PATRAM - A Handwritten Word Processor for Indian Languages

Kamesh Madduri, K.H. Aparna, V.S. Chakravarthy
Indian Institute of Technology Madras
{kamesh, ee01m03, schakra}@ee.iitm.ernet.in

Abstract

In this paper, we present PATRAM, a comprehensive handwritten wordprocessing system specially designed for Indian languages. In typical existing ink capture programs, handwriting is entered in a 'paint-on-a-wall' style without any syntactical structure imposed on the handwritten text. The PATRAM document page is classified into text (ruled space) and drawing (unruled space) areas. While inking the text region, the writer stays within the lines. The drawing region does not have such a restriction. While this arrangement accomodates unconstrained inking (in drawing area), it provides a more intuitive and convenient editing style for the text area. The standard operations including Insert, Cut, Copy, Paste, Delete, Select, Justify etc. are defined which operate on text or drawing objects in characteristic fashion. The document is saved in a specific format (.pat) which makes off-line character recognition possible. We finally demonstrate recognition capability in Tamil language.

1. Introduction

Handwritten character recognition (HCR) has assumed a lot of significance in recent times with the advent of PDAs and tablet PCs. These devices provide the feature of 'taking down notes' - i.e., the user can store data in his own handwriting using an electonic pen or a stylus as the input device [1]. However, most state-of-the-art systems pursue a 'write anywhere' philosophy encouraging unconstrained inking. Such an approach may be natural and convenient for short documents ('notes') but is not conducive to standard editing operations (copy, paste delete etc) which are the very strength of word processing. Pen-based input is a particular attractive alternative to keyboard input for Indian languages. However, until HCR in Indian languages reaches maturity, we foresee a need for a purely handwriting based approach to word-processing. We thus envisage a 'middle path' between the rigid structure of tranditonal text-based documents, with its powerful editing capabilities, and the naturalness and ease of pen input, as a temporary exigency for some of the problems that beset Indian language word-processing.

PATRAM is a completely handwritten wordprocessing program for Indian languages. The data hierarchy used by PATRAM enables most of the standard functions like Insert, Cut, Paste exactly as they behave in a font-based wordprocessor. A novel feature of PATRAM which distinguishes it from other ink-capture programs is that in PATRAM, the document page is pre-classified into text and drawing regions. In the text portion, the writer stays (or is forced to stay) within the lines. The structure imposed thus is conducive to intuitive text-editing: when a text portion is deleted, the content below (to the right of) it scrolls up (to the left), which does not happen in usual ink-capture programs. Our intuition or belief that a text-like editing capability is essential to any handwritten editor particularly while dealing with large documents is the basis of PATRAM.

PATRAM is also designed to serve as a generic word-processor for Indian languages. Presently, word-processing in Indian languages is a vexing experience, considering the restriction on use of the regular keyboard, designed for English. Elaborate keyboard mapping systems are normally used, but these are very inconvenient. A comfortable solution would be to let the user write in a natural, normal fashion using a suitable pen-like device and let the computer do the rest. That transfers the burden of learning keyboard mappings from the user to the computer. In the current work, we have tried out data-processing in Tamil script, a popular language in South India. The language recognition is carried out using the Tamil language recognizer presented in [6].

This paper is organized as follows: Section 2 explains the framework for representing handwritten data. Section 3 introduces PATRAM and its features. The paper concludes with a discussion of future plans in Section 4.

2. Document Structure

The following concepts and data entities have been designed to provide a unified framework for all Indian scripts. Hence all the properties may not be relevant to every script. Some definitions are in line.

Stroke: The trajectory of the pen between a pen-down event and a pen-up event. All the unique strokes of a script are manually identified and given unique labels. It is the smallest physically identifiable unit in online handwriting.
Proximity: Criteria for spatial proximity are necessary for grouping strokes into larger structures. There are 3 rules for representing 'proximity' of strokes.
• Contact or nearness: two strokes are proximal if they intersect, make contact or if the pair of nearest points on the respective strokes are closer than a threshold value.
• Enclosure: A stroke encloses another stroke (this situation does not arise in Tamil but is useful for other Indian scripts)
• X-overlap: Two strokes overlap in horizontal direction. Note that this proximity rule is a less restricted one than the two above and subsumes them.
Temporal proximity is not considered at present.
Stroke Group: Any random collection of strokes. Usually it refers to a set of strokes which form a unit with lesser significance than a character. A stroke group can also have a label.
In practice, a Stroke Group object may be used to group strokes in the following 2 ways:
• To group syntactically meaningful subsets of the full set of strokes forming a composite character (such grouping turns out to be more useful in formulating compositional rules for other Indian scripts like Hindi, Telugu etc.).
• To group physically connected ('proximal'') subset of strokes forming a composite character.
Thus a Stroke Group is meant to be used in a flexible way depending on the context as illustrated above.
Horizontal Block: This is a set of strokes grouped with the X-overlap proximity criterion described above. This definition is necessary because very often in Indian scripts all the strokes that comprise a composite character form a horizontal block.
Character: A character is the smallest segment of handwriting which can be associated with a syntactic code like, say, the ISCII code. Typically, it is one or several contiguous horizontal blocks. Note that all the characters in a line have the same vertical span ('linespace') but may however have different horizontal spans ('xspans'). The character code used here is the Indian Script Code for Information Interchange (ISCII), which is specially designed for representing Indian script systems.
Word: A word is an array of characters. The spacing between two characters determines the beginning of a new word or continuation of a word.
Line: A horizontal array of words interspersed with spaces.
Page: A vertical array of lines constitute a page. PATRAM also allows for insertion of 'drawing areas' which are associated with the page structure.
Document: An array of pages.

Each object in the data hierarchy above is represented in a coordinate system defined by the immediate parent object, e.g., a 'character' object is represented in the coordinate system defined by the 'word' object that contains it. Such a nested representation is necessary to easlily delete and insert data objects in any context.

The resemblence of the above structure to other standard pen data formats like [2] or [3] is obvious. A new feature in our representation is the strokegroup which is an intermediate level between a character and a stroke. Our hierarchy does not reach the substroke level.

3. PATRAM Features

3.1. Application Layout

PATRAM has been developed using GUIDE - the GUI Development Environment of MATLAB (version 6.1). Fig. 1 gives a screen-shot of PATRAM menu. The File and Edit options occupy the top part of the application and the central part of the application is the 'active area'. Fig. 2 shows some handwritten data and an inserted drawing area in the active area. The File I/O commands and the Edit commands can be executed from the menu as well as clicking on the corresponding icons in the Tool Bar. The top-left toolbar consists of various File and Edit icons. The 'Status Bar' in the center indicates the current mode of operation - Insert or Modify. The Insert and Select icons are present on top of the Status Bar. To the right, we have the current page and line number indicator and a tool bar for navigating through the document.

3.2. File I/O

The file I/O functions include Open, Save, Save As and Close options. The entered data is saved as a format defined by us (.pat) whenever the Save command is executed. The name of the currently open file and the Save status are indicated on the application's titlebar.

3.3. Document subregions

Document content is segmented into two regions: text and drawing regions (Fig. 2). Each of these subregion categories has a specific data hierarchy. The text subregion has the hierarchy of page-line-word-character strokegroup-stroke. The drawing area presently has a simple structure - an entire drawing area is represented as a stroke group. We plan to introduce a more elaborate hierarchy in the drawing area structure in future.

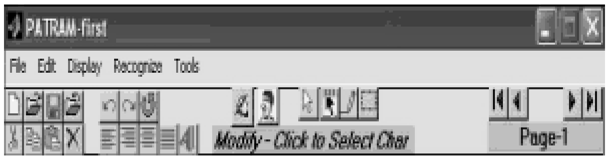

PATRAM-first
File Edit Display Recognize Tools
Modify - Click to Select Char
Page-1

Figure 1. PATRAM menu bar

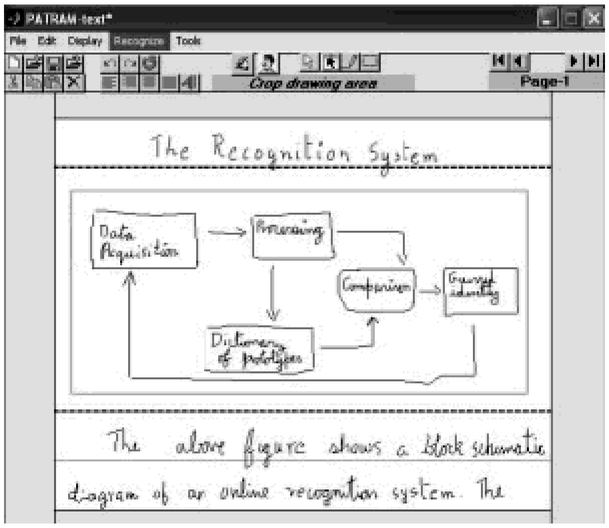


Figure 2. A PATRAM document page with text and drawing (flow-chart) regions

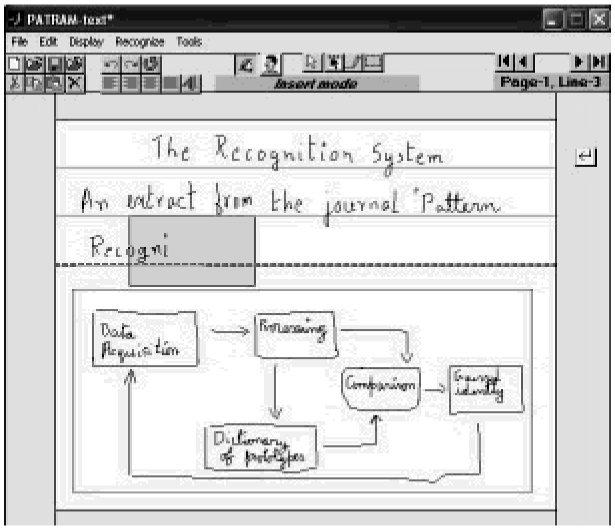


Figure 3. Insert mode - 'writeable area'

In Insert mode, text can be entered only in the highlighted box ('the writeable area'). The stroke being written must be entirely confined to the writeable area; else it will not be added to the document.

3.4. Modes of Operation

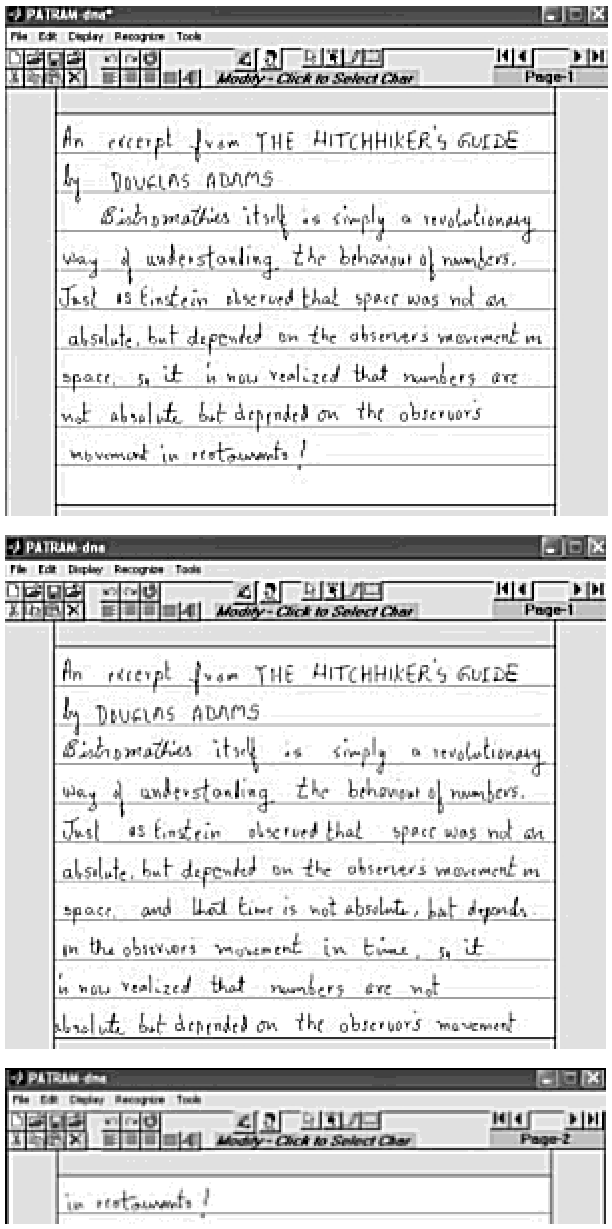
Two basic modes of operations are defined for PATRAM - the Insert mode and the Edit/Modify mode. Text and drawings can be inserted in the Insert mode. All the Edit operations like Cut, Copy etc. are disabled in the Insert mode. The default mode of operation is the Edit/Modify mode. The Select function is extensively used for editing.

3.4.1. INSERT mode. The Insert mode in PATRAM is used for entering text and drawings in the document (see Fig. 2). The Insert icon is a toggle button which activates the Insert mode (for text) when pressed. To insert drawing, the user needs to click on the picture icon next to the Insert-text button.

Inserting Text: When in Insert mode, the first step is to select the point of insertion, i.e., the point where the writer starts writing from. The default number of lines in a document are eight. Once the user clicks in the active area, the line number in the current page is identified and page/line display bar is updated. Once the Insert point is selected, the user can write only inside a selected area (the 'writeable area') on the screen. This region is shown in a highlighted rectangle as in Fig. 3. Strokes falling outside this highlighted area are erased and not added to the document structure. If the writer desires to insert spaces between words, he or she can click on the Insert toggle button and select a new point of insertion.

As the writer keeps writing, the document data structure is updated after every stroke. The writeable area is also updated and moves along with the pen. This feature is reminiscent of Giovanni Seni's 'Treadmill Ink' [4], which is meant specifically to enable continuous pen input in small devices. In our case, the moving writeable are' is necessary to force the user to stay between the lines. The 'xspan' (associated with a character, word and a line) is calculated and updated after every new stroke. Depending on the language, the default 'character to character' and 'word to word' gaps are applied. The bounds of the previous stroke and the current stroke are calculated and used to determine the gap between the strokes. If the gap is greater than the 'character to character' gap, the current character is wound up and a new character is initiated. Similarly, a new word is created when the gap is greater than the 'word to word' gap.

When the writer reaches the end of a line, i.e., if the previous stroke's coordinates are close to the boundary of the active area, the line is updated and the writeable area window moves to the next line. Similarly, when the user is on the last line of the page, a new page is created and the writeable area is updated.


PATRAM-dna*
File Edit Display Recognize Tools
Modify - Click to Select Char
Page-1
PATRAM-dna
Page-1
PATRAM-dna
Page-2

An excerpt from THE HITCHHIKER's GUIDE
by DOUGLAS ADAMS
Bistromathics itself is simply a revolutionary
way of understanding the behaviour of numbers.
Just as Einstein observed that space was not an
absolute, but depended on the observer's movement in
space, so it is now realized that numbers are
not absolute but depended on the observor's
movement in restaurants!

An excerpt from THE HITCHHIKER's GUIDE
by DOUGLAS ADAMS
Bistromathics itself is simply a revolutionary
way of understanding the behaviour of numbers.
Just as Einstein observed that space was not an
absolute, but depended on the observer's movement in
space, and that time is not absolute, but depends
in the observors movement in time, so it
is now realized that numbers are not
absolute but depended on the observor's movement

in restaurants!

Figure 4. An illustration of Insert operation

The phrase *and that time is ... in time* was inserted into the passage shown in the top figure. Note that the text scrolled down and two words in the passage have actually moved to the next page (the next two figures).

Another issue to be taken care of is that when the Insert point is above a picture or already entered data, the display area and the data structure must be updated accordingly. The old data has to be moved - to the right in the same line, to the next line or to a new page - in order to accommodate the newly inserted data (see Fig. 4). Similarly, if the user clicks on an existing word, the insert cursor should move to the end of the current word. The current version of PATRAM implements both these features. Another useful feature in the Insert mode is that, on pressing the return key, existing data is shifted down by one line and the cursor is moved to the next line.

Inserting drawing: To insert a drawing, the first step is to select the Insert point as in the case of handwriting insertion. This point specifies the top-left corner of the drawing area. Next, a dialog box is initiated which prompts the writer for height and width of the drawing area and creates one accordingly. The lines of the page overlapping with the area are identified and if any data is already present, it is shifted down. Fig. 2 shows a drawing area occupying lines 2-6 of the document page. We are currently working on the 'resize' and 'crop' functions on the drawing area. The existing Edit operations can be applied on the drawing area as a whole as well as on individual entries in the drawing area.

Our main aim of introducing a drawing area has been to facilitate saving data like formulae, drawings or pictures along with handwritten data. Exporting images from external files to the drawing area is a feature that will be added in future. To draw in the drawing area, the user has to click on the 'paintbrush' icon in the toolbar and then select the active drawing area, and then start drawing. The lines occupied by the drawing area are flagged so that they are not passed for recognition.

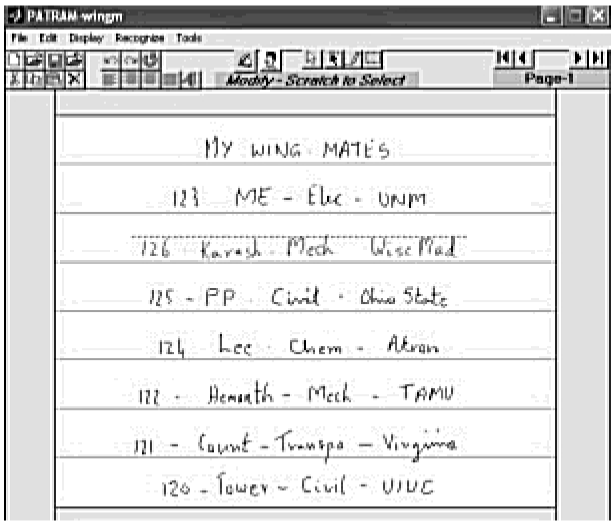
3.4.2. EDIT Operations. The Edit/Modify mode is the default mode of operation of PATRAM. All the Edit operations are carried out at the word level, i. e., only a single word or a group of words can be copied, pasted etc. The key feature that enables various editing operations is the select feature.

Select options: The Edit functions for handwritten data - Cut, Copy, Paste, Justify etc. are enabled only when data is selected. By default, no data is selected and the default select mode is the 'Select character' mode. In all select options, the selected data is highlighted in red. Also, when new data is selected, the previously selected data is deselected first.

• *Select Character*: This enables the user to select a specific character in a word. If it is cut or copied, it is then treated as a new word and not as a character any more. When a character is deleted from a word, the character array of the word is also simultaneously updated.

• *Select Word*: This option enables the user to select a single word by clicking on any character of the word.

• *Scratch select*: This option is used to select multiple words on the same line. Any horizontal line drawn by the user which overlaps with one or more words on a line is considered 'scratching' and selects the scratched words. (see Fig. 5)


PATRAM-wingm
File Edit Display Recognize Tools
Modify - Scratch to Select
Page-1
MY WING MATES
123 ME - Elec - UNM
126 Karesh - Mech WiscMad
125 - PP Civil - Ohio State
124 Lee Chem - Akron
122 - Hemanth - Mech - TAMU
121 - Count - Transpo - Virginia
120 - Tower - Civil - UIUC

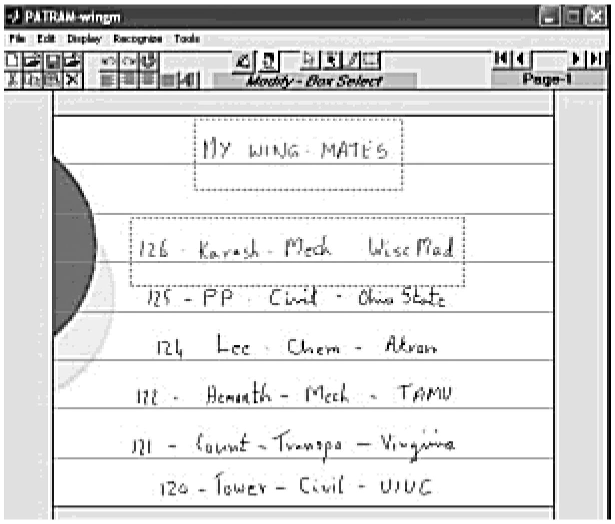

PATRAM-wingm
File Edit Display Recognize Tools
Modify - Box Select
Page-1
MY WING MATES
126 - Karesh - Mech WiscMad
125 - PP Civil - Ohio State
124 Lee Chem - Akron
122 - Hemanth - Mech - TAMU
121 - Count - Transpo - Virginia
120 - Tower - Civil - UIUC

Figure 5. Illustrations of Scratch (top) and Box select options

• *Box Select*: This function is used to select multiple words spanning more than one line. The line drawn by the user serves as the diagonal of a selection rectangle and all the words completely enclosed in this rectangle are selected.

Cut, Copy, Paste and Erase: These functions operate in an intuitive way in a usual sense. When a portion of a document is cut or deleted the portion below (or to the right) moves up (or to the left) to fill the blank space. Similarly when a paste function is run, the content below (or to the right) is moved down (or pushed to the right) appropriately.

Align and Justify: The left align, right align and center functions (Fig. 6) give the user the option to set the word to word gaps and also organize the handwriting in a better manner. If data from multiple lines is selected and one of these commands is executed, then the corresponding lines are aligned accordingly. Otherwise, the user can specify the lines to be aligned or justified in a dialog box.

Undo and Redo: A one-step Undo and Redo feature has also been implemented.

3.5 The 'Beautify' feature

One of the main motivations behind the creation of PATRAM is to present it as a cyber alternative to letter-writing. With increasing use of telephony and email, (handwritten) letter writing is fast becoming an obsolete tradition. Along with it all the ethos that surrounds the act of writing a letter with one's own hand, and the pleasant wait for a letter from a dear one, is also becoming socially irrelevant. We strongly intend to revive this human dimension of letter writing through PATRAM.

To this end, in the future versions of PATRAM, we wish to present visually pleasing background themes (watermarks) in the active area, similar to 'design templates' in Microsoft PowerPoint. The themes will be designed to suit various social occasions or contexts. The writer's stroke data will be mildly preprocessed so as to give a more beautiful, calligraphic appearance. While the essence of an individual's writing style is preserved, minimal treatment is given to it so as to make it look more appealing.

The beautification described by Julia & Faure is used in the process of graphical design using pen input [5]. Here we intend to use beautification to give a minimal calligraphic quality to a person's handwriting, which adds grace to a handwritten letter.

3.6. Character Recognition for Tamil

The Recognition feature is currently operational for handwritten data in the Tamil script (Fig.7). A preliminary version of this algorithm has been discussed in [6]. The entire document is passed as an input to the recognizer. Since the document structure does not depend on the language, the recognition and data capture parts are truly independent.

4. Conclusion

We present PATRAM, a completely handwritten word-processing program for Indian languages. Since the document page in PATRAM is segmented into text and drawing areas, a more convenient, intuitive editing is possible in the text area, while the drawing area allows unconstrained inking.

One of the prime motivations behind creation of PATRAM (the word means 'letter' in Sanskrit) is to

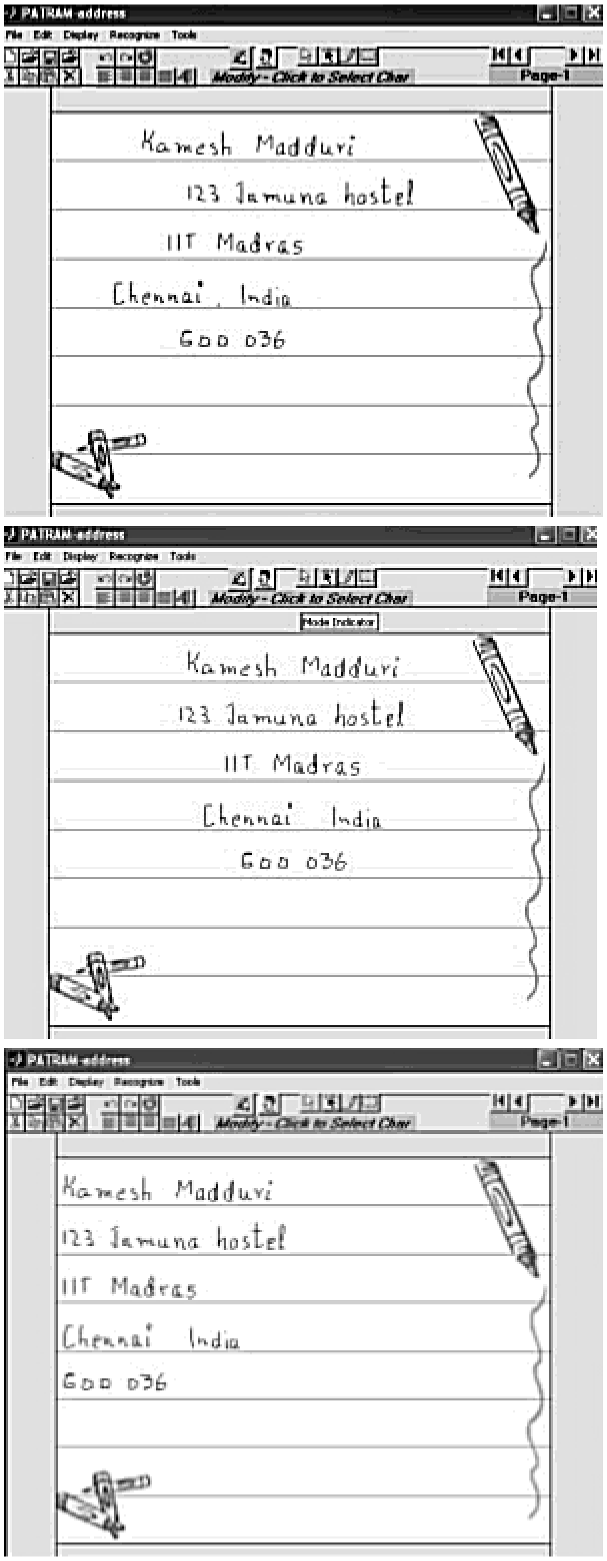


Figure 6. Illustration of the the Left align and center align options on unaligned data

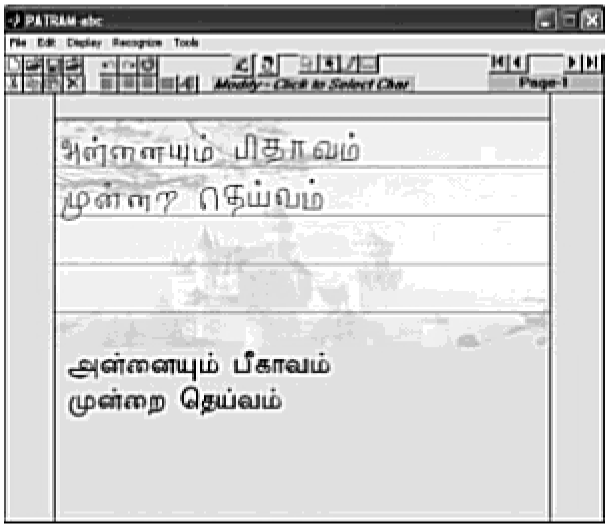

PATRAM-abc
File Edit Display Recognize Tools
Modify - Click to Select Char
Page-1
அன்னையும் பிதாவும்
முன்றை தெய்வம்

Figure 7. Character Recognition in Tamil

shape it as a tool for reviving the diminishing tradition of letter-writing. Future versions of PATRAM will have elegant, aesthetic and ornate document pages. We also wish to develop algorithms for 'beautifying' handwritten text, where minimal treatment is given to enhance appearance while preserving the personal essence. The PATRAM document, saved in .pat format, can be submitted for character recognition in an offline fashion. The current version has this capability only for Tamil. Future versions will progressively include all Indian languages and also multi-lingual support.

5. References

[1] R. Plamondon and S.N. Srihari, “On-line and Off-line Handwriting Recognition: A comprehensive survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence,” Vol. 22, No. 1, January, 2000.

[2] Ink Markup Language, http://www.w3.org/TR/InkML

[3] Guyon, I. and Schomaker, L. and Plamondon, R. and Liberman, R.and Janet, S, “UNIPEN project of on-line data exchange and recognizer benchmarks,” In Proc. ICPR '94., pp 29-33, Jerusalem, Israel.

[4] Giovanni Seni, “TreadMill Ink - Enabling Continuous Pen Input on Small Devices,” http://hci.stanford.edu/cs547/abstracts/01-02/020412-seni.html.

[5] L. Julia, C. Faure, “Pattern recognition and beautification for a pen based interface,” ICDAR'95, August 14 - 15, 1995, Montréal, Canada.

[6] B.J. Manikandan, Gowrishankar, V. Anoop, A. Datta, and V. S. Chakravarthy, “LEKHAK: A System for Online Recognition of Handwritten Tamil Characters,” International Conference on Natural Language Processing (ICON), December 2002.