Chapter 7

Feature Extraction using Snakes

7.1 Introduction

In the previous chapter, an effective methodology to localize the license plate of the moving vehicle is dealt with. Recognizing the characters is the next challenge to pursue. The researchers are continuously publishing new and efficient approaches to address the problem of text (alpha-numerals) recognition. A system could be envisaged which would identify the word directly from the image presented, but the task of the recognition system is greatly simplified by preprocessing the image, organizing the information and representing it in a more accessible manner. The processing to be carried out before recognition consists of two major parts-normalization and representation. The first of these attempts to remove variations in the images which do not affect the identity of the word, and the second then expresses the salient information contained in the image in a concise way, suitable for processing by a pattern recognition system. The process of alpha numerals recognition of any image can be broadly broken down into five stages:

1. Pre-processing

2. Segmentation

* The entire chapter in the form of paper has been published in "International Journal of Artificial Intelligence and Mechatronics", Volume 2, Issue 2, pp 31-36, Sept 2013.

3. Feature extraction

4. Classification

5. Post-processing

The preprocessing stage is a collection of operations in the form of successive transformations to be applied on an image. As a result, it takes in a raw image, reduces noise and distortion, removes skewness and performs skeltonizing of the image thereby simplifying the processing of the rest of the stages.

The segmentation stage takes in a image and separates the different logical parts, like text (containing alphanumerals) from graphics, lines of a paragraph, and characters of a word. The feature extraction stage analyzes a text segment and selects a set of features that can be used to identify the text segment uniquely. The selection of a stable and representative set of features is the heart of pattern recognition system design. Among the different design issues involved in building an alpha numerals recognition system, perhaps the most consequential one is the selection of the type and set of features. The classification stage is an important decision making stage of an alpha numerals recognition system and uses the features extracted in the previous stage to identify the text segment according to preset rules. The post-processing stage, which is the final stage, improves recognition by refining the decisions taken by the previous stage and recognizes words by using context. It is ultimately responsible for outputting the best solution and is often implemented as a set of techniques that rely on character frequencies, lexicons, and other context information.

Extraction of proper features is the main key to correctly recognize an unknown character. A good feature set contains discriminating information, which can distinguish one object from other objects. It must also be as robust as possible in order to prevent generating different feature codes for the objects in the same class. The selected set of features should be a small set whose values efficiently discriminate among patterns of different classes, but are similar for patterns within the same class.

The snakes' algorithm is an image segmentation process that utilizes energy-minimizing splines. The energy terms are obtained from discrete points that make up a snake and the gray-level values in an image. Snake is an energy minimizing, deformable spline influenced by constraint and the image forces pulling it towards object contours. Snakes are greatly used in applications like object tracking, shape recognition, segmentation, edge detection, stereo matching. Snakes may be understood as a special case of general technique of matching a deformable model to an image by means of energy minimization. Snake is an "active" model as it always minimizes its energy functional and therefore exhibits dynamic behavior.

7.2 Related work

7.2.1 Normalization

The style of writing alpha-numerals varies in many different ways, for different purposes. One way of reducing the variation is to identify certain parameters of the alpha-numerals that may vary to give a different appearance. Then a procedure must be determined to estimate each of these parameter values from the samples and finally another procedure must be found to remove the effects of the parameter from the word. The most obvious parameters include height, slant angle, slope, stroke width and rotation. The general steps followed in normalization are shown in figure 7.1.

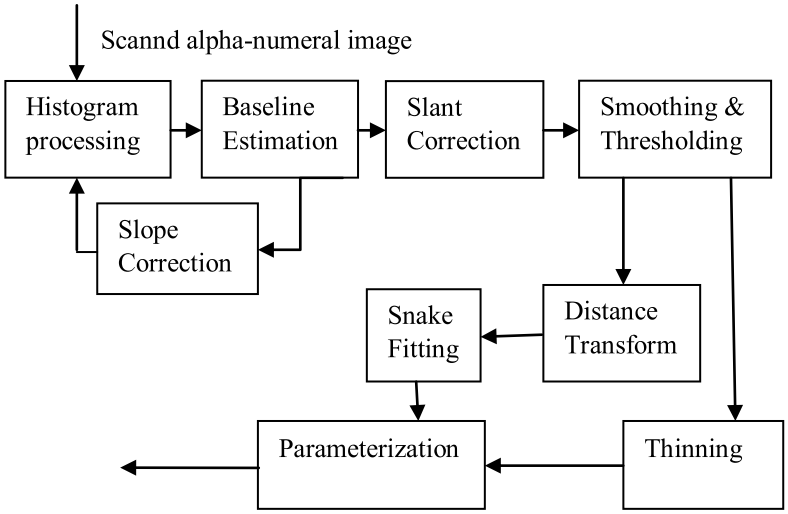


Figure 7.1: Preprocessing operations for image normalization before encoding

7.2.1.1 Base line estimation an slope correction

The character height is determined by finding the intuitively important lines which are shown running along the top and bottom ends of lower case letters as in figure 7.2 - the upper and lower base lines respectively, with a centre line between the two. With these lines, the ascenders and descenders which are used by human readers in determining word shape can also be identified.

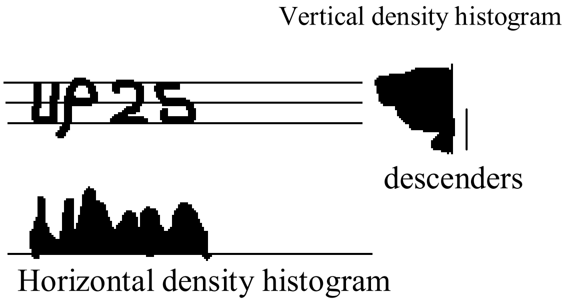


Figure 7.2: Histograms, centre line & base lines

The heuristic used for base line estimation consists of the following steps:

1. Calculate the vertical density histogram by counting the number of black pixels in each horizontal line in the image. Vertical and horizontal density histograms are shown on the right and bottom edges of the figure 7.2.

2. Reject the part of the image likely to be a hooked descenders (as in the letters 'ghy'). Such a descender is indicated by a peak in the vertical density histogram. The minimum in the histogram above this point is found and the image is cleared from that point downwards.

3. Find the lowest remaining pixel in each vertical scan line.

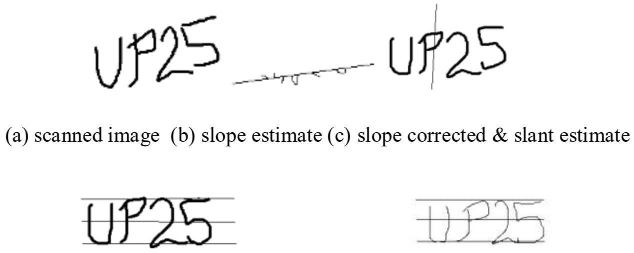
4. Retain only the points around the minimum of each chain of pixels.

5. Find the line of best fit through these points.

Reject the outlying points and calculate the new line of best fit. This is now considered to be the base line of the character.

7.2.1.2 Slant correction

Bozinovic ans Srihari's [193] algorithm commences by eliminating all horizontal rows in a alpha-numerals which contain horizontal strokes. These are identified as any rows which contain long runs of black pixels. The maximum number of consecutive black pixels which can be permitted before a line is eliminated is a parameter which must be specified. After each such row is eliminated, the remaining image is in horizontal strips, some of which are too narrow to use and are eliminated. The remaining strips are divided into boxes, containing separate near vertical-strokes in each of which the centroids of the upper and lower halves are determined and the slant of the line between the two is calculated. Averaging the slants across all such strokes gives an estimate of an average overall slant of the alpha-numeral. The slant is corrected with a shear parallel to the x axis.



(a) scanned image (b) slope estimate (c) slope corrected & slant estimate

(d)slant corrected with base line estimate (e) skeleton

Figure 7.3: Normalization stages

7.2.1.3 Smoothing and thinning

To remove noise from the image, either from the original image, from scanned image, or before/after applying shear transforms to discrete images, it is useful to smooth the image. This is carried out by convolution with one-dimensional Gaussian filter. Having normalized and smoothed the image, it is thresholded to leave every pixel black or white. Next an iterative, erosive thinning algorithm is applied to reduce the strikes in the writing to a width of one pixel so they can be followed later. This is the skeleton of the text shown in figure 7.3(e).

Skeletonization is a notoriously difficult problem to solve well, and many algorithms have been written, with a variety of properties. A comprehensive review with 138 references is presented in [194]. Despite this difficulty, because the skeleton is to be coarsely parameterized later, a simple algorithm was found to work well, and other algorithms that were tried in [195, 196] did no better. There is enough scope for more work on identifying a suitable thinning algorithm for alpha numerals.

7.2.1.4 Parameterization

Note that the image has been reduced to a standard form, which highlights invariants of the world and suppresses spurious variations; the normalized image needs to be parameterized in an appropriate manner for input to the network which is to carry out the recognition process [197].

The method of parameterization used is to code the skeleton of the alpha numeral so that information about the lines in the skeleton is passed on to the recognition system.

In the skeleton coding scheme, the area covered by the alpha-numeral is first divided into a grid of rectangles. The vertical strips (frames) are of a fixed width for the whole word (alpha numeral), a length determined by the height estimate of the character. Typically there are 6 frames in the horizontal space occupied by one character height. This assumes that the character height is proportional to the character width, which is a valid assumption for normal alpha numeral.

The vertical resolution of the grid is chosen so that the word is divided into seven regions, each of which can be identified as playing a definite, but distinct role in the representation of alpha numerals.

7.3 Finding alpha numerals Features

Section 2 of this part describes how the original alpha numeral image can be normalized and encoded in a canonical form so that different images of the same alpha numerals are encoded similarly. However, the coding only represents low-level information about the alpha numeral, and coded it fairly coarsely to reduce the information burden. The performance of the recognizer can be improved by passing it more information about salient features in the alpha numeral. The most obvious parameters are given in section 7.2.1.

7.3.1 Finding strokes

The centers of strokes are those parts which are farthest from the edges, so a natural choice of representation to consider is the distance transform. This assigns a value, $D(x, y)$, to each pixel (x, y) in the threshold image, which is the distance of that pixel from the nearest background pixel, zero if the pixel is itself part of the background. Thus circles in the image become cones in the distance transform, the transform increasing further a point from the edge, and strokes become ridges. Now detecting stroke centers becomes a problem of finding ridges in the distance transform. Here method chosen to find these ridges is snakes.

7.3.2 Using Snake model

The development of active contour models, or snakes, results from the work of Kass *et al.* [198] and Cipolla & Blake[199]. The active contour model is defined by an energy function. The energy functional, which is minimized, is a weighted combination of internal and external forces. The internal forces emanate from the shape of the snake, while the external forces come from the image and/or from higher-level image understanding process. The solution is found using techniques of variational calculus. The energy function is defined as:

$$E_{snake} = \int\limits_0^1 E_{snake}(V(s))ds$$

$$= \int\limits_0^1 E_{int}(V(s)) + E_{image}(V(s)) + E_{con}(V(s))ds \tag{7.1}$$

where,

$E_{int}(V(s))$: represents the internal energy of the spline due to bending

$E_{image}(V(s))$: represents the image forces

$E_{con}(V(s))$: represents the external constraint force

Internal Energy:

$$E_{int} = \frac{\alpha(s)|V_s(s)|^2 + \beta(s)|V_s(s)|^2}{den}$$

The first-order term makes the snake act like a membrane.

The second-order term makes it act like a thin plate.

The internal energy is composed of first-order term and second-order term. The first-order term, which is controlled by $\alpha(s)$, adjusts the elasticity of the snake. The second-order term, which is controlled by $\beta(s)$, adjusts the stiffness of the snake.

Image Forces given by

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term}$$

Line Functional given by

$$E_{line} = I(x, y)$$

Edge Functional given by

$$E_{edge} = -|\nabla I(x, y)|^2$$

$$E_{edge} = \frac{\partial \theta}{\partial n_\perp} = \frac{\frac{\partial^2 \theta}{\partial n_\perp^2}}{\frac{\partial \theta}{\partial n_\perp}}$$

The shapes of snakes are governed by cubic B-splines. A series of N control points $p_i : i = 0, \ldots, N-1$ is defined in a 2D plane and the actual spline path generated is an interpolation of these points, each point $x(s)$, $s \in [0, N-1]$ on the path being a weighted sum of the nearest control points' positions. $B(s)$ is a polynomial function which determines how much weight is given to each control point, according to the parameter s which increases from one end of the curve to the other. The B-spline is forced to terminate at the end control points by generating 'phantom' control points $p_{-1} = 2p_0 - p_1$ and $p_N = 2p_{N-1} - p_{N-2}$.

$$x(s) = \sum_{i=-1}^{N} B(s+2-i)p_i$$

$$B(s) = \begin{cases} \frac{1}{6}s^3 & 0 \leq s \leq 1 \\ \frac{2}{3} - \frac{1}{2}s - 2^3 - s - 2^2 & 1 < s \leq 2 \\ \frac{2}{3} + \frac{1}{2}s - 2^3 - s - 2^2 & 2 < s \leq 3 \\ \frac{1}{6}4 - s^3 & 3 < s \leq 4 \\ 0 & elsewhere \end{cases}$$

Given the positions of the control points, the snake can be located on an image. According to the features in the image, how it moves, must be defined. A potential function $f(x, y)$ is defined on the pixels (x, y) where the snake gets attracted to curves of f at its high values . Depending on the relevancy, f can be intensity, contrast, distance transform. We use city-block metric $D = |\Delta x| + |\Delta y|$.

The spline curves are sampled so that M samples are generated per unit in s. At each sample point s_k, the perpendicular to the curve is searched for minimum of $-f$ in the scope of some distance I either side. The minimum displacement is recorded for each sampling point and these are added to the control points to move the snake towards local maxima. In case of four control points, each sample point is a weighted sum of the nearest four control points:

$$x(s_k) = B(s_k+2-i)p_i + B(s_k+1-i)p_{i+1} + B(s_k-i)p_{i+2} + B(s_k-1-i)p_{i+3}$$

The displacement d(s) is distributed among these control points:

$$p_i(t+1) = p_i(t) + \frac{1}{M}\sum_k B(s_k+2-i)ds_k$$

If a snake with n control points is placed on S samples of a particular feature, like the short vertical stroke of $'i'$, the positions of the control points can be recorded and the related statistics can be gathered. If the k^{th} example feature has position

$s_k = (p_{k,0}, \ldots, p_{k,n-1})^T$, the centroid of that example can be found:

$$\bar{p_k} = \frac{\sum_i p_{k,i}}{n}$$

So, the mean distance of each point from the centroid can be calculated by taking average after substracting the centroids:

$$\delta s_k = (p_{k,0} - \bar{p_k}, \ldots, p_{k,n-1} - \bar{p_k})^T$$

$$\bar{\delta s} = \frac{\sum_k \delta s_k}{s}$$

where, $\bar{\delta s}$ is the mean shape of the feature and represents a typical example. The deviation of a particular example from the means shape of a feature is found as

$$\Delta s_k = \delta s_k - \bar{\delta s}$$

It can be considered as a vector of 2n coordinates and the $2n \times 2n$ covariance matrix $\sum$ of the shapes can be found as

$$\sum = \frac{\sum_k \Delta s_k \Delta s_k{}^T}{S}$$

Principal Component Analysis (PCA) can be carried out to determine the modes of variation in the system [200, 201]. This is done by diagonalization of the covariance matrix. Each eigenvector shows a correlation in the variation of the point coordinates - a 'mode' of variation in which the points concerned have linearly related displacements. The eigenvalues give the extent of variation in the direction of the corresponding eigenvector, so eigenvector having the largest eigenvalue's captures most of the variation in the model shape. These models are shown in Cootes *et al.* [202]. The major modes of variation of two feature models are shown in figure 7.4.

After determination of these variations, they can be used to constrain the variation of a snake. The centroid of the snake is calculated from the new control points as explained earlier. Transforming this difference into coordinate frame of

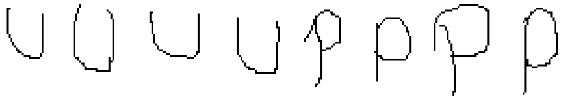


Figure 7.4: Snake models of 'U' & 'P' features showing the major mode of variation within ±1.5σ of the mean

the principal components gives the deviation from the means in each direction. Variation in the minor modes is suppressed since this represents deviation from the space of typical stroke shapes. The Mahalanobis distance $d^2(\Delta s) = \Delta s^T \sum^{-1} \Delta s$ shows much the snake deviates from the model. This distance scales down variation along the principal axes, giving the measures of standard deviations the snake lies from the mean, assuming that these deviations of snakes from the mean are distributed as a Gaussian ellipsoid. If the distance is too great, it can be reduced by scaling down all the components of the deviation. The constrained deviation is then transformed back to the original coordinates, and added to the centroid to generate a new snake which will have a shape similar to those observed in the training set.

Because the displacement to find the distance transform maxima and the application of the constraints are two separate processes, and because the image space is quantized, it is possible that the snake enters a cycle of displacing onto the maximum and being constrained to its original position. The snake thus never reaches a stable position. To avoid this case, the fitting process is stopped after a maximum of 10 iterations, though a match is usually found after just 2 or 3 iterations.

The use of these models for isolated character recognition for postcode reading has been investigated in [203-204]. Here a model is produced for each of 36 alphanumeric characters and these models are matched to pre-segmented images of written characters from a postcode database. Each model is compared with each image, and the best match is chosen. The authors in [203-204] do not use the distance transform for the match, but instead rely on the skeleton, which can often distorted away from the actual strokes at intersections.

7.3.3 Finding feature matches

Having created a model for each of the features to be found, the next step is to find all occurrences of each feature in the alpha numeral. The methods described above will find a feature match if one lies close to the starting position of the snake, so snakes must be placed at regular intervals along the word to detect all the features present. A snake, whose shape is initially the mean shape for the model, is placed at the left edge of the word, and permitted to deform to match the distance transform potential, but with the deformation being constrained to lie within k standard deviations of the mean shape - so the shape will always be similar to shapes already taken by that feature before. (For k, a value of 1 has been used here.) A best match under given the constraints is found by iterating for a limited number of times or until the snake ceases to move. Should the snake move above or below the band where it is normally found, for instance a '&' stroke feature matching the top of an 'r', then it is rejected. Otherwise, the degree of match between the snake and image is determined.

The degree of match, M, is defined as the difference of two components, representing the degree of support that the data provides for the model and the amount of deformation of the model required to fit the data. The support is the sum of two components: the sum of the distance transform along the length of the snake plus an extra weight, w, for all points that are not background points, and the deformation is measured with the Mahalanobis distance $d(\Delta s)$ of the match shape from the mean shape of the feature. M is given by

$$M = \sum_k f(X(s)) + w - d(\Delta s)$$

where

$$w_k = \begin{cases} w & \text{if } f(x(s_k)) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Snakes with score greater than a threshold should be accepted as feature matches, and the remainder are rejected. The extra weight acts as a penalty for the model crossing area that are not strokes. Its value can be determined empirically and the value of the threshold is adjusted in accordance with this value and the mean value of the distance transform. This makes the matching process independent of the width of the strokes since thick strokes give ridges with higher distance transform values than thin strokes. The mean value of the distance transform is used to indicate the stroke width in the modified slant detection algorithm, and also to give the spatial frequency parameter for Canny edge detector.

After each match, the shape and height of the snake is re-initialized to the mean and is displaced to the right by half its width, where the procedure is repeated till the whole text has been searched for that feature. In this way each feature is matched across the whole of each text in the training set. It is possible that two successive placements of a snake will converge to the same feature, but multiple matches of this sort can be rejected on the basis of the x coordinates of the centroids being much closed.

7.4 Conclusion

In this chapter, keeping in view, the importance of feature extraction for recognition of alpha-numeral test, we have introduced an approach to find large-scale features with snakes and Principal Component Analysis. Snakes are active contour models. We have illustrated the procedure to train the system with the feature models and shown the methodology to find best match for the features. Additionally, the steps for normalization of image have also been shown.