In *Bildverarbeitung in der Medizin, München*, pages 356–360, March 2000.

[13] C. Bregler and S. M. Omohundro. Nonlinear Image Interpolation using Manifold Learning. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7. MIT Press, pages 973–980, 1995.

[14] H. Burkhardt, A. Fenske, and H. Schulz-Mirbach. Invariants for the Recognition of Planar Contour and Gray-Scale Images. Technical Report TR-402-92-003, Technische Informatik I, TU Hamburg, Germany, 1992.

[15] Q. Chen, M. Defrise, and F. Deconinck. Symmetric Phase-Only Matched Filtering of Fourier-Mellin Transforms for Image Registration and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1156–1168, December 1994.

[16] K. Cheung, D. Yeung, and R. Chin. Recognition of handwritten digits using deformable models. In *Proceedings of the Fifth International Workshop on Frontiers in Handwriting Recognition*, Colchester, England, pages 259–262, September 1996.

[17] J. Dahmen, K. Beulen, M. O. Güld, and H. Ney. A Mixture Density Based Approach to Object Recognition for Image Retrieval. In *Proceedings of the 6th International RIAO Conference on Content-Based Multimedia Information Access, Paris, France*, pages 1632–1647, April 2000.

[18] J. Dahmen, K. Beulen, and H. Ney. Objektklassifikation mit Mischverteilungen. In *20. DAGM Symposium Mustererkennung 1998*, Stuttgart, Germany, pages 167–174, 1998.

[19] J. Dahmen, J. Hektor, R. Perrey, and H. Ney. Automatic Classification of Red Blood Cells using Gaussian Mixture Densities. In *Bildverarbeitung in der Medizin, München*, pages 331–335, March 2000.

[20] J. Dahmen, D. Keysers, M. O. Güld, and H. Ney. Invariant Image Object Recognition using Mixture Densities. In *Proceedings 15th International Conference on Pattern Recognition*, Barcelona, Spain, September 2000. In press.

[21] J. Dahmen, D. Keysers, M. Pitz, and H. Ney. Structured Covariance Matrices for Statistical Image Object Recognition. In *22. DAGM Symposium Mustererkennung 2000*, Springer, Kiel, Germany, September 2000. In press.

[22] J. Dahmen, R. Schlüter, and H. Ney. Discriminative Training of Gaussian Mixture Densities for Image Object Recognition. In W. Förstner, J. Buhmann, A. Faber, and P. Faber, editors, *21. DAGM Symposium Mustererkennung 1999, Bonn*, Informatik aktuell. Springer, pages 205–212, 1999.

[23] J. Dahmen, T. Theiner, D. Keysers, H. Ney, T. Lehmann, and B. Wein. Classification of Radiographs in the 'Image Retrieval in Medical Applications' System (IRMA). In *Proceedings of the 6th International RIAO Conference on Content-Based Multimedia Information Access, Paris, France*, pages 551–566, April 2000.

[24] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algoritm. *Journal of the Royal Statistical Society Series B*, 39:1–38, 1977.

[25] P. A. Devijver and J. V. Kittler. *Pattern Recognition. A Statistical Approach.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

[26] H. Drucker, R. Schapire, and P. Simard. Boosting Performance in Neural Networks. *International J. Pattern Recognition Artificial Intelligence*, 7(4):705–719, 1993.

[27] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[28] H. Esser and H. T. Jongen. *Differentialgleichungen und Numerik für Informatiker und Physiker.* Verlag der Augustinus Buchhandlung, Aachen, 1995.

[29] R. Fablet, P. Bouthemy, and P. Perez. Statistical Motion-Based Video Indexing and Retrieval. In *Proceedings of the 6th International RIAO Conference on Content-Based Multimedia Information Access, Paris, France*, pages 602–619, April 2000.

[30] V. S. Fayn, V. N. Sorokin, and V. S. Vaynshteyn. Continuous-Group Pattern Recognition. *Engineering Cybernetics*, 6:97–106, 1969.

[31] S. R. Fountain and T. N. Tan. Rotation Invariant Texture Features from Gabor Filters. *Lecture Notes in Computer Science*, 1352:57–64, 1997.

[32] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Computer Science and Scientific Computing Academic Press Inc., San Diego, CA, 2nd edition, 1990.

[33] D. Gabor. A New Microscopic Principle. *Nature*, 161:777–778, May 1948.

[34] G. Grosche, V. Ziegler, D. Ziegler, and E. Zeidler. *Teubner Taschenbuch der Mathematik*. Teubner, Leipzig, 1996.

[35] M. O. Güld. Bildzugriff mittels statistischer Objekterkennung. Diploma thesis, Chair of Computer Science VI, RWTH Aachen, Aachen, Germany, July 2000.

[36] R. Haralick, K. Shanmugam, and I. Deinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.

[37] T. Hastie and P. Simard. Metrics and Models for Handwritten Character Recognition. *Statistical Science*, 13(1):54–65, January 1998.

[38] T. Hastie, P. Simard, and E. Säckinger. Learning Prototype Models for Tangent Distance. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Inf. Proc. Systems*, volume 7. MIT Press, pages 999–1006, 1995.

[39] T. Hastie and R. Tibshirani. Discriminative Adaptive Nearest Neighbor Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):607–616, June 1996.

[40] T. Hastie, R. Tibshirani, and A. Buja. Flexible Discriminant and Mixture Models. In J. Kay and D. Titterington, editors, *Proceedings of Neural Networks and Statistics conference*, Oxford University Press, Edinburgh, pages 1–23, 1995.

[41] S. Haykin. *Neural Networks - A Comprehensive Foundation*. MacMillan, New York, 1994.

[42] A. Heyden and K. Åström. Computer Vision. Slides to a PhD course in Computer Vision given at the centre for Mathematical Sciences, Lund University, August 1999. URL: `http://www.maths.lth.se/matematiklth/vision/visit/`.

[43] G. E. Hinton, P. Dayan, and M. Revow. Modeling the Manifolds of Images of Handwritten Digits. *IEEE Trans. on Neural Networks*, 8(1):65–74, January 1997.

[44] G. E. Hinton, M. Revow, and P. Dayan. Recognizing Handwritten Digits Using Mixtures of Linear Models. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Adv. in Neural Inf. Proc. Systems*, volume 7. MIT Press, pages 1015–1022, 1995.

[45] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, August 1981.

[46] M. K. Hu. Visual Pattern Recognition by Moment Invariants. *IEEE Transactions on Information Theory*, 8:179–187, 1962.

[47] D. P. Huttenlocher, R. H. Lilien, and C. F. Olson. View-Based Recognition Using an Eigenspace Approximation to the Hausdorff Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:951–955, September 1999.

[48] D. Keysers. Bildsuche mit holographischer Dynamik. Seminar paper, Hauptseminar medizinische Bildverarbeitung, February 1999.

[49] D. Keysers. Texturanalyse von Farbbildern mit Gaborfiltern. Studienarbeit, RWTH Aachen, Institut für Medizinische Informatik, Aachen, 1999.

[50] D. Keysers, J. Dahmen, and H. Ney. A Probabilistic View on Tangent Distance. In *22. DAGM Symposium Mustererkennung 2000*, Springer, Kiel, Germany, September 2000. In press.

[51] D. Keysers, J. Dahmen, T. Theiner, and H. Ney. Experiments with an Extended Tangent Distance. In *Proceedings 15th International Conference on Pattern Recognition*, Barcelona, Spain, September 2000. In press.

[52] J. I. Khan. *Attention Modulated Associative Computing and Content-Associative Search in Image Archive*. PhD thesis, University of Hawaii, August 1995.

[53] J. I. Khan. Intermediate Annotationless Dynamical Objectindexed Based Query in Large Image Archives with Holographic Representation. *Journal of Visual Communication and Image Representation*, 7(4), 1997.

[54] J. I. Khan and D. Y. Yun. Holographic Image Archive. *Journal of Computerized Medical Imaging and Graphics*, 20:243–257, 1997.

[55] J. I. Khan and D. Y. Yun. A Parallel, Distributed and Associative Approach for Searching Image Patterns with Holographic Dynamics. *Journal of Visual Languages and Computing*, 8:303–331, 1997.

[56] J. I. Khan and D. Y. Yun. Characteristics of Multidimensional Holographic Associative Memory in Retrieval with Dynamically Localizable Attention. *IEEE Transactions on Neural Networks*, 9(3):389–406, May 1998.

[57] J. Kittler, M. Hatef, and R. Duin. Combining Classifiers. In *Proceedings of the International Conference on Pattern Recognition, Vienna, Austria*. IEEE Computer Society Press, pages 897–901, 1996.

[58] J. M. Kleinberg. Two Algorithms for Nearest-Neighbor Search in High Dimensions. In *29th ACM Symposium on Theory of Computing*, 1997.

[59] M. Kohnen, F. Vogelsang, B. Wein, M. Kilbinger, R. Günther, F. Weiler, J. Bredno, and J. Dahmen. Kategorisierung von digitalen Röntgenbildern mit parametrisierbaren Formmodellen. In *Bildverarbeitung in der Medizin, München*, pages 366–370, March 2000.

[60] T. Kohonen. *Associative Memory, a System-Theoretical Approach*. Springer, Berlin, 1st edition, 1977.

[61] T. Kohonen. *Content-Adressable Memories*. Springer, Berlin, 1980.

[62] T. Kohonen. *Self-Organization and Associative Memory.* Springer, Berlin, third edition, 1989. [First edition, 1984].

[63] J. Laaksonen. Local Subspace Classifier. *Lecture Notes in Computer Science*, 1327:637–642, 1997. Proceedings of ICANN'97, Lausanne, Switzerland.

[64] J. Laaksonen. Subspace Classifiers in Recognition of Handwritten Digits. *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series, No. 84*, 1997. Dr. Tech. Thesis, Helsinki University of Technology.

[65] T. Lehmann, W. Oberschelp, E. Pelikan, and R. Repges. *Bildverarbeitung für die Medizin: Grundlagen, Modelle, Methoden, Anwendungen.* Springer, Berlin, 1997.

[66] T. Lehmann, B. Wein, J. Dahmen, J. Bredno, F. Vogelsang, and M. Kohnen. Content-based Image Retrieval in Medical Applications: A Novel Multi-step Approach. In *Procs. Int. Society for Optical Engineering (SPIE)*, volume 3972(32), pages 312–331, February 2000.

[67] E. Levin and R. Pieraccini. Dynamic Planar Warping for Optical Character Recognition. In *ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume III, pages 149–152, March 1992.

[68] P. Meinicke and H. Ritter. Local PCA Learning with Resolution-Dependent Mixtures of Gaussians. In *Proceedings of ICANN'99, 9th International Conference on Artificial Neural Networks, Edinburgh, UK*, Institution of Electrical Engineers, London, UK, pages 497–502, 1999.

[69] T. P. Minka. A Statistical Learning/Pattern Recognition Glossary. www document, April 1999. URL: `http://vismod.www.media.mit.edu/ tpminka/statlearn/glossary/`.

[70] B. Moghaddam, C. Nastar, and A. Pentland. A Bayesian Similarity Measure for Direct Image Matching. In *Proceedings of the International Conference on Pattern Recognition, Vienna, Austria.* IEEE Computer Society Press, pages 350–358, 1996.

[71] B. Moghaddam and A. Pentland. Probabilistic Visual Learning for Object Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, July 1997.

[72] R. K. Moore. A Dynamic Programming Algorithm for the Distance Between Two Finite Areas. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(1):86–88, January 1979.

[73] H. Ney. Mustererkennung und Neuronale Netze. Script to the lecture on Pattern Recognition and Neural Networks held at RWTH Aachen, 1999.

[74] H. Ney. Digitale Signalverabeitung für Sprache und Bilder. Script to the lecture on Signal Processing for Speech and Images held at RWTH Aachen, 2000.

[75] Y. Normandin. Maximum Mutual Information Estimation of Hidden Markov Models. In C.-H. Lee, F. Soong, and K. Paliwal, editors, *Automatic Speech and Speaker Recognition.* Kluwer Academic Publishers, Norwell, MA, pages 57–81, 1996.

[76] C. Palm, D. Keysers, and K. Spitzer. Gabor Filtering of Complex Hue/Saturation Images for Color Texture Classification. In *Joint Conference on Information Sciences – International Conference on Computer Vision, Pattern Recognition, and Image Processing, Atlantic City, NJ, USA*, volume 2, pages 45–49, February 2000.

[77] R. Perrey. Affin-invariante Merkmale für die 2D-Bilderkennung. Diploma thesis, Chair of Computer Science VI, RWTH Aachen, Aachen, Germany, February 2000.

[78] J. Pösl. *Erscheinungsbasierte statistische Objekterkennung.* PhD thesis, Universität Erlangen Nürnberg, Erlangen, 1998. Shaker Verlag, Aachen.

[79] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, Cambridge, second edition, 1992.

[80] D. Psaltis and F. Mok. Holographic Memories. *Scientific American [International Edition]*, 273(5):70–76 (Int. ed. 52–58), November 1995.

[81] B. Schölkopf, P. Simard, A. Smola, and V. Vapnik. Prior Knowledge in Support Vector Kernels. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Inf. Proc. Systems*, volume 10. MIT Press, pages 640–646, 1998.

[82] E. G. Schukat-Talamazzini. *Automatische Spracherkennung.* Vieweg, Braunschweig, 1995.

[83] H. Schulz-Mirbach. On the Existence of Complete Invariant Feature Spaces in Pattern Recognition. In *Proceedings of the 11th International Conference on Pattern Recognition, Conference B: Pattern Recognition Methodology and Systems*, volume II, Den Haag, The Netherlands, pages 178–182, 1992.

[84] H. Schwenk and M. Milgram. Transformation Invariant Autoassociation with Application to Handwritten Character Recognition. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Inf. Proc. Systems*, volume 7. MIT Press, pages 992–998, 1995.

[85] S. Siggelkow and H. Burkhardt. Image Retrieval based on Local Invariant Features. In *Proceedings of the IASTED International Conference on Signal and Image Processing*, Las Vegas, Nevada, USA, October 1998.

[86] S. Siggelkow and M. Schael. Fast estimation of invariant features. In W. Förstner, J. Buhmann, A. Faber, and P. Faber, editors, *21. DAGM Symposium Mustererkennung, Bonn*, Informatik aktuell. Springer, pages 181–188, September 1999.

[87] P. Simard, Y. Le Cun, J. Denker, and B. Victorri. Transformation Invariance in Pattern Recognition — Tangent Distance and Tangent Propagation. In G. Orr and K.-R. Müller, editors, *Neural networks: tricks of the trade*, volume 1524 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pages 239–274, 1998.

[88] P. Simard. Efficient Computation of Complex Distance Metrics Using Hierarchical Filtering. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Inf. Proc. Systems*, volume 6. Morgan Kaufmann Publishers, Inc., pages 168–175, 1994.

[89] P. Simard, Y. Le Cun, and J. Denker. Efficient Pattern Recognition Using a New Transformation Distance. In S. Hanson, J. Cowan, and C. Giles, editors, *Advances in Neural Inf. Proc. Systems*, volume 5, Morgan Kaufmann, San Mateo CA, pages 50–58, 1993.

[90] P. Simard, Y. Le Cun, J. Denker, and B. Victorri. An Efficient Algorithm for Learning Invariances in Adaptive Classifiers. In *Proceedings of the International Conference on Pattern Recognition, The Hague, The Netherlands.* IEEE Computer Society Press, 1992.

[91] P. Simard, B. Victorri, Y. Le Cun, and J. Denker. Tangent Prop—A Formalism for Specifying Selected Invariances in an Adaptive Network. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Inf. Proc. Systems*, volume 4. Morgan Kaufmann Publishers, Inc., pages 895–903, 1992.

[92] S. J. Smith, M. O. Bourgoin, K. Sims, and H. L. Voorhees. Handwritten Character Classification Using Nearest Neighbor in Large Databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):915–919, September 1994.

[93] S. Y. Sohn. Meta Analysis of Classification Algorithms for Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1137–1144, November 1999.

[94] P. Spellucci. Numerical Analysis for Engineers and Scientists. Lecture Notes to a lecture held at TU Darmstadt.

[95] T. Theiner. Inhaltsbasierter Zugriff auf große Bilddatenbanken. Diploma thesis, Chair of Computer Science VI, RWTH Aachen, Aachen, Germany, February 2000.

[96] S. Uchida and H. Sakoe. A monotonic and continuous two-dimensional warping based on dynamic programming. In *Proceedings of 14th IAPR International Conference on Pattern Recognition (ICPR'98)*, pages 521–524, 1998.

[97] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[98] V. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, New York, 1998.

[99] V. Vapnik. An Overview of Statistical Learning Theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, September 1999.

[100] N. Vasconcelos and A. Lippman. Multiresolution Tangent Distance for Affine-invariant Classification. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Inf. Proc. Systems*, volume 10. MIT Press, pages 843–849, 1998.

[101] J. Wood. Invariant Pattern Recognition: A Review. *Pattern Recognition*, 29(1):1–17, January 1996.

Appendix A

Complements

A.1 Further Experiments

"And are you not," said Fook leaning anxiously forward, "a greater analyst than the Googleplex Star Thinker in the Seventh Galaxy of Light and Ingenuity which can calculate the trajectory of every single dust particle throughout a five-week Dangrabad Beta sand blizzard?" "A five-week sand blizzard?" said Deep Thought haughtily. "You ask this of me who have contemplated the very vectors of the atoms in the Big Bang itself? Molest me not with this pocket calculator stuff."

[1]

This section briefly describes some further results obtained during the experiments carried out for this work.

Data Multiplication via Thinning

Since data multiplication using image shifts led to significant improvements in classification performance, it was also tested, if an improvement could be gained by using *thinning*, which is a morphological operator defined for binary images [65, pp. 223ff].[1] The images were binarized with a given threshold value, then thinning was applied to yield virtual training data. The obtained results are shown in Table A.1. It can be observed that the results are significantly worse than the reference of 3.4% respectively 3.3% for simple training data. This result complies with the binarization experience (see page 103), which shows that the best result on binary images is obtained

[1] The used software for the thinning was implemented by Mark Oliver Güld.

Table A.1: Experiments with multiplication via thinning

Binarization Threshold [% of max]	ER 1-NN [%]	ER KD [%]
25	3.8	3.7
50	4.1	4.0
75	5.4	5.4

for low thresholds, leading to the 'thickest' images, which explains that thinning does not lead to improvements.

Restricting the Movement in $\hat{M}$

A few experiments were conducted to investigate the effect of the restriction of the movement of the projected point in the tangent subspace in tangent distance. this corresponds to a setting of $\gamma < \infty$ in Equation (5.6). For that, the optimal parameters α were determined, that is, the projection into the tangent subspace was performed. Then, the projection was recalculated using the corresponding new parameter

$$\alpha' = \frac{\gamma^2}{1+\gamma^2}\,\alpha$$

No gains in classification performance could be obtained using different values for γ. This is in compliance with the statement of HASTIE & SIMARD, who observed that it was "unnecessary to restrict the transformations to be local, since the degradation of the linear approximation far from the origin produced images that were extremely distorted." [37] Furthermore, "in high-dimensional image spaces, it is unlikely that images will have large projections within the tangent space and small projections off it." [43]

Using Squared l_p-Norms

As few maybe even one large differences in pixel values can mislead classifiers based on squared error distances (see e.g. [100]), it can be advisable to introduce a local threshold which limits the maximum contribution of a single pixel to the distance between two images. (Note that this thresholding implies a minimum probability for any observation with respect to any reference and therefore the probability density function is not normalizable any more.) This is justified by a priori domain knowledge, e.g. when it is known that the patterns may be subject to artifacts that do not affect class-membership, like noise or changing scribor[2] position in radiographs. On the other hand, when looking at relatively small images of digits, one notices that e.g. changing only a few pixels can be significant for discriminating between the handwritten digits '4' and '9'. Here it can be useful to enlarge the contribution of a single pixel difference generalizing the used norm and use a squared l_p-norm (see Equation (2.15)) instead of the squared Euclidean norm with values $p > 2$. Fig. A.1 shows the obtained error rate versus p for the kernel density based tangent distance classifier, without extended data. Figure A.2 shows two examples, which visualize the positive effect an increased value for p can have on classification of images of handwritten digits. In both cases only few pixels are different between the test image and the incorrect reference. The two images were correctly classified using $p = 3$, but incorrectly classified with $p = 2$. The higher value for p was especially effective for 15 times multiplied training data with tangent approximation with obtained error rates of 3.4% ($p = 2$) respectively 3.1% ($p = 3$). The use of higher values for p did not lead to an improvement of the best single classifier, but the classifier mentioned above was used in the bagging experiment which obtained the best overall result.

Reducing the Training Set

There are several methods known for reducing the number of samples in the training set. Among them are the editing and condensing techniques [25, 32, pp. 358ff]. In *editing* samples of the training

[2]Scribor refers to the area in a radiograph where patient and examination data is printed.

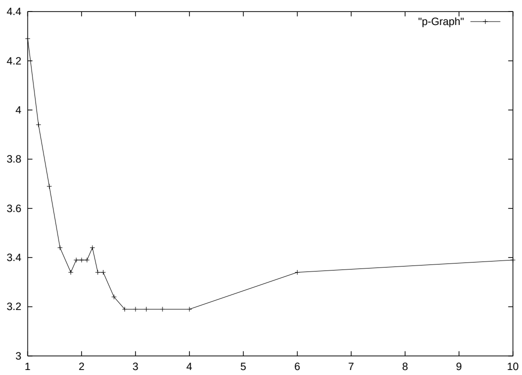

Figure A.1: Error rate of basic KD classifier using tangent distance on USPS (ordinate: p, abscissa: error rate [%])

Figure A.2: Example for two digits that were correctly classified using $p = 3$, but incorrectly classified with $p = 2$. (left: test image, center: best fitting reference, right: best fitting reference of the correct class)

set that are misclassified using a separate part of the training data as references (partitioning the training set) are discarded. In *condensing* the training set is built up from the empty set by adding more training samples in the case they cannot be correctly classified by the existing ones. This leads to a systematic removal of the ineffective samples. These methods can be applied to reduce the number of samples (for speedup) or to improve the classification by finding samples that lead to misclassification rather than to correct classification. The latter was the purpose of the experiment performed. In the first variant of the editing technique, the training set was classified using a leaving one out approach and the number of times were counted, that each sample led to a correct classification respectively a misclassification of another sample. A given training sample was then rejected, that is removed from the training set, if the number of times it led to a misclassification exceeded the number of times it led to a correct classification in a 1-NN classifier. This method yielded the set of 'bad' training samples shown in Figure A.3. Then the approach was extended to take into account not only the number of times a given sample led to misclassifications, but the relative proportion of probability weight in the summation for the KD conditional density. This approach led to a much larger set of 'bad' training samples shown in Figure A.4. Finally, from the two sets four seemingly 'very bad' examples were chosen, shown in Figure A.5. Then, the training set was reduced using each of the shown reduction sets, but the approach did not improve classification results in any of the experiments carried out, except for the 1-NN performance with the hand selected reduction set, which gave a 0.1% decrease in error rate. For the kernel density based classifier no improvement was obtained. This may be seen as an indication for the fact that a kernel density based classifier can overcome the influence of 'bad' training samples, because the decision of the classifier is not based on only one sample, as is the case for the 1-NN classifier. A

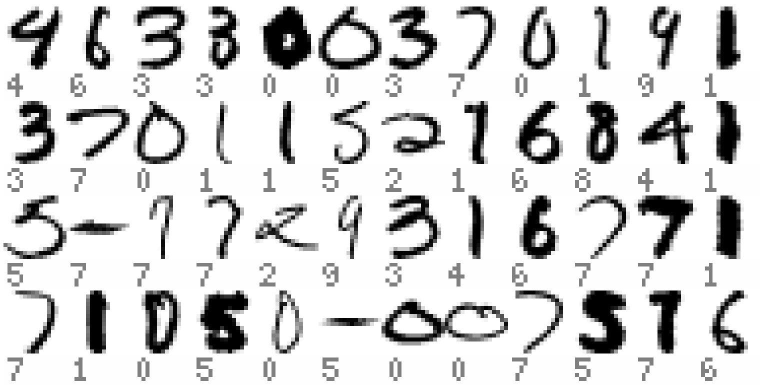

Figure A.3: Automatically constructed reduction set 1

Table A.2: Confusion Matrix for best single classifier result on USPS corpus

Test pattern from class	Classified as										
	0	1	2	3	4	5	6	7	8	9	$\sum$
0	356	0	0	0	0	3	0	0	0	0	359
1	0	260	0	0	2	0	1	1	0	0	264
2	1	1	192	1	1	1	0	1	0	0	198
3	2	0	0	158	0	4	0	0	0	2	166
4	0	2	0	0	191	1	0	1	0	5	200
5	1	1	1	1	0	155	0	0	0	1	160
6	0	0	0	0	0	1	169	0	0	0	170
7	0	1	1	0	3	0	0	142	0	0	147
8	3	1	1	0	0	1	0	0	160	0	166
9	0	0	0	0	0	1	0	0	0	176	177

possible application of the described method could be to automatically construct sets of possible label mistakes, which can then be reconsidered by an expert, reducing the number of samples that need to be looked at.

A.1.1 Confusion Matrix for Best Single Classifier on USPS

Table A.2 shows the confusion matrix for the USPS database for the best single KD classifier using tangent distance and nine times multiplied training and test data constructed with image shifts. It can be seen that most mistakes were made for the interpretation of ‘4’ as ‘9’ followed by interpretation of ‘3’ as ‘5’.

A.2 Implementation

Tricia had been quite impressed with herself, but also very impressed with the computer system she was working on. Using a computer workstation on Earth the task would probably have taken a year or so of programming.

[5]

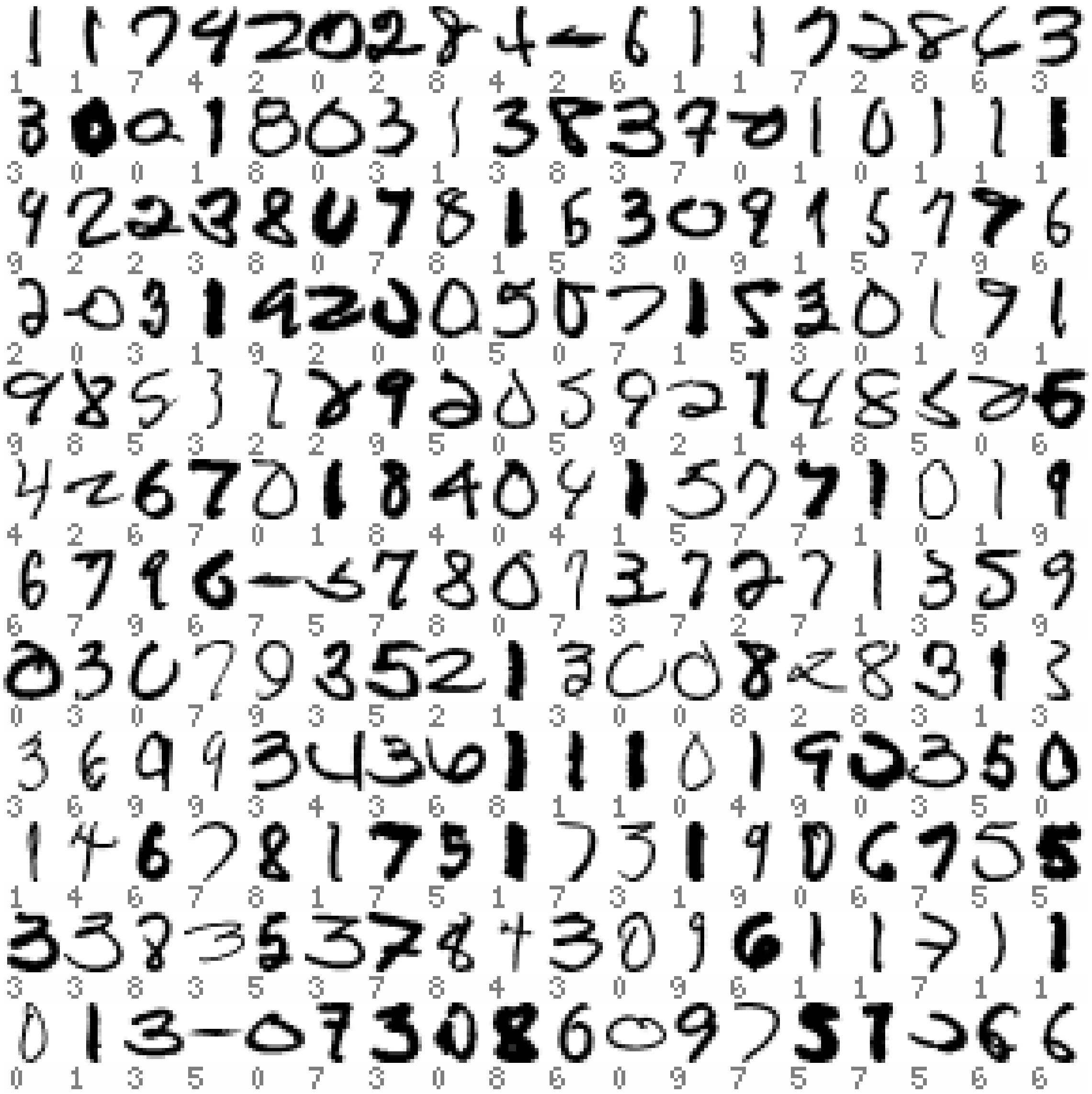

Figure A.4: Automatically constructed reduction set 2

Figure A.5: Four training examples manually chosen for reducing from the automatically constructed sets

This section of the appendix will briefly summarize the different programs developed and modified in the course of this work. The programs were written in the C programming language and compiled and executed on DEC Alpha and Intel Pentium processors. The programming received a lot of help from Mark Oliver Güld and Alexander Crämer, who among other things provided the software for classifier combination. Moreover some procedures from [79] were used, e.g. `jacobi.c` and `dsvdcmp.c` for eigenvalue and eigenvector computation as well as computation of orthonormalized subspaces. For some rather mathematical problems the Maple environment was very helpful, e.g. for symmetry analysis for holographic classification, and for considerations concerning the bilinear equation system and the linear difference (and differential) equations in manifold modeling.

Some typical time requirements on the mentioned processors for the calculation of SVD and tangents are given here for the USPS database. For single sided Tangent distance tangent calculation consumed about 5% of the total time (with about 0.2 ms per image) and SVD required about 60% of the total time and about 2.5 ms per orthonormalization of image tangent subspace performed. For double sided tangent distance an SVD needs to be performed for each distance computation separately (or equivalently a least squares problem must be solved) and the dimension of the subspace is doubled. One SVD call consumed about 9 ms time and the total percentage of the time needed for calls of the SVD was about 84%.

In the following some of the developed programs are listed with a brief description.

`makeholo.c` constructs a hologram from a given dataset with class labels

`decode.c` classifies a set of test patterns given a hologram

`tangent.c` calculates the tangents to a set of images

`nn_tangent.c` performs NN or KD classification using tangents or structured covariance matrices (lots of parameters)

`nn_tangent_svdgesamt.c` for the double sided tangent distance a seperate clasifier was developed

`centroid.c` calculates the different centroid / tangent subspace models for a given set of images

`leven2D.c` calculates the 2D Levenshtein distance following [72]

`inflate.c` inflates covariance matrices for smaller size images consistently for use with larger images

`four_norm.c` calculates an image normalization in the Fourier domain by setting the imaginary phases of the lowest frequencies to zero consistently

`labelfehler.c` determines training set reduction sets based on different criteria

`nni.c` IRMA classification program based on `nn_tangent.c` and the works of Thomas Theiner for [95] (lots of parameters added)

A.3 Complement to the Proof in Section 5.1.1

"Simple. I got very bored and depressed, so I went and plugged myself in to its external computer feed. I talked to the computer at great length and explained my view of the Universe to it," said Marvin.
"And what happened?" pressed Ford.
"It committed suicide," said Marvin and stalked off back to the Heart of Gold.

[1]

The results obtained in 5.1.1 can also be shown without using maximum approximation. Comments on the calculations are given there.

$$p(x|\mu) = \int p(x, \alpha|\mu) \, d\alpha$$

$$
\begin{aligned}
&= \int p(\alpha|\mu)\; p(x|\alpha,\mu)\; d\alpha \\
&= \int p(\alpha)\; p(x|\mu_\alpha)\; d\alpha \\
&= \int \frac{1}{\sqrt{2\pi\gamma^2}^L} \exp\left(-\frac{1}{2\gamma^2}\sum_l \alpha_l^2\right) \cdot \\
&\qquad \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}(\mu + \sum_l \alpha_l\mu_l - x)^T\Sigma^{-1}(\mu + \sum_l \alpha_l\mu_l - x)\right)\; d\alpha \\
&= \frac{1}{\sqrt{2\pi\gamma^2}^L}\frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \cdot \\
&\quad \int \exp\left(-\frac{1}{2}\left(\frac{1}{\gamma^2}\sum_l \alpha_l^2 + (\mu + \sum_l \alpha_l\mu_l - x)^T\Sigma^{-1}(\mu + \sum_l \alpha_l\mu_l - x)\right)\right)\; d\alpha \\
&= \frac{1}{\sqrt{2\pi\gamma^2}^L}\frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \cdot \\
&\quad \int \exp\left(-\frac{1}{2}\left(\frac{1}{\gamma^2}\sum_l \alpha_l^2 + (\mu - x)^T\Sigma^{-1}(\mu - x) + (\mu - x)^T\Sigma^{-1}(\sum_l \alpha_l\mu_l)\right.\right. \\
&\qquad \left.\left. +(\sum_l \alpha_l\mu_l)^T\Sigma^{-1}(\mu - x) + (\sum_l \alpha_l\mu_l)^T\Sigma^{-1}(\sum_l \alpha_l\mu_l)\right)\right)\; d\alpha \\
&= \frac{1}{\sqrt{2\pi\gamma^2}^L}\frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}\left((\mu - x)^T\Sigma^{-1}(\mu - x)\right)\right) \cdot \\
&\quad \int \exp\left(-\frac{1}{2}\left(\sum_l \alpha_l^2(\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l) + 2(\mu - x)^T\Sigma^{-1}(\sum_l \alpha_l\mu_l)\right)\right)\; d\alpha \\
&= \frac{1}{\sqrt{2\pi\gamma^2}^L}\frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}\left((\mu - x)^T\Sigma^{-1}(\mu - x)\right)\right) \cdot \\
&\quad \int \exp\left(-\frac{1}{2}\left(\sum_l (\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l)\left(\alpha_l + \frac{(\mu - x)^T\Sigma^{-1}\mu_l}{(\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l)}\right)^2\right.\right. \\
&\qquad \left.\left. -\sum_l \frac{((\mu - x)^T\Sigma^{-1}\mu_l)^2}{(\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l)}\right)\right)\; d\alpha \\
&= \frac{1}{\sqrt{2\pi\gamma^2}^L}\frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}\left((\mu - x)^T\Sigma^{-1}(\mu - x) - \sum_l \frac{((\mu - x)^T\Sigma^{-1}\mu_l)^2}{(\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l)}\right)\right) \cdot \\
&\quad \int \exp\left(-\frac{1}{2}\left(\sum_l (\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l)\left(\alpha_l + \frac{(\mu - x)^T\Sigma^{-1}\mu_l}{(\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l)}\right)^2\right)\right)\; d\alpha \\
&= \frac{1}{\sqrt{2\pi\gamma^2}^L}\frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}\left((\mu - x)^T\Sigma^{-1}(\mu - x) - \sum_l \frac{((\mu - x)^T\Sigma^{-1}\mu_l)^2}{(\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l)}\right)\right) \cdot \\
&\quad \int \prod_l \mathcal{N}(\alpha_l| -\frac{(\mu - x)^T\Sigma^{-1}\mu_l}{(\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l)}, (\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l)^{-\frac{1}{2}})\; d\alpha \\
&= \frac{1}{\sqrt{2\pi\gamma^2}^L}\frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left(-\frac{1}{2}\left((\mu - x)^T\Sigma^{-1}(\mu - x) - \sum_l \frac{((\mu - x)^T\Sigma^{-1}\mu_l)^2}{(\frac{1}{\gamma^2} + \mu_l^T\Sigma^{-1}\mu_l)}\right)\right) \cdot
\end{aligned}
$$

$$
\begin{aligned}
& \prod\nolimits_l \sqrt{2\pi}(\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l)^{-\frac{1}{2}} \\
= \quad & \frac{1}{\sqrt{\gamma^2}^L} \frac{1}{\sqrt{(2\pi)^D |\Sigma|}} \exp\left(-\frac{1}{2}\left((\mu - x)^T \Sigma^{-1} (\mu - x) - \sum\nolimits_l \frac{((\mu - x)^T \Sigma^{-1} \mu_l)^2}{(\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l)}\right)\right) \cdot \\
& \prod\nolimits_l (\frac{1}{\gamma^2} + \mu_l^T \Sigma^{-1} \mu_l)^{-\frac{1}{2}}
\end{aligned}
$$

The last few steps made use of the properties of the normal distribution. In the last expression it can be seen that only the exponential term $\exp(\cdots)$ depends on x such that the results arrived at without maximum approximation are essentially the same as those in 5.1.1.

Index