Handwriting Recognition:
A Progressive Elimination Approach

Vaibhav Mittal
Electrical Engineering Department, Delhi Technological University (formerly DCE),
Main Bawana Road, Delhi-110042, India
gemini.vaibhav@gmail.com

Sushant Kumar Singh
Electrical Engineering Department, Delhi Technological University (formerly DCE),
Main Bawana Road, Delhi-110042, India
sushantkrsingh08@gmail.com

Abstract

In this paper, a new offline pattern recognition method has been presented, as a need for time management in lower order real time systems has been felt. The paper aims to present an algorithm for recognition of patterns in need-based applications where the focus has been laid on matching and eliminating the patterns with a defined set of characters. Since the algorithm is real time action based, the processing power used and the time lag decreases with every stroke. The proposed algorithm is applied in different scenarios and the results show considerable reduction in incorrect matches.

Keywords: Pattern Recognition, Text Mining Algorithm, Data Matching, Touchscreen Application.

1. Introduction

An enhanced low-cost user interface using touch is a valuable feature for a variety of consumer, medical, automotive and industrial devices. In many consumer applications, designers prefer expensive capacitive touch screens to resistive technologies because they can track a large number of fingers and offer a friendlier interaction with the user. At present, low cost resistive technologies fill a market niche where only a single touch is required, extremely accurate spatial resolution is paramount, a stylus facilitates specific functionality—such as English-language character recognition, or in environments where users must wear gloves. The conventional approach of taking complete input and then processing, has been replaced by piecewise processing, enhancing the operation speed and improved output.

2. Component Review

2.1. *Resistive touchscreen*

The classical 4-wire resistive touch screen is popular for single touch applications because of its low cost. When two touches occur, a segment of resistance from the passive screen plus the resistance of the touch contacts is paralleled with the conducting segment of the active screen, so the impedance seen by the supply is reduced and current increases[1].

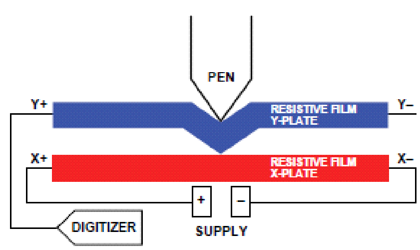

Fig. 1. Electrical Contact at user touch

The idea behind pattern recognition can be better described using a *pinch* as an example. A pinch starts with touches by two well-separated fingers. This produces a double contact, which reduces the impedance of the screen and thus, the voltage difference between the plates of the active layer. As the fingers are brought closer, the paralleled area decreases, so the impedance of the screen increases, as does the voltage difference between the plates of active layer. When tightly pinched, the parallel resistance approaches zero and (R_u+R_d) increases to the total resistance so the voltage increases to:

$$V_+ - V_- = I(R_u + R_d) = I \times R_{layer} \quad (1)$$

The voltage between the electrodes of one of the layers is constant while the other layer shows a step decrease when the gesture starts, followed by an increase as the fingers come closer. In case of a slant, both voltages show steep decrease and slow recovery. The ratio between the two recovery rates, normalized by the resistance of each layer, can be used to detect the angle of the pattern.

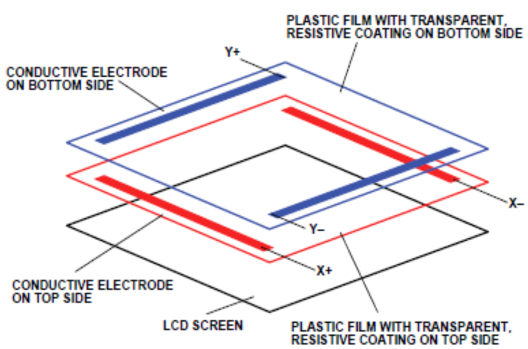

Fig. 2. Construction of a Resistive Touch Screen

2.2. *Capacitive touchscreen*

Touchscreen is a four -layer glass. The two sides of the glass substrate are coated with uniform conductive ITO (Indium Tin Oxide) coating. A 0.0015 millimeter thick silicon dioxide hard coating is done on the front side of ITO layer. There are electrodes on the four corners for launching electric current.

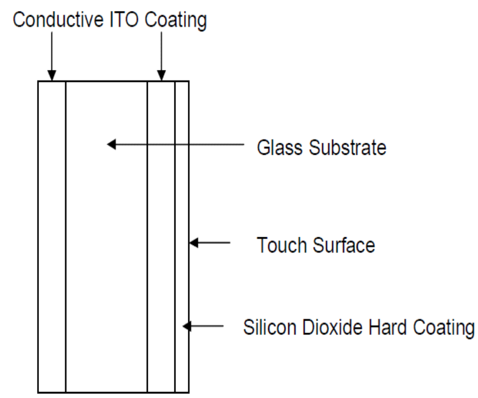

Fig. 3. Construction of a Capacitive Touch Screen

A small amount of voltage is applied to the electrodes at the four corners . A human body is an electric conductor, so when the screen is touched with a finger, a slight amount of current is drawn, creating a voltage drop. The current drifts to the electrodes at the corners. Theoretically, the amount of current that drifts through the four electrodes should be proportional to the distance from the touch point to the corners. The controller precisely calculates the proportion of the current passed through the four electrodes and figures out the (x,y) coordinate of the touch point.

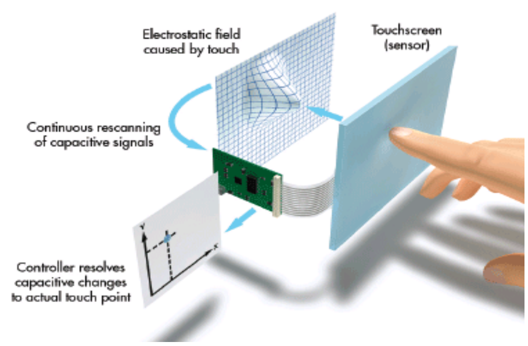

Fig. 4. Working Priciple of a Capacitive Touch Screen

3. Digitizer Smoothing

While modern pen digitizers are very accurate, some distortion is still produced. In particular the “staircase effect” that results from snapping the pointer position to the nearest pixel or hardware grid unit (see Figure 1) can complicate segment angle measurements.

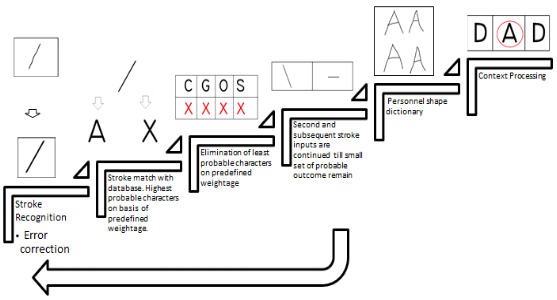

Fig. 5. Proposed Algorithm for Real-time Stroke-by-Stroke Recognition

The first stage of pre-processing uses a simple smoothing algorithm to remove this kind of distortion. To remove jitter from the handwritten text, we replace every point ($x(t)$, $y(t)$) in the trajectory by the mean value of its neighbours:

$$x'(t) = \frac{x(t-N) + \ldots + x(t-1) + \alpha x(t) + x(t+1) + \ldots + x(t+N)}{2N + \alpha} \qquad (2)$$

ar

y' (3)

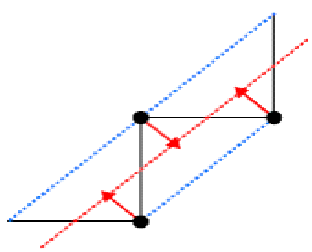

Fig. 6. Staircase approximation: These points exhibit the staircase effect. The blue lines connect every other point and are used in the algorithm to produce the smoothed red line[3]

The parameter α is based on the angle subtended by the preceding and succeeding curve segment of ($x(t)$, $y(t)$) and is empirically optimized. This helps to avoid the smoothing of sharp edges, which provide important information when there is a sudden change in direction. In our experiments, smoothing has improved our overall recognition rate by about 0.5%

3. Determination of completion of stroke

A pen-based computer needs to process a handwritten message as it is produced. The steps, ranging from various shape classification processes to ultimate shape recognition, have to cope with one of the most difficult problems of determining the beginning and ending of individual characters. The most common approaches used nowadays are unsupervised learning and data- driven knowledge. Some strategies start bottom-up, directly from the basic strokes that have been used to write a specific character. These strokes are generally hidden in the signal due to anticipation or time-superimposition effects.

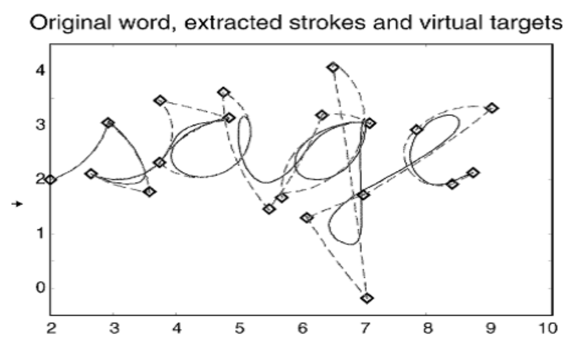

Fig. 7. Original Word, Extracted Strokes and Virtual Targets

Several operational approaches have been proposed to define and represent these basic strokes: segmentation at the point of maximum curvature, at a vertical velocity zero crossing, at minima of coordinates or at minima of absolute velocity. Some methods use a scale-space approach or a component-based approach. Others focus on perceptually important points, on a set of shape primitives etc. Model-based approaches start from a handwriting generation model and use nonlinear regression techniques to recover full parametric description of each stroke[4].

4. Stroke Matching with Database

After Stroke Extraction, it is matched with an existing database of alphabets (each alphabet in the database has been subdivided into number of strokes). Each stroke and virtual target is matched, and each alphabet in the database is allotted a score. For example, after first stroke which is a straight line inclined towards right, 'A' and 'X' will be allotted score of 26 and rest of the alphabets will be allotted zero. After second stroke is entered which is straight line inclined towards left originating from the same point as the first stroke, 'A' will be given another 26 points while 'X' will be given zero, thus bringing the total of 'A' to 52 compared to the second best score of ‘X’ to 26. Third and final Stroke which is a horizontal in is another match and it is accurately determined that letter is ‘A’. A more complex example can be taken to explain better. Suppose the first stroke entered is a right hand semicircle. After matching with the database, it is seen that alphabets ‘B’, ‘P’, ‘R’ and ‘O’ are constructed of similar strokes and allotted a score of 26, while rest alphabets are allotted zero. After the second stroke, which is a straight line, ‘B’, ‘P’ and ‘R’ are given another 26 while ‘O’ is not allotted any points. Now there is no further input from the user, leaving us with ‘B’, ‘P’ and ‘R’ having equal probability. From the database, we see that for character ‘B’ to be complete we need another semicircle and for character ‘R’ to be complete another straight line should have been entered. Therefore we can allocate another 26 points to ‘P’, because the number of stroke required for completion of the letter equals the number of inputs. To further improve accuracy, we use context processing by trying to place ‘B’, ‘P’ and ‘R’ in the word entered, we try to determine if the word formed is meaningful. We can clearly see that by entering a character stroke by stroke and allocating points to each alphabet we have significantly reduced error by eliminating the letters which have zero probability. A major benefit is that the system becomes more reliable when number of stroke entered is large(i.e. highly complex characters), because as we enter more strokes, the scoring system allows larger point difference between the most probable and the second most probable character. Another advantage is error detection as we can match the number of strokes entered and the score of the highest probable character. If it is more than the maximum possible score determined by the number of strokes, there has been an undesired stroke.

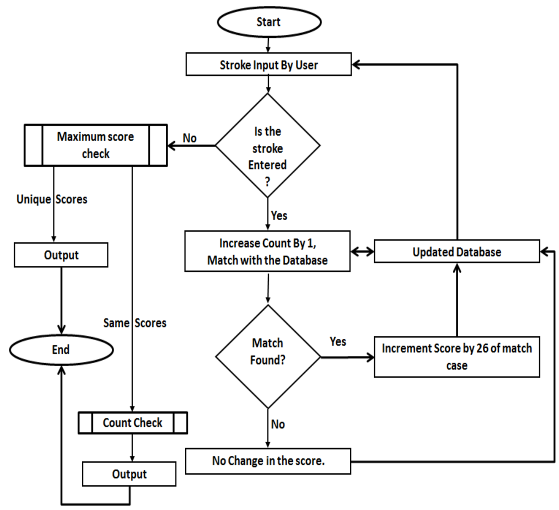

Fig: 8. Algorithm Flowchart

4. Word Context

To aid recognition of common words, we use a list of common words and their expected frequencies, if one is available. When a character is written, the letters surrounding it are used to find matching words within the frequency list. The samples of the potential letters are then rated by how frequently they appear. These ratings serve to boost recognition rates, but are not a major part of the final score. This component also suggests whether the character is likely to be a lower- or uppercase letter or a number judging from the preceding character.

5. Applications

5.1. *Signature Verifiers*

Signature verification refers to a specific class of automatic handwriting processing: the comparison of a test signature with one or a few reference specimens that have been collected as a user enrolls in a system. It requires the extraction of writer-specific information from the signature signal, irrespective of its handwritten content. This information has to be almost time-invariant and effectively discriminant. This problem has been a challenge for about three decades.

Signature verification tries mainly to exploit the singular, exclusive and personal character of the writing. In fact, signature verification presents a double challenge. The first is to verify that what has been signed corresponds to the unique characteristics of an individual, without necessarily caring about what was written. A failure in this context, i.e., the rejection of an authentic signature, is referred to as a Type I error. The second challenge is more demanding and consists of avoiding the acceptance of forgeries as being authentic. The second type of error is referred to as a Type II error. In the age of chip cards and the possibility of implanted ID transponders, on-line signature verification systems occupy a very specific niche among the identification systems. On the one hand, they differ from systems based on the possession of something (key, card, etc.) or the knowledge of something (passwords, personal information, etc.) because they rely on a specific, well-learnt gesture. On the other hand, they also differ from systems based on the biometric properties of an individual (fingerprints, voice print, retinal prints, etc.) because the signature is still the most socially and legally accepted means for identification. Its unique, self-initiated, motoric act provides an active means to simultaneously authenticate both a transaction and a transactioner[5].

6. Conclusions

We have developed a more practical offline handwriting character recognition technology based on exisiting technonology by adding new features of real time stroke by stroke elimination method. It is a hybrid adaptation of context processing and integration of adaptive classification (lower or higher order priority), which is also a new feature and realizes a higher recognition performance than these two methods individually. As a result, in one experiment, our proposed algorithm achieved a higher recognition accuracy. We are making continous efforts to improve the recognition accuracy and usability of hand writing interfaces.

7. References

1. Javier Calpe, Italo Medina, Alberto Carbajo, María José Martínez, "AD7879 Controller Enables Gesture Recognition on Resistive Touch Screens".
2. R. Plamondon, *"Online and off-line handwriting recognition: a comprehensive survey"* Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, Jan 2000.
3. Shashank Mathur, Vaibhav Aggarwal, Himanshu Joshi, Anil Ahlawat, "Offline handwriting recognition using Genetic Algorithm".
4. On-Line and Off-Line Handwriting Recognition:A Comprehensive Survey ReÂ jean Plamondon, Fellow, IEEE, and Sargur N. Srihari, Fellow,IEEE
5. CellWriter: Grid-Entry Handwriting Recognition Michael Levin,December 6, 2007 .