

Fast Handwritten Recognition Using Continuous Distance Transformation

Joaquim Arlandis and Juan-Carlos Perez-Cortes

Universitat Politècnica de València, Camí de Vera s/n 46022 València, Spain
{jarlandi,jcperez}@disca.upv.es

Abstract. The Continuous Distance Transformation (CDT) used in conjunction with a k -NN classifier has been shown to provide good results in the task of handwriting recognition [1]. Unfortunately, efficient techniques such as kd -tree search methods cannot be directly used in the case of certain dissimilarity measures like the CDT-based distance functions. In order to avoid exhaustive search, a simple methodology which combines kd -trees for fast search and Continuous Distance Transformation for fine classification, is presented. The experimental results obtained show that the recognition rates achieved have no significant differences with those found using an exhaustive CDT-based classification, with a very important temporal cost reduction.

1 Introduction

Statistical non-parametric methods, such as k -nearest neighbor classifiers (k -NN) provide very good results on many pattern recognition tasks (e.g.[11, 4]). One of the basic requirements for these methods to obtain good performances, however, is the use of a very large database of labeled prototypes. In some tasks, like handwritten character recognition, collecting a large number of examples is not as hard as in other applications, but searching through the whole database to find the nearest objects to a test image is time-consuming, and has to be done for every character in a document. This has been a recurring argument against the use of k -NN classifiers for this task, since a character recognizer is supposed to carry out many classifications per second on a moderately powerful machine to be useful and competitive. On other hand, the use of complex distances, aggravates the problem of the speed in the classification phase, where a distance computation for each test character to every prototype in the training set must be done.

In order to avoid searching over the whole training set, kd -tree data structures can be used. A kd -tree is a binary tree which allows fast and approximate search in large databases, providing results very similar to exhaustive search. Unfortunately, only certain metrics, including L -norms, can be naturally used as a distance function by traditional kd -tree search algorithms.

In this work, the problem of the computational complexity reduction associated to a k -NN classifier using complex distance functions has been approached

in a simple way: find a number k' of nearest neighbors using a fast approximate kd -tree search in a first feature space and, then, use a more complex or qualified dissimilarity measure applying exhaustive k nearest neighbors search to the subset of k' neighbors selected by the kd -tree. This results in a huge reduction on the the number of heavy computations needed to be performed. CDT-based features and distance functions have been selected in this paper because of the good results reported in previous works [1] for the particular task of handwriting recognition.

2 The Continuous Distance Transformation

Obtaining feature maps from images, where the distance relationships among their pixels are taken into account is the goal of a well-known technique usually referred to as *Distance Transformation* or DT [10]. The Distance Transformation is traditionally defined as an operation that transforms a binary image consisting of feature and non-feature pixels into a distance map, where all non-feature pixels have a value corresponding to the distance (any suitable distance function on the plane) to the nearest feature pixel [6]. Unfortunately, binarization is a necessary step in order to compute the classical Distance Transforms from continuous-valued images, causing a loss of information.

Recently, a generalization of the DT, the Continuous Distance Transformation (CDT), has been presented as a technique to compute distance maps from continuous-valued images [2]. Applicable to gray-level images, the CDT technique avoids binarization process and make use of the whole information content of the original range of representation,

Taking the definition of Distance Transformation as a basis, an item (i, j) of a “Distance Map to the Nearest White Pixel” holds the distance from pixel (i, j) on the image to the nearest white pixel. Note that this value can be interpreted as the number of fringes expanded from (i, j) until the first fringe holding a white pixel is reached, where a “fringe” is defined as the set of pixels that are at the same distance of (i, j) .

A parallelism between a distance map of binary images and one whose pixel values are defined in the gray-scale domain $[0..MaxBright]$ implies the replacement of the “white pixel” concept by the “maximum bright value” and actions as “find the nearest white pixel” by “accumulate a maximum bright value on an expanding neighborhood”. Moreover, the value of an item on the continuous distance map is a function of the pixel value itself, as well as, of the number of fringes expanded until an accumulated bright value reaches a threshold according to a certain criteria of bright value accumulation, which is applied to the pixels belonging to each fringe analyzed. Then, the concept of “distance to the nearest white pixel” is substituted by the concept of “distance from a pixel to the limit of their minimum area of brightness saturation”.

Two types of CDT-based maps can be defined: Continuous Distance Map to Direct Scale Saturation or Θ^D and, Continuous Distance Map to Reverse Scale Saturation or Θ^R depending on if a maximum value of bright intensity or

a maximum value of reverse bright intensity is accumulated, respectively. Both maps provide distinct information about a point and its surrounding area. In [2], detailed descriptions of these concepts are presented. Given an image, either a Θ^D map or a Θ^R map are more or less descriptive depending on its brightness distribution. The cost of a CDT map computation is in $\Omega(m^2 \times n^2)$ for an image of $n \times m$ pixels, but, in practice, it is much lower.

Several distance and dissimilarity measures based on the Continuous Distance Transformation can be used to take advantage of the full possibilities of the representation obtained. These measures are collected in three generic, families including several distance functions each [2]:

- Continuous Fringe Distance measures including Fringe Distance using Θ^D maps (FDD), Fringe Distance using Θ^R maps (FDR) and Symmetrical Fringe Distance (SFD) which uses both Θ^R and Θ^D maps.
- Continuous Pixel Distance measures or PDL $_p$, which use the L -norm metric in its computation along with Θ^R and Θ^D maps¹.
- L -norm between CDT maps. Three sub-families can be computed depending on the CDT maps taken: the L $_p$ D metrics if the L -norm of Θ^D maps is computed; the L $_p$ R metrics if the Θ^R maps are used instead; and the L $_p$ DR metrics if both maps are employed.

3 Fast Approximate Search of k Nearest Neighbors Using kd -Trees

The nearest neighbor search problem can be formulated in several distinct domains: from Euclidean vector spaces to (pseudo)metric spaces. Most algorithms intended for vector spaces are directly based on the construction of a data structure known as kd -tree [7, 5].

A kd -tree is a binary tree where each node represents a region in a k -dimensional space. Each internal node also contains a hyper-plane (a linear subspace of dimension $k - 1$) dividing the region into two disjoint sub-regions, each inherited by one of its children. Most of the trees used in the context of our problem divide the regions according to the points that lay in them.

In many cases, an absolute guarantee of finding the real nearest neighbor of the test point is not necessary. In this sense, a number of algorithms of approximate nearest neighbor search have been proposed [3, 8].

In a kd -tree, the search of the nearest neighbor of a test point is performed starting from the root, which represents the whole space, and choosing at each node the sub-tree that represents the region of the space containing the test point. When a leaf is reached, an exhaustive search of the b prototypes residing in the associated region is performed. Unfortunately, the process is not complete at this point, since it is possible that among the regions defined by the initial partition, the one containing the test point be not the one containing the nearest prototype. It is easy to determine if this can happen in a given configuration,

¹ In previous works, this family of measures was named GPD $_p$.

in which case the algorithm backtracks as many times as necessary until all the regions that can hold a prototype nearer to the test point have been checked.

If a guaranteed exact solution is not needed, the backtracking process can be aborted as soon as a certain criterion is met by the current best solution. In [3], the concept of $(1 + \epsilon)$ -approximate nearest neighbor query is introduced, along with a new data structure, the BBD-tree. A point p is a $(1 + \epsilon)$ -approximate nearest neighbor of q if the distance from p to q is less than $1 + \epsilon$ times the distance from p to its nearest neighbor.

4 Combining kd -Trees and the Continuous Distance Transformation

A simple methodology combining kd -trees and Continuous Distance Transformation for handwriting recognition is proposed. In a first step, fast search using kd -trees is applied to a test observation in order to get a number k' of nearest prototypes. Secondly, an exhaustive search of the k nearest neighbors, $k < k'$, among the k' pre-selected prototypes using specific features and distance functions is carried out to assess better performances.

For handwritten character classification we tested CDT-based distance functions due to its good results reported in previous works [1]. The computational cost of this combined methodology is significantly lower than that of exhaustive search over the whole training set. The computational cost of the method will depend on k' , the number of neighbors pre-selected, because it will determine the number of computations of the second step.

5 Experiments

The main goal of the experiments was to show if the proposed technique performs similarly to exhaustive search method in the task of handwritten character recognition, and quantify the computation time improvements obtained.

In this context, several configurations of the classification parameters can be identified. In order to analyze the behavior of the system in each of those possible settings, three experimental phases were planned: *a)* Selection of the features to use in the kd -tree pre-classifier; *b)* Test of the performance of different CDT variations and *c)* test for different values of k and k' .

5.1 Datasets, Preprocess, and Algorithms

The well-known NIST Special Database 3 (SD3) contains a large number of isolated handwritten characters: lower-case, upper-case letters, and digits. In this work, the 44951 upper-case letters from the SD3 were chosen for the experiments. The characters are stored as 128×128 binary pixels images, segmented and labeled by hand. The database was split into two sets used for error estimation: the first 39941 upper-case letters for training, and the last 5009 for test. No writer appeared in both sets.

Table 1. Results of phase *a*). Error rates of the three methods using different feature maps, for $k'=100$, $k=4$, PDL_3 distance function, and $\epsilon=1.5$.

Method \ Map	Image	Θ^D	Θ^R
<i>kd</i> -tree	6.05	8.66	4.99
<i>kd</i> -tree & CDT	3.87	4.07	3.85
Exhaustive CDT		3.85	

To obtain a usable representation of the images in a lower dimensional space, common resampling and normalizing procedures were applied. These techniques generate gray-level images that keep most of the original information. Thus, the character images were sub-sampled from 128×128 binary pixels into 28×28 gray value by first computing the minimum inclusion box of each character, keeping the original aspect ratio, and then accumulating into each of the 28×28 divisions the area occupied by black pixels to obtain a continuous value between 0 and 1.

The Continuous Distance Transformation was applied to the subsampled images, obtaining the CDT maps. Some parameters that have influence in this process have been fixed to present comparative results: the maximum number of fringe expansions was set to 3; the fringe value function chosen was “the maximum pixel value on the fringe”; and, the L_∞ , or chessboard distance, was the metric on the plane used for computing the pixels belonging to each fringe².

In order to make use of the *kd*-tree, Principal Component Analysis (PCA) was performed on the representations to reduce its dimensionality to 40 [9]. The approximate nearest neighbor algorithm used was based on [3].

5.2 Phase *a*)

In the first phase, the main goal was to compare performances using the following three classification methodologies: 1) Approximate k -NN search using *kd*-trees with $k=4$; 2) Approximate k' nearest neighbors search using *kd*-trees with $k'=100$ followed by exhaustive k nearest neighbors search with $k=4$ using a CDT-based measure over the 100 pre-selected neighbors; and 3) Exhaustive k nearest neighbors using a CDT-based measure with $k=4$ over the whole of the training set. In the *kd*-tree classifier, the image, and the Θ^D and Θ^R maps were tested as feature vectors.

The approximation parameter ϵ used for approximate search in *kd*-trees was 1.5. It has been shown that this value reduces the computation time without affecting recognition rates in other related experimental contexts [9]. A common dissimilarity measure, PDL_3 , was chosen for tests which involve CDT evaluations. The results are shown in Table 1 as error rates at zero rejection.

Notice the excellent error rate of 4.99% obtained from the *kd*-tree technique using the Θ^R map as features, and the 3.85% error rate obtained from our

² Some tools and functions in ANSI-C related to the Continuous Distance Transformation can be found at <http://tafetan.disca.upv.es/sw/cdt>

Table 2. Results of phase *b*). Error rates for CDT-based measures using the combined methodology for (if applied), $k'=100$, $k=4$, and $\epsilon=1.5$. The kd -tree features were the Θ^R map.

Distance \ p	1	2	3	4	5	6	7	8	9	10	11	12
L_pD	9.42	8.64	8.48	9.00	9.32							
L_pR	5.77	4.83	4.51	4.39	4.29	4.11	4.19	4.23	4.31	4.39		
L_pDR	6.19	4.97	4.39	4.23	4.07	4.19	4.13	4.01	4.05	4.17	4.15	4.09
FDD							7.77					
FDR							4.83					
SFD							8.80					
PDL_p	5.39	4.13	3.85	3.81	3.73	3.75	3.89	3.85	3.77	3.91		

combined methodology. Both rates improve significantly the error rate obtained from kd -trees when it uses images map as features.

5.3 Phase *b*)

According to the results of the previous phase, the combined kd -tree and CDT methodology was chosen to be analyzed in more detail. Thus, in order to get the best CDT-based measures using our methodology in the task of handwriting recognition, tests over all CDT families of measures were done. For those measures having a p potency (related to the L -norm of the CDT maps), a range of values of p between 1 and 12 were tested. For the Fringe Distance measure family, the three existing functions were used. The results presented in Table 2 show that the Pixel Distance family gives the best performance with very low error rates, followed by the L_pDR and L_pR metrics.

5.4 Phase *c*)

This phase is intended to analyze the combined kd -tree and CDT system performance in function of the number of pre-selected nearest neighbors k' . Several tests using PDL_5 distance function for an exhaustive search through a number of pre-selected prototypes between 25 and 500 were performed. For each k' value tested, error rates for several k values between 1 and 7 are provided too. The results are shown in Figure 1.

For a given value of k , the performance of the combined method approaches the exhaustive CDT rates as k' increases. In practice, for a moderately large k' the system performs as the exhaustive search. In the context of the experiments presented, good results are achieved from around 75 neighbors. Nevertheless, this is expected to be a function of the size of the training set, growing as the size of the training set increases.

The recognition times per character (in a Pentium III, 800 MHz.) for the three methodologies presented were: 3.17 ms. using the kd -tree search, 7.78 ms. using kd -trees and CDT with $k'=100$, and, 287.5 ms. using exhaustive CDT

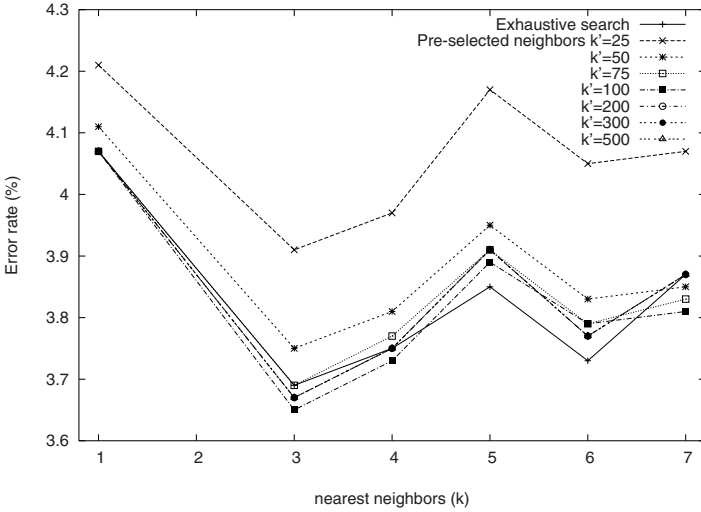


Fig. 1. Results of phase *c*). For values of $k' \geq 75$, no difference with exhaustive search is found.

Table 3. Error rates of the three methods for $k'=100$, $k=3$, and $\epsilon=1.5$.

Method \ Distance	L ₆ R	L ₉ DR	FDR	PDL ₅
Exhaustive CDT	4.17	3.93	5.05	3.69 (287 ms/char)
kd -tree (Θ^R) & CDT	4.17	3.93	5.01	3.65 (7.78 ms/char)
kd -tree (image)	6.05 (3.69 ms/char)			
kd -tree (Θ^R)	4.99 (3.17 ms/char)			

search. Further, the computation times for different values of k' in the combined methodology were directly proportional to the k' value used, and perfectly competitive in a real application.

5.5 Summary

In Table 3, a summary of results is shown. Error rates are presented for the best measure from each CDT-based family, obtained in phase *b*) -PDL₅, L₉DR, L₆R and FDR, excluding the L_pD distances- using a k value of 3 (the best one from phase *c*), and a value of $k' = 100$. Results for the kd -tree technique using image and Θ^R features are also shown. A significant reduction of 1.06% can be achieved using the Θ^R map as input features to the kd -tree instead of directly the image features (PCA to 40-D is always applied as the last feature extraction step). The error reduction reaches 2.4% when the combined methodology is applied using the Pixel distance with $p=5$.

As shown in Figure 1, values of k' lower than 100 can be used when a higher recognition speed is needed, at the expense of a small increase of the error rate.

6 Conclusions

A combined methodology using *kd*-trees and the Continuous Distance Transformation is presented. In a first step, fast search using *kd*-trees is applied to each observation in order to get a selected number of nearest prototypes. Then, exhaustive search among the pre-selected prototypes using more complex measures, such as the CDT-based ones, is carried out to refine the result. The error rates obtained are equivalent to those from exhaustive search on the whole training set. The execution times reported are an order of magnitude lower than those of exhaustive search and only moderately higher than those of approximate *kd*-tree search methods, with a significantly lower error rates that clearly compensate for this small cost increase in most cases.

References

- [1] Arlandis, J., Perez-Cortes, J.C., Llobet, R.: Handwritten Character Recognition Using Continuous Distance Transformation. Proceedings of the 15th. International Conference on Pattern Recognition **1** (2000) 940–943
- [2] Arlandis, J., Perez-Cortes, J.C.: The Continuous Distance Transformation: A Generalization of the Distance Transformation for Continuous-valued Images. Pattern Recognition and Applications **56** (2000) 89–98
- [3] Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.: An optimal algorithm for approximate nearest neighbor searching. Journal of the ACM **45** (1998) 891–923
- [4] Beis, J.S., Lowe, D.G.: Indexing without Invariants in 3D Object Recognition. IEEE Trans. on PAMI **21(10)** (1999) 1000–1015
- [5] Bentley, J.L., Weide, B.W., Yao, A.C.: Optimal Expected Time Algorithms for Closest Point Problems. CM Trans. Math. Software **6** (563–580)
- [6] Borgefors, C.: A New Distance Transformation Approximating the Euclidean Distance. Proceedings of the 8th. International Conference on Pattern Recognition **1** (1986) 336–338
- [7] Friedman, J.H., Bentley, J.L., Finkel, R.A.: An algorithm finding best matches in logarithmic expected time. ACM Trans. Math. Software **3** (209–226)
- [8] Perez, J.C., Vidal, E.: An Approximate Nearest Neighbours Search Algorithm Based on the Extended General Spacefilling Curves Heuristic. Lecture Notes in Artificial Intelligence **1451** (1998) 697–706
- [9] Perez-Cortes, J.C., Arlandis, J., Llobet, R.: Fast and Accurate Handwritten Character Recognition using Approximate Nearest Neighbours Search on Large Databases. Lecture Notes in Artificial Intelligence **1876** (2000) 767–776
- [10] Rosenfeld, A., Pfaltz, J.L.: Sequential Operations in Digital Picture Processing. Journal of the ACM **13(4)** (1966) 471–494
- [11] Tomasi, C., Manduchi, R.: Stereo Matching as a Nearest-Neighbor Problem. IEEE Trans. on PAMI **20(3)** (1998) 333–340