Sādhanā Vol. 28, Part 6, December 2003, pp. 1037–1046. © Printed in India

Recognition of Pitman shorthand text using tangent feature values at word level

P NAGABHUSHAN and S MURALI

Department of Studies in Computer Science, University of Mysore,
Mysore 570 006, India
e-mail: Pnagabhushan@hotmail.com; nymurali@yahoo.com

MS received 6 November 2000; revised 16 December 2002

Abstract. Recognition of text recorded in Pitman shorthand language (PSL) is an interesting research problem. Automatic reading of PSL and generating equivalent English text is very challenging. The most important task involved here is the accurate recognition of Pitman stroke patterns, which constitute "text" in PSL.

The paper describes automatic recognition of the strokes of the PSL at word level. A pen-down to pen-up sequence makes a stroke, which is a composition of primitives. The words are separated based on pen-down and pen-up points. The features that form a word (a stroke) are grouped first. Next, primitives and their sequence are identified and passed to a recognizer which identifies the word. A tangent-based vector through the contour of a stroke identifies the consonant primitives. Any other marks close to the stroke but not associated with the contour of a stroke represent the vowel markers.

Keywords. Pitman shorthand; pitman word (stroke); consonant primitives; vowel markers; tangents; contour traversal; thinning.

1. Introduction

Recognition of Pitman shorthand is a challenging pattern recognition problem because of the variety of pattern shapes involved with it (Chen & Lee 1992; Leedham & Nair 1992; Hemanth Kumar 1998). Pitman shorthand was invented by Sir Isaac Pitman in 1840 (Pitman 1989). In this method of documenting, speech is directly converted into phonetic strokes, where each phonetic stroke is a composition of consonants and vowels. These consonants are called consonant primitives or simply primitives. Basically there are 26 primitives. All 26 consonants are represented by means of simple geometric forms as shown in figure 1. All the words in shorthand are composition of mainly these primitives (Pitman 1989). Vowel

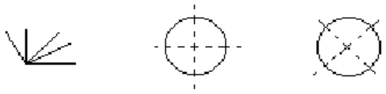


Figure 1. Geometric forms of the primitives used in PSL.

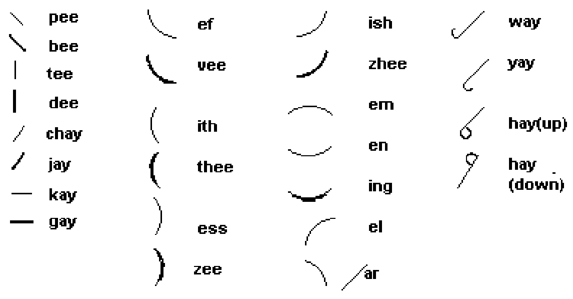


Figure 2. Primitives in Pitman shorthand representation.

markers are thick/thin dots/dashes and they are placed near a primitive (above, below, at the top, middle or end of the primitive), depending upon the phonetic-syntax of PSL.

Five straight-line primitives and eight curved primitives (arcs) make the fundamental set of primitives. These primitives are also characterized and identified with their thickness, i.e. strokes may be thin or thick. Four hook-structured primitives are used as consonants. Figure 2 gives the list of primitives and the corresponding phonetic composition.

In PSL, the primitives are concatenated to form a stroke to represent an English word. To recognise these strokes, we need to identify the primitives that compose a stroke (Hemanth Kumar 1998). The basic primitive patterns, as can be seen in figure 1, are straight lines, arcs and hooks. Thus we have to choose a technique that recognises these three types of fundamental shapes (Babaguchi & Aibara 1987; Rosenfield 1995) accurately.

In literature, very little information can be found concerned to Pitman shorthand recognition (Leedham & Qiao 1992; Hemanth Kumar 1998). Recognising primitives by the Hough transformation (Nagabhushan & Murali 2000) takes a lot of computational time and requires a large amount of memory. There are several techniques to recognise these basic patterns (lines, arcs) (Chen & Lee 1992; Anin *et al* 1997; Foggia *et al* 1997) individually. However, it is necessary to recognise the basic patterns when they are concatenated to form a stroke. These basic shapes are concatenated such that they 'grow' without overlapping. There is only one pen-up point for a stroke in a Pitman word. The next word commences with the next pen-down point.

We have proposed a method to identify fundamental shapes in a stroke by defining tangential vectors at two levels – tangent feature value (TFV) at the first level and derived feature vector (DFV) at the second level, ultimately resulting in a primitive vector (PV), which enables the actual recognition.

Section 2 explains a procedure for feature extraction and starting point identification. In § 3, a method for contour traversal and generation of TFVs is explained. In § 4 we explain a method for word separation. Section 5 presents a procedure to identify the corners in a stroke and a method to identify the primitives constituting a stroke. Finally, some of the examples taken from a stenographer's notes are illustrated.

2. Feature extraction

2.1 *Feature points identification*

A point (non-white pixel that contributes to make a stroke) in a PSL document image is represented by its (x, y) co-ordinates. A word (stroke) in PSL is a set of non-white pixels, which has a single pen-down point and a single pen-up point. Some English words in sentences

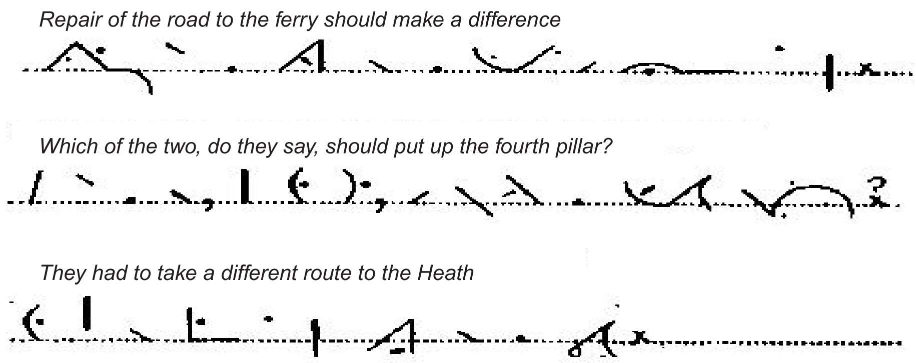

Repair of the road to the ferry should make a difference
Which of the two, do they say, should put up the fourth pillar?
They had to take a different route to the Heath

Figure 3. Pitman shorthand sentences for same English statements.

and their equivalent strokes in PSL are shown in figure 3. All the points between pen-down point and pen-up point are the feature points corresponding to a word. The first step in recognising a word and its primitives would be to have the skeleton of the image, which is a prerequisite for directional code measurement (Chen & Lee 1992). The handwritten PSL document is subjected to thinning to obtain the skeleton (Leedham & Qiao 1993; Tappert *et al* 1990). We have followed the thinning technique suggested by Chen & Lee (1992).

2.2 *Starting point detection*

It is necessary to identify the starting point of a stroke to initiate the contour traversal through a stroke. This is achieved by scanning the thinned image vertically in the x-y plane. A black pixel (which is part of the stroke) detected first is the starting point. Sometimes the vowel markers that are placed above the first primitive of the stroke may be detected first. If the first point detected is a vowel marker, the contour terminates after 6 or 7 pixels as shown in figure 4. It is assumed that a dash vowel will not take more than 6 or 7 pixels. A true starting point of a word will have a contour that sustains beyond 7 pixels.

3. Contour traversal and tangents as feature vector

3.1 *Contour traversal*

The sequence of primitives in a PSL word always proceeds or grows in one direction, i.e., there is no overlap of primitives. In other words, a word in PSL is an one-stroke word without any cross-over. Hence the contour traversal technique is adopted to traverse through a word. Hook

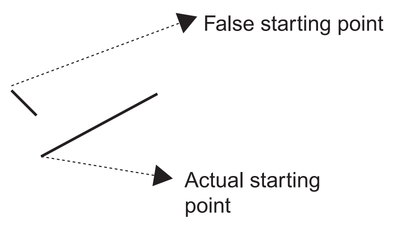
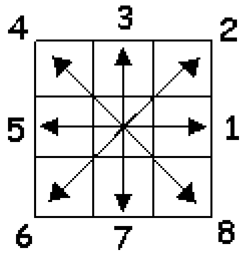


Figure 4. Starting point identification in a stroke.



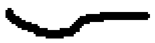


Figure 5. **(a)** Eight-direction code **(b)** PSL stroke with *en* and *kay* primitives. The direction codes for the figure 5(b) are 8818181811181111 111121111212121222311111111111111111111111.

primitives appear either at the beginning or at the end of the stroke, and they certainly do not hinder contour traversal. An 8-directional chain code method (Leedham & Qiao 1993; Amin *et al* 1997) is used to traverse along the contour of the word. Figure 5 shows the directional code and an example to illustrate the coding method for a word formed by the concatenation of two primitives *en* and *kay*.

The primitives in a stroke may be thin or thick. Concatenation of these two types of primitives is most common. Thick primitives are normally wider than thin primitives. If the thinning technique is applied to a stroke irrespective of type, the skeleton will be of single pixel width. If the image is considered a binary one, after thinning both types of primitives will have single pixel width skeletons. It is difficult to differentiate two different types of primitives. Normally, in handwritten documents, if a character is thick, the writer uses more pressure on his pencil than when writing thin primitives. Hence, thin and thick are differentiated by different levels in gray values rather than by their thickness.

3.2 *Tangent feature vector along the contour*

The direction between two successive pixels is represented by the chain code. It represents the direction of the second pixel with respect to the first one. The set of possible angles measured in degrees is {0, 45, 90, 135, 180, 225, 270, 315}, since there are only 8 directions possible as shown in figure 5a. The angle between the line formed by two successive pixels and the x-axis has a value that belongs to the above set.

In the proposed method, the primitives are recognised based on the direction of growth of the contour from the starting point.

One method to achieve this is by measuring the slopes. The slope between two points $\tan^{-1}[(Y2 - Y1)/(X2 - X1)]$ would give the angle subtended by the line joining them and the x-axis. If the slope between two successive points is considered, then the set of possible slopes is (0, 45, 90, 45).

The slope between two successive points gives the direction of the second point from the first. If the slope between a point and its successor to successor point were measured, it represents the direction of the third point from the first point. Since the three points are close to each other and consecutive, their contribution to the shape of the stroke is unique, i.e. they are almost linear with respect to the shape of the stroke. This linearity may exist for 'n' successive points. In a typical case of a primitive of length 18 pixels, the value of n is 4 or 5. In other words the consecutive 4 or 5 pixels in a primitive are linear. The slopes between $(p1, p2)$, $(p1, p3)$, $(p1, p4) \ldots$, $(p1, pn)$ take almost similar values as shown in figure 6. Such a vector of slopes is called an ITV (initial tangent vector).

The ITV$_i$ at an ith point on the contour is defined as the set of slopes of the secants $\approx$ tangents through the closely placed pair of points (P_i, P_j) for $j = i + 1, i + 2, \ldots i + n$. ($n \leq 5$ in this case).

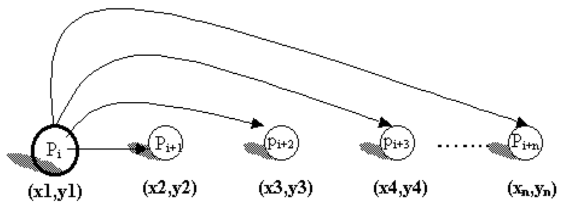


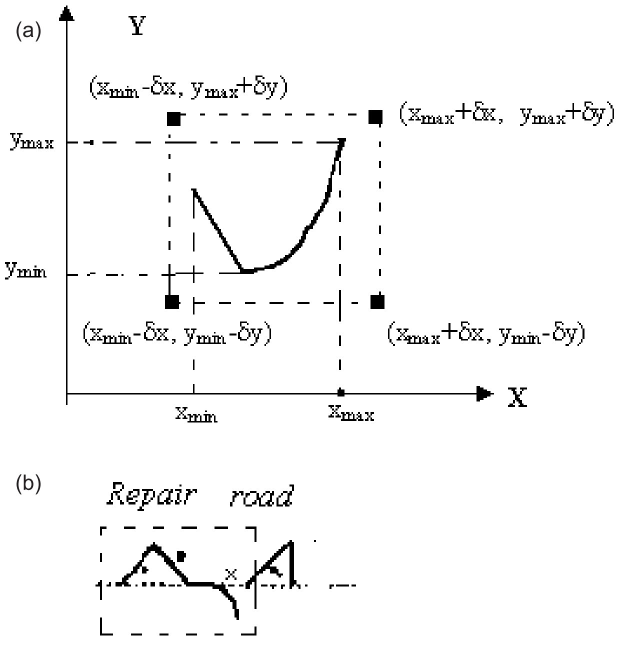
Figure 6. *n* successive points to obtain TFV.

The ITV$_i$ is indicised to represent a single value called TFV$_i$ – tangent feature value of the ith point, which in our case is the average of all elements in ITV$_i$.

4. Word separation

Typically shorthand is written on a special paper which has horizontal lines called reference lines. Strokes may be written over or through the line depending on the composition of primitives. A line may have 10 to 15 stroke patterns.

A word in PSL is a trace of points on the contour from a pen-up point to a pen-down point. Typically any word is a composition of primitives and vowels. It is necessary to separate the vowel markers corresponding to a word. To identify the vowel markers associated with a word, the area that covers the word (rectangle) is considered. Along the contour minimum and maximum values of x and y co-ordinates $x_{\max}$, $x_{\min}$, $y_{\max}$, $y_{\min}$ are measured. The rectangle with corner points $(x_{\min} - \delta x,\ y_{\min} - \delta y)$, $(x_{\max} + \delta x,\ y_{\min} - \delta y)(x_{\min} - \delta x,\ y_{\max} + \delta y)(x_{\max} + \delta x,\ y_{\max} + \delta y)$ is considered. δx and δy are the offset values that are used to accommodate vowels at the first or last position of the first and last primitives of the stroke respectively.



(a)
Y
(x_{min}-δx, y_{max}+δy)
(x_{max}+δx, y_{max}+δy)
y_{max}
y_{min}
(x_{min}-δx, y_{min}-δy)
(x_{max}+δx, y_{min}-δy)
X
x_{min}
x_{max}
(b)
Repair road

Figure 7. **(a)** A stroke and an imaginary box covering the stroke. **(b)** An example with overlapping box.

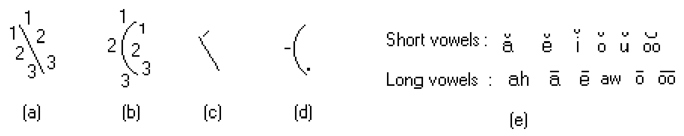


Figure 8. Decision tree for recognising primitives.

An example of a stroke with two primitives covered by a rectangle is shown in figure 7. In a typical PSL text, this rectangle may cover portions of other strokes, but these should be identified and discarded. If vowel markers of other strokes appear in this rectangle, these markers should not be mistaken as in association with the current stroke, since a vowel marker can appear only in 6 possible positions as shown in figure 8. An example with the portion of the second stroke *(road)* being included in the rectangle drawn for first stroke *(repair)* is shown in figure 7b. This included portion could be mistaken for a dash vowel marker with reference to the 3rd primitive of the stroke under consideration. However, this cannot actually happen since the expected vowel point is indicated by the 'X' mark in figure 7b, which is far away from the pseudo-dash. Thus the pseudo-dash is eliminated.

5. Primitive selection

5.1 *Corner detection*

The primitives in PSL text are normally of 18 to 23 pixels length with 150-dpi resolution and correspond to 2/3 of an inch. The arc primitives practically require more pixels than line primitives do. Length of a primitive is one of the features which may be used to differentiate one primitive from another. There is some variation between the length of the primitive in handwritten text and the corresponding standard length of the primitive. The lengths of the handwritten primitives vary significantly and sometimes they are too short to be recognised as primitives.

The length of the primitive as a feature is not enough to identify the meeting point of two primitives in a stroke, especially in handwritten text. In handwritten text the primitive lengths vary slightly from stroke to stroke. At the corner point (primitives meeting point) a TFV has significant change compared to its neighbours (neighbours of corner points, both before and after). By considering both length and TFVs, it is possible to detect corner points.

5.2 *Primitive recognition*

To recognise a primitive, it is divided into five blocks. Each block has m number of TFVs where $m = p/5$ (rounded off to an integer), where p is the number of pixels in the primitive. The number of TFVs in each block may vary by $\pm$ 1, to adjust the fractional part of $p/5$ to m blocks.

DFVi (derived feature vector) is the set of TFVs in a block and is defined as

$$\mathrm{DFV}_{i=1,2,3,4,5} = \{\mathrm{TFV}_{k+1}, \mathrm{TFV}_{k+2}, \ldots, \mathrm{TFV}_{k+m-1}\}, \text{ where } k = (i-1) * m.$$

The DFV$_i$ is indicised to represent a single value called BFV$_i$ – block feature value of the ith block, which in our case is the average of all elements in DFV$_i$.

The vector of BFVs is referred to as the PV (primitive's vector) that contains the features of a primitive which are unique to each primitive. This PV is the input to the recogniser. Figure 9 shows a primitive *ess* and its IFVs, TFVs, DFVs, BFVs and PV.

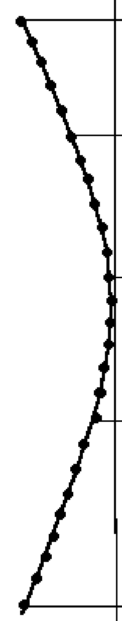
primitive ess	Position	Direction Code	IFV	TFV	DFV	BFV	PV
	67 91	8	{315.00,333.43,326.31,323.13,}	324.47			
	68 90	1	{360.00,333.43,326.31,323.13,}	335.72			
	69 90	8	{315.00,315.00,315.00,315.00,}	315.00	{324,335,315,312,310,302}	316	
	70 89	8	{315.00,315.00,315.00,306.87,}	312.97			
	71 88	8	{315.00,315.00,303.69,306.87,}	310.14			
	72 87	8	{315.00,296.56,303.69,296.56,}	302.95			
	73 86	7	{270.00,296.56,288.43,284.04,}	284.76			
	73 85	8	{315.00,296.56,288.43,296.56,}	299.14			
	74 84	7	{270.00,270.00,288.43,284.04,}	278.12	{284,299,278,284,296}	288	
	74 83	7	{270.00,296.56,288.43,284.04,}	284.76			
	74 82	8	{315.00,296.56,288.43,284.04,}	296.01			
	75 81	7	{270.00,270.00,270.00,270.00,}	270.00			
	75 80	7	{270.00,270.00,270.00,270.00,}	270.00			{316,288,267,248,229}
	75 79	7	{270.00,270.00,270.00,270.00,}	270.00	{270,270,270,266,261}	267	
	75 78	7	{270.00,270.00,270.00,255.96,}	266.49			
	75 77	7	{270.00,270.00,251.57,255.96,}	261.88			
	75 76	7	{270.00,243.44,251.57,255.96,}	255.24			
	75 75	6	{225.00,243.44,251.57,243.44,}	240.86			
	74 74	7	{270.00,270.00,251.57,255.96,}	261.88	{255,240,261,252,234}	248	
	74 73	7	{270.00,243.44,251.57,243.44,}	252.11			
	74 72	6	{225.00,243.44,236.31,233.13,}	234.47			
	73 71	7	{270.00,243.44,236.31,233.13,}	245.72			
	73 70	6	{225.00,225.00,225.00,225.00,}	225.00			
	72 69	6	{225.00,225.00,225.00,}	225.00	{245,225,225,225,225}	229	
	71 68	6	{225.00,225.00,}	225.00			
	70 67	6	{225.00,}	225.00			

Figure 9. Primitive *ess* with starting point at (67, 91) in x-y plane and IFVs,TFVs,DFVs,BFVs and PV.

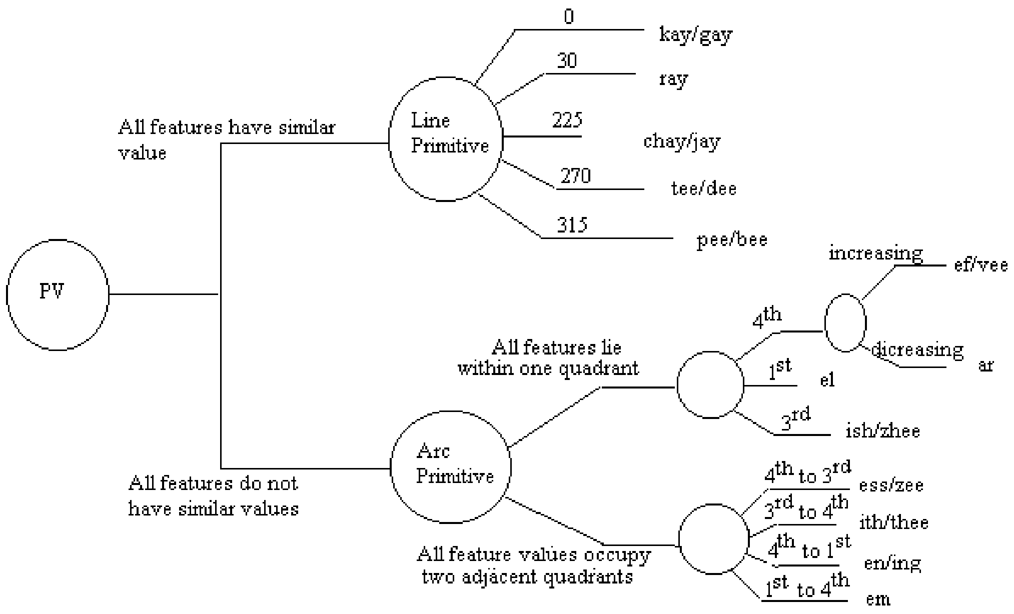


Figure 10. Vowels and their positions.

The PV of a primitive provides the input to a trained system that recognises the primitive. Figure 10 gives the decision tree for recognising a primitive based on PV. If the elements of PV are of the same numerical value or vary marginally ($\pm$ 5 in this case), then it can be concluded that a straight-line primitive exists, otherwise it is an arc primitive.

5.3 *Detection of vowels*

Vowels are appended to the consonants to make meaningful words. In Pitman shorthand there are two types of vowels – short and long. Thick dots and dashes represent long vowels, whereas thin dots and dashes represent short vowels. These vowels are placed at different positions near a consonant primitive, namely beginning, middle and end. These vowels are placed below or above the primitive to read either before or after the consonant. Figures 8a and 8b show the positions of vowel markers for the primitives *pee* and *ith*. The short vowel dash for the primitive *pee* is shown in figure 8c. The short vowel dash and long vowel dot for the primitive *ith* are shown in figure 8d. Short vowels and long vowels are shown in figure 8e.

To detect vowels we have created a simple data structure, which is a set of pixels, those connected and not part of a stroke, but lying close to the stroke (pixels used for consonants are all connected from pen-down to pen-up and the remaining ones are obviously vowel markers). The pixels which form a dash are grouped; similarly for a dot. A typical dash has a length of four to six pixels and dots are of length one to three pixels.

Any consonant primitive has a starting point, a middle point and an end point. Normally, vowels are written within 5-pixel distance from the beginning, middle or end of the consonant. Distances between a vowel's starting point with 3 different points (starting, middle and end) of the primitive are computed. This is continued for all combinations of vowels and primitives in a stroke. Vowel and primitive combinations with distance less than 5 pixels are considered.

6. Overall algorithm

Input: Single stroke image of 128×128 with 256 colours.
Output: Stroke primitives.

(1) Extract feature points from the image (for one line)
(2) Obtain skeleton of the image.
(3) Find next starting point of the stroke to initiate contour traversal and repeat step 3 to 9 till the end of line is encountered.
(4) From the starting point move to the next point and continue traversing till the end point. During each movement record next point co-ordinates (x, y) and direction code.
(5) Identify the feature points lie within the rectangle points $(x_{\min} - \delta x, y_{\min} - \delta y)$, $(x_{\max} + \delta x, y_{\min} - \delta y)(x_{\min} - \delta x, y_{\max} + \delta y)(x_{\max} + \delta x, y_{\max} + \delta y)$, where $x_{\max}, x_{\min}, y_{\max}, y_{\min}$ are minimum and maximum values of x and y along the contour and δx and δy are the offset values. These features are considered for vowel recognition.
(6) For each point on the contour perform the following.
    (a) ITV$_i$ (initial tangent vector) measured at every point on the contour is defined as the set of slopes of the secant $\approx$ tangent through the closely placed pair of points (P_i, P_j) for $j = i + 1, i + 2, \ldots i + n$.
    (b) For each vector calculate an index value which is the average of vector elements called TFV (tangent feature value).
(7) Along the contour identify any significant change in the TFV from that of its neighbours. Designate that point as the corner point.
(8) TFVs between the corner points are divided sequentially into 5 groups. The average of TFVs in the group is termed BFV (block feature value).
(9) The set of BFVs termed PV (primitive vector) is sent to the recogniser. (Distinct PV for every primitive.)

7. Implementation and experimental results

We have implemented and tested the algorithm using Turbo '*C*' language and used 256-colour images (Windows Bitmap standard), since thick and thin primitives are to be differentiated based on their gray levels. Handwritten samples are taken from a scanner with a resolution of 150 dpi.

In some of the handwritten curved primitives, a corner point may not be identified even after traversing a standard number (primitive's average length in pixels) of pixels along the primitive from the previous corner point. The average slope after this point should be observed even for small variations. Concatenation of primitives such as *pee* and *tee* in a word may pose a problem, since there is not much variation in the average slope at the meeting point of these two primitives.

The experiment is conducted on 500 handwritten words (strokes). These words are included in appended format to verify separation of words. We have tested this algorithm with approximately 5 to 6 strokes per line; 100 words with 2 primitives, 200 words each with 3 primitives and 4 primitives were taken for testing. The algorithm developed could recognise 377 of them exactly. The other 123 were recognised partially (some of the primitives in the words were recognised exactly). Table 1 gives performance analysis of the algorithm by considering 500 handwritten strokes.

Table 1. Performance analysis of the algorithm.

			Part. recogn.				
No. of primitives in the stroke	No. of strokes tested	No. of primitives recognised exactly	No. of primitives recognised 1	2	3	Completely unrecognised strokes	Successful recognition rate in percentage
2	100	85	9	–	–	6	85
3	200	159	5	26	–	10	80
4	200	133	9	24	22	12	67
Total	500	377	–	–	–	–	75

8. Conclusion

The proposed method of separating words along with its vowels is adequate for carrying out the task of recognising text written in PSL. Detection of consonant primitives and vowel markers in a word by using tangent-based feature vector approach is able to recognise most handwritten PSL words. The recognised consonant primitives and vowels are stored in a standard data structure suggested by Nagabhushan & Anami (1997). The recognition rate can be increased with some preprocessing activity after feature extraction, as the primitives set is limited (26 types). The efficiency of the algorithm can also be improved by having variable group size while calculating DTV's. The tangent-based feature vector method proposed here is also tested with English capital letters and the recognition rate is found to be greater than for the Pitman shorthand strokes.

References

Amin A, Kim S, Sammut C 1997 Hand printed Chinese character recognition via machine learning. *4th Int. Conf. on ICDAR*, vol. 1, pp 123–156

Babaguchi N, Aibara T 1987 Curvedness of a line picture. *J. Pattern Recogn.* 20: 273–280

Chen B, Lee J H 1992 Recognition of handwritten Chinese characters via short line segments. *J. Pattern Recogn.* 25: 543–552

Foggia P, Sansone C, Tortorella F 1997 Character Recognition by Geometrical Moments on Structural Decomposition. *4th Int. Conf. on ICDAR*, vol. 1, pp 6–10

Hemanth K G 1998 *On automation of text production from Pitman shorthand notes.* Ph D thesis University of Mysore, Mysore

Leedham C G, Nair A 1992 Evaluation of dynamic programming algorithms for the recognition of short forms in Pitman's shorthand. *J. Pattern Recogn. Lett.* 13: 343–350

Leedham C G, Qiao 1993 Segmentation and recognition of handwritten Pitman shorthand outlines using an interactive heurtistic search. *J. Pattern Recogn.* 26: 433–441

Nagabhushan P, Anami B S 2000 A knowledge-based approach for recognition of grammalogues and punctuation symbols useful in automata. National Conference on Recent trends in Advanced Computing, Tirunelveli, pp 173–185

Nagabhushan P, Murali S 2000 Linear Hough transformation for recognition of stroke primitives in Pitman shorthand text, National Conference on Recent trends in Advanced Computing, Tirunelveli, pp 126–128

Pitman I 1989 *Pitman shorthand instructor and key* (Wheeler)

Rosenfield A 1995 Geometric properties of set of lines. *J. Pattern Recogn. Lett.* 16: 643–648

Tappert C C, Suen C Y, Wadhara T 1990 The state of the art in on-line handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 12: 1029–1058