

On Structural Modelling for Omnifont and Handwritten Character Recognition

Nadeem A. Khan and Hans A. Hegt

Eindhoven University of Technology,
Department of Electrical Engineering, SES-Group, EH 5.25/5.28
P.O.box 513, 5600 MB, Eindhoven, The Netherlands
{N.A.Khan, J.A.Hegt}@ele.tue.nl

Abstract. A novel scheme for structural modelling of multifont and handwritten characters is presented. The focus is on constructing such structural models that can be hierarchically interpreted leading to a multistage recognition scheme. This can form a basis of a high speed but reliable classifier which leaves the task of detailed discrimination between confusion classes to the secondary stage. The proposed class descriptors or prototype shape models utilise a certain "well-thought" set of shape primitives. They are "simplified" enough to ignore the inter-class variations in font-type or writing style yet retaining enough details for discrimination between the samples of the similar classes at the secondary stage. This type of modelling when combined with a good scheme of checking the spatial interrelation of features results in a powerful character recognition system utilising minimal prototypes per class. It also proves to be robust against various distortions and degradation like touching and broken characters.

1 Introduction

The performance of the current OCR systems does not degrade gracefully on poor quality and mixed font text [1]. This owes to the fact that they frequently employ character shape definitions which are either incomplete or incorporate nonessential shape details or both. Optimal shape descriptors are pre-requisite for constructing a robust high speed omnifont or hand-written character recognition system. In our approach we accept and utilise the notion that hand-crafted prototypes (structural class descriptors) based on character skeletons serve as a good basis for such type of modelling. This also avoids a priori training which can be cumbersome at times.

Some interesting approaches based on similar concepts have been proposed in the past [2] and especially [3], [4], [5], which makes use of Feature Description Graphs to express the configuration of features of both the prototypes and the candidate characters. The scheme employs homeomorphic graph matching. Our approach, based on structural analysis, also employs attributed graph matching of extracted geometric features. However, it departs from the existing approaches in the following

ways: Firstly, the identified paths in the candidate graph for one-to-one correspondence with the prototype graph edges can be partially overlapping. This permits mapping of even those input samples of a class to a single prototype which are not homeomorphic to the prototype class. Secondly, we directly make use of poly-line fitting of the character skeleton without attempting to interpret the candidate features as straight or curved without the absence of the prototype context. Thirdly, the allowed primitive types are restricted to only a few. Further to that the scheme employs a meaningful hierarchical interpretation of these primitives. This permits accommodation of inter-class variation into a single prototype on one hand though maintaining enough details for the discrimination process of similar classes. The scheme lets the system automatically focus on the intrinsic discriminating aspects of a class at the secondary stage for resolving the class ambiguity. Thus, the cumbersome task of manually supplying the additional specific constraint set is highly minimised if not at all eliminated.

2 Modelling and Representation of the Class Descriptors

The prototypes (or structural class descriptors) are based on geometric modelling of character strokes. Hand-crafting of the prototypes is done based on only the following limited set of geometric features types which we will regard as primitives for our scheme. They are: (i) line segment primitive; (ii) quarter circle primitive; (iii) semicircle primitive and the (iv) full circle or loop primitive.

Associated with each of these primitives is its own attribute set. The set of primitives used for the line segment primitive comprises of end-points' co-ordinates, length, orientation etc. Curve primitives are also attributed in a similar way by making use of the reference cord as shown in Fig. 1. For a loop primitive it is merely a convenient diameter cord. In contrast to the semicircle-circle primitive, a quarter circle primitive is used to model those curved parts of the character (but not loops) which, if replaced by a straight segment, will not necessarily change the class of the character (for example the curve in 'J').

Looking to the set of primitives, it would be clear that modelling the exact geometric shape of a character class is not the goal. To come up with a minimum number of prototypes per class it is essential to keep in view the variation in font or writing style and use appropriate primitives which not only geometrically best resemble the character strokes but also provide the best (hierarchical) interpretation covering the possible variations.

The prototypes (and the also candidates) are represented as directed attributed feature graphs using the above mentioned feature types or primitives. The edges of the graph correspond to features where the labels of the edges provide detailed feature attributes. Nodes represent the feature end points. Shared end-points by multiple features are indicated by multiple overlapping nodes in the figures. The suggested prototype set based on the above scheme for the upper-case English alphabet and numerals is shown in Fig. 2.

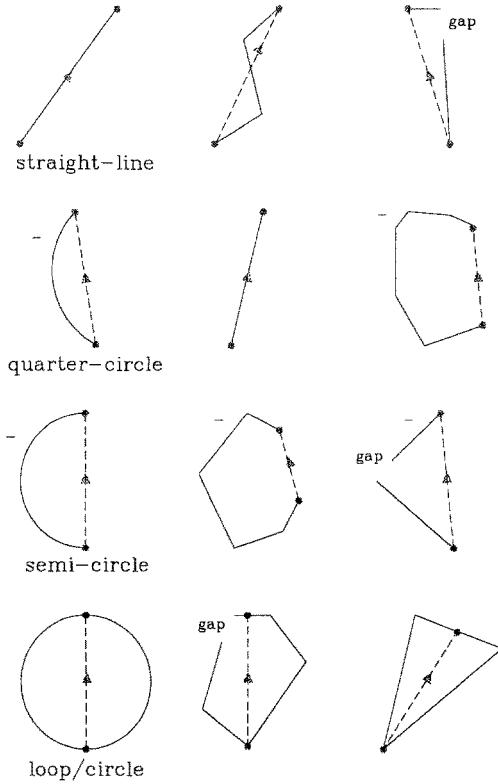


Fig. 1. Primitive set with examples of candidate sub-graphs that can match them at the first stage of classification; ‘-’: curve on left side of chord; ‘+’: curve on right side of chord

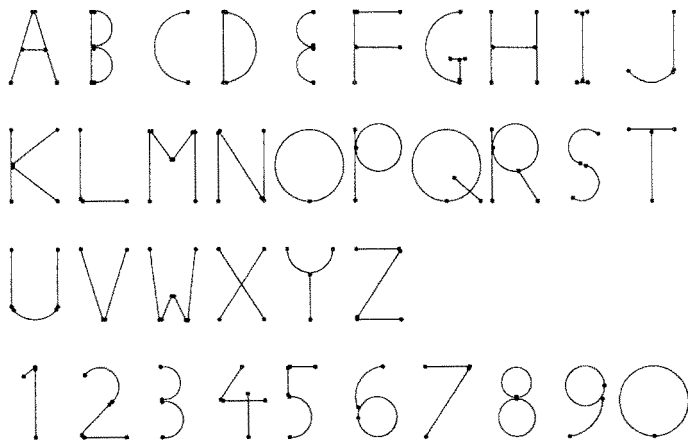


Fig. 2. Proposed set of prototypes represented as feature graphs; the curve primitives in ‘J’, ‘U’, ‘6’ and ‘9’ are quarter circles and the rest of the curve primitives are either loops or semi-circles

3 The Matching Scheme Based on the Hierarchical Interpretation of the Prototypes

The candidate graph is derived from the input image of the character that has been pre-processed (for noise removal and enhancement) and then thinned and poly-line fitted with straight line segments. For better matching, vertices are also added at junctions, crossings and where the end-point of a line segment is in close proximity to another line segment. It is then expressed as a feature graph utilising line segment primitives. We define a path in the candidate graph as an ordered set of adjacent edges of the feature graph. Gaps of reasonable size are permitted between the adjacent edges of a path and no edge is included more than once in the same path.

The classification is then reduced to graph matching based on multiple-to-one matching of the candidate graph-features to the features in the prototype. For recognition, correspondence is established between the paths of the candidate graph and the prototype graph features such that (a) corresponding matching paths in the candidate graph are found for all prototype features (as discussed below in more detail) (b) they satisfy the spatial interrelationship as depicted by the features in the prototype graph. Flexibilities of the sort as illustrated in Fig. 3 are permitted. The related transformation costs are then calculated. The given sample is thus classified to the prototype class for which the total transformation cost is minimum among others.

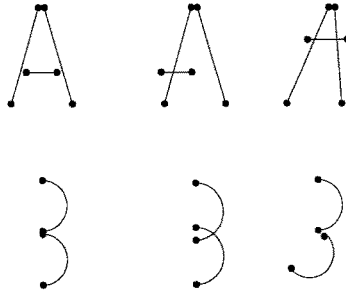


Fig. 3. Allowed flexibility by the feature spatial location checks

A multi-stage matching process is proposed based on the view that not all the prototype classes are equally well-separated from each other in the feature space. For instance, in case of the character 'A' the top sharp corner can be made smooth. However, smoothing the bottom corner of 'V' will convert it into a 'U'. To cope with this situation loose interpretation of the primitives is used during matching at the first stage to manage with the least number of prototypes (one in most cases). However, if the candidate seems to belong to a class belonging to a confusion set of similar classes, the secondary stage is invoked which permits less flexibility to deviate from the exact shape definition of the prototype and specifically focuses on the discriminating features of each individual class.

The salient features of the permitted flexibility for the first stage matching of a

path (shape) to a prototype feature shape is also illustrated in Fig. 1. Transformation costs are calculated based on the deviation w.r.t. size, geometrical shape, orientation and inclusion of gaps in a path. Note that the quarter circle primitive can also be matched to a single straight edge. A match is not found if the deviation regarding any of the above attribute exceeds a certain selected limit. A (straight) edge in the candidate graph can be included in more than one path. Though, no two paths which are established as matches of two prototype features should fully comprise of the same set of strokes. This permits matching in case the adjacent strokes of a character partially overlap. A (straight) edge in the candidate can be split into two edges for use in two different paths corresponding to two different primitives of the prototype graph.

In the second pass, for the disambiguation process the allowed flexibility is reduced based on the following principles which are stated in increasing order of priority:

- (i)- A more exact matching is performed by increasing weights for all transformation cost. Generally speaking, the deviations in shapes and sizes require more attention (and hence more penalty) than the deviations in feature orientation. The quarter-circle primitive can still be matched with both a straight line segment as well as with a curved path of multiple strokes.
- (ii)- Multiple inclusion of a candidate graph edge in more than one path is severely penalised. Cutting a candidate graph edge into two for inclusion in two different paths is still permitted.
- (iii)- Extra focusing on the dissimilar parts of the prototypes of a confusion set by an additional increase of weights of the transformation cost related to them.

It should be noted that weight values corresponding to various transformations (determining their relative importance) have been initialised arbitrarily by making a reasonable guess. They were then coarsely "tuned" by processing a limited set of samples and revising their values. In certain cases the transformation rules were also revised. The system is not found to be too sensitive for the exact choice of weights. This demonstrates the robustness and generality of the adopted approach. However, in future some fine tuning through automatic training will be added for the disambiguation of very confusing cases as sometimes exists in totally unconstrained hand-written characters.

4 Modelling of Interclass Diversity Versus Discrimination

In this section we will try to show how utilisation of only a few primitives combined with a flexible matching proves beneficial in shedding the unnecessary details related to inter-class variations. Still enough information for class discrimination in confusion classes is retained which is explored by the hierarchical interpretation of primitives.

Illustrated in Fig. 4 with the help of a few examples is how the careful choices made in constructing the prototype set with the prescribed primitives can prove useful in keeping the prototypes per class to a minimum (one in most cases). Especially to

note is the advantage of postponing the abstraction process of portions of the poly-line fitted skeleton into curve primitives to the classifier stage. Also note the advantage gained in permitting the use of a candidate stroke in multiple paths. Thus, in cases where homeomorphic graph matching [3] can fail or result in a poor match this matching scheme can succeed. The following examples throw light on the disambiguation process:

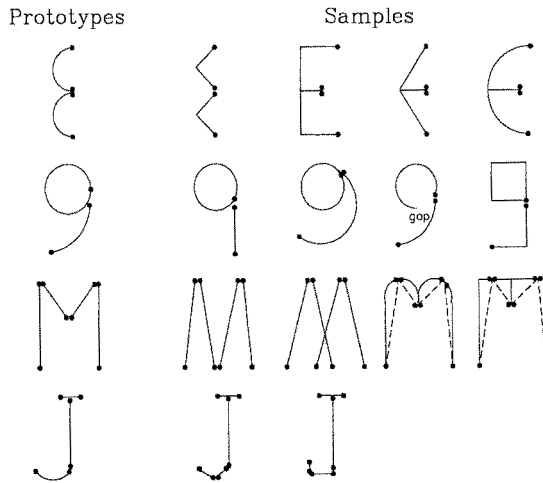


Fig. 4. Mapping of class diversities into a single (or minimal number of) prototypes

Example 1: The ambiguity of interpretation at the first stage regarding an input sample from the confusion set (2,Z) is resolvable in the second pass. The tighter checking rules at the secondary stage will let the system focus on determining whether the upper stroke in the input sample is closer to a curve or a straight segment. It is the key discriminating feature in this case.

Example 2: Fig. 5 shows some distorted and/or varied styled samples belonging to the ambiguous set of {5, S}. Since candidates have shapes not exactly matching to either character, the primary classification is not decisive. The secondary stage can classify them in a reasonable way but with low confidence level.

Finally, it is better to add additional prototypes per class in case of doubt rather than risking misclassification. Regarding the confusion set (0 ,O) we believe that additional specific constraints and context information is required for reliable classification. In case of totally unconstrained hand-written characters, context information is indispensable especially for confusion classes.

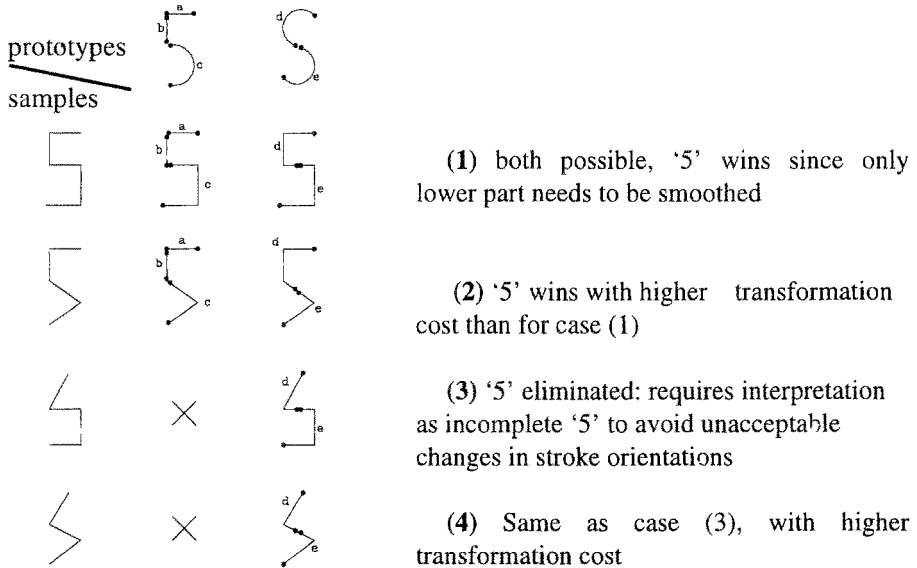


Fig. 5. Matching of some patterns to prototypes '5' or 'S' at the 2nd stage with stricter penalties

5 Experimental Results

The method has been implemented in C language on a PC platform. Though the system is still in the final stage of completion to be followed by a proper performance evaluation, preliminary results are promising. Fig. 6(a) and (b) presents some examples in connection to the ability of the system to recognise alphanumeric characters. The system can cope with segmentation errors, extraneous strokes, gaps, distortions introduced in original shape due to the thinning and polyline fitting etc. In Fig. 6(c) some laboratory-prepared examples of numeral characters with segmentation errors and/or extraneous strokes are shown that are also correctly recognised by the system.

When tried on a test database comprising of few hundreds of pre-segmented totally unconstrained hand-written numerals of postal codes extracted from real-life mail, a correct classification rate of slightly above 98% was achieved for the zero rejection rate. It also includes binary images of numeral characters with large gaps in their bodies as shown in Fig. 6 (d).

6 Final Remarks

The structural approach presented in this paper also attempts to analyse the issue of which distortions are important for humans during flexible matching of patterns and

at which stage of the recognition process. The system is in its final stage of completion. Future work includes research on additional generalised rules for better hierarchical classification and discrimination and accelerating the processing speed by more intelligently avoiding fruitless matching options. Other feature extraction methods than skeletonisation can also be explored for the disambiguation process. The specific features or properties depicted in the prototypes to be checked in the given sample need not to be obtained only from character skeletons. Finally, context and syntax information can be also be used once an application is specified.

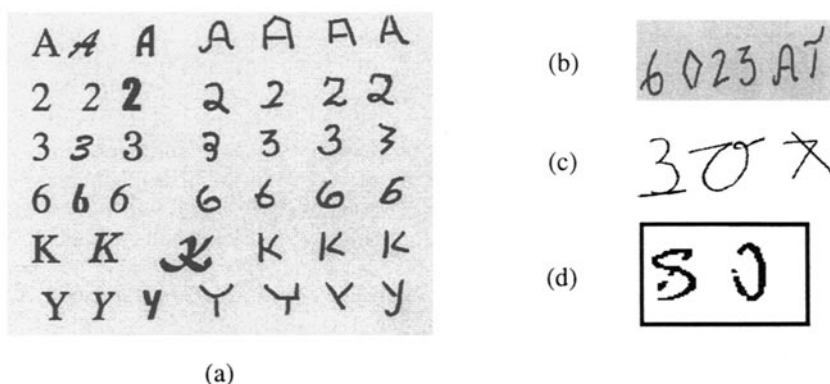


Fig. 6. (a) Examples of font and style variations in machine-printed and hand-written characters that the system can cope with; (b) Sample of hand-written postal code whose pre-segmented characters are recognisable by the system; (c) Examples of hand-written numerals with segmentation problems that are correctly recognised by the system; (d) images of hand-written numerals '8' and '0' with gaps

References

1. Tian, Q., et.al.: Omnifont Printed Character Recognition. SPIE, Vol.1606. Visual Communication and Image Processing, (1991) 260-268
2. Anisimovich, K., et.al.: Using Combination of Structural, Feature and Raster Classifiers for Recognition of Handprinted Characters. Proc. Fourth International Conference on Document Analysis and Recognition, Vol. 2, (August 1997) 881-885
3. Rocha, J., Pavlidis, T.: A shape analysis model with applications to a character recognition system. IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. 16. No. 4, (April 1994) 393-404
4. Zhou, J., Pavlidis, T.: Discrimination of characters by a multi-stage recognition process. Pattern recognition, Vol. 27. No.11, (1994) 1539-1549
5. Wang, L., Pavlidis T.: Direct gray-scale extraction of features for character recognition. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 15. No. 10, (October 1993) 1053-1067