International Journal of Image Processing (IJIP)

Book: 2008 Volume 2, Issue 3
Publishing Date: 30-06-2008
Proceedings
ISSN (Online): 1985-2304

IJIP Journal is a part of CSC Publishers
http://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers

Table of Contents

Volume 2, Issue 3, May/June 2008.

MAHII: Machine And Human Interactive Interface

G. Sahoo drgsahoo@yahoo.com
Department of Computer Science & Engineering
Birla Institute of Technology, Mesra,
Ranchi: 835 215, India

Bhupesh Kumar Singh kumar.bhupesh04@gmail.com
Department of Computer Science & Engineering
Lingaya's Institute of Mgt. & Technology,
Faridabad: 121002, India

Abstract

Free-style sketching is more natural than drawing a sketch using mouse and palette based tool. A number of sketching systems have been developed but not much attention is paid towards friendliness of the system. We present here an efficient Sketching system MAHII for sketching the strokes from multiple domains. The proposed paper demonstrates how a sketching system can be made user friendly and user interactive too. We also present the architecture of our system that solves many flaws encountered in the previously built systems. The interface should be such that it minimizes the gap between a human being and a machine. Our system provides a robust architecture that proves to be efficient. It also allows the user to sketch the diagram freely. And at the same time it edits the sketches as well as recognizes the sketches efficiently. This paper represents more robust and natural environment to the user. The MAHII provides the user an interface that is built in accordance with a novice user. MAHII provides free style sketching, recognizes the sketch produced by the user with accurate results.

Keywords: MAHII, sketch, interface, stroke, multi-domain, Heuristics.

1. INTRODUCTION

Sketching and drawing are being used in various aspects of life. In the recent years, many sketching systems have been developed, but there is always a gap between how a user draws a sketch and how a machine interprets the sketch. A little has been thought about the development of interactive human-machine interface. However, for some specific purpose many pen-based sketching systems and sketch recognition systems have been developed in the literature. But these systems suffer from the basic problem of naturalness of the sketching environment [2]. The sketching interface must be such that it allows the user to draw sketches freely and naturally. It must know when the system should take input from the user, edit the sketch as well as when it should recognize the sketch.

For sketching and recognizing the user drawn sketches, we propose here a system known as Machine And Human Interactive Interface (MAHII). The proposed system gives a solution to the problems that have been encountered in previously built sketching interfaces. Further, our system works on multi-domain still maintaining the efficiency and naturalness of the system. We present the interface architecture of our system whose framework has already been explained in our earlier work [1]. This system MAHII is built so that it proves to be useful to a computer-expert and even to a novice person. It contains some of the known commands of computer as well as normal behavioral commands that a layman can use in sketching the diagrams. The System MAHII also gives the user more liberty to draw sketches freely and naturally. Moreover, the user need not to learn every capability of the system but still he can draw the sketch and get accurate results of recognition.

The paper is organized as follows. In the next section we discuss the design issues of the sketching systems followed by a brief overview of the proposed system. Section 4 and 5 take care of the presenting the abstract view and layered architecture of the proposed system. Section 6 describes the interface commands followed by experiments and user feedback in the section 7. The comparison study has been discussed in section 8 followed by the conclusion and future work in the following sections.

2. DESIGN ISSUES

Various design issues have been predicted from previously sketch-based systems. We arrive at the following design issues that can help in overcoming these issues successfully.

- i) The system should display recognition result only when the user completes sketching.
- ii) The system should have some editing layers so that an editing layer can perform editing after every sketch input
- iii) The recognition should take place continuously and simultaneously but the result can be displayed only when the waiting time of the user is more than average waiting time between strokes.
- iv) The recognition process should be restricted to a single domain until automatic domain detection becomes feasible.

This can be achieved by the use of some interfacing mechanism, which is our main concern in this work. The primary goal of the interfacing mechanism is to connect the domain class description to the recognition engine only when its few corresponding elements are recognized by the recognition engine through Heuristic engine using a database containing shapes. With or without giving more stress to the recognition efficiency, one should go for describing a system that is more natural having user-friendly environment. Further, the system should posses a powerful and robust architecture and can be more useful for the user.

3. THE PROPOSED SYSTEM

In this section we give the overview of our proposed system called Machine And Human Interactive Interface (MAHII). By defining stroke as a set of points from pen-up, we categories strokes that are drawn or sketched as:

1. Single stroke input
2. Multi strokes input

And then we say that

- i) The system MAHII recognizes both single stroke input and multi-strokes input as well.
- ii) Further, it provides a new style of interface that takes care of human- machine interaction as well as free-style sketch recognition.
- iii) Even by considering a single window for both input and output in our designing process we still maintain a low complexity regarding the architecture of the system. For this we provide a three-layered architecture that we will explain in subsequent sections.

In our construction, we consider two modes – sketch mode and edit mode. MAHII provides the user more friendly environment because it does not require the user to select modes –sketch mode or editing mode. When the user draws sketches, the system shows him the edited images simultaneously based on which he/she can modify the respective sketch. So, he/she does not need to switch between sketch mode and edit mode [4], [13]. User can draw whenever he wishes to draw. MAHII can work in single domain as well as multi-domains. Even the user is not required to select the domain which he wants to draw in. For this, we provide domain class descriptions for each domain and an interface mechanism through which the system links to the particular domain automatically. MAHII automates all the functionalities. MAHII does not use specialized tools like palette based tools etc. and provides features for novice as well as expert users. So, it fulfils the demands of all types of users.

Again it can be pointed out here that some work has been done on Human machine interaction [5] but in solitude; similarly many pen-based sketching systems have been developed but without paying much attention towards human machine interaction. However, the proposed system explores the user aspects of a recognition based sketching system that efficiently recognizes the naturally drawn sketches and displays the result on the same window. MAHII reduces the complexity and increases the efficiency of the sketching system and gives the result with accuracy. It changes itself according to the user environment.

4. ABSTRACT VIEW OF THE SYSTEM FRAMEWORK

The system framework (Figure 1) that has been discussed in [1] gives the information about the working of the system. Now, this framework is implemented as the architecture of the system and the corresponding interface is named as MAHII. The architecture is explained below.

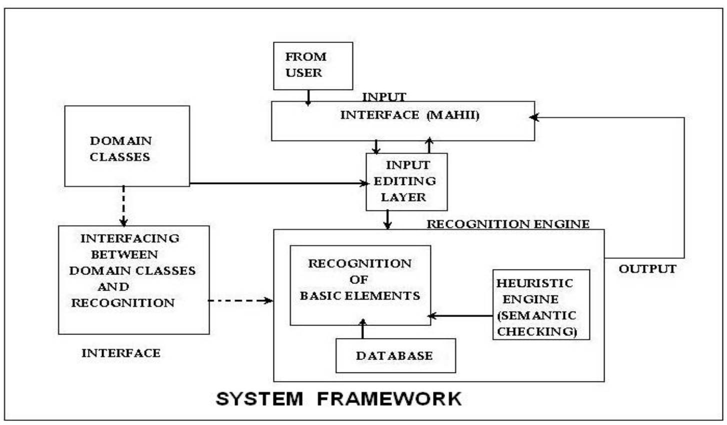


FIGURE 1: System framework

User provides input to the system by drawing sketches on the MAHII. This interface gets the sketch input from the user and sends it to the input refining block where input is refined by removing noise from the sketch input and then it is sent to the recognition engine as well as to the MAHII window. The edited or refined sketch is sent back to MAHII window because it displays the user the edited version of the sketch drawn at previous time instant. If it is not according to the user’s requirement, user can change it by using simple commands that are of common behavior. We explain such commands in a later section. When the edited input is sent to MAHII, it is also sent to the recognition engine also. After recognition, the output of the recognition is sent to the MAHII window. The flexibility and ease use of this system reflects here that user needs not to know when he should change the sketch mode, edit mode or recognition mode. The system takes care of these things by itself.

If a user draws a stroke say in 2 seconds and next stroke after 4 seconds then he draws the third stroke after 5 seconds and so on, the system takes the average of the first few time gaps between the consecutive strokes. Thus, it calculates the average of time difference then sets the systems' recognition time greater than the calculated average waiting time. The system does this for every user. So, if a user is faster in sketching, the system provides him results in less time and for a novice user, it displays the result according to the time taken by the user to draw next stroke. Thus, the system changes itself according to the user. It reduces the problem when the system should or should not recognize sketches.

The interface, which we represent in the framework that is between domain class description and recognition engine, helps the system to work in a multi-domain environment while it works actually over a single domain. In comparison to the earlier multi-domain sketch systems [4], [12], [13] where the user had to select explicitly the domain that he wants to use, MAHII does not impose any such requirement. The recognition engine actually requires the domain descriptions. Moreover, the database contains knowledge about recognized shapes and their particular domain. So, when the system needs to select a particular domain, it identifies few basic shapes or elements that are parts of a particular domain followed by the class description with the recognition engine. By using this modular approach we not only increase the system efficiency, naturalness of the interface but also reduce the complexity of the system and enable the system to work in a multi-domain environment. For an example, let us consider three domains as one domain for mechanical engineering, one domain for electronic circuit designing and one domain for dancing choreography. Initially, the user draws the input in a particular domain and gets the edited and refined sketch at the next instant of time. If he draws 3-4 basic shapes, the system relates them to the particular domain and gets to know that it belongs to, for instance, choreography domain. It gets this basic information, of course, from the corresponding database. Then, it interfaces the choreography domain with the recognition engine.

5. LAYERED ARCHITECTURE OF MAHII

The MAHII interface uses modular approach. Basically the architecture of MAHII consists of three layers namely

i) Input layer or sketching layer or presentation layer
ii) Editing layer or middle layer or refining layer
iii) Core layer or main processing layer or recognition layer

and are depicted in Figure 2.

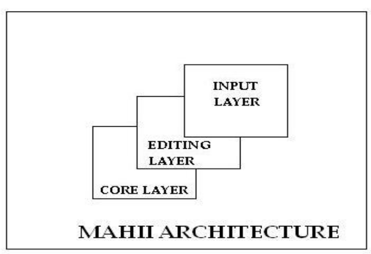


FIGURE 2: MAHII architecture

5.1 Input layer

The input layer is the window that is visible to the user for drawing the sketches. The user can draw frequently the sketches and these are used as input to the system. The user even may not have the knowledge of the functionality of the computer. As the user draws the symbols or sketches, he gets the edited image as well. The user need not wait for the system as and when to draw a sketch on a pen-based system.

This window after taking the input from the user not only displays the smooth sketch but also displays the recognition layer's result. We consider many issues and tackle various problems encountered in the earlier sketch interfaces.

The sketch interface MAHII consists of a single window. However, the editing and recognition takes place in the backend. The window represents simple, congestion-free screen to the user.

Though it contains menus for the computer known user still it contains many commands that a novice user can use.

We now describe some of the advantages of MAHII over existing system like SkRUI. The system SkRUI which was developed earlier contains two separate windows [4]. User draws in one window and gets the result in the other window. So, user switches explicitly between the two. MAHII makes this task easier. It uses a single window for all purposes and it does not have any synchronization problem. Further, many interfaces provide menu based tools like edit, undo, redo, cut, copy, paste. We make these tools available in our system in addition to some extra natural commands that can easily be used by novice.

In SkRUI, user cannot draw new items when it is in edit mode whereas MAHII does editing by itself and allows the user to draw the new sketches as well as to modify the previously drawn items. For this purpose, it uses buffers for temporary storage of previously drawn sketches and whenever the user wants to copy a sketch, he just needs to redraw the sketch over the existing one. The MAHII interface shows large buttons (menu based) so that user can easily identify them and the button is labeled with the pictures of what it does.

5.2 Editing Layer

This layer performs the editing of the sketch input. This layer includes the functioning of image filtering, image segmentation and image editing. It provides input to the recognition layer. It gets the input from the input layer and refines it by removing the noise from the sketch and increases the contrast of faded image/sketch. Any unclosed or unconnected Figures or basic shapes are connected by the system itself [6]. For example, if user wants to draw a circle, but incomplete, the editing layer does this for the user (Figure 3).

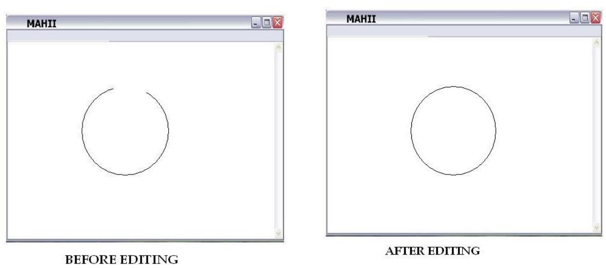


FIGURE 3: Working principle of editing layer

This layer takes input strokes one by one from the input layer as soon as the user draws the strokes on the MAHII interface. If the user draws a stroke at t time instant, the editing layer gets the input at say (t+Δt) time instant and edits during (t+2Δt) and it returns the result at (t+3Δt) time instant to both the input layer and recognition layer (Figure 4). The recognition layer does its job during 2Δt units of time. The final output of recognition layer gets displayed at (t+5Δt). Here Δt represents a very small delay of the order of microseconds.

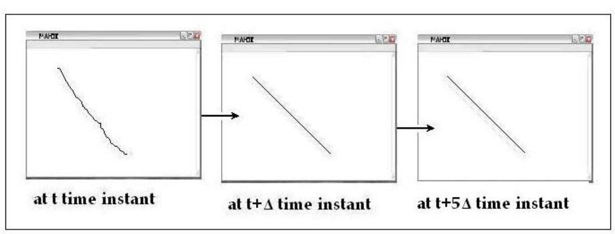


FIGURE 4: Process of editing at different time instants

Previously developed systems have a drawback that system should wait until the user finish sketching to display editing or recognition results. But since both of these activities take place simultaneously at the backend so, neither user nor the system should wait for sketching or responding respectively.

Here, an alternation that we want to provide to the user to know about the mode is that one can use pen-based editing, that is, when the cursor is in pen-mode, then the user can draw sketches and when it is in cursor mode it means the editing mode. Thus, sketching and editing use distinct pen motions.

5.3 Recognition layer

This layer is the core of our proposed system. This is the only layer that is given the power of recognition (the output) and named as recognition layer. It gets the input from the editing layer as mentioned above and uses heuristic engine to perform semantic checking, and searches for the relevant details from the system database. Finally, it shows the result to the MAHII interface or input window.

For processing, the system uses the domain class descriptions through an interface. The heuristic engine is being used here for structure matching and semantic checking. We can say that the recognition layer introduces the concept of domain independence. It recognizes the basic elements belonging to a particular domain using the Heuristic engine and database of samples of shapes and then identifies to which domain those elements belong to and then selects a particular domain class description out of many and connects it to the recognition engine. Thus, it makes the architecture as a disconnected type of architecture. When a new domain is referred to, the previous domain interface gets destroyed. It also helps the domain descriptor to define separate domain class descriptions and different domains never mingle with each other. This also reduces the complexity of domain description plus interprets the results with accuracy.

The recognition layer recognizes the output and displays the result to the input-output layer of the interface (MAHII). The recognition layer provides the result in real time because, it calculates the average waiting time for each user and then responds accordingly. It does not wait for editing or recognition by user, it starts working as soon as it gets the input.

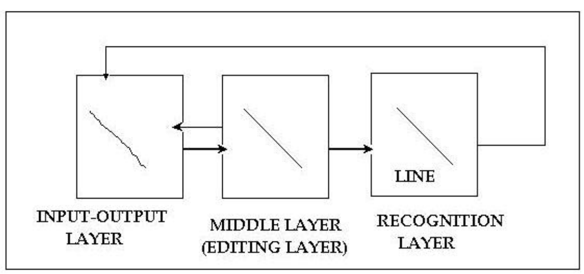


FIGURE 5: Abstract layered view

Based on the input-output, we represent a simple view of layered architecture of MAHII interface. The Figure 5 reflects the simple view of a three layered architecture through which we can estimate the real time responses. We use a single window for getting input from the user as well as for showing the output of editing and recognition of the sketch. This is an efficient approach since the user uses a single interface there is no need to switch between the sketch mode and the edit mode.

6. INTERFACE COMMANDS

Our system promises to be a user-friendly interface by indulging itself according to the user's choice. It can respond the way the user behaves. For a computer-literate user, it provides menu-based buttons and tools. At the same time, it provides normal or generally used commands [7] that a novice can use it as he/she draws the sketch on the paper with a pen.

If a user draws a sketch on a paper and if it is not drawn according to the user's choice then he/she simply cuts it using a X symbol (Figure 6(a)). In the similar way, this command deletes the figure. If a user wants to copy the same figure he/she needs to redraw it over the figure itself (Figure 6(b)). This creates a copy in the buffer of that window area wherever the user leaves the cursor or with a click the sketch can be pasted. For the same purpose, the user can use the command C, say, for copy and P for paste (Figure 6(c)).

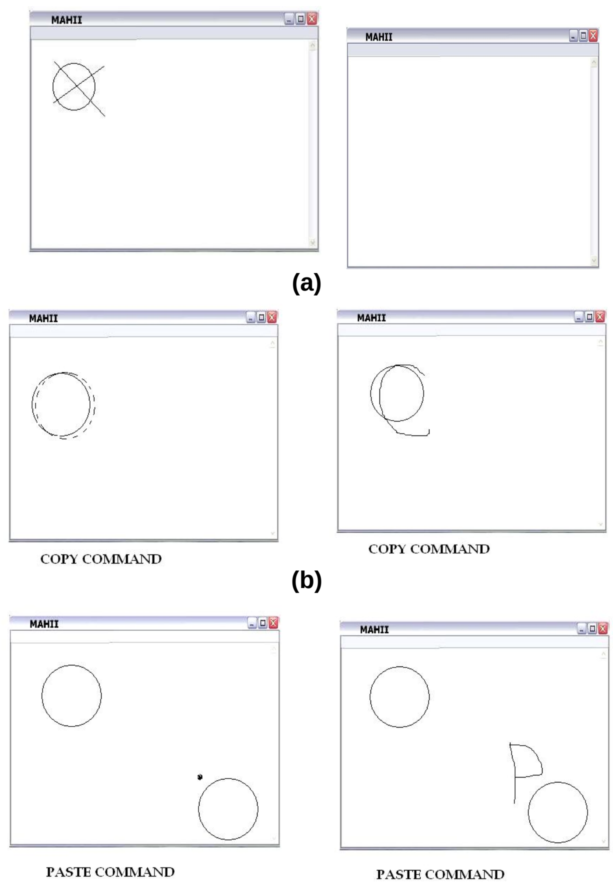


FIGURE 6: Demonstration of (a) Delete command, (b) Copy command, (c) Paste command

If user wants to cut a sketch and paste it somewhere else he needs to delete the figure after copy and paste. If user wants to redo, he can simply do it by using the command like R or r over the window area which will redo the last undo (Figure 7) part.

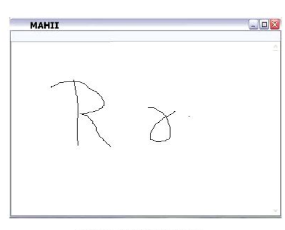


FIGURE 7: Redo command

To undo, the user needs to use the symbol U or u. This undoes the thing what the user has done just before (Figure 8).

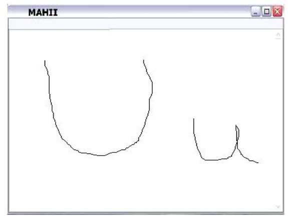


FIGURE 8: Undo command

To rub a portion of the sketch, a zigzag command can be used (Figure 9).

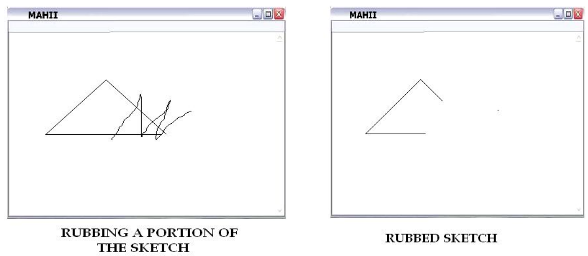


FIGURE 9: Rubbing a portion of a sketch

7. EXPERIMENTS AND USER FEEDBACK

We have provided the system to be used by any type of users. There are two groups of people to use the system. Some of them are good enough in computer skills and some have never any interaction with the computer before. Our basic aim is to get the idea of how the system MAHII is interactive and natural to the users. There are no instructions or demonstration given how MAHII basically works; instead it provides amazing results of much satisfactory and fulfilling the target of natural or user-friendly environment.

The experiment was done as follows. There were 12 users who were asked to draw sketches of various domains. There were 6 novice users and 6 were computer known people. Novice users found our system simple and easy to operate. They draw sketches of electronic circuits and shortest-path graphs without explicitly switching between the domains. Our system was successful in finding the specific domain itself. This provided the novice users enough flexibility. Even MAHII provided self-editing feature that the users appreciated. Since people did not know any command that MAHII provides, so they used normal behavioral commands e.g. to delete a figure they just crossed over it. So, MAHII proves to be successful as natural, simple and user friendly.

Then we tried to find out how it encourages the computer known people. They were also asked to draw sketches from various domains. Our system proved successful for them as well. They sketched strokes and they found it easy to use the menu based commands since the menu based commands were based on buttons that were labeled with the corresponding images rather than the command name. MAHII was successful for expert users as well in all respects.

Table 1 shows our efficiency measurement in terms of ease of use.

USER	Efficiency in Terms of Ease of Use
Novice	93%
Expert	94%

Table 1: Efficiency of MAHII for various users

8. RELATED WORK AND COMPARISON STUDY

In this section we compare the efficiency and effectiveness of the proposed system withy other existing systems defined for sketching. In Table 2 we have given the recognition rate of MAHII for different inputs, and domains.
Comparison with previous work shows that MAHII improves the sketching, editing and recognition. This shows its capability better than the earlier systems.

Input Type	Static	97%
	Dynamic	95%
Domain	Single Domain	96%
	Multi Domain	87%

Table 2: Recognition rate

Various sketch systems like discussed as in paper [11] have shown the efficiency (6.5 / 7) of their system to 92.8%. CALI [10] gives recognition efficiency up to 95.3% for an overall recognition rate. The recognition rate for static inputs has been improved to 95.1%, but for dynamic input, it is limited up to 70% only. Bayesian approach [9] that gives the accuracy rate till 94% is based on the probability. Whereas using the Fuzzy Logic approach the efficiency is up to 92% for a single domain system and not much work has been done to increase the efficiency of multi-domain sketching system [8]. However, MAHII gives 87% efficiency for multi-domain sketching.

Further, for the purpose of various comparison aspects we consider here the existing systems CAD, SkRUI, SketchRead, FIRST and depict the corresponding graphs in Figures 10 and 11.

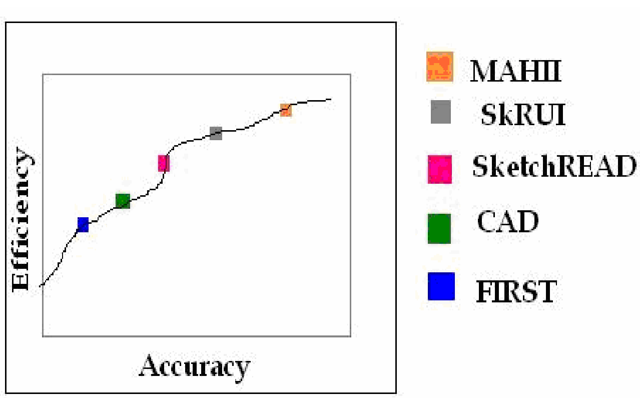
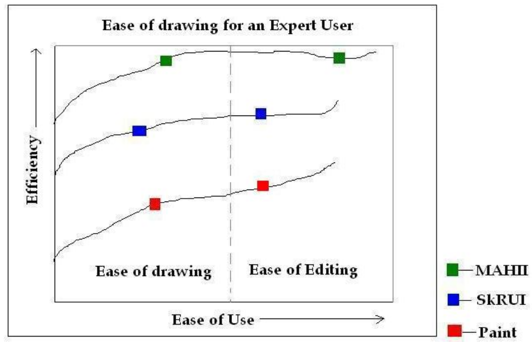
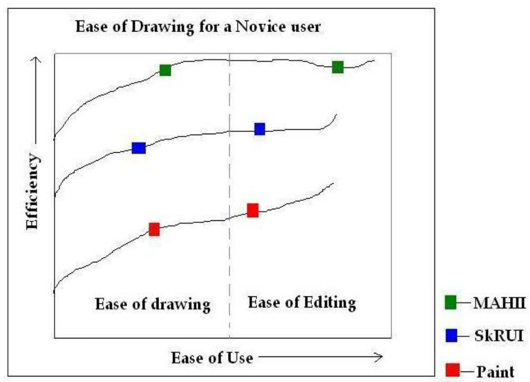


FIGURE 10: The comparison graph respect to sketching, editing and recognition



(a)



(b)

FIGURE 11: Efficiency curves respect to easy of drawing for (a) Expert user, (b) Novice user

From the above Table and Figures we can say that the system MAHII has a capability in sketching, editing and recognition over the existing systems. It provides better results with a higher rate of usefulness: its automatic editing feature makes it a perfect sketching system.

9. CONCLUSION AND FUTURE WORK

MAHII interface presented here is a new system that exhibits the powerful yet more natural environment for the user. It takes care about the every kind of user and the people of various domains. Users need not to learn computer to use this system. It reduces the complexity still showing more freedom to the user in drawing the sketches.

MAHII satisfies the requirements of novice user and it can further be enhanced to provide more real time responding and friendly environment. It may use large amount of buffer area, but satisfies the demand of multi domain recognition without the knowledge of user. This work can be carried out later.

10. REFERENCES:

1. G. Sahoo, Bhupesh Kumar Singh, “A New Approach to Sketch Recognition using Heuristic”, International Journal of Computer Science and Network Security, Vol. 8 No. 2, 2008., PP. 102-108.,
2. Mankoff, J. Hudson, S.E., and Abowed. G. D., “Providing integrated toolkit-level support for ambiguity in recognition-based interfaces”. In proceedings of CHI 2000 conference on Human Factors in Computing Systems, 368-375.

3. Newman. M.W., Lin. J., Hong. J. I., and Landay. J. A., “DENIM: An Informal web site design tool inspired by observations of practice.” Human-Computer Interaction 18(3): 259-324
4. Alvarado C., “Sketch Recognition And User Interfaces: Guidelines for Design and Development”, AAAI Fall Symposium on Pen-Based Interaction, 2004.
5. Landay. J.A., and Myres, “Interactive Sketching for the early stages of user interface design”. In proceedings of CHI 95, 43-50. B.A.1995
6. Mahoney. J. V., and Fromherz. M. P. J., “Three main concerns in Sketch recognition and an approach to addressing them”. AAAI Spring Symposium on Sketch Understanding, 2002.
7. Rubine D. “Specifying gesture by example”. SIGGRAPH 1991, Las Vegas, pp.329-327
8. Fonseca. M., Jorge. J., “Using Fuzzy Logic to recognize geometric shapes interactively”. Proceedings of the 9th IEEE conference on Fuzzy systems, pp. 291-296, 2000.
9. Shilman. M., Pasula. H., Russell. S., and Newton. R. “Statistical visual language models for ink parsing.” Proceedings of AAAI spring Symposium on Sketch Understanding, Stanford University, pp. 126-132, March 2002.
10. Fonseca. M.’ Pimetal. C., “CALI: an online scribble recognizer for calligraphic interface”. Proceedings of AAAI Spring Symposium on sketch Understanding, Stanford University, PP.51-58, March 2002.
11. Wolin. A., Smith. D., and Alvarado. C., “A Pen—based Tool for efficient labeling of 2D sketching”, EUROGRAPHICS workshop on sketch based Interfaces and Modeling (2007).
12. Alvarado. C. and Oltmans. M., Davis. R.’ “A Framework for multi-domain sketch recognition”. Proceedings of AAAI Spring Symposium on sketch Understandings, Stanford University, pp.1-8, March 2002.
13. Alvarado. C.’ “Multi-domain sketch understanding”’ PhD thesis, Department of Electrical Engineering and computer science, Massachusetts Institute of Technology, September 2004.

Segmentation of Handwritten Text in Gurmukhi Script

Rajiv K. Sharma rajiv.patiala@gmail.com
Sr. Lecturer, SMCA
Thapar University,
Patiala, 147002
Punjab, India

Dr. Amardeep Singh amardeep_dhiman@yahoo.com
Reader, UCoE,
Punjabi University,
Patiala, 147004
Punjab, India

Abstract

Character segmentation is an important preprocessing step for text recognition. The size and shape of characters generally play an important role in the process of segmentation. But for any optical character recognition (OCR) system, the presence of touching characters in textual as well handwritten documents further decreases correct segmentation as well as recognition rate drastically. Because one can not control the size and shape of characters in handwritten documents so the segmentation process for the handwritten document is too difficult. We tried to segment handwritten text by proposing some algorithms, which were implemented and have shown encouraging results. Algorithms have been proposed to segment the touching characters. These algorithms have shown a reasonable improvement in segmenting the touching handwritten characters in Gurmukhi script.

Keywords: Character Segmentation, Middle Zone, Upper Zone, Lower Zone, Touching Characters. Handwritten, OCR

1. INTRODUCTION

In optical character recognition (OCR), a perfect segmentation of characters is required before individual characters are recognized. Therefore segmentation techniques are to apply to word images before actually putting those images to reorganization process. The simplest way to segment the characters is to use inter – character gap as a segmentation point[1]. However, this technique results in partial failure if the text to be segmented contains touching characters. The situation becomes grim if text consists of handwritten characters. The motivation behind this paper is that to find out a reasonable solution to segment handwritten touching characters in Gurmukhi script. Gurmukhi script is one of the popular scripts used to write Punjabi language which is one of popular spoken language of northern India. Because our work is related with segmentation of Gurumukhi script, so it is better to discuss some characteristics of the said script so that the reader can have a better idea of the work.

2. CHARACTERISTICS OF GURMUKHI SCRIPT

Gurmukhi script alphabet consists of 41 consonants and 12 vowels[2] as shown in FIGURE 2. Besides these, some characters in the form of half characters are present in the feet of characters. Writing style is from left to right. In Gurmukhi, There is no concept of upper or lowercase characters. A line of Gurmukhi script can be partitioned into three horizontal zones namely, upper zone, middle zone and lower zone. Consonants are generally present in the middle zone. These zones are shown in FIGURE 1. The upper and lower zones may contain parts of vowel modifiers and diacritical markers.

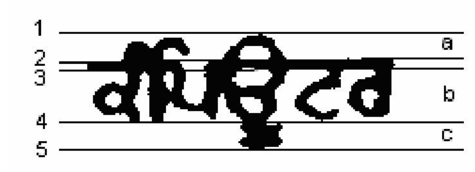


FIGURE 1 : a) Upper zone from line number 1 to 2, b) Middle Zone from line number 3 to 4, c) lower zone from line number 4 to 5

In Gurmukhi Script, most of the characters, as shown in FIGURE 2, contain a horizontal line at the upper of the middle zone. This line is called the headline. The characters in a word are connected through the headline along with some symbols as i, I, A etc. The headline helps in the recognition of script line positions and character segmentation. The segmentation problem for Gurmukhi script is entirely different from scripts of other common languages such as English, Chinese, and Urdu[3] etc. In Roman script, windows enclosing each character composing a word do not share the same pixel values in horizontal direction. But in Gurmukhi script, as shown in FIGURE 1, two or more characters/symbols of same word may share the same pixel values in horizontal direction.

This adds to the complication of segmentation problem in Gurmukhi script. Because of these differences in the physical structure of Gurmukhi characters from those of Roman, Chinese,

Consonants (Vianjans)

ੳ ਊੜਾ (ūṛā) u, ū, o	ਅ ਐੜਾ (aiṛā) a, ā, ai, au	ੲ ਈੜੀ (īṛī) i, ī, e	ਸ ਸੱਸਾ (sas'sā) sa [sə]	ਹ ਹਾਹਾ (hāhā) ha [hə]
ਕ ਕੱਕਾ (kakkā) ka [kə]	ਖ ਖੱਖਾ (khakhkhā) kha [kʰə]	ਗ ਗੱਗਾ (gaggā) ga [gə]	ਘ ਘੱਗਾ (ghaggā) gha [gə]	ਙ ਙੰਙਾ (ṅaṅṅā) ṅa [ŋə]
ਚ ਚੱਚਾ (caccā) ca [tʃə]	ਛ ਛੱਛਾ (chachchā) cha [tʃʰə]	ਜ ਜੱਜਾ (jajjā) ja [dʒə]	ਝ ਝੱਜਾ (jhajjā) jha [dʒə]	ਞ ਞੰਞਾ (ñaññā) ña [ɲə]
ਟ ਟੈਂਕਾ (ṭaiṅkā) ṭa [ʈə]	ਠ ਠੱਠਾ (ṭhaṭhṭhā) ṭha [ʈʰə]	ਡ ਡੱਡਾ (ḍaḍḍā) ḍa [ɖə]	ਢ ਢੱਡਾ (ḍhaḍḍā) ḍha [ɖə]	ਣ ਣਾਣਾ (ṇāṇā) ṇa [ɳə]
ਤ ਤੱਤਾ (tattā) ta [tə]	ਥ ਥੱਥਾ (thaththā) tha [tʰə]	ਦ ਦੱਦਾ (daddā) da [də]	ਧ ਧੱਦਾ (dhaddā) dha [də]	ਨ ਨੱਨਾ (nannā) na [nə]
ਪ ਪੱਪਾ (pappā) pa [pə]	ਫ ਫੱਫਾ (phaphphā) pha [pʰə]	ਬ ਬੱਬਾ (babbā) ba [bə]	ਭ ਭੱਬਾ (bhabbā) bha [bə]	ਮ ਮੱਮਾ (mam'mā) ma [mə]
ਯ ਯੱਯਾ (yayyā) ya [jə]	ਰ ਰਾਰਾ (rārā) ra [rə]	ਲ ਲੱਲਾ (lallā) la [lə]	ਵ ਵੱਵਾ (vavvā) va [wə]	ੜ ੜਾੜਾ (ṛāṛā) ṛa [ɽə]
ਸ਼ ਸ਼ੱਸ਼ਾ (śaśśā) śa [ʃə]	ਖ਼ ਖ਼ੱਖ਼ਾ (k̲h̲ak̲h̲k̲h̲ā) k̲h̲a [χə]	ਗ਼ ਗ਼ੱਗ਼ਾ (ġaġġā) ġa [ɣə]		
ਜ਼ ਜ਼ੱਜ਼ਾ (zazzā) za [zə]	ਫ਼ ਫ਼ੱਫ਼ਾ (faffā) fa [fə]	ਲ਼ ਲ਼ੱਲ਼ਾ (l̲al̲l̲ā) ḷa [ɭə]		

FIGURE 2 a) : Consonants (Vianjans)

Vowels and Vowel diacritics (Laga Matra)

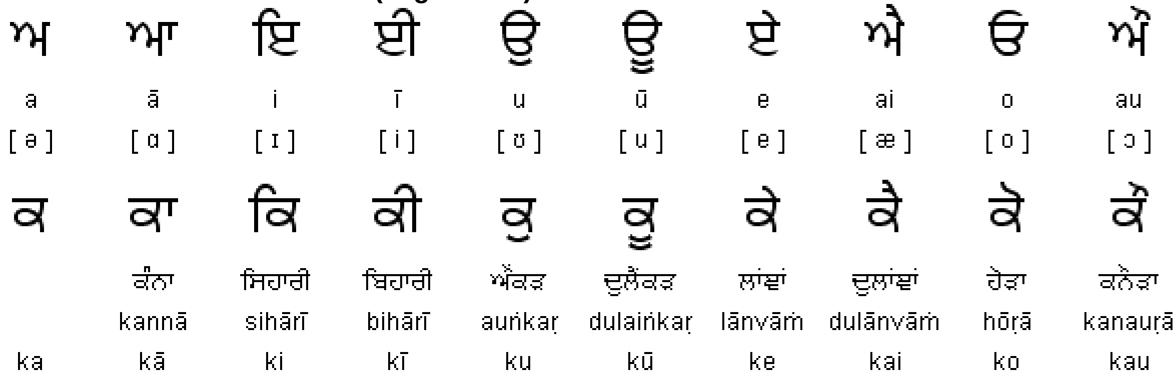


FIGURE 2 b) : Vowels and Vowel diacritics (Laga Matra)

Other symbols

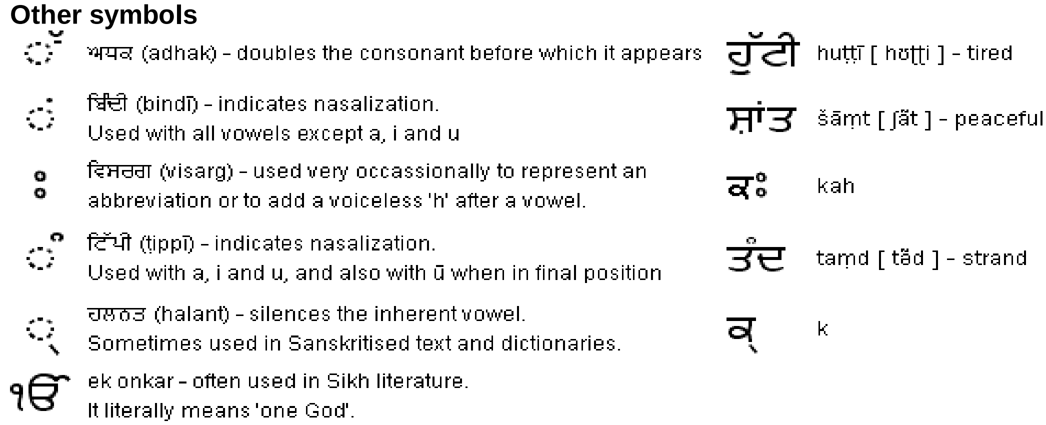


FIGURE 2 c) : Other symbols

Japanese and Arabic scripts, the existing algorithms for character segmentation of these scripts does not work efficiently for handwritten Gurmukhi script.

3. PREPROCESSING

Preprocessing is applied on the input binary document so that the effect of spurious noise can be minimized in the subsequent processing stages. In the present study, both salt and peeper noise have been removed using standard algorithm [4]. It is supposed that height and width of document can be known easily. The image is saved in the form of an array. For that purpose a 2-D array with number of rows equal to height of the document and number of columns equal to width of the document is created. Calculate the maximum intensity of pixels in the document using any standard function available in the tool used for the implementation, it is getRGB() method available in java. Scan every pixel of document and compare its intensity with the maximum intensity. If the intensity is equal to maximum intensity, store one in the array at that location, and if it is not equal store zero in the array.

4. PROPOSED PROCEDURES TO SEGMENT LINE, WORD and CHARACTER

Line Detection

The following procedure is implemented to find the location of lines in the document.

i. Create an array of size equal to height of the document and with two columns.

ii. Start from the first row and count the number of 1's in that row. If it is zero, move to next row. And if it is not zero, that is the starting location of that line. Store that location in the array.
iii. Check consecutive rows until we get 0.The before we get zero is the ending location of that line. Store that value in the array.
iv. Also calculate the location of maximum intensity in each line and store it in the second column before that line. It would be used as the starting position of characters.
v. Repeat step (ii) to (iv) for the whole document.

Word Detection

The following procedure is implemented to find location of words in each line.

i. Create a 2-D array.
ii. For each line move from 0^{th} pixel up to width.
iii. Calculate the number of one's in first column from the starting location of line to the ending position of line.
iv. If number of 1's are not zero, that is the starting location of word. Save that location in that array. Keep on moving to the right until we get no one in any column. The column with 0 1's is the ending location of the word. Store that location in array too.
v. Repeat this until we reach the width.
vi. And repeat step (ii) to (v) for each line.

Character Detection

The following procedure is implemented to find the location of character in each word.

i. Create a 3-d array. Its first index will represent line number. Second index will represent word number and third index will contain the location of character. This array will be created dynamically.
ii. Repeat the step (iii) to (iv) for each line and each word detected so far.
iii. Move from starting position of the word to the ending position of the word.
iv. Start from the starting position of line and move downwards to the ending position. Count the number of one's in that column leaving the location of line with maximum intensity. If it is not zero, that is the starting position of character. Move to right until we get column with no ones. that will be the ending location of character.

This process will generate the location of characters.

The above approach was put to number of documents; the image of one such scanned document is given below.

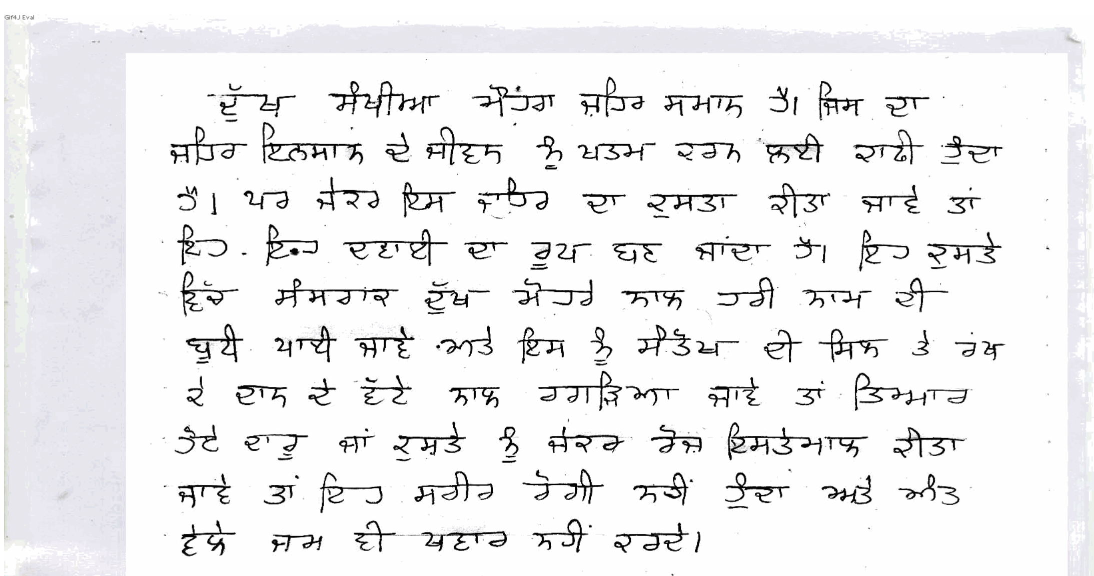


FIGURE 3: Scanned Image of a Document

The result of the scanned document after processing is given below.

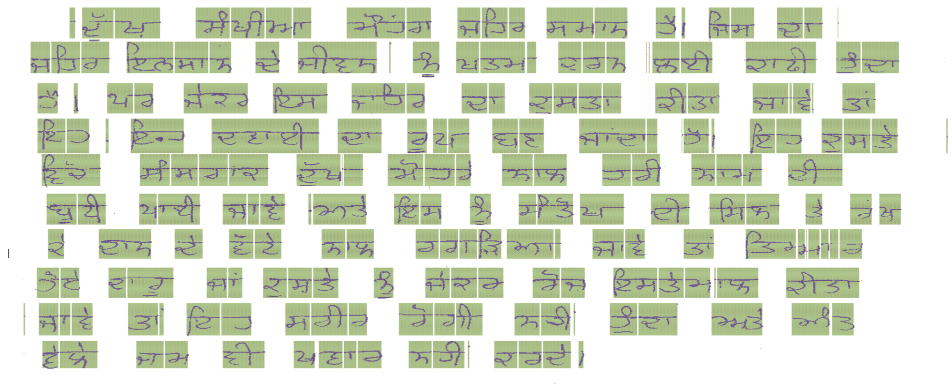


FIGURE 4: Processed Document

The main objective of the work was to segment the lines, words and to segment the touching characters present in handwritten document in Gurmukhi script. We obtained the following table after putting handwritten Gurmukhi documents for segmentation. The results are summarized as in following tables:

Document	No of Lines	Correctly Detected	Inaccurate segmentation	Accuracy
Doc1	5	4	1	80%
Doc2	8	7	1	87.5%
Doc3	10	8	2	80%
Doc4	13	11	2	84.61

TABLE 1: ACCURACY for Line Segmentation

Document	No of Words	Correctly Detected	Inaccurate segmentation	Accuracy
Doc1	38	32	6	84.21%
Doc2	56	49	7	87.5%
Doc3	95	79	16	83.15%
Doc4	110	90	20	81.81

TABLE 2: ACCURACY for Word Segmentation

Document	No of Characters	Correctly Detected	Inaccurate segmentation	Accuracy
Doc1	79	71	8	89.8%
Doc2	168	145	23	86.30%
Doc3	224	175	49	78.12%
Doc4	289	232	57	80.27

TABLE 3: ACCURACY for Character Segmentation

5. CONCLUSION AND FUTURE WORK

This work was carried out to detect lines present in scanned document in handwritten Gurumukhi script. So firstly we are to find out the lines present in the document then to find words present in each line detected at the first step. Using the detected words it is to segment characters present in each word. Therefore using line detection algorithm (the first approach) lines were detected. Mostly we found the correct lines, but some were not detected correctly. The reason behind this may be the writing style of Gurumukhi script as shown in FIGURE 1. So the words presents in the lower zone were considered as a different line. The correctly detected lines were further put to word detection algorithm. Here the results were good, but sometimes when the words were not joined properly then that was detected as a different word. The locations of the detected words were used to segment the characters. At few point segmentation was good but at few point it was not upto the expectations. This may be because of the similarity in the shapes of few characters. All these issues can be dealt in the future for handwritten documents written in 2-dimensional script like Gurumukhi by making few changes to proposed work.

6. REFERENCES

1. Y. Lu. "Machine Printed Character Segmentation – an Overview". *Pattern Recognition,* vol. 29(1): 67-80, 1995

2. M. K. Jindal, G. S. Lehal, and R. K. Sharma. "Segmentation Problems and Solutions in Printed Degraded Gurmukhi Script". IJSP, Vol 2(4),2005:ISSN 1304-4494.

3. G. S .Lehal and Chandan Singh. "Text segmentation of machine printed Gurmukhi script". Document Recognition and Retrieval VIII, Proceedings SPIE, USA, vol. 4307: 223-231, 2001.

4. Serban, Rajjan and Raymund. "Proposed Heuristic Procedures to Preprocesses Character Pattern using Line Adjacency Graphs". Pattern recognition, vol. 29(6): 951-975, 1996.

5. Veena Bansal and R.M.K. Sinha. "Segmentation of touching and Fused Devanagari characters, ". Pattern recognition, vol. 35: 875-893, 2002.

6. R. G. Casy and E. Lecolinet. "A survey of methods and strategies in character segmentation". IEEE PAMI, Vol. 18:690 – 706,1996.

7. U. Pal and Sagarika Datta. "Segmentation of Bangla Unconstrained Handwritten Text". Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR), 2003.

8. U. Pal, S. Sinha and B. B. Chaudhuri. "Multi-Script Line identification from Indian Documents", Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR) 2003.

9. Rajean Plamondon, Sargur N. Srihari. "On – Line and Off – Line Handwritting Recognition: A Comprehensive Survey", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol 22(1). Janurary, 2000.

10. Giovanni Seni and Edward Cohen. " External word segmentation of off – line handwritten text lines". Pattern Recognition, Vol. 27(1): 41-52, 1994.

OBJECT-ORIENTED IMAGE PROCESSING OF AN HIGH RESOLUTION SATELLITE IMAGERY WITH PERSPECTIVES FOR URBAN GROWTH, PLANNING AND DEVELOPMENT

Afroz Shaik Mohammed[1*] Email:smafroz@yahoo.com
Deccan college of Engg. and Technology,
(Affiliated to Osmania University)
Dar us Salam, Near Nampally, Hyderabad-500 001, (A.P), India.
Mobile: 9959732140, Landline: 914023535891

Dr Shaik Rusthum[2] srgisace_2k7@rediffmail.com
Professor & Principal,
VIF College of Engg. & Technology, Gandipet, Hyderabad.
Mobile: 9848530370

Abstract

The management of urban areas by urban planners relies on detailed and updated knowledge of their nature and distribution. Manual photo-interpretation of aerial photographs is efficient, but is time consuming. Image segmentation and object-oriented classifications provide a tool to automatically delineate and label urban areas. Here single pixels are not classified but objects created in multi-resolution segmentation process, which allows use of, spectral responses but also texture, context and information from other object layers. This paper presents a methodology allowing to derive meaningful area-wide spatial information for city development and management from high resolution imagery. Finally, the urban land cover classification is used to compute a spatial distribution of built-up densities within the city and to map homogeneous zones or structures of urban morphology.

Key words: Object oriented, Classification, Segmentation, Spatial information, Accuracy assessment, Urban morphology

1. INTRODUCTION

Human land use decisions on the environment are influenced by socioeconomic factors which can be represented by spatially distributed data. The accelerating urban sprawl, often characterized by a scattered growth, has rarely been well planned, thus provoking concerns over the degradation of our environmental and ecological health[2]. Up-to-date and area-wide information management in highly dynamic urban settings is a critical endeavor for their future development. Thematic assessments of urban sprawl involve procedures of monitoring and mapping, which require robust methods and techniques[3]. Conventional survey and mapping methods cannot deliver the necessary information in a timely and cost-effective mode. Limited spatial information within the built-up zone hinders urban management and planning. Especially in growing and altering cities lack of up-to-date data is apparent. The challenge of classifying urban land cover from high resolution remote sensing data arises from the spectral and spatial heterogeneity of such imagery. There to the high dissimilarity of functions like industrial or residential areas as well as parks or agricultural regions causes problems in terms of an indirect inferring of land use [7,8].

2. STUDY AREA, METHODOLOGY AND RESULTS

2.1 Study Area

The study site, Vijayawada city, known as the political capital of the State, located in the south-east of India is the third largest city of Andhra Pradesh state. Vijayawada is located on the banks of the sacred Krishna River and is bounded by the Indrakiladri Hills on the West and the Budemeru River on the North.

The other details of Vijayawada city are given in Table 1.

Table 1: Details of Vijayawada city

1)State	Andhra Pradesh	6)Time zone	IST (UTC+5:30)
2)District	Krishna	7)Population	10,35,536
3)Coordinates	16.30° N 80.37° E	8)Density of Population	17,854 /km²
4)Area	58 km²	9)Postal code	5200xx
5)Elevation	125 m	10)Telephone code	+91 866

2.2 Data

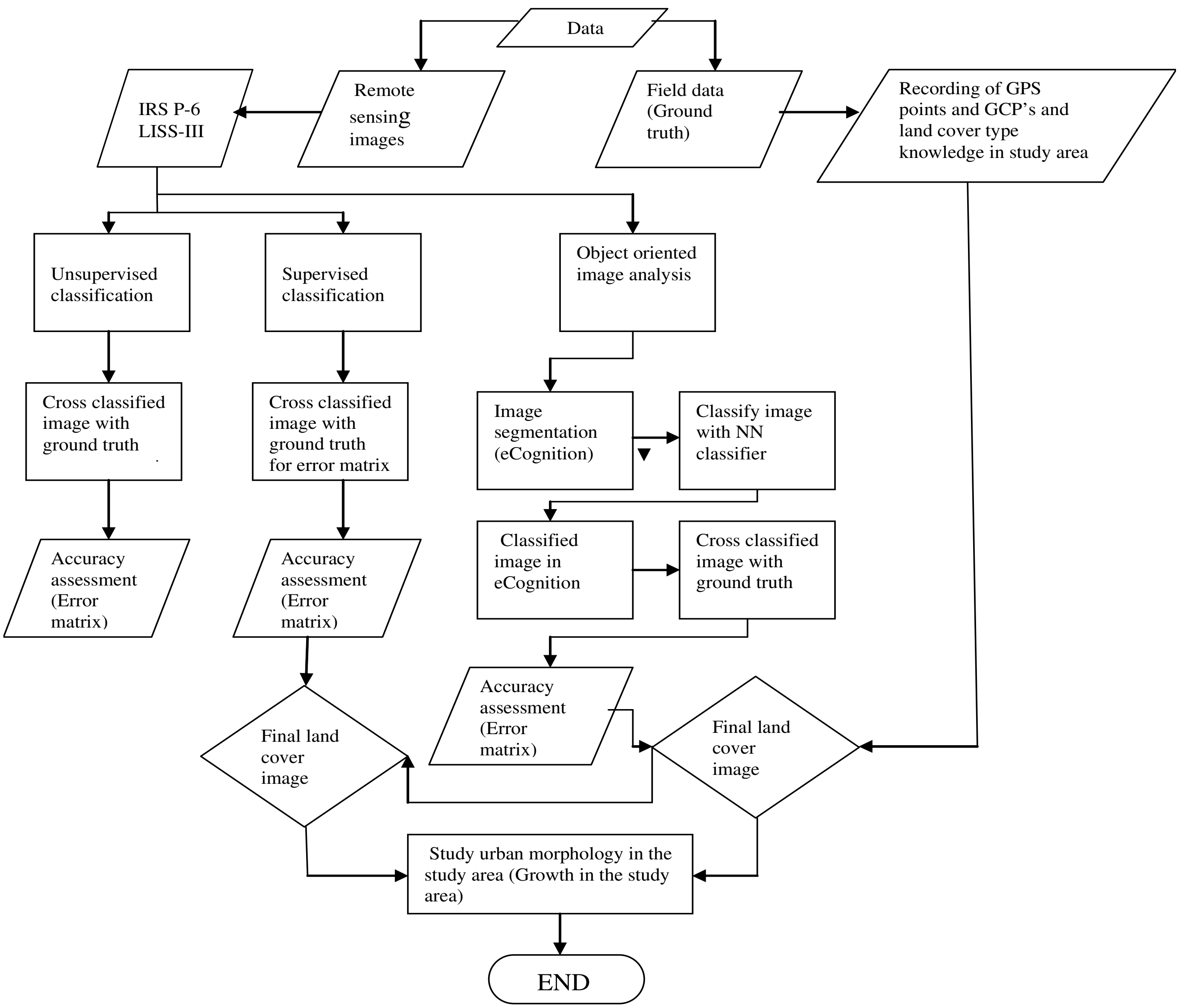
High resolution multispectral IRS P-6 LISS-3(Band 2,3,4 &5) images were taken. This satellite carries three sensors (LISS-III, AWiFS & LISS-IV) with 5.8m, 23.5m & 56m resolutions and fore-aft stereo capability. The payload is designed to cater to applications in cartography, terrain modeling, cadastral mapping etc., These images were supplied by NRSA, Hyderabad, India.(http://www.nrsa.gov.in)

Global Positioning System (GPS) receiver has been used for ground truth data that records the coordinates for the polygons of homogeneous areas, and also it records the coordinates that will be used for geometric correction. The GPS is in existence since the launch of the first satellite in the US Navigation System with Time and Ranging (NAVSTER) system on February 22, 1978, and the availability of a full constellation of satellites since 1994. The US NAVSTAR GPS consists of a constellation of 24 satellites orbiting the Earth, broadcasting data that allows a GPS receiver to calculate its spatial position (Erdas imagine, 2001).

Ground truth data is used for use in image classification and validation. The user in the field identifies a homogeneous area of identifiable land cover or use on the ground and records its location using the GPS receiver. These locations can then be plotted over an image to either train a supervised classifier or to test the validity of a classification.

2.3 Methodology

Here, the description about the land cover types and their distributions of the study area is given. Except this, the remote sensing images, ground truth used in this study are described in detail and also the data preprocessing before conducting the classification is described. Methodology to perform this research is given in flow chart 1.



Flow chart 1: Methodology

2.4 Results

2.4.1 Unsupervised and Supervised classification

The basic premise for unsupervised classification is that spectral values within a given land cover type should be close together in the measurement space, whereas spectral data in different classes should be comparatively well separated (Lillesand, 2001). Unsupervised classification is fast and has the ability to analyze the image spectral statistics completely and systematically, thus unsupervised classification can give useful indication of detectable classes for supervised classification (Mather, 1987).

Supervised classification result of the study area (Vijayawada city) with different land cover types is presented in Plate 1.

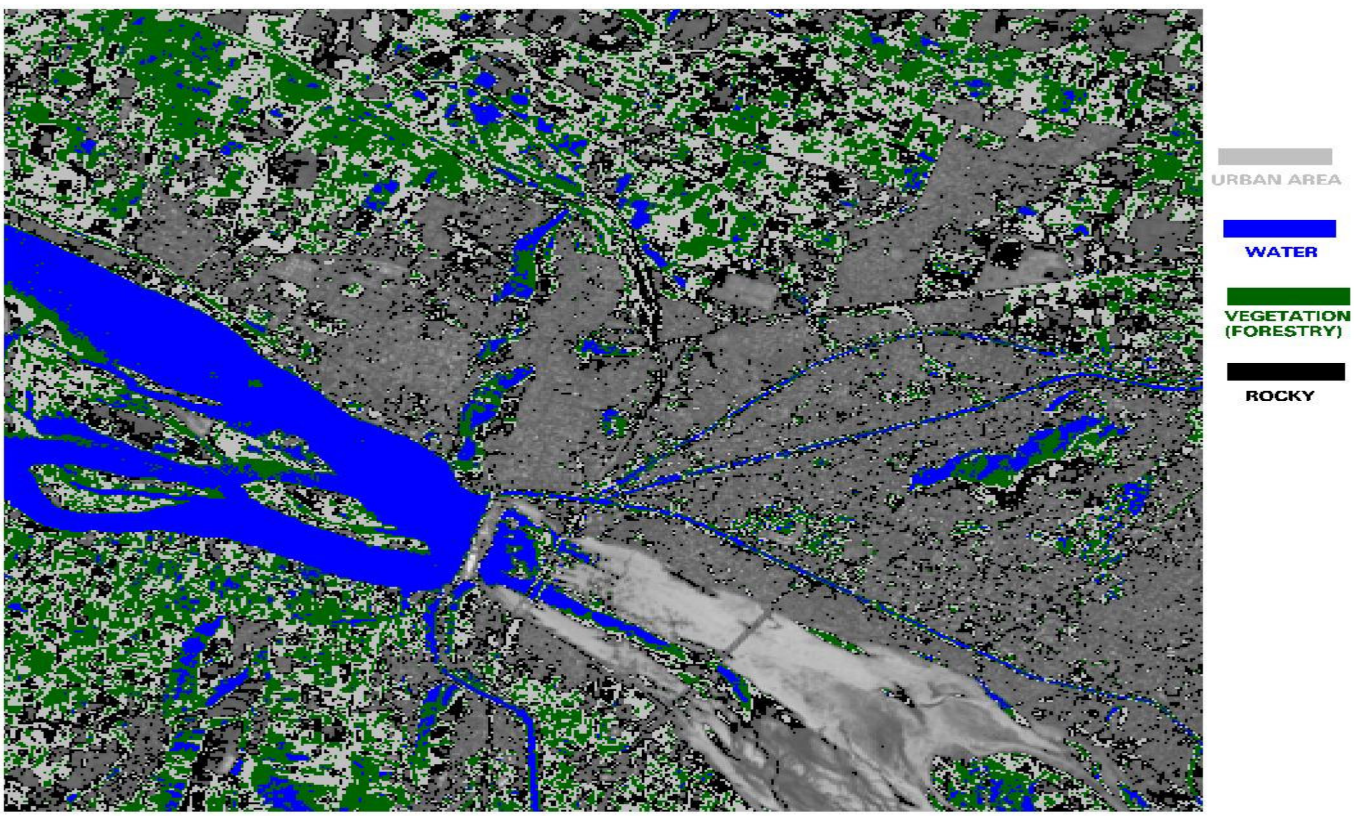


Plate 1: Supervised classification result of Study area(Vijayawada city) from IRS P-6 LISS-III Imagery

2.4.2 Object oriented image analysis

Using the object oriented image analysis approach to classify the image is performed in eCognition. Object oriented processing of image information is the main feature of *eCognition*. The first step in *eCognition* is always to extract image object primitives by grouping pixels. The image objects will become building blocks for subsequent classifications and each object will be treated as a whole in the classification. Multi-resolution segmentation is a basic procedure in eCognition for object oriented image analysis. The segmentation rule is to create image objects as large as possible and at the same time as small as necessary. After segmentation, a great variety of information can be derived from each object for classifying the image. In comparison to a single pixel, an image object offers substantially more information.

2.4.3 Comparison of segmentation results with different scale parameters in the study area

Plate 2 is the original image of the study area. Plates 3, 4 and 5 show the effect of segmentation results using different segmentation parameters. Except scale difference, the other parameters that influence the segmentation result are color, shape, smoothness and compactness but these are kept constant. Plate 3 is the segmentation result with a scale parameter 5.Comparing this segmentation result with the original image, it is found that neighbor pixels are grouped into pixel clusters-objects, and because of the low value of scale parameter, there are too many small objects. Plate 4 is the segmentation result with scale parameter 10. It is found by comparing it with Plate 3 that higher scale parameter value generates larger objects. Plate 5 is the segmentation result with scale parameter 20. By visual comparison, a scale parameter of 10 is selected because the segmentation result fits the information class extraction best. Based on these parameters, segmentation process is performed.

2.4.4 Image classification

Classification is the process of connecting the land cover classes with the image objects. After the process of classification, each image object is assigned to a certain (or no) class. In eCognition, the classification process is an iterative process. The classification result can be improved by editing the result: defining unclassified objects with the correct classes, correcting wrongly classified objects with the correct classes, etc.

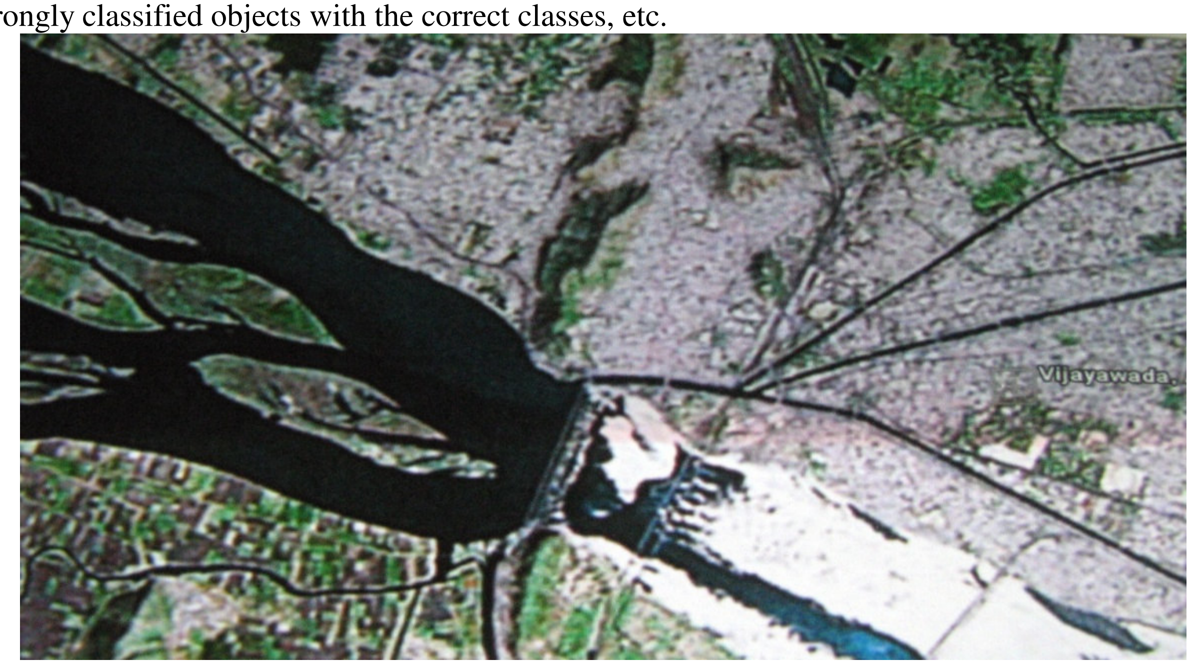


Plate 2: Original image of the study area (Vijayawada city)

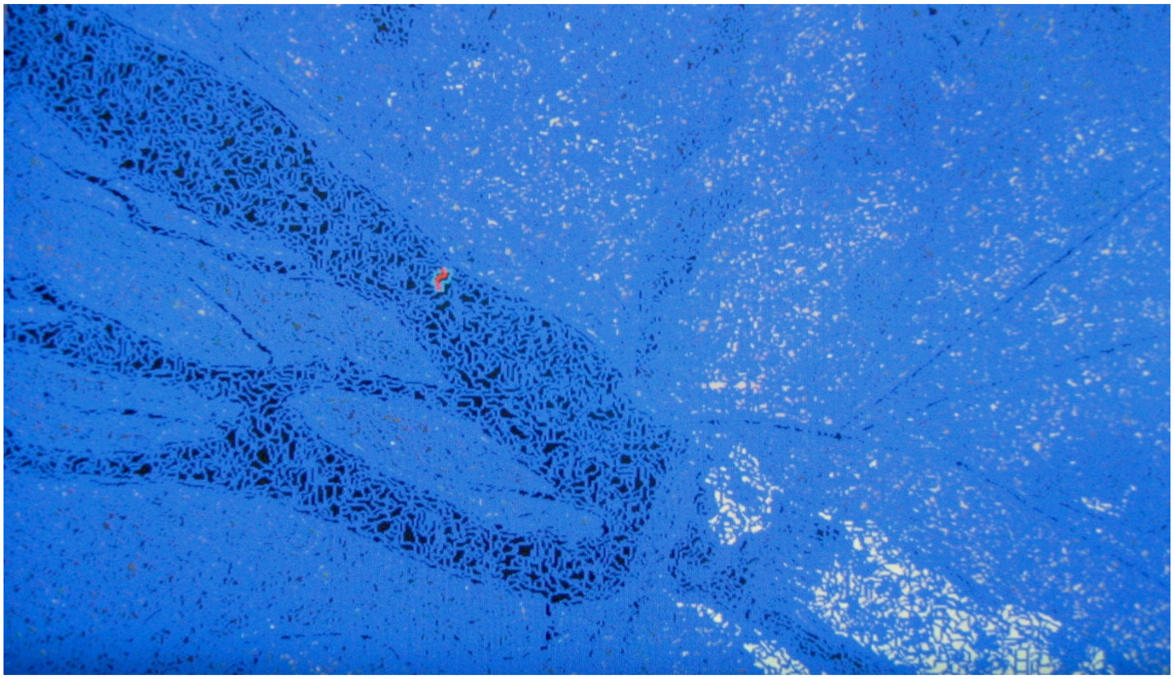


Plate 3: Segmentation result with scale parameter 5, color 0.8, shape 0.2, smoothness 0.9, & compactness 0.1

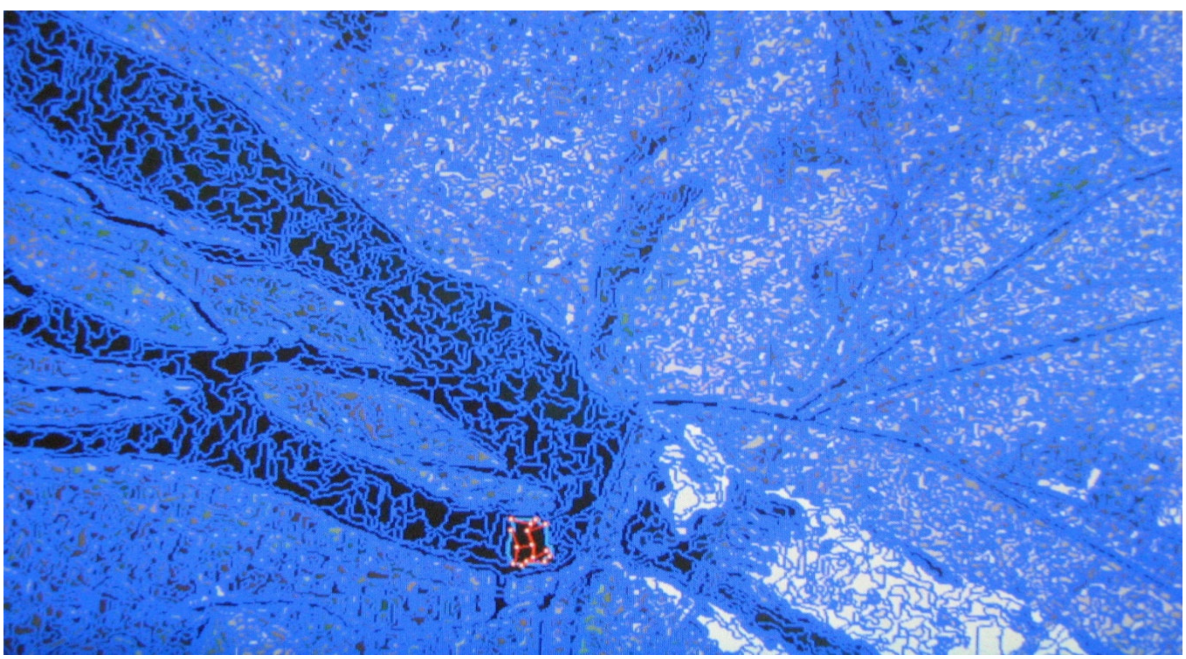


Plate 4: Segmentation result with scale parameter 10, color 0.8, shape 0.2, smoothness 0.9, & compactness 0.1

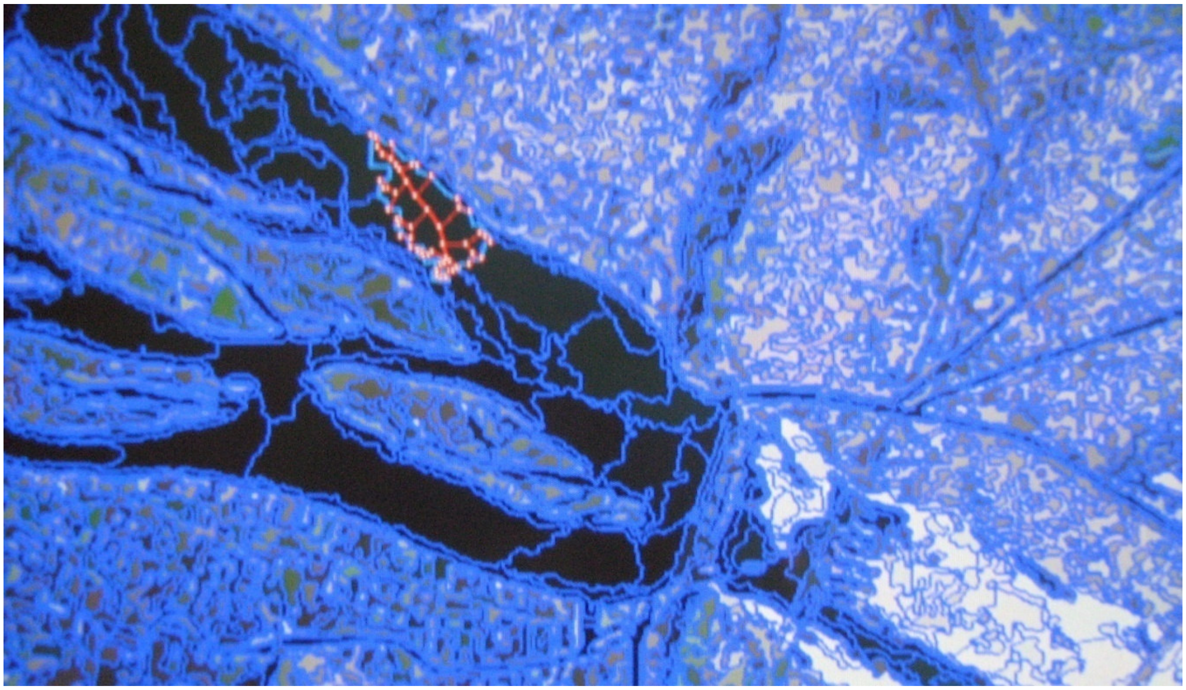


Plate 5: Segmentation result with scale parameter 20, color 0.8, shape 0.2, smoothness 0.9, & compactness 0.1

2.4.5 Accuracy assessment

Accuracy assessment values were generated in eCognition by creating a test area and training mask (TTA) as shown in table 2. The TTA mask contained 52 "Urban," "Vegetation," and "rocky" objects and 25 "water" objects. These objects, representing actual land cover were compared against the classified identity of these objects. The "water" class was very accurately classified, and was therefore limited to 25 testing objects in order to reduce its inflationary effect on the accuracy statistics.

Table 2: Error matrix and Accuracy statistics

	Reference data*				
Classification Data	**U.A**	**W**	**V**	**R**	**Total**
U.A	12766	0	0	951	13717
W	5168	59897	0	0	65065
V	0	0	17600	0	17600
R	0	0	1180	30336	31516
Total	17934	59897	18780	31287	127898

*U.A-Urban Area, W-Water, V-Vegetation, R-Rocky

Producer's accuracy can be calculated using the formula:

$$\textbf{PA (class I)} = a_{ii} \div \sum_{i=1}^{n} a_{ki}$$

Producer's accuracy (%):
Urban area=12766/17934=71.18
Water=59897/59897=100
Vegetation=17600/18780=93.7
Rocky=30336/31287=96.9
User's accuracy can be calculated using the formula:

$$\textbf{UA (class I)} = a_{ii} \div \sum_{i=1}^{n} a_{ik}$$

User's accuracy(%):
Urban area=12766/13717=93
Water=59897/65065=92
Vegetation=17600/17600=100
Rocky=30336/31516=96.2

Over all accuracy can be calculated using the formula:

$$\mathrm{OA}=\sum_{k=1}^{n} a_{kk} \div \sum_{i,k=1}^{n} a_{ik} = 1/n\sum_{k=1}^{n} a_{kk}$$

Over all accuracy= (12766+59897+17600+30336)/127898=94.2%

Kappa Statistics can be computed as:

$$K = N \sum_{i=1}^{n} a_{ii} - \sum_{i=1}^{n}(a_{i+} * \dot{a}_{+i}) / N^2 - \sum_{i=1}^{n}(a_{i+} * \dot{a}_{+i})$$

Where

n=no. of the rows in the matrix

$\dot{a}_{ii}$=the no. of observations in row *i* and column *i* (on the major diagonal)

$\dot{a}_{i+}$=total of observations in row *i*

$\dot{a}_{+i}$= total of observations in column *i*

N=total no. of observations included in matrix

Therefore

Kappa Statistics:

$$K= \frac{127898(12766+59897+17600+30336)-((13717*17934)+(65065*59897)+(17600*18780)+(31516*31287))}{127898*127898-((13717*17934)+(65065*59897)+(17600*18780)+(31516*31287))}$$

=0.91

An overall accuracy of 0.942 and a Kappa Index of Agreement (KIA) of 0.91 are fairly reasonable and good accuracy levels. However, it is felt that there is still much misclassification that can be improved upon. It is hoped that this can be improved by exploiting some class related features and topological relationships.

Histogram for this classification is given in figure 1.

Statistics of the classified image are given in table 3.

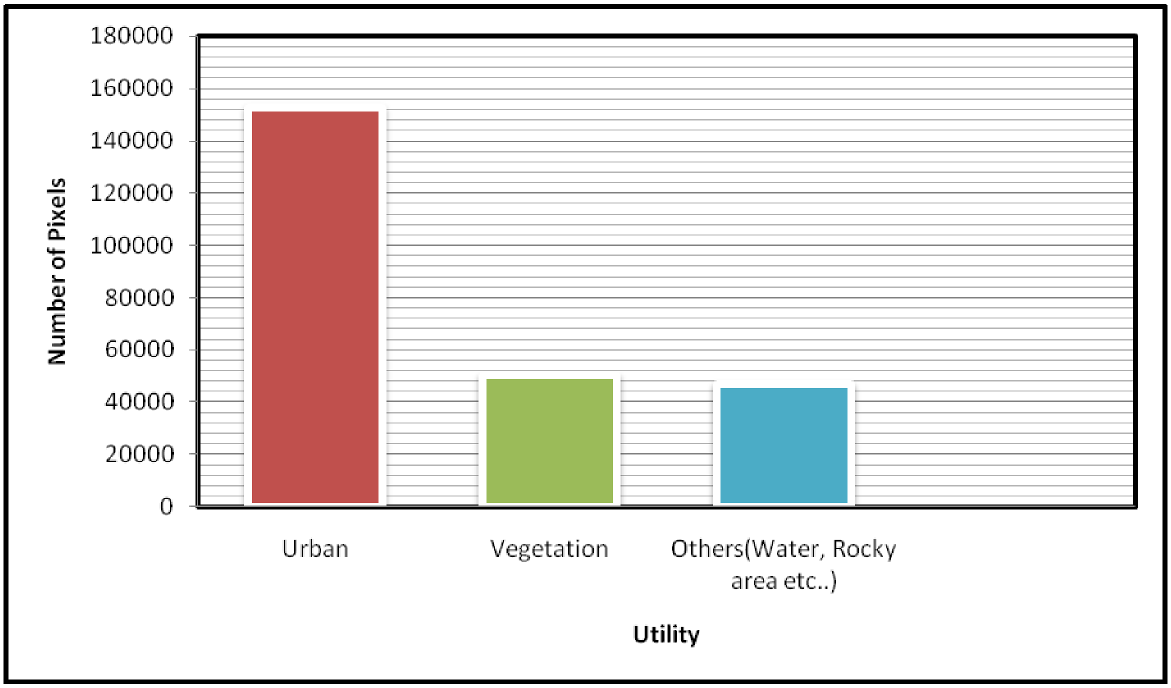


Figure 1: Histogram of the classified image

Table 3: Statistics of classification result

Land cover lasses	**Pixel number**	**Pixel no. P(%)**	**Area(Sq.Km)**
1)Urban	152746	10.54	34.6
2)Vegetation(Forestry)	50326	3.47	11.4
3)Others(Water, Rocky area etc..)	46795	3.23	10.6

From the Histogram of this classification it is clear that out of the 58 sq.kms of the study area the urban area covers 34.6 sq.km which includes residential, commercial, industrial, traffic and transportation, public utility etc., the vegetation (trees, plants, shrubs etc.,) covers 11.4 sq.km and water, rocky area etc., covers 10.6 sq.km.

3. CONCLUSIONS, RECOMMENDATIONS AND PROPOSALS

3.1 PLANNING EFFORTS

The way with which the city is growing and developing due to the migration of population from rural areas for employment and other opportunities, it has been proposed that the ultimate land use structure of the Vijayawada urban area in the coming 20 years should be around 130 sq.km. The residential area is proposed to cover about 48% followed by transport and recreation uses. The land use pattern for the coming 20 years should definitely be far more balanced compared to the prevailing situation if the authorities concerned look in to the following recommendations and proposals.

3.2 RECOMMENDATIONS AND PROPOSALS

- The proposals aim at municipal performance improvement of environmental infrastructure and aims at socio-economic development.
- The proposals for municipal reforms are aimed at enhancing the efficiency, effectiveness and service delivery with accountability.
- The reform proposals should include privatization of advertisement tax collection, revenue improvement, town development, operation and maintenance of critical infrastructure investment.
- The environmental infrastructure proposals aim at improvement of infrastructure in the prioritized poor settlements as per poverty and infrastructure deficiency matrices and linked infrastructure for poor settlements.
- These include rehabilitation of existing infrastructure provision of water supply, roads, drains, sanitation and street lighting based on community prioritization and construction of drains to improve the living environment.
- The social development proposals aim at addressing the socio-economic needs identified and prioritized through participatory micro planning process.
- These proposals cover areas of health, education, livelihood, vulnerability and strengthening of SHGs (Self help groups), with focus on gender issues.
- This leads to the reduction of poverty and improvement in living conditions of the people in the poor settlements.

4. REFERENCES

1. "Erdas Imagine, 8.5", User Guide, 2001
2. T.M. Lillesand, and R.W. Kiefer. "*Remote Sensing and Image Interpretation*". 4th ed, John Wiley and Sons, inc. USA, 2001, ISBN: 0471255157, 2001
3. Mather, M.Paul. "*Computer Processing of Remotely-Sensed Images*". St Edmundsbury Press Ltd., Bury St Edmunds, Suffolk, Wiley and Sons, ISBN: 0-471-90648-4,1987
4. "ECognition user guide", Concept and Methods, 2001
5. U.C. Benz, P. Hofmann, G. Willhauck, I. Lingenfelder, M. Heynen. "*Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS - ready information*". *ISPRS* Journal of Photogrammetry and Remote Sensing, 58: 239-258, 2004
6. N. Goward, F. Huemmrich, and H. Waring. "*Visible-near infrared Spectral reflectance of landscape components in western Oregon*". Remote Sensing environment, 47:190-203, 1994

7. D. Klimešová, E. Ocelíková. "*Spatial data in land management and local government"*. Proceedings of the 4th Conference of the European Federation for Information Technology in Agriculture. Debreceen, 363-368, 2003
8. S.P.S. Kushwaha, S.K. Subramanian, G.CH. Chennaiah, J. Ramana Murthy, S.V.C. Kameswara Rao, A. Perumal, and G. behera. "*International Remote Sensing and GIS methods for sustainable rural development*". International Journal of Remote Sensing, 17: 3055-3069, 1996
9. J.C. Price. "*How unique are spectral signatures?"*. Remote Sensing of Environment, 49: 181- 186, 1994

A Simple Segmentation Approach for Unconstrained Cursive Handwritten Words in Conjunction with the Neural Network.

Amjad Rehman Khan amjadbzu2003@yahoo.com
PhD researcher
Department of Computer Graphics and Multimedia
University Technology Malaysia
Skudai, 81310, Malaysia

Dr. Zulkifli Mohammad dzullkilfli@utm.my
Associate Professor
Department of Computer Graphics and Multimedia
University Technology Malaysia
Skudai, 81310, Malaysia

Abstract

This paper presents a new, simple and fast approach for character segmentation of unconstrained handwritten words. The developed segmentation algorithm over-segments in some cases due to the inherent nature of the cursive words. However the over segmentation is minimum. To increase the efficiency of the algorithm an Artificial Neural Network is trained with significant amount of valid segmentation points for cursive words manually. Trained neural network extracts incorrect segmented points efficiently with high speed. For fair comparison benchmark database IAM is used. The experimental results are encouraging.

Keywords: Image analysis, Segmentation, Neural Network, Preprocessing, Pattern matching.

1. INTRODUCTION

An extensive research has been done in the field of handwriting recognition in the last few decades [1]. It seems that the research has been reached to its maturity for the recognition of isolated characters recognition, hand printed words recognition, automatic address processing and bank check reading (holistic approaches) [2-4]. In contrast for the analytical approaches where the word is segmented into its component characters, recognition results for unconstrained handwriting is still low due to the poorly segmented words. Segmentation errors mislead classifier during character recognition [5-7]. In fact segmentation problem has persisted for nearly as long as handwriting recognition problem itself. In literature, segmentation algorithms for unconstrained handwritten words can be generalized into two categories.[7] External segmentation and Holistic segmentation. In the former category letter boundaries are determined prior to recognition while in the latter, segmentation and recognition are carried out at the same time and the final character boundaries are determined dynamically by semantic analysis and classification performance.[8-10].

Higher the segmentation accuracy, the more beneficial it is to the recognition rates [11]. Hence the segmentation is the backbone of the recognition process and is still active research topic. Researchers have acknowledged the important role that segmentation plays in handwriting recognition process [7, 12-13]. That is why more innovative, accurate and fast methods need to be employed and compared to the work of other researchers using benchmark databases.

In most of the existing segmentation algorithms, human writing is evaluated empirically to deduce rules [15]. Sometimes the rules derived are satisfactory but there is no guarantee for their optimum results in all style of writing. Because human writing varies from person to person and even for the same depending on mood, speed, environment etc. On the other hand researchers have employed artificial neural networks, hidden Morkov models, statistical classifiers etc to extract rules based on numerical data [16-21, 36-37]

This research attempts to integrate, rule based segmentation approach and intelligent method for the character segmentation of unconstrained handwritten words.

A simple but efficient, rule based segmentation algorithm is presented that performs character segmentation very well with high speed but some characters are over-segmented. Therefore an ANN is integrated with the proposed approach as artificial neural networks have been successfully used in the field of pattern recognition [14, 20-23]. To verify the segmentation points marked by proposed algorithm, an artificial neural network is trained with correct and incorrect segmentation points for the words images taken from benchmark database [24].

The rest of the paper is organized in four sections. Section 2 presents proposed segmentation algorithm along with segmentation results. In section 3, neural based experimentation is performed and results are discussed. Finally, conclusion and future work is drawn in section 4.

2. PROPOSED SEGMENTATION APPROACH

In this section segmentation algorithm and preprocessing steps are presented. Artificial neural network is trained manually for the correct and incorrect segmentation points obtained from the proposed segmentation technique. MATLAB 7.0 is used for all experiments performed on system of 1.6 GHz processor and 1 GB DDR RAM.

2.1 Preprocessing and proposed segmentation algorithm

The original grey scaled image is binarized using Otsu algorithm by selecting automatically a threshold value for a given image [22]. If required, following binarization, slant correction is performed [23]. Finally, image is converted to skeleton format to allow users verity of writing device, pen tilt and to suppress extra data. The proposed segmentation algorithm is explained in the figure 1.

Step 1. Take word image from database.
Step 2. Perform pre-processing.
Step 3. Calculate sum of foreground pixels (white pixels) for each column. Save those columns as candidate segment column (CSC) for which sum is 0 or 1 only.
Step 4. By previous step, we have more candidate segmentation columns than actual required. Hence threshold (approximate character width) is selected empirically from candidate segment columns to come out with actual segment columns.

FIGURE 1: Proposed Segmentation Algorithm

Due to the simplicity of the proposed segmentation technique, it is very fast and performs well in most of the cases. For few characters such as m, n, u, v and w over segmentation occurs and this technique fails to find accurate character boundaries. Segmentation results by the proposed segmentation technique are shown in figure 2**.**

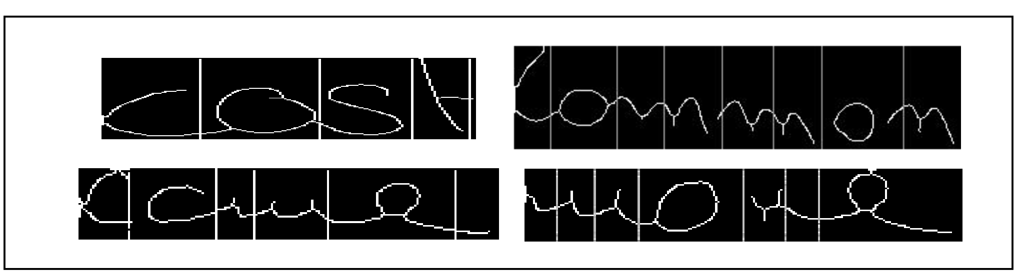


FIGURE 2: Segmentation Results by Proposed Segmentation Approach.

It can be seen from the results that segmentation is good except for few characters, where over segmentation occurs. Therefore it is required to integrate this technique with some intelligent method to increase its performance. In this regard a trained neural network is employed.

It is mention worthy that over segmentation is minimum and occurs for few characters only. Hence it lessened burden of the classifier used and therefore processing speed increased.

2.2 Handwriting database

For the fair comparison, patterns are selected from IAM V3.0 benchmark database [24]. A few samples for segmentation, training and testing of the ANN are shown in figure 3. The reason for selecting this database is that, it is freely available for researchers to comparing their results.

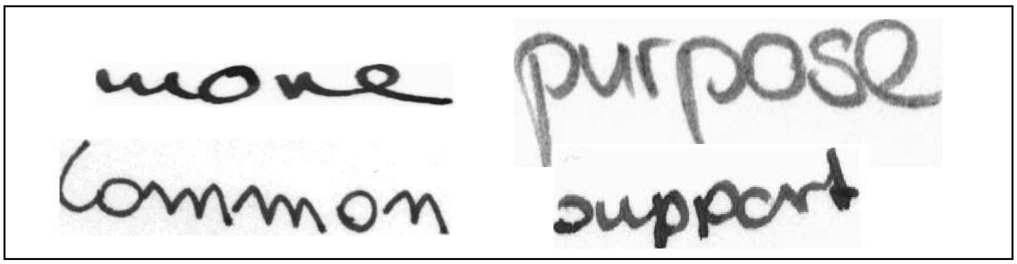


FIGURE 3: Samples of Word Images from IAM Database

3. EXPERIMENT AND RESULTS

3.1 Training Artificial Neural Network.

A simple program in MATLAB 7.0 is developed to detect co-ordinates of all segmentation points given birth by the proposed segmentation technique for each pattern. These segmentation points are divided into correct and incorrect categories manually and stored in a training file. Data is preprocessed prior to use for ANN training.

For training, ANN with standard back propagation algorithm is used. A number of experiments with different structures, weights, epochs, momentum and learning rate are performed to enable ANN to distinguish between correct and incorrect segmentation points. The ANN trained with 25072 training patterns (segmentation points) taken from 2678 words. The optimal structure of ANN thus found contained 235 to 310 inputs, 25 to 38 hidden units and one output (correct or incorrect segmentation point) with 300 epochs. Learning rate and momentum was set to 0.2 and 0.6 respectively. MATLAB 7.0 is used for implementation. Trained neural network operates in figure 4.

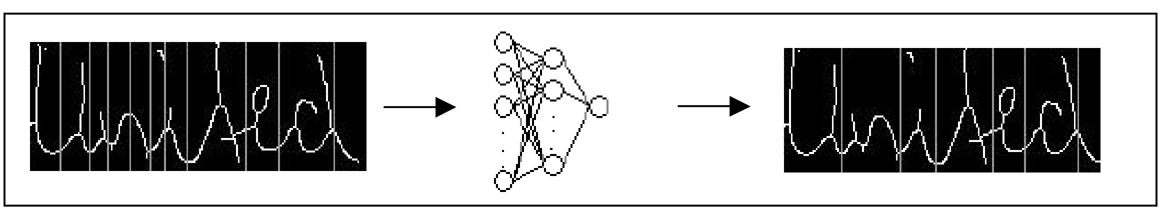


FIGURE 4: Incorrect Segmentation Points are rejected by Trained Neural Network.

3.2 Performance of the neural rule-based Segmentation Technique.

After training, the testing phase occurs. For testing phase 2936 samples are selected from the IAM database. These new patterns are segmented by the proposed algorithm. Segmentation points thus obtained are fed to the train ANN for their classification into correct and incorrect categories. Finally, correct are left and incorrect are rejected by trained ANN as shown in figure 4. Segmentation results for test set are presented in table 1.

Correct segmentation rate.	2678/2936
% Correct segmentation rate.	91.21 %
% Miss segmentation rate.	5.38 %
% Over segmentation rate.	3.20 %

TABLE 1: Segmentation Rates

3.3 Analysis and discussion of results

The neuro rule-based segmentation algorithm achieved recognition rate of 91.21% for valid identification of 2936 segmentation points pattern. Two problems are found during the analysis of the results. Firstly, noisy characters, so some additional preprocessing is done before training ANN. Secondly, touched/ overlapped characters. This type of problem is very hard to deal with. When two characters are tight together, ligatures can't be found and therefore they can't be segmented. Hence overall correct segmentation results decreased.

It is very hard to compare segmentation results with the other researcher because segmentation is intermediate process of recognition. In addition to that many researchers report recognition rates only. Moreover different researcher used different database and report results under some constrains.

Segmentation results are reported in literature is presented in table 2 for fair comparison.

Author	Segmentation method	Segmentation rate	Database used	Comments
Tappert et al [25]	Feature based + Rule based	81.08%	CEDAR	number of words not mentioned
Srihari [26]	ANN	83%	Handwritten zip codes	No alphabetic
Han and Sethi [27]	Heuristic algorithm	85.7 %	Latin handwritten Words on 50 real mail envelopes	Only 50 mail envelopes are taken.
Lee et al [28]	ANN	90%	Printed latin alphanumeric characters	Printed alphanumeric characters used
Eastwood et al [29]	ANN	75.9 %	Cursive latin handwritten from CEDAR database	100,000 training pattern used
Blumenstein and Verma [30]	ANN + conventional method	81.21 %	2568 words from CEDAR	
Yanikoglu and Sandon [31]	Linear Programming	97 %	750 words	No bench mark database used
Nicchiotti and Scagliola. [32]	Rule-based	86.9 %	CEDAR	850 words used only.
Verma and Gader [33]	Feature based + ANN	91	CEDAR	words number not mentioned
Blumenstein and Verma [34]	Feature based+ ANN	78.85%	CEDAR	words number not mentioned
Verma[35]	Feature based + ANN	84.87	CEDAR	300 words only
Cheng et al [36]	Feature based + ANN	95.27	CEDAR	317 words only
Cheng et al [37]	Enhanced feature based+ ANN	84.19	CEDAR	317 words only

TABLE 2: Comparison of Segmentation Results in the Literature.

4. CONCLUSION & FUTURE WORK

Segmentation is the important step of analytical approaches employed to handwritten word recognition. Hence it is the base of most modern approaches. It is admitted fact that no segmentation method can directly locate character location accurately without an intelligent method. In this paper, proposed segmentation algorithm is integrated with neural network using standard back propagation. Initial experiments exhibit very encouraging results with segmentation accuracy up to 91.21%. Speed is another important factor, overlooked in many past researches. Due to the minimum over segmentation, neural network is least burdened and therefore speed is optimum. This paper has briefly described one stage on our progress towards final goal of unconstrained cursive words recognition with higher recognition rate and speed.

ACKONOLOGEMENT

This research work is fully supported by the Ministry of Science, Technology and Innovation Malaysia under grant vote 79258. The author is also thankful to colleague researcher Fajri Kurniawan for his help in experimentation.

5. REFERENCES

1. H.Bunke. "*Recognition of cursive roman handwriting, past, present and future*". In Proceeding of 7th International Conference on Document Analysis and Recognition, 448-461, 2003.

2. C.Y.Suen, R.Legault, C. Nadal, M.Cheriet and L.Lam. "*Building a new generation of handwriting recognition systems*". Pattern Recognition Letters, 14: 305-315, 1993.

3. S.W. Lee. "*Multilayer cluster neural network for totally unconstrained handwritten numerical recognition*". Neural Networks, 8: 783-792, 1995

4. H. I. Avi-Itzhak, T. A. Diep and H.Garland. "*High accuracy optical character recognition using neural networks*". IEEE Trans. Pattern Analysis and Machine Intelligence, 18: 648-652, 1996.

5. S.B. Cho. "*Neural networks classifiers for recognition totally unconstrained handwritten numericals*". IEEE Trans. On Neural Networks, 8.

6. S.W. Lee. "*Off-line recognition of totally unconstrained handwritten numericals using multilayer cluster neural network*". IEEE Trans Pattern Analysis and Machine Intelligence, 18: 648-652, 1996.

7. R.G. Casey, E. Lecolinet. "*A survey of methods and strategies in character segmentation*". IEEE Trans. Pattern Analysis and Machine Intelligence, 18: 690-706, 1996.

8. R. Farag. "*Word–level recognition of cursive script*". IEEE Trans. Computing 28: 172-175, 1979.

9. J. M. Bertille, M. E. Yacoubi. "*Global cursive postal code recognition using hidden Marko models*". In Proceeding of the First European Conference Postal Technology, France, 129-138, 1993.

10. J. Wang, J. Jean. "*Segmentation of merged characters by neural networks and shortest path*" Pattern Recognition 27(5): 649-658, 1994.

11. C. K. Cheng, M. Blumenstein. "*The neural-based segmentation of cursive words using enhanced heuristics*". In Proceedings of Eighth International Conference on Document Analysis and Recognition (ICDAR'05), 650-654, 2005.

12. S.N. Srihari. "*Recognition of handwritten and machine printed text for postal address interpretation*". Pattern recognition letters, 14: 291-302, 1993.

13. M. Gilloux. "*Research into the new generation of character and mailing address recognition systems at the French post office research center*". Pattern Recognition Letters.14: 267-276 1993.

14. M. Blumenstein, B. K. Verma. "*An artificial neural network based segmentation algorithm for off-line handwriting recognition*". In proceedings of International Conference on Computational Intelligence and Multimedia Applications (ICCIMA' 98), Gippsland, Australia, 1997.

15. X. Xiao, G. Leedham. "*Knowledge–based English cursive script segmentation*". Pattern recognition letters 21: 945-954, 2000.

16. M. Giloux. "*Hidden Markov Models in Handwriting Recognition*". Fundamentals in Handwriting Recognition, S.Impedovo ed., NATO ASI Series F: Computer and System Science, 24: Spinger Verlang, 1994.

17. M. El. Yacoub, M. Gilloux and J. M. Bertille. "*A statistical approach for phrase location and recognition with in a text Line*". An application to Street Name Recognition, IEEE Trans. Pattern Analysis and Machine Intelligence, 24(2): 172-188, 2002.

18. A. Khotanzad, J. Lu. "*Shape and texture recognition by a neural network*". Artificial Neural Networks in Pattern Recognition, Elsevier Science Publishers B.V., Amsterdam, Netherlands, 109-131, 1991.

19. B. Zheng, W. Qian and L.Clarke. "*Multistage neural network for pattern recognition in mammogram screening*". IEEE ICNN, Orlando, 3437-3448, 1994.

20. A.D. Kulkarni. "*Artifical neural networks for image understanding*". Van Nostrand Reinhold, New York. 154-270, 1994.

21. K. Han, I. K. Sethi. "*Handwriting signature retrieval and Identification*". Pattern Recognition Letters, 17, 83-90, 1996.

22. N. Otsu. "*A threshold selection method from gray level histograms*". IEEE transactions on systems, Man and Cybernetics, 9(1): 62-66, 1979.

23. S. Knerr, E. Augustin. "*A neural network–hidden markov model hybrid for cursive word recognition*". In Proceedings of International Conference on Pattern Recognition, Brisbane, 2: 1518-1520, 1998.

24. U. Marti, H. Bunke. "*The IAM database: An English sentence database for off-line handwriting recognition*". International Journal of Document Analysis and Recognition, 15: 65-90, 2002.

25. C. C. Tappert., C. Y. Suen and T. Wakahara. "*The state of the art in on-line handwriting recognition*". IEEE Trans. Pattern Analysis. Machine. Intelligence. 12: 787-808, (1990)

26. S. N. Srihari. "*Recognition of handwritten and machine-printed text for postal address Interpretation*". Pattern Recognition Letters 291-302, 1993.

27. K. Han, I. K. Sethi. "*Off-line cursive handwriting segmentation*". ICDAR 95, Montreal, Canada, 894-897, 1995

28. S-W. Lee, D-J. Lee and H-S Park. "*A new methodology for gray–scale character segmentation and recognition*". IEEE Transaction on Pattern Analysis and Machine Intelligence, 1045-1051, 1996.

29. B. Eastwood, A. Jennings and A. Harvey. "*A feature based neural network segmenter for handwritten words*". ICCIMA'97 Australia, 286-290. 1997

30. M. Blumenstein, B. Verma. "*A segmentation algorithm used in conjunction with artificial neural networks for the recognition of real-word postal addresses*". In Proceeding of International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'97), Gold Coast, Australia. 155-160, 1997.

31. B. Yanikoglu, P.A.Sandon. "*Segmentation of off-line cursive handwriting using linear programming*". Pattern Recognition, 31: 1825-1833, 1998.

32. G. Nicchiotti, C.Scagliola. "*Generalized projections: a tool for cursive handwriting normalisation*". In Proceedings of 5th International Conference on Document Analysis and Recognition, Bangalore, 729-733, 1999.

33. B. Verma, P. Gader. "*Fusion of multiple handwritten word recognition techniques"*. Neural Networks for Signal Processing X, 2000. In Proceedings of the IEEE Signal Processing Society Workshop, 2: 926-934, 2000.

34. M. Blumenstein, B. Verma. "*Analysis of segmentation performance on the CEDAR benchmark database*". In Proceedings of Sixth International Conference on Document Analysis and Recognition (ICDAR'01), 1142, 2001.

35. B. Verma. "*A contour character extraction approach in conjunction with a neural confidence fusion technique for the segmentation of handwriting recognition*". In Proceedings of the 9th International Conference on Neural Information Processing. 5: 18-22, 2002.

36. C. K. Cheng., X. Y. Liu., M. Blumenstein and V. Muthukkumarasamy. "*Enhancing neural confidence-based segmentation for cursive handwriting recognition*". In Proceeding of 5th International Conference on Simulated Evolution and Learning (SEAL '04), Busan, Korea, SWA-8, 2004.

37. C. K. Cheng, M. Blumenstein. "*Improving the segmentation of cursive handwritten words using ligature detection and neural validation*". In Proceedings of the 4th Asia Pacific International Symposium on Information Technology (APIS 2005), Gold Coast, Australia, 56-59, 2005.

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA