Convex Hull based Approach for Multi-Oriented Character Recognition from Graphical Documents

Partha Pratim Roy[1], Umapada Pal[2], Josep Lladós[1], Fumitaka Kimura[3]
[1]*Computer Vision Center, Universitat Autònoma De Barcelona, 08193, Bellaterra,Spain*
[2]*Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata – 108, India*
[3]*Graduate School of Engineering, Mie University, 1577 Kurimamachiya, Mie 514-8504, Japan*
Email: partha@cvc.uab.es

Abstract

In this paper, we present a scheme towards recognition of English character in multi-scale and multi-oriented environments. Graphical document such as map consists of text lines which appear in different orientation. Sometimes, characters in a single word may follow a curvi-linear way to annotate the graphical curve lines. For recognition of such multi-scale and multi-oriented characters a Support Vector Machine (SVM) based scheme is presented in this paper. The feature used here is invariant to character orientation. Circular ring and convex hull have been used along with angular information of the contour pixels of the character to make the feature rotation invariant. We tested our proposed scheme on two different datasets. Combining circular and convex hull feature we have obtained 96.73% and 99.56% accuracy in these two datasets.

1. Introduction

In graphical documents, such as map, text characters are often in different scale to give importance to specific locations in some regions. Also the characters could appear in different orientations other than the usual horizontal and vertical directions. They are printed thus to annotate graphical objects such as river, road etc. We show a map in Fig.1 to illustrate the problem. The interpretation of graphical documents does not only require the recognition of graphical parts but the detection and recognition of multi-oriented text.

Optical character recognition (OCR) systems available commercially are not capable of handling different ranges of images containing multi-sized and multi-oriented characters. There are many research papers on the recognition of the normal text characters with different font and size but to the best of our knowledge, there exists only a few papers towards the recognition of multi-oriented characters from a single document. Adam et al. [1] used Fourier Mellin Transform for multi-oriented symbol and character recognition from engineering drawings. The image is convolved by a set of filters and then the system tries to locate the pixel for which the response is pre-specified. Some of the multi-oriented character recognition systems consider character realignment. The main drawback of these methods is the distortion due to realignment of curved text. Parametric eigen-space method is used by Hase et al. [2]. Xie and Kobayashi [3] proposed a system for multi-oriented English numeral recognition based on angular patterns. Pal et al. [6] proposed a modified quadratic classifier based recognition technique for handling multi-oriented characters.

In this paper, we present an improved scheme of [6] towards recognition of English character in multi-scale and multi-oriented environments. To make the system rotation invariant, the features are mainly based on the angular information of the external and internal contour pixels of the characters, where we compute the angle histogram of successive contour pixels. Note that, angle between successive pixels will be similar even if the character is rotated. For illustration, see Fig.2. In Fig.2(a) a component of length 5 pixels is shown. Let the name of these 5 pixels are A, B, C, D and E. The angle between three consecutive pixels ABC (<ABC) is 180º, <BCD is 225º, <CDE is 180º. Let the component be rotated 90º in clock-wise direction, then its rotated version is shown in Fig.2(b). It can be noted that, after the rotation the <ABC is 180º, <BCD is 225º and <CDE is 180º and these angles are similar to earlier version. We compute frequencies of different angles obtained from the contour pixels of a character and the frequencies are used for recognition. It may be noted that such frequency of a character will be similar even if the character is rotated. Circular ring and convex hull have been used to divide a character into several zones and zone wise angular histogram is computed to get higher dimensional

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

feature for better performance. A SVM classifier has been used for recognition purpose.

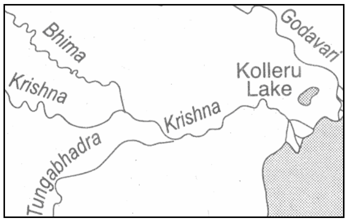
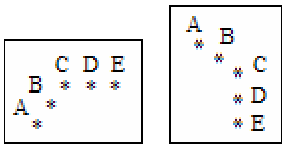


Fig.1. Example of a river map from India



(a) (b)

Fig.2.(a) Example of a 5 pixel component and (b) its rotated version (90˚ in clockwise direction).

2. Data collection and pre-processing

For the experiment of present work, we considered two sets of data. First set of data is from graphical document only and its size is 8250. An automatic method [7] has been used to extract characters from map towards the development of this dataset. Second set of data was the same data used in Pal et al. [6] and the size of this data is 18232 and its collection is mainly from journals, magazines, newspaper, advertisements and computer printouts. Data are scanned at 300 dpi and we have used a histogram based global binarization algorithm [4]. The digitized image may contain spurious noise points and irregularities on the boundary of the characters, leading to undesired effects on the system. For removing this sort of noises we have used a method discussed in [4]. To get an idea of data quality, we have shown some samples in Fig.3.

Both uppercase and lowercase alphanumeric characters were considered for our experiment, so we should have 62 classes (26 for uppercase, 26 for lowercase and 10 for digit). But because of shape similarity of some characters/digits, here we have 40 classes. We are considering arbitrarily rotation (any angle up to 360 degrees) so, some of the characters like “p” and “d” are considered same since, we will get the character “p” if we rotate the character “d” 180 degrees.

3. Feature extraction

Details of the feature extraction method are discussed as follows.

3.1. Global feature based on angular information

For an input image internal and external contour pixels are computed and they are used to determine the angular information feature of the image.

Given a sequence of consecutive contour pixels $V_1 \dots V_i \dots V_n$, of length n (n>7), the angular information of the pixel V_i is calculated from the orientation of vector pairs V_{i-k} , V_i and V_i , V_{i+k} . For better accuracy, we take the average of 3 orientations for each pixel, considering k=1, 2 and 3. From each pixel we will get two angles (one from background side and other from foreground side) and the angle corresponding to background side is considered here. The angles obtained from all the contour pixels of a character are grouped into 8 bins corresponding to eight angular intervals of 45 degree (337.5 degree to 22.5 degree as bin no. 1, 22.5 to 67.5 as bin no. 2 and so on). For a character, frequency of the angles of 8 bins will be similar even if the character is rotated at any angle in any direction. For illustration, see Fig.4. Here, we compute the histogram of the angles corresponding to 8 angular information of two rotated shapes of the character ‘W’. From the figure it can be noted that angle histogram of two characters is similar although the character has different rotation.

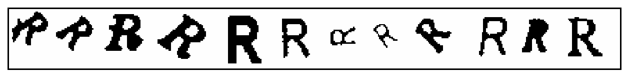


Fig.3. Some images of character ‘R’ from the dataset are shown.

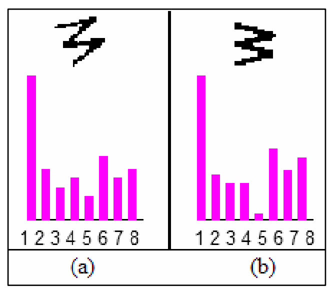


Fig.4. Input images of the character ‘W’ in 2 different rotations and their angle histogram of contour pixels are shown. The numbers 1-8 represent 8 angular bins.

We may feed this global angular histogram of the characters into the classifier for recognition but to get higher accuracy, we divide a character into several zones and zone-wise angular information is computed and hence higher dimensional features are found. Circular ring and convex hull have been used for this purpose. Different dimensional features are computed for the comparison of recognition accuracy and their procedures are discussed as follows.

3.2. Higher dimensional feature detection using circular ring and convex hull

Circular ring based division: A set of four circular rings is considered here and they are defined as the concentric circles considering centre as the centre of minimum enclosing circle (*MEC*) of the character and the minimum enclosing circle as the outer ring. The radii of the rings are in arithmetic progression. Let R_1 be the radius of *MEC* of the character, then the radii (outer to inner) of these four rings are R_1, R_2, R_3 and R_4, respectively. Where R_1-R_2 = R_2-R_3 = R_3-R_4 = R, where R is $R_1/4$. See Fig.5(a), where four rings are shown on the character 'E'. These rings divide the *MEC* of a character into four zones.

Convex hull based division: Convex hull rings are computed from the convex hull boundary. We compute 4 convex hull rings and we consider the outermost convex hull ring (say C_1) as the convex hull itself. Other 3 convex hull rings are similar in shape and computed from C_1 by reducing its size. The 2nd ring can be visualized by zooming out the C_1 with R pixels inside. Other 2 rings are computed similarly. Four convex hull rings are shown in Fig.5(b) on the character 'W'.

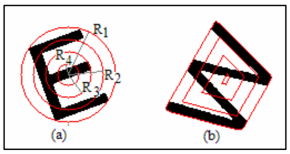


Fig.5.(a) The circular rings and (b) The convex hull rings.

We have used this convex-hull representation since it has some interesting properties as given below. These properties help us to get higher dimensional feature.

Residue area: The area of residue is defined by the number of points inside a convexity defect zone.

Residue height: The height of a residue is the depth of convexity defect from the convex hull.

Residue surface width (*RSW*): The width of the residue is the distance between the point of the contour where the defects begin and the point of the contour where the defects end.

Mid-Point of residue surface width: This is defined as the mid-point of the residue surface width. See Fig.6 where residue and its different parameters are shown.

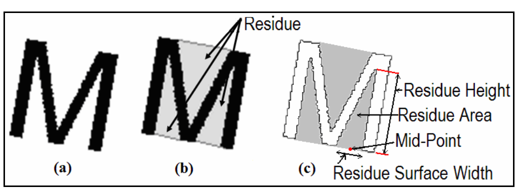


Fig.6. (a) Image of the character "M". (b) Three concave residua from the convex hull of "M". (c) Different parameters of convex hull.

Reference point and reference line detection: If we compute information of angular histogram on the character portions in each of the ring, then we will get 32 (4 rings × 8 angular information) dimensional feature. To get more local feature for higher accuracy, we have divided each ring into few segments. To do such segments, we need a reference line (which should be invariant to character rotation) from a character. The reference line is detected based on the background part of the character using convex hull property. The mid-point of *RSW* of the largest (in area) residue of a character is found and the line obtained by joining this mid-point and the centre of *MEC* of the character is the reference line of the character. If there are two or more largest (in area) residue then we check the height of these largest residue and the residue having largest height is selected. If the heights of the largest residue are same, then we select the residue having maximum *RSW*. The mid-point of the *RSW* of the selected residue and centre of *MEC* of the character is the reference line. The mid-point of *RSW* is the reference point. If no residue is selected by above, we consider the farthest contour point (P_f) of the character from the centre of *MEC* and the line obtained by joining P_f and the centre of *MEC* is the reference line. Here, P_f is the reference point.

Initially, we used principal component analysis to get a reference line and we obtained lower accuracy. Hence, we computed reference line as discussed above.

A reference line can segment each ring into two parts and if we compute the feature on each of the segment, then we will get 64 dimensional features (4 rings × 2 segments × 8 angular information). If we take another reference line perpendicular to this reference line, then each ring will be divided into 4 segments and as a result, we will get 128 dimensional features (4 rings × 4 segments × 8 angular information). See Fig.7, where two reference lines PP´ and QQ´ are shown. To get 256 dimensional features, we consider 2 more reference lines which are angular bisectors of these lines. Here, we denote *fc, fh* as the feature obtained from circular ring and convex hull, respectively. Also, by *fc*(k) and *fh*(k), we mean the feature of dimension k obtained from circular and convex hull segments. In our experiment, we considered *fc*(32), *fc*(128), *fc*(256), *fh*(32), *fh*(128) and *fh*(256) to get comparative results. We also used the combination of these features in our experiment. For example, we get 512 dimensional combined features when we add *fc*(256) and *fh*(256) features together.

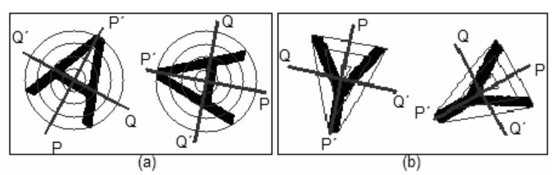


Fig.7. Reference lines PP´ and QQ´ are shown with (a) circular ring division in character 'A' (b) convex hull ring division in character 'Y'.

To get the different segment sequentially we consider the segment that starts from the reference point as segment number 1 (say S_1). To get 256 dimensional features each ring will be divided into 8 block segments. Starting from S_1 if we move anti-clockwise then the segments obtained from outer ring block (R_1-R_2) are designated as 1st, 2nd...8th. Similarly, from the (R_2-R_3)

ring block, we will get 9th, 10th.....16th segment. Other segments are obtained in similar way.

To get size independent features we normalize them. For normalization we divide the number of pixels in each segment by the total number of contour pixels. Hence we get these feature values between 0 and 1.

4. SVM Classifier

We use Support Vector Machine (SVM) classifier for recognition. The SVM is defined for two-class problem and it looks for the optimal hyper-plane which maximizes the distance, the margin, between the nearest examples of both classes, named support vectors (SVs). Given a training database of M data: {xm| m=1,...,M}, the linear SVM classifier is then defined as:

$$f(x)=\sum_j \alpha_j x_j \cdot x+b$$

Where, $\{x_j\}$ is the set of support vectors and the parameters α_j and b have been determined by solving a quadratic problem [5]. The linear SVM can be extended to a non-linear classifier by replacing the inner product between the input vector x and the SVs x_j, to a kernel function k defined as: $k(x,y)=\phi(x)\cdot\phi(y)$. This kernel function should satisfy the Mercer's Condition [5]. Some examples of kernel functions are polynomial kernels $(x\cdot y)^p$ and Gaussian kernels exp(-||x-y||2/c), here c is a real number. We use Gaussian kernel for our experiment. Details of SVM can be found in [5].

5. Results and discussions

As discussed in Section 2, we consider two sets of data. Both uppercase and lowercase letters of different fonts including Times New Roman and Arial are considered. Variable sized characters with different orientations were used for the experiment. Both the dataset have been tested using cross validation technique. For this purpose, we divided the dataset into 5 parts. We trained our system on 4 parts of the divided dataset and tested on remaining part of the data. From the first dataset, we have obtained 96.54% (95.78%) recognition accuracy and from the 2nd dataset we achieved 99.45% (99.40%) accuracy, using circular (convex hull) based feature of dimension 256. Recognition accuracy obtained from circular and convex hull features with their different dimension are given in Table1. From the experiment we noted that better accuracy can be achieved combining circular and convex hull features. Combining circular and convex hull features of 256 dimension each we got 512 dimension feature. Using this 512 dimensional combined feature we achieved 96.73% and 99.56% accuracy from our SVM classifier in the two datasets mentioned above.

We noticed that most of the errors occurred due to similar shape structures. We noted that highest error occurred from the character pair 'f' and 'r'. This is because of their shape similarity. Other errors occurred mainly from noisy data where residue from convex hull has not been extracted properly. This wrong residue detection sometimes influences error.

Characters with font-size between 10 to 64 points are considered for experiment. From the experiment we also noticed that better results were obtained in case of bigger font-size characters. We obtained better accuracy when documents written in 16 or more font-size were considered. Also, from the experiment we noticed that upper-case characters provided better results than lower-case characters.

Comparison of results: Adam et al. [1] received 97.5% accuracy on English characters. The recognition rate in [6] for the 2nd dataset has been reported as 98.34% whereas from the same dataset, our proposed approach shows more than 99% (99.45% using circular feature and 99.40% using convex hull) accuracy.

Table 1. Recognition accuracy with different feature dimension

Data Set	Feature type	Feature Dimension		
		32	128	256
Set 1	*fc (circular)*	90.54	96.01	96.54
	fh (convex hull)	82.77	93.76	95.78
Set 2	*fc (circular)*	98.30	99.39	99.45
	fh (convex hull)	92.25	98.86	99.40

6. Acknowledgement

This work has been partially supported by the Spanish projects TIN2006-15694-C02-02 and CONSOLIDERINGENIO 2010 (CSD2007-00018).

References

[1] S. Adam, J. M. Ogier, C. Carlon, R. Mullot, J. Labiche and J. Gardes,"Symbol and character recognition: application to engineering drawing", IJDAR,(3), pp. 89-101, 2000.

[2] H. Hase, T. Shinokawa, M. Yoneda and C. Y. Suen, "Recognition of Rotated Characters by Eigen-space", Proc. 7th ICDAR, pp. 731-735, 2003.

[3] Q. Xie and A. Kobayashi, "A construction of pattern recognition system invariant of translation, scale-change and rotation transformation of pattern", Trans. of the Society of Instrument and Control Engineers,(27), pp.1167-1174, 1991.

[4] B. B. Chaudhuri and U. Pal, "A complete printed Bangla OCR system", *Pattern Recognition,* (31), pp. 531-549, 1998.

[5] V. Vapnik, "The Nature of Statistical Learning Theory", *Springer Verlang,* 1995.

[6] U. Pal, F. Kimura, K. Roy and T. Pal, "Recognition of English Multi-oriented Characters", ICPR (2) pp. 873-876, 2006.

[7] P. P. Roy, J. Lladós and U. Pal, "Text/Graphics Separation in Color Maps", ICCTA pp. 545-551, 2007.