

# Frame Deformation Energy Matching of On-Line Handwritten Characters

Jakob Sternby

Centre for Mathematical Sciences,  
Sölvegatan 18, Box 118,  
S-221 00, Lund, Sweden  
`jakob@maths.lth.se`

**Abstract.** The coarse to fine search methodology is frequently applied to a wide variety of problems in computer vision. In this paper it is shown that this strategy can be used to enhance the recognition of on-line handwritten characters. Some explicit knowledge about the structure of a handwritten character can be obtained through a structural parameterization. The Frame Deformation Energy matching (FDE) method is a method optimized to include such knowledge in the discrimination process. This paper presents a novel parameterization strategy, the Dijkstra Curve Maximization (DCM) method, for the segments of the structural frame. Since this method distributes points unevenly on each segment, point-to-point matching strategies are not suitable. A new distance measure for these segment-to-segment comparisons have been developed. Experiments have been conducted with various settings for the new FDE on a large data set both with a single model matching scheme and with a  $k$ NN type template matching scheme. The results reveal that the FDE even in an ad hoc implementation is a robust matching method with recognition results well comparing to the existing state-of-the-art methods.

## 1 Introduction

Explicit usage of the structural information inherent to handwritten characters is highly uncommon in state-of-the-art recognition methods today. The tantalizing idea of automatic optimization of all kinds of features by means of statistical methods such as Neural Networks (NN) and/or Hidden Markov Models (HMM) seems to have caused researchers to abandon the more straightforward discrimination methods based on template matching [10]. The most successful of the template matching methods, which still seems to have some followers [13], is the Dynamic Time Warping (DTW) method which improves on static matching by enabling a dynamic match between different dimensions. Under equal training circumstances Hidden Markov Models seems to provide a higher hitrate than DTW [3]. It is probable that HMM can be somewhat less sensitive to the non-linearity of the variations of handwritten data since it models smaller segments of each character with hidden states based on features [6]. However, the Markov characteristic of such systems may also make it difficult to construct features that

can discriminate between some models that have similar sets of hidden states but of varying durations [1]. Recently Bahlmann et al. [2] have shown how DTW and HMM relate to each other.

This paper presents considerable enhancements to the new template matching method called Frame Deformation Energy (FDE) matching. The method is based on a structural parameterization obtained by extracting a structurally defined subset of points called *core points* in two stages. A outer *core point frame* is obtained as the set of local extreme points in the direction orthogonal to the writing direction and then a fixed number of points is added to each such segment. This paper introduces a new method, here called the Dijkstra Curve Maximization (DCM) method to extract the fixed number of points on each segment of the core point frame. Previously the extraction of such sets of *interesting* points have been performed mainly for segmentation of cursive script [9, 11]. It has previously been shown that a structural parameterization enhances the recognition performance for Euclidean matching. In this paper, implementation of a new curve segment matching strategy developed for the DCM parameterization method produces the highest recognition results obtained for the FDE strategy so far. In particular the method seems robust and delivers very reliable results for top two candidates.

## 2 Structural Reparameterization with Core Points

In the field of handwriting recognition (HWR) most of the techniques for extracting the most *interesting* points on a curve have been developed for segmentation of cursive script [8]. It has previously been shown that this strategy can be used also to decompose isolated characters into smaller segments of simple curves. The achievement of such a decomposition is that it enables a description of the non-linear variations of handwritten data into smaller less non-linear parts. In this paper a very simple yet effective method of dividing samples into segments has been studied. The extreme points in the direction orthogonal to the writing direction (normally  $y$ ) define a subset here called the **core point frame**. Once such a method for fixing this greater structure of segments has been chosen, the problem of fixing a parameterization for the intermittent points can be addressed. Independently of the method chosen for accomplishing this the new parameterization of a core point frame with intermittent points, will be called the *core points*  $C(X)$  of a sample  $X$ .

### 2.1 Choosing a Fixed Number of Intermittent Points

The most basic approach for picking middle points is to sample each segment in the core point frame by arclength. The weakness of this method is that segments may require a varied number of points in order to be described correctly. Of course one can choose to pick many points but this also effects the time complexity. This rudimentary method is here referred to as the *Segment Arclength* (SA) method. Aiming at enabling an upper bound for the required number of points on each segment, methods that try to approximate the segment by a few

number of points have also been investigated below. For handwritten characters a study of the various allographs of the script reveals that there is a maximum of significant points on the segments. For the Latin-1 character set it has been empirically observed that this number is three on any individual segment. For this reason a fixed number of three points have been placed on each segment for both methods of choosing points described below.

Since each segment of the core point frame should be described as well as possible a method that chooses the most *interesting* points on this curve segment is needed. As stated previously such methods have been used in the past for cursive script segmentation [8]. A method inspired by these ideas have been tried and will be referred to as the *Curvature* (C) method. Instead of just spacing the points evenly on the segment as one would if one were to use conventional arclength parameterization, a search for points that have a significant curvature is performed first. This search is done recursively by picking any point with a diversion from the line between the start and end point that exceeds a threshold. If the number of curvature points chosen in this manner is less than the fixed number of points per segment, points are added in a manner that spaces them as evenly as possible.

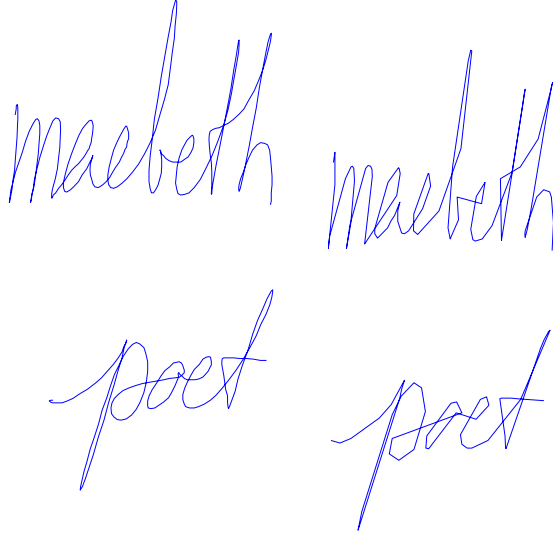
Choosing the  $n$  points on a piece of a curve that *best* approximates it is a problem that has been thoroughly studied in the field of discrete geometry [5]. There it is common to refer to the best approximation the so called min- $\epsilon$  solution in terms of the uniform metric i.e. the  $n$  points on the (discrete) curve  $X = \{x_i\}_{i=0}^m$  resulting in the smallest maximum distance between the removed points and the resulting piecewise linear curve.

Below we present a fundamentally different approach for finding an approximation of an  $m$ -polygon with  $n$  points. With a Euclidean metric the subset that maximizes the linear segment function is:

$$(x_{p_1}, \dots, x_{p_n}) = \underset{(p_1, \dots, p_n) \subset (1, \dots, m)}{\operatorname{argmax}} \sum_{i=1}^n \|x_{p_i} - x_{p_{i-1}}\|. \quad (1)$$

We call the method of finding  $(x_{p_1}, \dots, x_{p_n})$  on a  $m$ -polygon according to (1), the *Dijkstra Curve Maximization* (DCM) method since the set can be found by means of a modified version of the Dijkstras algorithm. This is a clear strategy with the appealing characteristic that it is independent of threshold values and other tuning parameters indispensable for the C-method described above.

One can easily show that the DCM and the min- $\epsilon$  curve approximations are similar under some circumstances. One equally easily realizes that there are many cases when they differ. One interesting example are the respective solutions of the min- $\epsilon$  approach and the DCM to picking one point on a sinus curve on the interval  $[0, 2\pi]$ . Here the DCM has two optimal solutions lying close to the respective extreme points, whereas the min- $\epsilon$  approach will choose the middle point. In particular one easily observes that their behavior differ greatly when the number  $n$  is less than the number of local extreme points on the curve. The DCM gets many solutions in this case, all aiming at choosing one of the prominent features of the curve whereas the min- $\epsilon$  solution gives the mean path.



**Fig. 1.** Two cursive words with the original sampling to the left and the core point reparameterized words to the right. Here the DCM technique of Section 2.1 is used to find the intermittent points in the core point frame.

Examples of the extracting core points with DCM on some connected character sequences are shown in Figure 1. Apparently the DCM provides a nice and smooth curve.

### 3 The Frame Deformation Energy Model

One of the limitations of template matching techniques such as DTW lies in the fact that the normalization is global. Even though DTW is successful at enabling matching between samples of varying dimension it is still dependent on normalization and thereby also sensitive to non-linear variations. In other words a handwritten sample  $X$  is in general not only the result of global transformation of the reference template but also of individual transformations of each segment of the core point frame.

These facts motivate the search for a method that tries to find both the local segment transformations as well as calculate the resulting distance to the transformed segments. In short the matching process of a sample  $X = \{x_i\}$  to a template (prototype)  $P = \{p_i\}$  can be divided into three stages:

1. Find the best global linear transformation  $A_P = \operatorname{argmin}_L \|P - LX\|$
2. Find the frame bending transformation  $B_P, p_i = B_P(x_i), \forall x_i, p_i$  in their respective core point frames
3. Calculate a distance value dependent on the transformations  $A_P, B_P$  and the remaining difference  $P - B_P(A_P(X))$

Analysis of samples of on-line handwritten characters clearly show that in-class global transformations of handwritten characters are constrained linear transformations. There are no reflections and only limited rotation and skew. The frame bending transformations are defined as the transformations identifying the corresponding core point frames.

### 3.1 Distance Calculation

A popular method to achieve exact transformations between templates in pattern recognition is thin-plate splines. Although there have been successful applications of thin-plate splines to the character recognition problem in the past [4] it has obvious shortcomings. The main problem is that common variations in handwritten patterns involves points on the extension of a line being distributed on either side of the line causing folding of the thin-plate spline. To counter this problem a much simpler energy model for the frame is introduced. Let each segment be modelled by a robust spring that is equally sensitive to compression and prolongation and let the connection between each segment be a coiled spring that is equally sensitive to torque in both directions. The most simple distance measure for the bending energy of the frame between a sample  $X$  and a template  $P$  of  $m$  core points, with frames  $F_X = (f_X(1), \dots, f_X(n))$  to  $F_P = (f_P(1), \dots, f_P(n))$  is then given by

$$E_B(X, P) = \sum_{i=2}^{n-1} k_x \left( \frac{\|f_X(i+1) - f_X(i)\|}{\|f_X(i) - f_X(i-1)\|} - \frac{\|f_P(i+1) - f_P(i)\|}{\|f_P(i) - f_P(i-1)\|} \right)^2 + \sum_{i=2}^{n-1} k_a \left( \frac{\theta_i^{F_X} - \theta_i^{F_P}}{\pi} \right)^2, \quad (2)$$

where  $k_x, k_a$  are the spring constants for the segment springs and the inter-segment springs respectively. The intermittent frame segment angles  $\theta_i^{F_X}$  are defined as  $\theta_i^{F_X} = \arg(f_X(i+1) - f_X(i), f_X(i) - f_X(i-1))$ . For notational convenience a modula  $\pi$  for the angle retrieved with the arg operator is implied. As described in the previous section the result of the bent frame  $B_{F_P}(F_X)$  is that  $\|B_{F_P}(F_X) - F_P\| = 0$ , however, the intermittent points are just Bookstein coordinates in their respective surrounding segment and will generally not be identical. To model their distance, the different models for selecting the intermittent points presented in Section 2.1, have been evaluated with various distance measures.

**Distance Measures for Intermittent Core Points.** From an implementation point of view the most simple way to model the energy of transforming points from one curve to the other is to find a model that corresponds to the Euclidean measure. This is achieved by imagining that each of the intermittent points are attached to the corresponding point in the sample being matched by elastic strings. This induces an energy measure for the intermittent points after matching the frame

$$E_M^{\text{Euc}}(B_P(A_P(X)), P) = \sum_{j=1}^m k_j \|B_P(A_P(x_j)) - p_j\|^2, \quad (3)$$

where  $k_j$  is the spring constant for the string attached to core point  $j$  in  $P$ . Evidently setting  $k_j = 1, j = 1, \dots, m$  gives the square Euclidean distance of the bent frame transformed sample  $\|B_P(A_P(X)) - P\|^2$ . This should only be suitable when there is a strong correspondence between points on the segments so it should not be used with selection methods that distributes points unevenly such as the DCM. Even though most of the parameterizational differences should have been depleted by the core point reparameterization one could also try a DTW measure on the intermittent points. However this has not been tested in this paper.

The DCM method presented in Section 2.1 is not suitable to use with either of these measures since points may be distributed anywhere on the curve segment. Instead some kind of curve comparison measure that is independent of the parameterization is needed. To accomplish a new distance function the *Dijkstra Curve* distance  $E_M^{DC}$ , consisting of two individual components is proposed. The first component is a Point-to-curve distance function  $d_{PC}(X, P)$ , used for matching the intermittent core points of one curve to some line segment in the other curve. It is a DTW method with transitions  $(1, 0), (1, 1)$  solving the problem of finding the correspondence function  $\Phi(k) = (\phi_x(k), \phi_p(k)), k = 1, \dots, m$  that optimizes

$$d_{PC}(X, P) = \min_{\Phi} \sum_{i=1}^m \mathbf{g}_{PL}(x_{\phi_x(i)}, \mathbf{p}_{\phi_p(i)}), \quad (4)$$

where  $\Phi(1) = (1, k_1), \Phi(m) = (m, k_m), k_i \leq k_{i+1}, \forall i \in (1, \dots, m)$ . Here  $\mathbf{g}_{PL}(x_k, \mathbf{p}_j)$  denotes the distance between point  $x_k$  and the line segment  $\mathbf{p}_j = (p_{j-1}, p_j)$ . Let  $l_j$  be the line passing through line segment  $\mathbf{p}_j$  and let  $x_{l_j}^\perp$  be the orthogonal projection of point  $x$  onto  $l_j$ , then

$$\mathbf{g}_{PL}(x_k, \mathbf{p}_j) = \begin{cases} \min(\|x_k - p_{j-1}\|, \|x_k - p_j\|), & \text{if } x_{(l_j, k)}^\perp \notin \mathbf{p}_j, \\ \|x_k - x_{(l_j, k)}^\perp\|, & \text{otherwise.} \end{cases} \quad (5)$$

The  $d_{PC}$  distance function from (4) is found through the following recursive algorithm

**Algorithm 1.**

```

 $D_j(1) := \mathbf{g}_{PL}(1, j), j = 1, \dots, m + 1$ 
for  $j = 1, \dots, m + 1$  do
  for  $k = 2, \dots, m$  do
     $D_j(k) := \mathbf{g}_{PL}(k, j) + \min(D_{j-1}(k-1), D_j(k-1))$ 
  end for
end for
 $d_{PC} := \operatorname{argmin}_j D_j(m)$ 

```

The second is a fuzzy DTW distance function computing a distance between the angles of consecutive core points. Denote the normalized angles of consecutive points in  $X$  and  $P$  by  $\{\theta_i^X\}_{i=1}^m$  and  $\{\theta_i^P\}_{i=1}^m$  respectively then the angular distance function corresponding to (5) is defined as

$$\mathbf{g}_A(x_i, p_j) = (\kappa_\theta(\theta_i^X - \theta_j^P)/\pi)^2 + (\kappa_\lambda(\lambda_i^X - \lambda_j^P))^2, \quad (6)$$

where  $\lambda_i^X = \frac{(x_i - x_1)^T (x_m - x_1)}{\|x_m - x_1\|}$  is the location of each angle in terms of the baseline. It has been found that results are improved if a certain fuzziness is added to the angular distance function. To enable this the definition of  $\mathbf{g}_A$  in (6) is extended to treat matching the angle on one curve segment to a flat segment on the other defined as a parameter  $t \in [0, 1]$  between two points as

$$\mathbf{g}_A(x_i, p_j + t) = (\kappa_\theta(\theta_i^X - \pi)/\pi)^2 + (\kappa_\lambda(\lambda_i^X - ((1-t)\lambda_j^P - t\lambda_{j+1}^P)))^2. \quad (7)$$

The recursive update rule for the algorithm finding the best DTW distance corresponding to the inner statement of Algorithm 1 in this case becomes

$$D_j^A(k) = \min \begin{cases} \mathbf{g}_A(x_j, p_k) + D_{j-1}^A(k) \\ 2\mathbf{g}_A(x_j, p_k) + D_{j-1}^A(k-1) \\ \mathbf{g}_A(x_j, p_k) + D_j^A(k-1) \\ 2\mathbf{g}_A(x_j, p_k) + \min_{r \in (1, \dots, j-1)} D_{j-r}^A(k-1) + \sum_{i=1}^r \mathbf{g}_A(x_{j-r+i}, p_{k-1} + i/(r+1)) \\ 2\mathbf{g}_A(x_j, p_k) + \min_{r \in (1, \dots, j-1)} D_{j-1}^A(k-r) + \sum_{i=1}^r \mathbf{g}_A(x_{j-1} + i/(r+1), p_{k-r+i}) \end{cases} \quad (8)$$

According to (8) the total angular distance will be  $d_A(X, P) = D_m^A(m)$ . Now the *Dijkstra Curve* distance  $E_M^{DC}$  can be written as

$$E_M^{DC} = d_{PC}(X, P) + d_{PC}(P, X) + d_A(X, P). \quad (9)$$

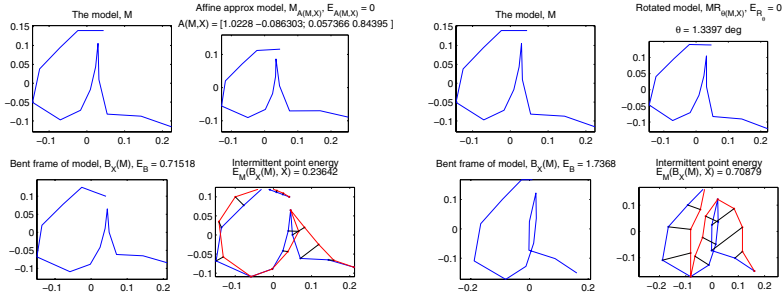
**The Frame Deformation Energy Distance (FDE).** Above we have described methods to account for the two steps of frame bending and curve segment comparison. It is not entirely obvious how to fit a suitable penalization of global transformations into this. On one hand global transformations are natural variations of isolated handwritten character data and on the other some kind of penalization is necessary since the energy  $E_B(X, P)$  of (2) is invariant to global rotation. One could try global transformations of rotation and of the triple scale, rotation and skew. For these parameters of scale  $(\lambda_x, \lambda_y)$ , rotation  $\theta$  and skew  $\eta$  one can define a distance function similar to that of the bending energy by setting

$$E_{RSS}(X, P) = k_\lambda \left( \left( \frac{\lambda_x}{\lambda_y} \right)^{t_\lambda} - 1 \right)^2 + k_\theta \left( \frac{\text{mod}(\theta, \pi)}{\pi} \right)^2 + k_\eta \left( \frac{\text{mod}(\eta, \pi)}{\pi} \right)^2, \quad (10)$$

where  $t_\lambda$  is 1  $\lambda_x < \lambda_y$  and  $-1$  otherwise. However in the experiments of this paper only the rotational component in (10) denoted by  $E_R(X, P)$  has been used.

Combining the distance components for global transformation, bending energy and curve segment into a weighted sum produces the following distance functions:

$$D_R^{\text{method}}(X, P) = w_A E_R(X, P) + w_{BE} E_B(A(X), P) + w_M E_M^{\text{method}}(A(B(X)), P) \quad (11)$$



**Fig. 2.** One in-class and one inter-class example of a mean model matched to a sample according to the scheme of Section 3

Matching with the distance function in (11) will be referred to as Frame Deformation Energy Matching (FDE). The optimization problem of finding the optimal set of parameters  $\{w\}, \{k\}$  could probably be solved by some Support Vector Machine inspired method but it is interesting enough to receive full attention in a separate paper.

## 4 Experiments

The recognition experiments in this paper have been conducted on the MIT single character database [7]. The set of 37161 samples from the  $w$  section (single characters from words) was selected as test set while the 2825 samples from the  $l$  section was selected as the training set.

For single models the new version of the template matching method FDE was compared to DTW as well as a Gaussian Active Shape Model (AS) such as it is described in thesis [12]. For the FDE and DTW methods a single model was constructed for each allograph as the mean of the samples belonging to that allograph class. For AS one model was built for each allograph class. The FDE was implemented with the  $D_R^{DC}$  measure in the most simple way by manually setting all the spring constants in (2), (3) as well as all of the weights in (11) to suitable values. Even with this simple ad hoc setting the results of template matching

**Table 1.** Results of  $k$ -NN matching on the MIT database. The different methods for selecting intermittent points are shown as CP-C (Curvature) and CP-DCM (Dijkstra Curve Maximization). Where available the recognition result for best-two candidates is also displayed to the right.

k-Distance Measure	Arclength	CP-C	CP-DCM
1-Euclidean	86.3%	89.6 / 91.0 %	-
1-DTW	91,3%	89,5 %	-
1- $D_{AFE}^{DC}$	-	-	88.2 / <b>94.7 %</b>



**Table 2.** Results of single model matching on the MIT database

Method	Original data	CP-DCM
AS	77.2 %	82.4 %
DTW-mean	<b>89.6 / 90.7 %</b>	-
$D_R^{DC}$ -mean	-	82.4 / 90.3 %

with single models performs as well as DTW on the original parameterization for top-two candidates Table 2 making it a promising method well worth further research. Especially since the results of the top two candidates when running the same FDE method on multiple models for each class as seen in Table 1. Although only two methods have been tried for second candidates in the 1-NN matching the significant increase in recognition accuracy for the FDE method indicates a strong potential for improvement even at the single-model stage.

## 5 Discussion and Conclusions

This paper presents a new parameterization and a distance measure for use with the novel Frame Deformation Energy (FDE) matching method. The main objective of the new method is to try to address the weak points of a global matching scheme by dividing the matching process of a handwritten character into natural segments called a core point frame. It has been shown that the new strategy provides a robust matching method with results comparable to state-of-the-art template matching methods such as DTW for top two candidates even in an ad hoc implementation of manually setting the spring constants of the energy model.

Further research will include automatic methods for optimizing spring constants for different allographs as well as hybrid methods for a final optimal recognition rate. It might be even more efficient to view the problem in a probabilistic way by determining the class  $C$  with a model  $M_C$  that has the highest probability  $P(C|A_X, B_X, B_X(A_X(M_C)) - X)$ . Since the novel FDE technique already at this early stage has shown a promising capacity for computationally efficient single models it will be especially useful in on-line cursive script systems based on segmentation graphs.

## References

1. J. Andersson. Hidden markov model based handwriting recognition. Master's thesis, Dept. of Mathematics, Lund Institute of Technology, Sweden, 2002.
2. C. Bahlmann and H. Burkhardt. The writer independent online handwriting recognition system *frog on hand* and cluster generative statistical dynamic time warping. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(3):299–310, March 2004.
3. E. J. Bellegarda, J. R. Bellegarda, D. Nahamoo, and K. Nathan. A fast statistical mixture algorithm for on-line handwriting recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(12):1227–1233, 1994.

4. S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(24):509–522, 2002.
5. M. T. Goodrich. Efficient piecewise-linear function approximation using the uniform metric: (preliminary version). In *SCG '94: Proceedings of the tenth annual symposium on Computational geometry*, pages 322–331, New York, NY, USA, 1994. ACM Press.
6. J. Hu, S.G. Lim, and M. K. Brown. Writer independent on-line handwriting recognition using an hmm approach. *Pattern Recognition*, (33):133–147, 2000.
7. R. Kassel. The MIT on-line character database. <ftp://lightning.lcs.mit.edu/pub/handwriting/mit.tar.Z>.
8. X. Li, M. Parizeau, and R. Plamondon. Segmentation and reconstruction of on-line handwritten scripts. *Pattern Recognition*, 31(6):675–684, 1998.
9. M. Parizeau and R. Plamondon. A handwriting model for syntactic recognition of cursive script. In *Proc. 11th International Conference on Pattern Recognition*, volume II, pages 308–312, August 31 to September 3 1992.
10. R. Plamondon and S. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(1):63–84, January 2000.
11. C. De Stefano, M. Garutto, and A. Marcelli. A saliency-based multiscale method for on-line cursive handwriting shape description. In *Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition*, pages 124–129, 2004.
12. J. Sternby. Core points - variable and reduced parameterization for symbol recognition. Technical report, 2005. Licentiate Thesis in Mathematical Sciences 2005:7.
13. V. Vuori. Adaptation in on-line recognition of handwriting. Master's thesis, Helsinki University of Technology, 1999.