Experiments with robust similarity measures for OCR *

Gilbert Maître
Institut Dalle Molle d'Intelligence Artificielle Perceptive
c.p. 592, CH-1920 Martigny
maitre@idiap.ch

February 1995, Revised June 1995

Abstract

This report describes a system for handprinted character recognition. The goals of this work are to understand better the role of shape, shape representations, and shape similarity in character recognition, and to develop character recognition methods that are more robust to degradations and distortions of character patterns than existing methods. In the system, methods previously used in visual object recognition are used to determine the similarity between input character patterns and stored character prototypes. The shape of character patterns is represented by their boundary, sometimes annotated with information about local orientation and curvature. The effects of different representations of shapes and different similarity measures are evaluated and compared in tests with character patterns from the NIST Special Database 3.

Keywords: OCR, shape representation, shape similarity, k-NN classification

*under supervision of Thomas M. Breuel, supported by grant no. 21-39576.93 from the Swiss National Foundation for Scientific Research

Contents

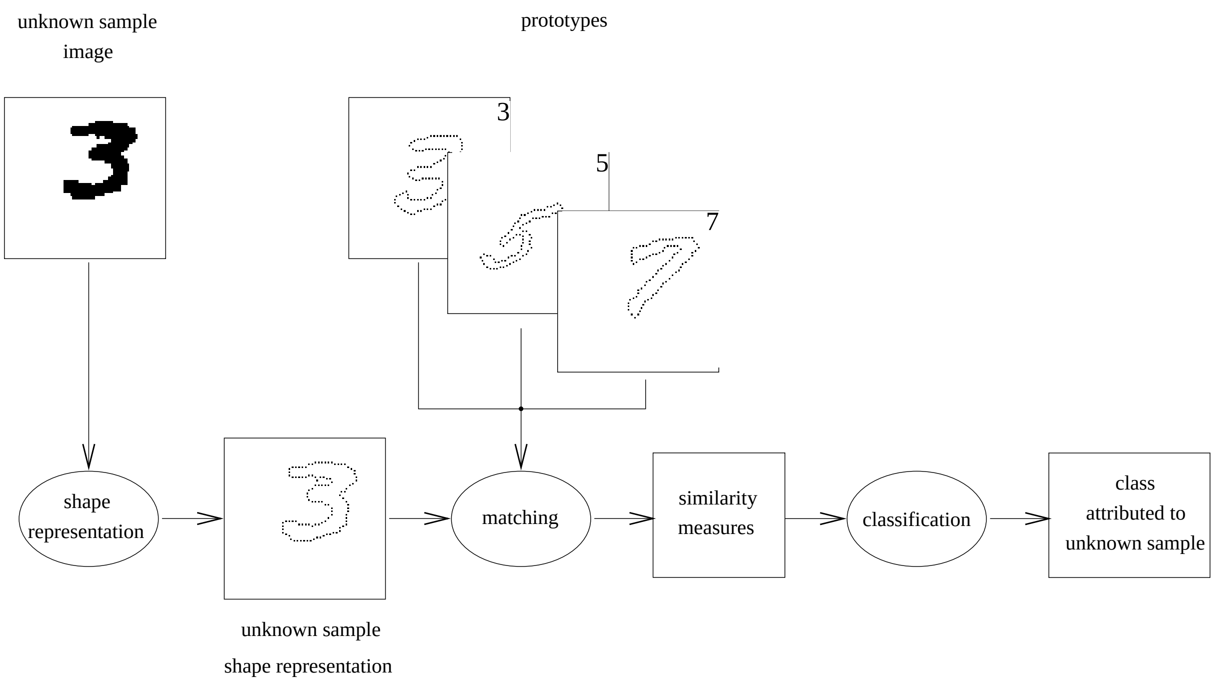

Figure 1: The character recognition system

1 Introduction

Limitations of existing handwriting recognition methods still prevent their application in many areas. We believe that these limitations can only be addressed by understanding better how the shape of character patterns should be described and compared.

Computer vision has had to cope with difficult problems relative to visual object recognition. Difficulties arise from uncertainty in measurements and from the occurrence of situations where objects are partially occluded and/or embedded in cluttered scenes. Computer vision researchers have developed methods to address these problems. We therefore investigate the use of shape representations and shape recognition algorithms from visual object recognition for achieving robust character recognition.

Applying visual object recognition techniques to character recognition problems has an additional reason. In visual object recognition, the evaluation of methods is made difficult because of a lack of databases available for experiments. In character recognition however, large amounts of data are available, letting us test the methods and evaluate their performances on a statistical basis.

We apply visual object recognition methods to character recognition as follows (see figure 1). First, a shape representation is computed from the raw input image containing an unknown character pattern. Second, the similarity between the extracted representation and stored prototypes is measured. Prototypes are character patterns selected in a set of training samples and represented in the same manner as the unknown character patterns. Finally, the unknown character pattern is classified according to the most similar prototypes. Each of these three steps is described in more details hereafter.

A character pattern is represented by its boundary. Such a representation contrasts significantly with others used for character recognition. Indeed, many systems are based on features which are specific to character recognition and often even to particular alphabets. Also, those features contain usually less information than the initial data. Representation through boundary, however, is general; it is also complete and should hence allow the best possible recognition scores.

The extraction of the character pattern boundary is preceded by a normalization step in order to reduce shape variability. We investigate the use of slant, size, and/or stroke width normalization.

Shape similarity between the unknown character pattern and the stored prototypes is measured by bounded error matching [Gri90, Bre92]. This shape comparison technique from visual object recognition lets us choose classes of geometric transformations under which shape is invariant. Here, we limit invariance to the class of (two-dimensional) translations. The benefits of bounded error matching for visual object recognition are tolerance to measurement uncertainties and robustness in the presence of occlusions and clutter. In applying this matching technique to character recognition, we expect similar benefits, i.e invariance to small distortions together with robustness to segmentation errors and to situations where non-character patterns (e.g. underlining) interfere.

The unknown character pattern is classified using a k-Nearest Neighbors (k-NN) rule, for which the similarity measures represent distances between the unknown sample and the prototypes. In some cases, we use a decision tree classifier as post-processing step to k-NN classification results. The use of this post-processing increases significantly the performance of the recognition system [Bre93].

For training, i.e. learning prototypes, the classification stage is made of a simple Nearest Neighbor (NN) classifier. The procedures used for learning prototypes can be interpreted as training set editing methods for NN classification [Das91].

2 Shape representation

Character recognition starts with the representation of the unknown character pattern by its boundary. The boundary itself has several possible representations. The one we use is made of a set of regularly sampled boundary points, sometimes complemented with associated boundary orientation and/or curvature. Such a representation contrasts with the one used in Fourier-based approaches, which is global and suffers therefore from a lack of robustness to degradations.

The extraction of the character pattern boundary from the input image is preceded by two processing steps applied in sequence to the input image. The first one is concerned with noise removal, the second with shape normalization of the character pattern. We describe hereafter in detail each of the three processing steps leading to the representation of the character pattern by its boundary. Numerical values in parentheses represent the parameter values we have used in the experiments reported in section 5.

- **Noise removal.** Connected components of small size (less than 5 pixels) are removed from the image.
- **Shape normalization.** The image is transformed so that some shape parameters of the character pattern get a preset norm value. We have tested following normalization procedures:
  - **Size Normalization (SN).** The image is scaled uniformly so that the pattern bounding box fits into a box of normal size (48x48).
  - **Slant Correction (SC).** Slant correction is an operation which shears the input image in a direction parallel to the horizontal axis to force the global slope of the character pattern to be vertical. In the present system, shearing parameters are determined by the conditions that the operation lets the image centroid unchanged and that the second order mixed moment of the output image is equal to zero.
  - **Stroke Width Normalization (WN).** First, the image is smoothed with a Gaussian and thresholded (relative threshold 0.3). Then, a thinning operation is applied ([Pav82]: algorithm 9.1). Finally, a dilation operation is performed with a disk as structuring element (radius 2).
- **Boundary extraction.** The character pattern boundary to be extracted from the image is similar to an edge. So, boundary extraction is essentially performed in applying to the image an edge detection method, as described in the next few lines. The image is convolved with a Gaussian (standard deviation 1.5). The Laplacian of the resulting image as well as the gradient are computed. Laplacian zero crossings which correspond to a non-zero gradient value are selected as boundary points.

  Further, on the purpose of data reduction, the boundary points are grouped into chains which are then under-sampled by a fixed integer factor (2). The remaining boundary points are retained to represent the character pattern.

When used, boundary orientation and curvature at the extracted points are computed from the image gradient. The boundary orientation is simply the gradient direction. The boundary curvature is estimated from two gradient directions taken at neighboring boundary points (as their difference in angle). In order to smooth curvature values, 1-D curvature signals are constructed based on the under-sampled boundary chains used before. The signals are convolved with a 1-D Gaussian (standard deviation 2).

3 Matching

Once the representation of the unknown character pattern is available, the matching stage measures the similarity between the unknown sample and each prototype, applying the bounded error matching method used elsewhere for visual object recognition.

Given a set of model features, a set of image features with associated error bounds, bounded error matching finds the geometric transformation (e.g. translation, rotation, and/or scaling) such that a maximal number of model features and image features are brought into correspondence under the given error bounds. Features are geometric entities which are variant to the classes of transformations that are considered. Invariant geometric properties and/or non-geometric properties associated to the features play the role of feature labels. In order to correspond, features must have the same label, we say they must be compatible. The correspondence between compatible model/image features, called quality of match, can be measured in different ways [Bre92].

Bounded error matching is used in following manner to measure the similarity between the unknown sample and a prototype. Model and image are, respectively, the prototype and the unknown sample. Also, the geometric transformations restrict themselves to the class of translations and the features are simply the boundary points. Since they are invariant to translations, boundary orientation and curvature play the role of feature labels. With the adaptation of bounded error matching described here-before, the measure of similarity between unknown sample and prototype is the maximal quality of match for all possible transformations between image and model.

For the rest of this section, we will use the general concepts, model, image, and features, to refer to prototype, unknown character pattern, and boundary points, respectively. First, we describe what measures we have chosen to evaluate the quality of match and hence the similarity between model and image. In a second part, we explain how the similarity measures are computed.

Choice of the similarity measures. The similarity between model and image is determined by the best quality of match. We investigate different possibilities for evaluating the quality of match and so the similarity. More precisely, we consider alternatives in regard of two aspects, described hereafter in their principles:

Bounded error vs. least error quality of match. Using an unique error bound value for all features lets us two basic choices for evaluating the quality of match. We can either fix the error bound and measure the quality of match by the amount of feature correspondences (bounded error) or fix the amount of correspondences and measure the quality of match by the minimal error bound allowing this amount (least error).

Symmetric vs. asymmetric quality of match. Feature correspondences can be measured either in a symmetric way, i.e. so that model and image are interchangeable, or in an asymmetric manner.

The combination of these alternatives defines four measures for the quality of match.

For a precise description of the similarity measures based on these alternatives, we introduce now following definitions. Let us first define the model matches as the set of model features that are within the error bound of some (compatible) image feature and the image matches as the set of image features for which there exists at least one (compatible) model feature within the error bound. (Note that, in this definition, image matches are not symmetric to model matches. However, image features are defined in that way to take advantage of the computation facilities described below.) Let us further define the symmetric common match rate as the minimum between the fraction of model matches and the fraction of image matches, and the asymmetric common match rate as the minimum between the fraction of model matches and the fraction composed by the number of model matches divided by the number of image features. The above defined concepts let us describe precisely the alternatives for the quality of match as:

Bounded error vs. least error quality of match. The bounded error quality of match is determined by the common match rate achieved for a given error bound while the least error quality of match is determined by the minimal error bound achieved for a given common match rate.

Symmetric vs. asymmetric quality of match. The symmetric quality of match uses the symmetric common match rate and the asymmetric quality of match uses the asymmetric common match rate. We note that the property of symmetry is not strictly verified since image matches are not exactly symmetric to model matches.

Computation of the similarity measures. The computation of the similarity measure necessitates to find the best (maximal) quality of match among the set built by all translations. This is linked to the problem of finding the translation of best quality of match.

- **Finding the best quality of match**

  In order to solve the bounded error matching problem, computer vision researchers have developed powerful algorithms [Bre92, Cas92], which search for the optimal transformation in the continuous transformation space. However, in the present recognition problem, such algorithms are not optimal for two reasons. First, we are not only interested in bounded error matching but also in least error matching. Second, we can restrict ourselves to a finite set of transformations, i.e. all translations of at most $\pm N$ pixels ($N = 10$ in our experiments). So, because of these conditions, the best quality of match is found instead by an exhaustive search among all translations. Note that the translation vector is measured with respect to model/image centroid overlap.

- **Transforming the representation of image features**

  Since the quality of match has to be computed once for each translation and therefore several times for each model, the operation has to be carefully optimized. An interesting gain in computation is obtained by transforming the representation of the image features with the help of the Euclidian Distance Transform (EDT). The EDT of a binary image with values "Off" and "On" is an image containing in each position the distance in the input image from the pixel of same position to the nearest pixel which is "On". Optionally, an image with the positions of the nearest "On" pixels is associated to the EDT image. We call this second image EDT source image.

Boundary representation is transformed with the help of the EDT in following manner. A 2-D array of images is created, one image for each quantified orientation and curvature value. Images are initialized at the "Off" value. Pixels with same positions as boundary points are set to "On" in the images of corresponding orientation and curvature values, as well as in images of neighboring values (9-neighborhood). The array of EDT images is computed. In the case of a symmetric quality of match, the array of EDT source images is also computed.

- **Computing the quality of match**

  Any of the quality of match measures can be easily computed from the histogram(s) of minimal distances between model and image points. In the case of an asymmetric measure, the histogram of minimal distances from model points is sufficient. For the symmetric measures, the histogram of minimal distances to image points has also to be computed.

  Histogram of minimal distances from model points. For each (translated) model point, the distance to the nearest compatible image point is read in the corresponding EDT image, which is selected according to model boundary orientation and curvature. The histogram of distances is constructed.

  Histogram of minimal distances to image points. For each (translated) model point, the position of the nearest compatible image point is read in the corresponding EDT source image, while the distance is read in the EDT image. The distance value is written into a buffer image at the image point position, if there is not already a distance value which is smaller than the new one. After each model point has been processed, the distances stored in the buffer image at the image points positions are read. The histogram of distances is constructed.

4 Classification

As the last processing stage, classification makes the decision on character identity, using the measures of similarity between each prototype and the unknown sample. The classification stage has two different structures for the two system working phases, training and testing. For training, i.e. learning prototypes, it is made of a simple NN (Nearest Neighbor) classifier, while, for testing, similarity measures are input into a k-NN classifier, the results of which are post-processed by a decision tree classifier. The classifier structure used for testing has been described in [Bre93]. We report here on the training phase.

Learning prototypes may be considered as a training set editing operation, known from the field of the k-NN classifiers [Das91]. Methods proposed in the specialized literature for reducing (editing) the training set are iterative and computation expensive, even the simplest ones [Har68, Gat72, Wil72]. Hence, in order to limit the computation time, we use instead a simple one-pass method, implemented and tested in two versions: a basic one and an enhanced one, the aim of which is to avoid storing bad samples as prototypes.

Basic prototype learning. One sample pattern of each character is arbitrarily chosen among the training set as a first prototype. Then, in an arbitrary order, each of the remaining samples is matched with the prototypes. If the classification given by the nearest neighbor (prototype of best match) is wrong, the current sample is memorized as a prototype. In the contrary, if it is right, the sample is "forgotten".

Elimination of bad prototypes. A score is attributed to each prototype and initialized at zero. When a prototype is the nearest neighbor of a new sample, its score is increased by one if the classification is correct, and decreased by one if the classification is incorrect. The prototype is eliminated when its score is below zero.

5 Experimental results

NIST database. The experiments have been carried out on the NIST (National Institute of Standards and Technology) Special Database 3 [GW92]. This database contains about 313000 handprinted character patterns available in separate binary images. For each image, there exists a reference classification. The character samples come from 2100 writers. They are mainly digits: about 223000 against 45000 upper-case and 45000 lower-case character patterns.

The database was built in scanning 2100 specially designed (identical) forms which were filled by paid US Census workers (each by a different writer). Character patterns were extracted from the images of the forms and classified automatically. The results were then verified and corrected by human operators such that the reference classification error rate is less than 0.1%.

Experimental database. For the experimentation, only a subset of the available patterns were used, involving first the reduction of the character set to the 10 digits. Second, from the available 223000 digits, only 20000 were used, those with the identifiers 0-19999. This experimental set was divided into a training set (0-15024), which represents 139 writers, and a test set (15025-19999) which represents 47 writers, all different from the writers of the training set.

Results. The performances of the character recognition system have been evaluated for different shape representations and different shape similarity measures. The achieved error rates, expressed in %, are represented in tables 1- 4, each time with several classifiers: k-NN alone for different k values, 10-NN with decision tree post-processing, 2-NN with rejection when the two Nearest Neighbors do not belong to the same class. In each case, the number of prototypes resulting from training (with single NN classifier) is also given.

We present now the results and discuss them with respect to the influence of shape representation and shape similarity. Finally, we compare the two prototype learning procedures described in section 4.

Normalization	proto.	1-NN	3-NN	5-NN	7-NN	9-NN	2-NN/rej.	2stages
N1	606	3.78	**3.34**	3.36	3.70	4.22	0.40/11.6	2.08
N2	417	3.08	2.73	**2.43**	3.14	4.02	0.28/10.1	1.28
N3	895	5.85	6.07	**5.33**	5.69	5.63	1.37/14.6	2.26

Table 1: Influence of the shape normalization on the classification errors

features	proto.	1-NN	3-NN	5-NN	7-NN	9-NN	2-NN/rej.	2stages
P1	746	5.59	4.18	**3.98**	4.10	4.44	0.99/12.2	
P2	417	3.08	2.73	**2.43**	3.14	4.02	0.28/10.1	1.28
P3	414	3.08	**2.11**	2.49	2.69	2.83	0.18/9.1	
P4	361	2.81	2.27	**2.19**	2.27	2.79	0.22/8.3	

Table 2: Influence of the boundary representation on the classification errors

meas.	param.	proto.	1-NN	3-NN	5-NN	7-NN	9-NN	2-NN/rej.	2stages
AB	2.9	583	3.70	3.24	**2.83**	3.14	3.28	0.40/11.6	1.64
AB	4.0	450	2.81	**2.69**	2.75	2.81	3.14	0.40/9.2	1.36
AB	5.0	417	3.08	2.73	**2.43**	3.14	4.02	0.28/10.1	1.28
AB	6.0	441	**2.58**	2.81	2.96	3.20	3.58	0.36/9.9	1.40
AB	7.0	559	3.72	2.79	**2.59**	2.59	2.85	0.52/9.5	1.30
SB	2.9	636	3.70	2.89	2.65	**2.37**	2.51	0.38/12.0	1.72
SB	5.0	441	3.20	3.09	**1.63**	1.71	1.81	0.40/8.8	1.58
AL	70%	680	4.46	3.22	2.69	**2.13**	2.15	0.62/12.0	2.22
AL	80%	478	3.38	2.67	**1.93**	2.01	2.21	0.32/9.8	1.70
AL	90%	426	2.61	**2.35**	2.65	3.02	3.40	0.38/7.9	1.32

Table 3: Influence of the similarity measure on the classification errors

Normalization	proto.	1-NN	3-NN	5-NN	7-NN	9-NN	2-NN/rej.	2stages
N1	532	3.20	3.44	**2.96**	3.24	3.88	0.34/10.5	
N2	357	2.43	**2.39**	2.53	3.22	3.76	0.28/8.4	
N3	537	3.74	**3.50**	3.52	3.78	4.54	0.95/8.5	

Table 4: Influence of the shape normalization on the classification errors, in the case where learning includes elimination of bad prototypes

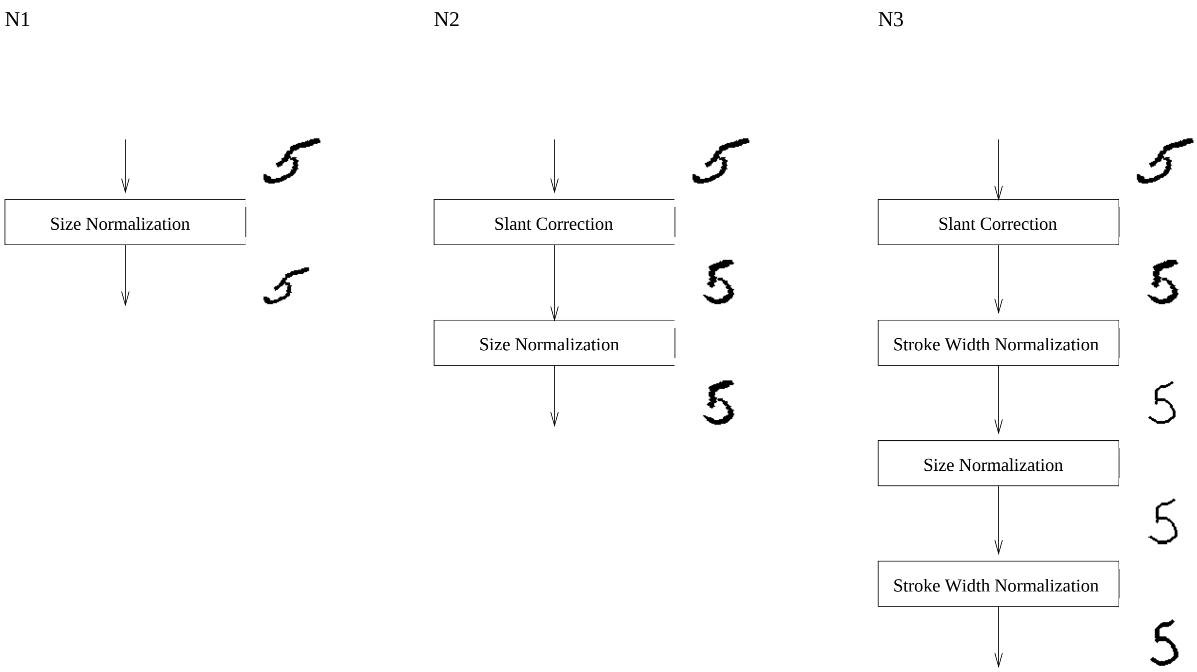

Figure 2: The experimented normalization variants

Shape representation. The experiments on shape representation have been performed with a single similarity measure: asymmetric bounded error with 5 as error bound value. With this similarity measure, we have investigated the influence of some normalization variants and some boundary representation variants on the recognition scores.

- **Normalization**

  We have tested three variants for normalization built with the processing blocks (SN,SC, and WN) described in section 2. The variants, showing increasing complexity, are: (see figure 2)

  (N1) size normalization.

  (N2) slant correction and size normalization. Slant correction is applied prior to size normalization.

  (N3) slant correction, size normalization, and stroke width normalization. The stroke width normalization procedure is applied two times. Once just before and once just after size normalization.

  Table 1 shows that the best results are achieved by variant N2 (slant correction and size normalization) as both the number of prototypes and the error rate are minimal. Size normalization alone does not reduce much the variation in shape between patterns of the same character. On the other hand, the use of stroke width normalization makes some character patterns more similar to patterns of other characters.

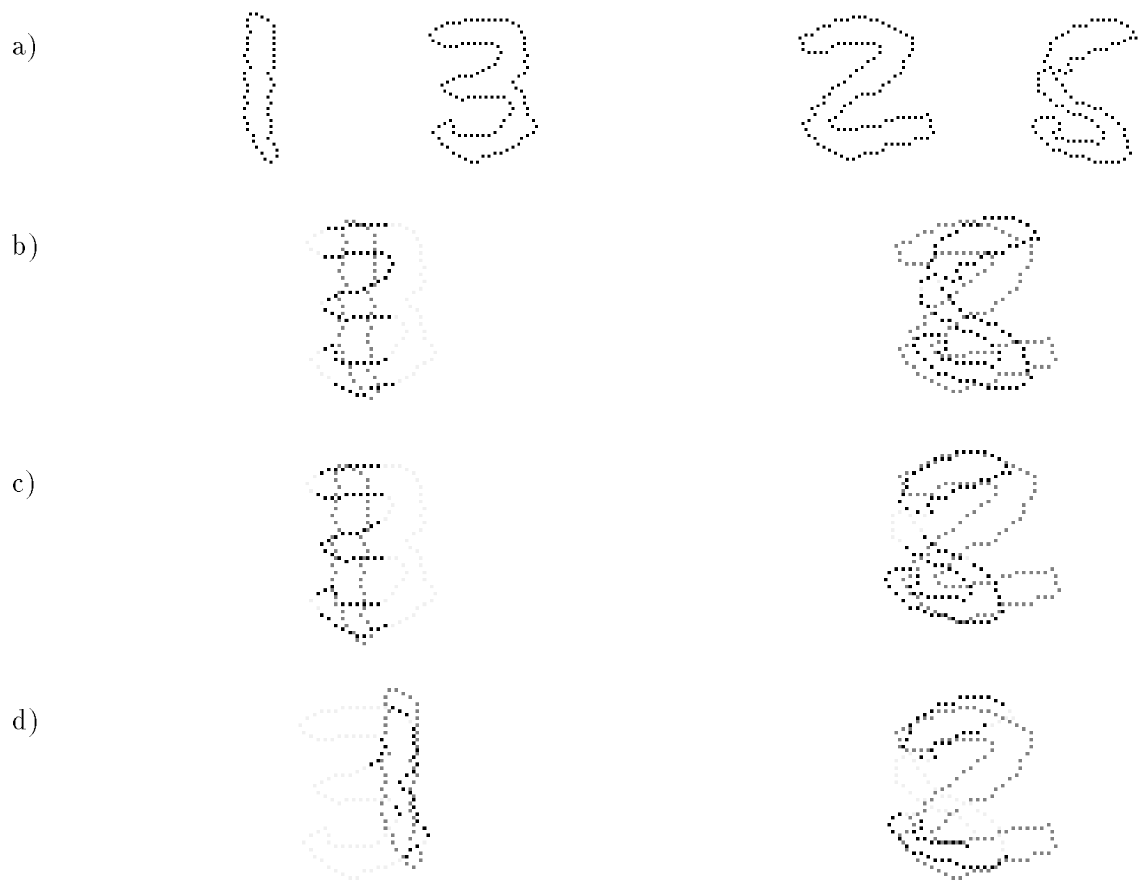

Figure 3: Influence of the boundary orientation on the best match. (Representation of matching: medium-grey: image points, black: matching model points, light-grey: non-matching model points.) (a) Matched shapes. In each pair, left: image, right: model. (b) Best match without taking into account the orientation. (c) Best match when the orientation is quantified in 4 levels. (d) Best match when the orientation is quantified in 16 levels.

- **Boundary representation**

  We have examined how the use of orientation and curvature in the representation of boundary influences the system performances. More precisely, we have tested four configurations (R1-R4) made by: use of orientation alone represented by 8 (R1), 16 (R2), and 32 (R3) quantification levels, use of orientation together with curvature sign (convex, straight, concave), the orientation being represented by 16 quantification levels (R4).

  Increasing the number of quantification levels in orientation and adding curvature to the representation of the boundary both cause an increase of the dissimilarity measure between unknown sample and prototype, since the number of compatible boundary points decreases.

  From the results represented in table 2, we conclude that the increase in dissimilarity measure produced in taking into account the boundary orientation is beneficial.

Moreover, the quantification of the orientation angle must be sufficiently fine (at least 16 levels). However, the improve in recognition scores produced by increasing the number of quantification levels from 16 to 32 or by taking into account boundary curvature is negligible.

Figure 3 (or figure 4 for a color representation) illustrates on two examples the influence of orientation on the best match.

Similarity measure. The experiments on shape similarity have been made with an unique shape representation involving the normalization variant N2 and the variant R2 for the representation of the boundary.

We have tested three of the four similarity measures described in section 3: asymmetric bounded error, asymmetric least error, and symmetric bounded error. These three similarity measures let us consider the alternatives: bounded error vs. least error and asymmetric vs. symmetric.

For each of the similarity measures, there exists a parameter (error bound or match rate) which sets the sensitivity of the similarity measure. There is of course an optimum to find. If the similarity measure is too sensitive, patterns of the same character appear dissimilar, while if it is not sufficiently sensitive, patterns of different characters appear similar. For each similarity measure, we have hence tested different parameter values. The results in table 3 show the existence of following optimal values. With the normalization in scale we have used (48x48), the optimal error bound value was about 5 pixels for asymmetric bounded error. The optimal match rate value seems to lie in the range of 80-90% in the case of asymmetric least error.

We consider now the alternatives: bounded error vs. least error and symmetric vs. assymetric.

- **Bounded error vs. least error**

  Comparing the results of asymmetric bounded error matching with those of asymmetric least error matching let us argue that there is no remarkable difference between bounded error matching and least error matching.

- **Symmetric vs. asymmetric**

  Symmetric bounded error matching shows better recognition scores than asymmetric bounded error matching (table 3). However, there is no evidence that symmetric matching performs better than asymmetric matching since there are more prototypes in the symmetric case than in the asymmetric case. Moreover, since symmetric matching is more computation expensive than asymmetric matching, this last solution is preferable.

Learning prototypes. Comparing table 4 with table 1 shows that learning with rejection of bad prototypes increases the performance of the classifiers. This is particularly emphasized in the case where stroke width normalization is performed (variant P3), since in that case some patterns are so much modified by the processing that they become very similar to patterns of a different character.

6 Conclusions

We have applied methods from visual object recognition to the problem of character recognition. We have experimented those methods on a limited set of character patterns (20000 samples) which were available in separated images. From this experimentation, we make following conclusions, first relative to the importance of shape for character recognition, second relative to the methods we have used for shape representation and shape comparison.

We have applied commonly used normalization operations to the character patterns to reduce shape variability among patterns of the same character. We have observed that normalization has to be performed carefully. In particular, we have observed that the morphological operations (e.g. thinning) involved in stroke width normalization can modify the shape of a character pattern in a way that makes it more similar (at least with the comparison method we have used) to samples of other characters. So, in our opinion, those operations should be avoided.

In our system, the recognition of characters occurs in comparing the pattern boundary of the unknown sample with the one of prototypes. We have investigated how the representation and the comparison of boundary shapes influence the recognition scores.

We have observed that the use of orientation in the boundary representation of character patterns allows to better discriminate patterns of different characters. These observations confirm the need of prefering "local coherent" matches for better discriminating shapes, as mentioned in [Bre92]. Indeed, taking into account the orientation in the representation of the boundary is equivalent to using a similarity measure that takes into account the co-occurrence of neighboring point matches.

We have tested alternatives for the similarity measure: symmetric vs. asymmetric error and bounded vs. least error. The results do not show any clear difference in recognition scores between them. However, least error has an advantage over bounded error since its parameter (match rate) is scale independent. Asymmetric error matching is preferable to symmetric error matching since it is less computation expensive.

One of our motivations in this work was to find methods which are robust to segmentation errors and to situations where non-character patterns (e.g. underlining) interfere. This robustness capability has not been proved yet. The next step in our work is therefore to test the matching methods in the case where several character patterns (or parts of them) are present in an unique image.

Aknowledgements

Made under supervision of Dr. Thomas M. Breuel, this work has been funded by the Swiss National Foundation for Scientific Research under project no. 21-39576.93.

References

[Bre92] Thomas M. Breuel. *Geometric aspects of visual object recognition.* PhD thesis, Massachusetts Institute of Technology, 1992.

[Bre93] Thomas M. Breuel. Recognition of handprinted digits using optimal bounded error matching. In *International Conference on Document Analysis and Retrieval (ICDAR)*, pages 493–496, Tsukuba Science City, Japan, 1993.

[Cas92] Todd A. Cass. Polynomial-time object recognition in the presence of clutter, occlusion, and uncertainty. In *Proceedings Image Understanding Workshop*, pages 693–704, 1992.

[Das91] Belur V. Dasarathy. *Nearest neighbor (NN) norms : NN pattern classification techniques*. IEEE Computer Society Press tutorial. IEEE Computer Society Press, Los Alamitos, California (etc.), 1991.

[Gat72] G.W. Gates. The reduced nearest neighbor rule. *IEEE Transactions on Information Theory*, IT-18(3):431–433, May 1972.

[Gri90] W. Eric L. Grimson. *Object recognition by computer : the role of geometric constraints*. The MIT Press series in artificial intelligence. MIT Press, Cambridge, MA, 1990.

[GW92] M. D. Garris and R. A. Wilkinson. Nist special database 3: Binary images of handwritten segmented characters, 5.25" cd-rom. available from: Standard Reference Data, National Institute of Standards and Technology, 221/A323, Gaithersburg, MD 20889, 1992.

[Har68] P.E. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, IT-14(3):515–516, May 1968.

[Pav82] T. Pavlidis. *Algorithms for graphics and image processing*. Springer, Berlin, Heidelberg, 1982.

[Wil72] D.L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2(4):408–421, July 1972.

Appendix

A Color figures

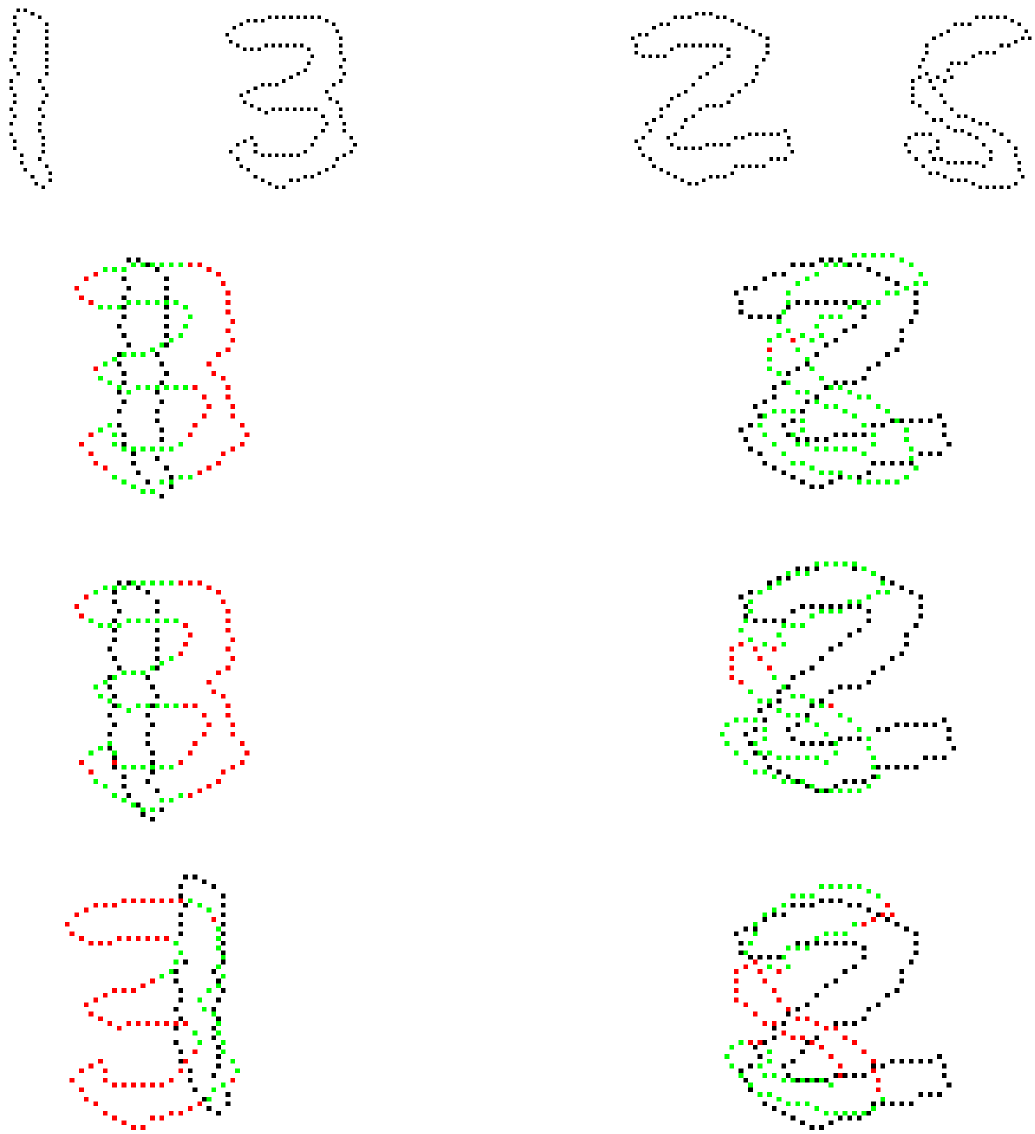

Figure 4: Influence of the boundary orientation on the best match. (Representation of matching: black: image points, green: matching model points, red: non-matching model points.) (a) Matched shapes. In each pair, left: image, right: model. (b) Best match without taking into account the orientation. (c) Best match when the orientation is quantified in 4 levels. (d) Best match when the orientation is quantified in 16 levels.