# Human Interaction with Digital Ink: Legibility Measurement and Structural Analysis

Timothy Simon Butler

A thesis submitted in partial fulfilment of the
requirements of the University of Hertfordshire
for the degree of Doctor of Philosophy

The programme of research was carried out in the
Department of Computer Science,
Faculty of Engineering and Information Sciences,
University of Hertfordshire

December 2003

# Abstract

Literature suggests that it is possible to design and implement pen-based computer interfaces that resemble the use of pen and paper. These interfaces appear to allow users freedom in expressing ideas and seem to be familiar and easy to use. Different ideas have been put forward concerning this type of interface, however despite the commonality of aims and problems faced, there does not appear to be a common approach to their design and implementation.

This thesis aims to progress the development of pen-based computer interfaces that resemble the use of pen and paper. To do this, a conceptual model is proposed for interfaces that enable interaction with "digital ink". This conceptual model is used to organize and analyse the broad range of literature related to pen-based interfaces, and to identify topics that are not sufficiently addressed by published research. Two issues highlighted by the model: digital ink legibility and digital ink structuring, are then investigated.

In the first investigation, methods are devised to objectively and subjectively measure the legibility of handwritten script. These methods are then piloted in experiments that vary the horizontal rendering resolution of handwritten script displayed on a computer screen. Script legibility is shown to decrease with rendering resolution, after it drops below a threshold value.

In the second investigation, the clustering of digital ink strokes into words is addressed. A method of rating the accuracy of clustering algorithms is proposed: the percentage of words spoiled. The clustering error rate is found to vary among different writers, for a clustering algorithm using the geometric features of both ink strokes, and the gaps between them.

The work contributes a conceptual interface model, methods of measuring digital ink legibility, and techniques for investigating stroke clustering features, to the field of digital ink interaction research.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

This chapter gives an introduction to the topic covered by the thesis, followed by an overview of the entire thesis, with reference to the detail contained in the following chapters.

## 1.1 Background

Digital ink is data that is input to a computer via an electronic pen, such as those used with graphics tablets, tablet PCs, or PDAs. The digital ink is usually either represented on screen by depicting the trace of the pen, or recognized by a handwriting recognition program and sent as input to an application.

There seem to be two approaches to designing a pen-based interface. The first is that commonly found on PDAs. Here the pen is used to point and select. Handwriting recognition of some form, or an on-screen keyboard, is used to enter textual input. These interfaces resemble those found on desktop PCs. The movement of the pen is treated like that of a mouse, and handwriting recognition replaces the keyboard.

The second approach is to construct an interface that resembles the use of a pen and paper. The idea is, that instead of treating a pen as a replacement for a mouse and keyboard, a pen-based interface can exploit the strengths inherent in the pen. Using a pen as a mouse and keyboard replacement has inherent drawbacks, for instance: When a pen is used like a mouse, the hand can obscure information on the screen; handwriting or the use of on-screen keyboards is also much slower than typing on a real keyboard. On the other hand, there are strengths of pen-use: A pen can be used to draw instantly, and can also describe intricate figures with a finer definition than a mouse.

We are interested in this second type of pen-based interface. These interfaces allow users to freely write anything, anywhere, without any hindrance. They do not usually rely on handwriting recognition. Algorithms are employed however, to recognize basic patterns formed by the digital ink. This allows users to manipulate the digital ink on the computer screen in a meaningful way and achieve more than they could with traditional pen and paper.

Although pen-based computers have been available for some time, almost all devices available commercially have interfaces designed following the first approach. Pen-based computer interfaces that reflect the real-world use of pen and paper appear to be few and far between.

Over the past ten years or so, many different ideas have been put forward concerning computer interfaces resembling pen and paper. As will be seen later, although sharing many common aims, published work appears fairly disparate in the topics covered. The overarching aim of this thesis is to find a way forward; to progress research that could lead to the commercial realization of computer interfaces resembling pen and paper.

### 1.1.1 A New Interface Model

The majority of this thesis centres around our conceptual model of interaction with digital ink. This is in contrast to the common approach of translating pen-movement into input for familiar computer interfaces. Our model is derived from a number of ideas present in literature on pen-based interfaces; in particular the concept of Informal Interaction. Informal Interaction is an emerging interaction paradigm that is founded on enhancing the tasks commonly carried out using pen and paper. Our conceptual model takes this idea and develops the practical requirements of implementing such an interface.

Although primarily a conceptual interface model, our model also provides a framework in which published literature concerning pen-based interfaces can be organized. This facilitates the identification of research topics, associated with this style of interaction, that have not yet been fully addressed. Two identified topics are chosen to demonstrate the utility of the model in isolating research topics: digital ink legibility, and digital ink structuring. These topics are explored, in turn, in the latter half of the thesis.

The rest of this chapter describes the motivation and aim of the work undertaken. This is followed by a summary of the methodology employed, and a short description of the content of each chapter.

## 1.2 Motivation

Published literature suggests that it is possible to design and implement pen-based interfaces that resemble the use of pen and paper. These interfaces appear to allow users freedom in expressing their ideas, and seem to be familiar and easy to use.

Interfaces like this have been demonstrated in the domains of: marking up documents [63]; note taking [10]; and web-page [37] and architectural [24] design. These prototypes have also added automatic editing, searching, or prototyping functionality that would not have possible using paper.

Over the past 10 years or so, many different ideas have been put forward concerning this type of interface. These include working with hand-drawn diagrams [12, 24, 62], working with handwritten script [10, 13], and recognizing the structure of information described by the digital ink on the page [36, 45, 46]. However despite the commonality of aims and problems faced, there does not appear to be a common approach to designing and implementing these interfaces.

Pen-based interfaces that resemble using a real pen and paper, could feel familiar and be easy to use. They may be able to assist people in expressing and exploring their ideas. They might be flexible, capable of integrating seamlessly with many different tasks. Despite this potential pen-based computer interfaces that reflect the real-world use of pen and paper appear to be few and far between. These possibilities present many interesting avenues of research.

## 1.3 Aim

This thesis focusses in on the topic of a pen-based interface that uses the pen in its standard capacity of a writing and drawing tool, rather than a pointing device similar to a mouse. Looking at the previous research in this area, projects have concentrated on: design work [24, 37]; note taking [10]; and document markup, adding notes or revision information [57, 63]. These are all "document based" tasks. Tasks that are frequently approached with pen and paper. These may not be surprising targets for computer interfaces that resemble the use of pen and paper.

The type of questions that could be asked are: What is standing in the way of these interfaces? Why, when there is such potential in these interfaces, are they not more common? Perhaps more importantly, how should research into these interfaces progress to further their development and exploitation?

The aim of this thesis therefore is to answer the question:

> "How can the principal barriers to an intuitive interface to draft and edit documents on a pen-based computer be overcome?"

## 1.4 Methodology

To address the aim, a number of stages are undertaken. The initial problem is to specify and define the exact nature of an "intuitive interface to draft and edit documents on a pen-based computer". Once this has been done the "principal barriers" to the interface are identified. Finally, work is demonstrated that investigates those barriers, providing an answer to the thesis question.

This process is commenced by presenting a conceptual model of such an intuitive interface. The model is derived from literature concerning pen-based interfaces and Informal Interaction. Informal Interaction is an emerging interaction paradigm that is founded on enhancing the tasks commonly carried out using pen and paper. The conceptual model is the lynchpin of the thesis on which the rest of the work presented hangs. Elements of the model are evaluated by the experimental work. The primary concern of the evaluation of the thesis is whether or not the use of this model was the most suitable strategy to adopt in answering the thesis question.

After the model has been presented, it is then used to organize literature concerning pen-based interfaces. Such a strategy is required for two reasons. Firstly, although there is a body of literature concerning pen-based interfaces, there is no obvious or generally accepted framework in which to present and understand it. Secondly, the model focusses on the requirements of the "intuitive interface" it depicts, draws in relevant literature not explicitly concerning pen-based interfaces, and in this establishes the whole context of the thesis.

Once the literature has been presented, the model is used again, this time to isolate the "principal barriers". A principal barrier could be either a topic that has not yet been conclusively addressed, or one that has not been addressed at all. This stage uncovers a number of topics to address. The principal barriers are identified with reference to a layer structure within the model. This indicates that those topics at lower layers support the functionality of the upper layers. Thus the topics of "digital ink legibility" and "digital ink clustering" identified at the two lowest layers are put forward as the "principal barriers".

Two experiments then follow, piloting work to address the barriers that have been identified. The purpose of these experiments is twofold. Firstly to investigate "how" the barriers may be overcome, and secondly to validate the division of concerns within the model. If each barrier can be addressed with reference only to elements within the layer in which it was identified, then the work may increase confidence in the usefulness of the model.

Finally the work presented is evaluated, conclusions are drawn, and further work proposed. Evaluation happens on a number of different levels including the interface model, each experiment, and the methodology applied in arguing the thesis.

## 1.5 Overview

This thesis is broken down into six chapters. The current chapter provides an introduction to the work done, presents the thesis question, and summarizes the methodology employed in answering it.

Chapter two is the literature review. The chapter commences by introducing the concept of Informal Interaction and briefly giving an overview of literature on pen-based interfaces. After this, the conceptual interface model is presented and used to structure a full review of relevant literature.

Chapter three briefly revisits the model, uncovering areas of potential work at each layer within the model. It concludes by analyzing these areas to identify the principal barriers to implementing an interface following the pattern depicted by the model.

Chapter four addresses the first principal barrier, that of digital ink legibility. The chapter presents an investigation into legibility measurement and proposes two ways of measuring legibility. The issue motivating the requirement to measure digital ink legibility is whether or not the display resolution on pen-based computers is adequate for displaying large amounts of handwritten script. The findings of this work suggest that the latest pen-based computers should have adequate screen resolutions, such that the legibility of handwriting is not degraded by poor screen resolution. However caveats are raised concerning smaller devices that may be required to display handwriting.

Chapter five tackles the second principal barrier. That of digital ink structuring, specifically the process of clustering digital ink strokes into words. Unlike digital ink legibility, digital ink clustering has been addressed before; however, the results presented were not adequate to determine whether or not ink-strokes can be *reliably* grouped into handwritten words. The work presented tackles this topic on three fronts. Firstly a general accuracy for geometric clustering methods is determined. Secondly, work is presented that aims to improve this result. Finally, work is undertaken to characterize different styles of handwriting, and relate them to factors used in clustering the ink-strokes. The chapter concludes by mapping out what could be done to take this work further.

Chapter six presents the conclusions to the thesis. It summarizes the work undertaken and its major findings.

# Chapter 2

# Literature Review

## 2.1 Informal Interaction

The term "Informal Interaction" was first coined by Moran et al. in 1995 in their paper *"Implicit structures for pen-based systems within a free-form interaction paradigm"* [46]. It is a paradigm for pen-based computer interfaces that is founded on enhancing the tasks commonly carried out using a pen. Inspiration is often taken from the use of (paper) notepads [80] and whiteboards [50].

An Informal Interface, a pen-based computer interface designed within the paradigm of Informal Interaction, could be considered as being an "intuitive" interface. This claim is based on the inspiration of the paradigm. The experience of using an Informal Interface should resemble that of using pen and paper, thus the interface may be considered familiar. In addition to this, an Informal Interface should implement further functionality to aid common pen-and-paper tasks, in effect "doing the right thing".

In 1995, Moran et al. [46] stated: "the notion of 'informal interaction' is somewhat vague". However our current review of literature on pen-based interfaces from the past decade suggests that Informal Interaction describes a pen-based interface exhibiting the following three attributes [6, 7]:

1. Freedom of expression.

2. Structured editing.

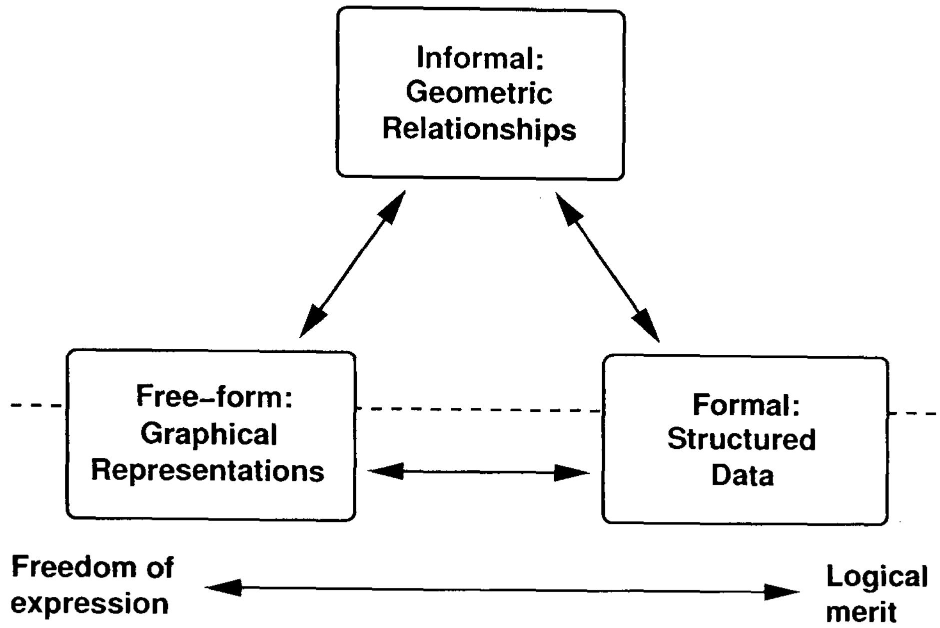



Figure 2.1: Interaction Styles

3. Integration with other applications.

This chapter is devoted to discussing and refining this definition.

### 2.1.1 Interaction Style

Informal Interaction combines the concepts of freedom of expression and structured editing. This is illustrated in Figure 2.1. The "free-form" and "formal" interaction styles describe very different types of computer application.

Free-form interaction covers applications like simple paint packages that handle user input only on the level of a coloured bitmap of pixels, they may have many painting tools but their use has no lasting effect in the sense that the applications store the canvas as one unstructured bitmap at all times. There is no formal structure on the canvas, but there are also no constraints on what the user can "paint".

Applications such as word processors are representative of formal interaction. The image on the screen is a graphic representation of an underlying structure: A list of words with their relative positioning and information content, and typographic attributes such as font and size. The graphical image facilitates the user's understanding of this underlying structure. Interaction is constrained to manipulating this structure using the graphical image for guidance only.

Informal Interaction applications use pen-strokes, rather than pixels as their basic elements. The users are free to write whatever they like with their pen. This constitutes *freedom of expression*. When a user wishes to interact with the *digital ink* on the page (screen), an Informal Interaction application will recognize the geometric relationships between the ink-strokes. This allows the application to support interaction with higher level structures, such as words, lists, or paragraphs. Thus enabling *structured editing* operations on free-form digital ink.

This "structure recognition" is one of the fundamental characteristics of an Informal Interface. Structure recognition identifies implicit *perceptual structures*. These are structures that are formed, possibly unsystematically, as ink-strokes are added to the page. The term 'perceptual" is used because these are structures that the user perceives, but has not explicitly specified. For instance, a collection of strokes may form a "word" structure, and many words may form a "paragraph" structure. The user however has simply written or drawn on the screen.

The goal of supporting perceptual structures also introduces the concept of *ambiguity*. Any collection of ink-strokes on a page may constitute many different perceptual structures, the interpretation then is ambiguous. An Informal Interaction application will be able to recognize, and allow the user to interact with, all possible structures within the ink-strokes. Choosing to interact with one particular perceptual structure does not fix this interpretation. Instead all possible interpretations can be used at all times. This concept is discussed later, and illustrated in Figure 2.3.

The final point to consider concerning interaction style is that of "interface modality". Interface modality describes the number of different modes an interface may enter. For instance an interface with a draw mode and an edit mode would have an interface modality of 2. Interface modality is commonly used in this fashion, to distinguish between data and commands. This could be implemented by pressing a button on the side of the pen when writing a command. There is a strong argument that Informal Interaction applications should implement 1-mode (non-modal) interfaces since this resembles the use of pen and paper.

### 2.1.2 Interface Characteristics

Now, after having looked at the concepts of "freedom of expression" and "structured editing", we can examine the attributes of Informal Interaction in more detail:

**Freedom of expression:** Just as with real pen and paper, users are free to mark any number of any shape of pen stroke, anywhere on a page. Users are not required to structure or specify any meaning associated with these free-form digital ink strokes. Ink remains as ink on the page and since explicit handwriting recognition is not employed, users are not distracted from their primary task of recording information on the page. All this helps to give the interface a familiar and informal feel, just like working with paper.

**Structured editing:** To manipulate, organize, and explore ideas on paper, requires redrawing or copying and using scissors and glue. This is not required when using a computer. Digital ink can be manipulated on-screen. Perceptual structures, such as a group of strokes forming a word, can be identified without any recognition of the word itself [10], simply by examining the spatial and temporal relationships of the individual ink-strokes.

While handwriting recognition may not be employed by Informal Interfaces, structural recognition is. Many authors agree that a large amount of information is contained in the implicit spatial relationships formed by the ink strokes on a page [36, 45, 46, 47, 61, 62]. The spatial relationships are implicit because they are not declared or defined, and they may change as further ink strokes are added to the page. For example, words may initially be written as a list, but as that list is expanded and detail added, list items become paragraphs. The expected response when interacting with these two different structures may be different. These differences should be detected and supported.

In recognizing this implicit structure as and when required, structured editing can be supported. For instance, groups of ink strokes representing words or diagrams can be selected and manipulated in a manner consistent with that interpretation. In the case of deleting a word, other words may re-flow to fill the space left, or in the case of moving a line in a diagram, it could be automatically positioned to join other lines.

It is worth stating at this point that previous research has tended to separate pen-based interfaces into either diagramming or note-taking interfaces. To fully support freedom of expression and structured editing, a functional Informal Interface will probably have to support both domains. In fact this division into two domains seems quite artificial. There is considerable cross-over between the two domains exhibited for instance in annotated diagrams, and structural marks in handwritten notes.

**Integration:** The third distinguishing attribute is the ability to integrate with other applications. If design work is carried out on paper it will often be transfered to a computer application once ideas have been developed to a certain stage [24, 37]. Likewise, people often print out documents to read and annotate them [63]. Informal Interfaces can import from, or export information to, other computer applications.

The processes of importing and exporting information to and from the Informal Interaction application are somewhat removed from the rest of the interface. Information import and export are likely to be two stages performed in their entirety at the beginning and end of completing a task with an Informal Interface. This is in contrast to the use the freedom of expression and structured editing functionality, use of which is likely to be interleaved with each other.

Since conversion is the user's primary task in import and export phases, user-mediated recognition techniques such as handwriting recognition, may be appropriate at these times.

## 2.2 Pen-Based Interfaces

Informal Interaction is still an emerging research area. As such there is no generally accepted structure to this paradigm. Research associated with Informal Interaction can be drawn from many different fields: ethnography, computer graphics, and handwriting recognition to name a few. There are no hard and fast rules stating what is and is not applicable to the study of Informal Interaction. To form a better idea of what may and may not be applicable, we will briefly review the literature associated with pen-based interfaces, published in the last 10–15 years.

The concept of interacting with a computer using a pen has been around for many years, from Bush's Memex machine in 1945 [5] and Kay's Dynabook in 1968 [29], to Microsoft's recent Tablet PC. The availability of pen-tablets with integrated displays in the late 1980s generated the contemporary research interest that we summarize here.

### 2.2.1 Pre Informal Interaction

In 1990 Tappert et al. recognized the suitability of pen-based interfaces for capturing and refining ideas when they published their review paper on *"The State of the Art in On-Line Handwriting Recognition"* [70]. Although more recent work has presented the case for avoiding handwriting recognition, Tappert et al. cite the reason for publication as a "renewed interest in on-line handwriting recognition ... for preparing a first draft and concentrating on content creation, ... editing, annotating, and other applications that are heavily interactive and that use direct pointing and manipulation." This 'renewed interest' was certainly evident in the late 80s shown by the publication of papers looking at the use of hand markings [21, 81].

1990 also saw the publication of work on *"A gesture based text and diagram editor"* [79]. The system was "designed to mimic the usual pen and paper type of editing" and seemed to do so, embodying many of the concepts of Informal Interaction. The interface was however quite modal, with text and diagrams, and input and editing all requiring different modes. Editing operations were recognized and executed immediately. The system also employed recognition of script and diagrams which seemed to be executed when a document was exported. However, the major feature distinguishing it from an Informal Interface, was that there was no recognition of any perceptual structure.

Research into text entry and document editing [20, 25] with pen-based interfaces continued through to 1993. There was also interest in diagramming, usually concentrating on automatic beautification and gesture driven editing [16, 67]. Pedersen et al. [50] also published work on their Tivoli system in 1993, an electronic whiteboard, working at the "stroke" level. They explored, among other topics, "the need to reconsider the basic assumptions behind the standard desktop GUI".

There are a number of common threads through all this work. Firstly, the recognition that the pen is actually a far better tool than a mouse and keyboard for certain tasks such as editing and drawing [21, 50]. Secondly, in implementing systems to demonstrate this they found that the look and feel of pen-based interfaces will depart from those used with mouse and keyboard [20, 50].

### 2.2.2 Post Informal Interaction

Recent work, from 1994 to 2002, associated with pen-based interfaces can be grouped into four major, but overlapping, topics: work with handwritten script [10, 36, 63, 80]; work with hand-drawn diagrams [12, 23, 24, 33, 37, 62]; work looking at perceptual structure [45, 46, 47, 61]; and work integrating pen-based interfaces with other applications [24, 47, 61]. The most recent work has concentrated on structure recognition and integration.

The DENIM application [37], built using the SATIN[1] toolkit [27], recognizes the structure of sketches of web pages and allows a designer to navigate their prototyped web-site. Gross and Do [24] also use the constraints of an application domain, and even allow end-user programming of these so that their interface can serve different tasks. However, Saund et al. [61] argue that the functionality of such systems relies on "prior constraints imposed from a targeted application domain", and target their work at supporting general structures within script and sketches. At an even more generic level, Moran et al. [45] describe techniques for manipulating digital ink based only on the boundaries of different regions.

The integration of pen-based interfaces with formal applications is also being addressed. As with structure recognition, there is a spectrum of different approaches. Integration is much more than exporting words and sketches to a word-processor or design package. Integration involves the exchange of information both ways. At a basic level Saund et al. [61], in their application ScanScribe,

[1] http://sourceforge.net/projects/informal/

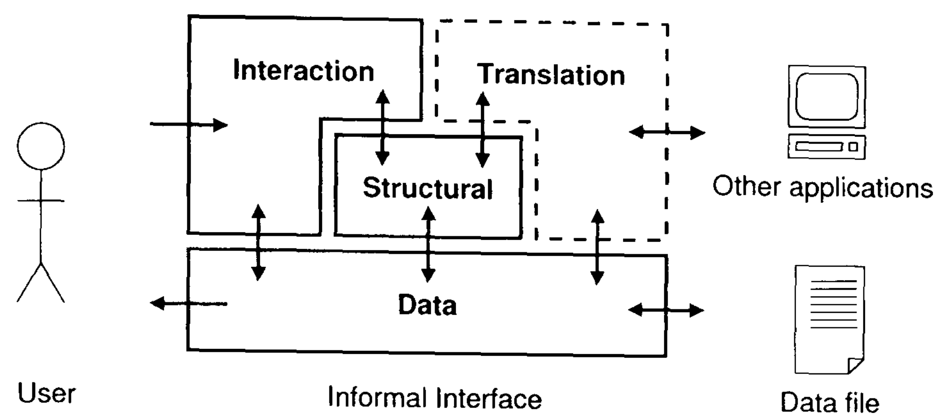



Figure 2.2: Informal Interface: Context and Layers

implement a way to import document images, which can then be manipulated by the user. Progressing from the idea of manipulating image data, Moran et al. [47] manipulate "Domain Objects". These objects are graphical *representations* of formal application data. The domain objects can then be manipulated with their interface, and resulting structures exported back to the original applications. Gross and Do [24] go yet another step further by enabling their interface to directly interact with formal applications. This creates a dual interface, where a sketch of an object will be recognized and created in a modeling application. Changes made within the formal application are then also represented in the sketch.

## 2.3 Informal Interface Structure

The aim of this section is to propose a conceptual interface which can then be used as a framework to organize and present literature concerning Informal Interfaces. The conceptual interface is formed in this section by identifying and outlining the main functional areas of an Informal Interface. The following sections define each area in greater detail and present the pertinent literature.

At this point we depart from our three-point attribute list and focus instead on what may be needed to implement that functionality. We will call these "functional layers". We have broken down the characteristics of an Informal Interface into four functional layers, which can be seen in Figure 2.2. This diagram also places the interface in some sort of context.

The four functional layers and the concepts they embrace are described below:

**The Translation** layer corresponds directly with the "integration" attribute from our working definition. This encompasses work on handwriting recognition, and other work on integrating pen-based interfaces with other applications. It is depicted in Figure 2.2 with a dotted border as its functionality only comes into play at certain times.

**The Interaction** layer does not correspond with any attribute from our working definition, but is required to facilitate interaction in the interface. It encompasses work such as selection and manipulation techniques, perhaps through the use of gestures.

**The Structural** layer corresponds roughly with the "structured editing" attribute from our working definition in that it supports the recognition and representation of possible structures within the interface.

**The Data** layer does not correspond directly to any attributes from our working definition. It could be thought of as supporting "freedom of expression" in that it encapsulates all data within the interface, whether structured or not.

Perhaps similar to this layered scheme, Hong and Landay, in their work on creating a toolkit for informal ink-based applications [27], analyze a number of different pen-based applications to identify and support common functionality. They identify 12 interrelated concepts, however these primarily concern functionality at our Data and Interaction layers. They implement some Structural functionality but no Translation layer functionality.

Our four layer structure stems from the work we have reviewed. The major differences between our structure and Hong and Landay's may be attributed to a simple fact: They looked to construct a framework to support *common* functionality in the applications they reviewed, whereas we propose a framework that attempts to embrace *all* the functionality of the work we reviewed. Our structure does however bear some resemblance to the structure of Gross and Do's "Back of an Envelope" work [24]. These similarities will be explored in the latter parts of this chapter, as well as in the conclusions of the thesis.

## 2.4 Data Layer

The Data layer constitutes the foundation of an Informal Interface. The layer encapsulates the digital ink data type and includes the implementation of: an abstract data structure, graphical representations, and a file format. With the exception of the SATIN toolkit [27], there is no work published in mainstream literature. There are clues to what has been implemented in other Informal Interface projects, but no details of the decisions taken or techniques used are available. We now take and examine the above-mentioned sub-areas in turn.

### 2.4.1 Abstract Data Structure

There is a consensus in literature that the basic element of online digital ink is the 'stroke', which is the stream of coordinate data describing the pen movement between pen-down and pen-up events [10, 24, 27, 36, 45, 50]. However it is clear that the Data layer must also be capable of handling other types of data [27] such as ASCII characters [63], or representations of objects from any arbitrary domain [47]. The layer must therefore implement a data structure to hold these different types of object. Each of these basic elements will include not only their raw data, but a graphical representation so they can be rendered and interacted with.

SATIN [27] implements a "scenegraph, a tree-like data structure that holds graphical objects", a concept taken from 3D modeling systems. The system employs an object oriented policy using "view objects", members of graphical objects, which dictate how the objects will be rendered dependent on their context. Object orientation allows the scenegraph to hold and manipulate any type of data object. This could even include audio files if they have an associated graphical representation.

### 2.4.2 Graphical Representation

The Data layer overlaps somewhat with the Interaction layer as it also handles the basic input–output feedback loop: As the pen is dragged across the screen, the digital ink appears underneath the pen tip. Consequently the Data layer includes rendering algorithms, which convert the coordinate pen data into a pixel-based graphical representation.

SATIN renders at low quality while drawing strokes, so that the feedback to pen movement is as quick as possible. After strokes are finished they are re-rendered at a higher resolution. Apart from this there is no other information in the public domain concerned explicitly with the application of ink rendering to pen-based computers or Informal Interfaces. There is however plenty of work concerning the modeling of writing materials in the general domain of computer graphics [34, 66]. Although these models may be overly complex for simple sketching and note-taking applications, the principles they explore may be applied at the Data layer to simulate familiar materials.

The Data layer could also include the graphical implementations of some visualization tools and interaction widgets such as: zooming [1], fish eye views [22], or marking menus [37]. Although some of these, particularly the marking menus, may be more appropriately situated in the Interaction layer.

### 2.4.3 File Format

The Data layer only holds the basic data objects, meta-data describing higher level structures is contained in the Structural layer. Consequently, when ink files are saved there will have to be some mechanism for accessing and encoding this information.

Surprisingly, there is no mention of file format in the work on SATIN, however work is progressing in the public domain through the World Wide Web Consortium's work on an Ink Markup Language[2]. Being an XML based format, extensions to add structural information and non-ink objects should be fairly trivial to implement.

## 2.5 Structural Layer

The Structural layer is central to an Informal Interface, with software interfaces to the other three layers. The purpose of the Structural layer is to apply different

[2]http://www.w3.org/TR/InkML/

structure recognition algorithms to the abstract data stored at the Data layer. It implements algorithms that reveal structure in a priori unstructured digital ink. At a basic level this could include algorithms: to segment ink data into script and diagrams; to group ink-strokes into words; and to recognize basic diagram components such as boxes and arrows. At a more advanced level it could include grammars that describe the logical behaviour of these basic objects. These grammars may be general or domain specific.

Visual grammars specify the set of valid spatial arrangements for visual languages. A visual language is a system of interpretation under which basic graphical objects have a particular meaning. For instance, a 'box' figure will have a different meaning in a UML[3] diagram than it does in an architectural floor plan.

In an application supporting visual languages, basic objects are analysed using a spatial parser which identifies the geometric constraints and relationships between the basic objects. The valid relationships are specified in the language's grammar. Research into the use of visual grammars is fairly mature [11, 65].

Visual grammars can be used to support three important behaviors of an Informal Interface. Firstly, different grammars can be used to specify and evaluate the interpretation of freeform digital ink [24]. Secondly, the application of a grammar over digital ink can allow a user to manipulate the ink in a meaningful way [36]. Finally, the use of grammars can facilitate the translation and export of freeform digital ink in to other applications [24].

The Structural layer does not force any particular interpretation or structure on the ink data, but it does provide all possible alternatives through a consistent software interface to the Interaction and Translation layers. In this way, the layer facilitates ambiguity in the interface.

Most work seems to concentrate on clustering strokes representing either handwritten script [10, 36], or diagrams [24, 62] into meaningful entities such as words or glyphs (eg. Figure 2.3), although some work does approach clustering at a general level [45, 50]. As well as stroke clustering algorithms, the Structural layer also implements grammatical constraints, such as the re-flowing of words in a paragraph after word deletion or insertion. To support basic functionality, grammars describing the behaviour of simple notes [36] and diagrams [61] will be im-

[3]Unified Modeling Language

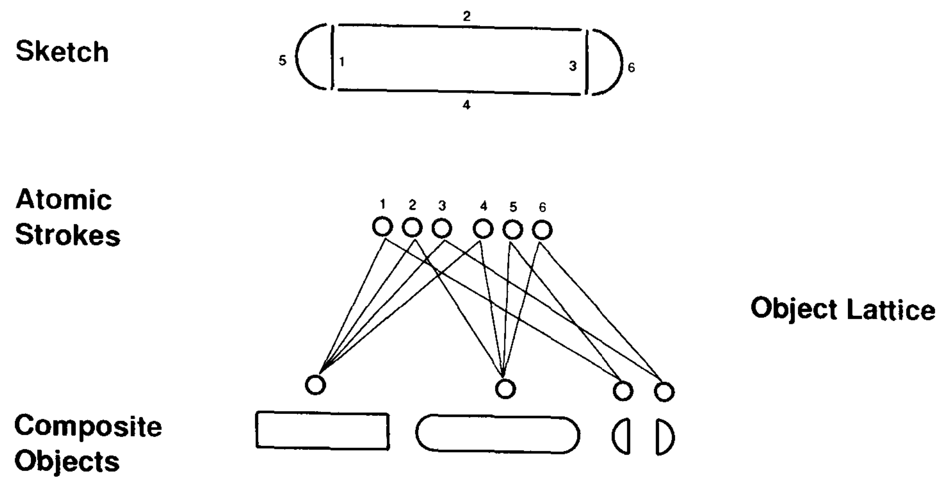



Figure 2.3: An Object Lattice from Saund and Moran [62]

plemented. This layer may be extend with grammars for tasks in specific domains as demonstrated by Gross and Do [24].

If different algorithms are used for script and diagrams, then one can either create algorithms to identify each type [40, 61] or employ a strategy that generates groupings with a "confidence" rating [62]. The former option seems less resource intensive, but the later lends itself more to the multiple interpretation, ambiguity, concept. A third approach is to have the user explicitly select a domain for the interpretation [49], however this would be a choice made at the Interaction layer.

## 2.5.1 Diagrams

The consensus among workers in the area of creating tools that support structures perceived by the human visual system [23, 24, 61, 62] is that to support multiple alternative perceptions, structures should be represented in an "Object Lattice". An Object Lattice contains information on the composition of higher level objects based on the membership of the basic strokes. An illustration of this concept from Saund and Moran's work [62], can be seen in Figure 2.3.

Saund and Moran [62] state that "substantial power derives from a rather modest set of rules underlying the grouping procedures". They state that these rules can be found in computer vision literature, and specifically implement rules to identify closure, parallelism, corners, and T-junctions. The results of these groupings on the basic strokes are then stored in the object lattice. This is echoed again in later work [61] which describes using many different approaches including matching stroke groups to domain-specific databases of shapes, relations, and semantic interpretations.

Gross and Do [23, 24] take a slightly different approach, and use a low-level recognition algorithm to classify strokes into types such as line, box, or circle. These basic components are then parsed by higher level, user programmable, algorithms which recognize the context and configuration of the classified strokes. The structures these algorithms generate can be general or domain specific. Their system is also capable of maintaining multiple interpretations until ambiguity is resolved.

### 2.5.2 Script

There seem to be two main approaches to grouping handwritten strokes into words: selection driven methods [15, 46] and clustering techniques [10, 36, 55]. Selection driven methods work by grouping ink strokes indicated by a selection gesture. This could either be those contained in an area [46] or those that intersect a line [15]. Although these may be effective methods of selection for individual words or paragraphs, they do not allow the system to build a perceptual structure from the ink strokes. Clustering techniques group strokes together if the distance between them is under a certain threshold.

Stroke clustering is not a new research area, but has been addressed by handwriting-recognition researchers [52, 70]. Plamondon and Srihari [52] state that "prior to any recognition, the acquired data is generally preprocessed to reduce spurious noise, to normalize the various aspects of the trace, and to segment the signal into meaningful units". This may include the extraction of lines [55] or words [70]. Tappert et al. [70] point to a number of papers [18, 32, 41] that describe methods that employ measures of time and space between strokes to group them into "words". More contemporary work [10, 36, 61] uses hierarchical agglomerative clustering, again using temporal and spatial components to calculate the distances between strokes.

### 2.5.3 Further Domains

As well as segmenting data and detecting perceptual structures, work at this layer also addresses the use of domain specific recognizers [24] and grammars [36]. This will allow Informal Interfaces to work in any domain. Work has been already been published demonstrating structure recognition in the domains of mathematical formulas [17, 30], computer aided software engineering [12], and web design [37].

## 2.6 Interaction Layer

The Interaction layer provides the interface to the user. Its functionality falls broadly into two different areas: interface components and interaction mediation. The interface components may be pen-specific or general components. These may include features like pen-gesture recognizers [59] or visualization techniques, such as page zooming [1]. Interaction mediation is the process of interpreting interaction, and has been demonstrated in different ways in the domain of diagramming [24, 42, 62].

### 2.6.1 Widgets and Visualization Components

Research published on interaction techniques with pen-based computers can be divided into papers that present research on particular techniques, and those that demonstrate the application of such techniques in some sort of technology demonstrator.

Research into pen-specific interaction techniques started around the late 80s and early 90s. The earliest work [21, 81] looked at the use of hand-markings, with a physical pen and paper, to indicate basic editing commands for use with text and diagrams. This was followed with work investigating the use of pen-based editing commands such as selection gestures and pie-menus [31], where users can select items from a radial menu by drawing the pen at the angle of the option to select. The authors of these investigations generally went on to demonstrate possible applications of their findings [25, 82].

As already mentioned, a popular way of interacting with pen-based computers is through the use of gestures. Gestures are interface commands specified by a single pen stroke. The benefit of using gestures is consistently cited as the ability to specify an operation and its parameters in a single easy-to-remember gesture. The seminal paper in this field is Rubine's work on *"Specifying Gestures by Example"* [59]. Rubine's recognizer has been implemented in many projects. It is free, easy to implement, and only requires a modest set of training examples.

More recent work has looked specifically at the design of gestures. In their first paper, Long et al. [39] address the computer recognition of gestures and suggest improvements that should be made to gesture design tools. Their primary conclusion is that gesture designers generally do not understand the subtleties of recognition algorithms, and that design tools should give active feedback to the designer on exactly where and why recognition difficulties arise along with explanations of how such ambiguities may be resolved.

In their second paper [38], they address human perception of gestures and by experimentation propose a computer model which predicts perceived gesture similarity. Such a model, they conclude, can help a gesture designer both identify gestures that users will find similar and may confuse, as well as assist in creating gestures that users perceive as being related, for accessing related functions.

Related to gesture recognition is the study of scribble matching. Scribble matching is the process of searching a document of digital ink for matching "scribbles" without the need to perform any character recognition on the data. Each stroke, or cluster of strokes, is characterized and compared with every other. This can be used for searching digital ink [19, 53] and creating indexes of digital ink documents [76].

There are no specific visualization techniques for pen-interfaces, however different interface demonstrators have implemented various techniques. These have included a zooming interface [2] in the SATIN toolkit [27], and the ability to move and squash segments in the Flatland whiteboard project [49]. Techniques that address working with large documents enabling a user to see detail in areas of interest, while still providing an overall awareness of the whole document [58, 60] also seem relevant to pen-based systems.

Other interface demonstrators have implemented pen-specific interaction techniques, such as: Cutting and pasting ink annotations [15]; abstracting annotations and their context to enhance document browsing [63]; and collapsing handwritten notes within a hierarchical structure [36].

### 2.6.2 Interaction Mediation

Mankoff et al. [42, 43, 44] have demonstrated a toolkit for interaction mediation or as they call it, ambiguity resolution. They break interaction mediation into three groups: *Recognition* mediation, when the identity of a command is ambiguous; *target* mediation, when location of a command is ambiguous; and *segmentation* mediation, when the grouping of inputs is ambiguous. They state that ambiguities can be resolved through: *Repetition* of input; *selection* from an n-best list of alternatives; or *automatically* through additional information such as the application of thresholds, rules, or historical statistics. Their work demonstrates mediation for the three different classes of ambiguity, primarily through selections from n-best lists. Their work provides us with a matrix of three ambiguity types and three resolution strategies. In an Informal Interface we prefer *automatic* mediation, since one of the central tenets of the paradigm is not to distract the user from expression.

Gross and Do [24], although not explicitly mediating interaction, do demonstrate automatic mediation. Their system attempts to recognize every input ink stroke. Each stroke, or glyph, has a certainty associated with its recognized identity. By analyzing the best recognized glyphs, both domains and configurations can be proposed, again with certainties. This contextual information can then be used to mediate the recognition of ambiguous glyphs. The process is applied repeatedly trying to map a diagram to different domains. In this, we can say that Gross and Do demonstrate the use of context to perform *automatic* mediation of *segmentation* and *recognition*.

Contextual information is obviously very important in automatic mediation. Gross and Do's system is programmed with an extensible hierarchy of domains and configurations. Other systems allow users to define specific areas as conforming to a certain domain [49]. The domain, or context, can only be changed by a specific command from the user. However, domain information is not sufficient. The Pegasus system [28], in the domain of geometric design, demonstrates *selection* to perform *target* mediation, i.e. finding the location and attitude of hand-draw "straight" lines and replacing them by perfectly drawn ones. Yet another approach is employed by Saund and Moran [62]. They use *selection* strokes to perform ink *segmentation*, before applying a command.

## 2.7 Translation Layer

The Translation layer operates alongside the Interaction layer. It allows information to be imported from, and exported to, other applications. This could be done either by file import/export or some sort of 'live' interprocess communication. The Translation layer facilitates a number of actions: Editing a formal document after importing it [61]; transitioning a document between an informal and formal definition by exporting it [52]; and providing an Informal Interface to a formal application through interprocess communication [24].

Every application or data format for which translation is required will require its own filter to import and to export through. If it is possible to communicate interactively with a third party application, then it may be possible to modify these filters to function interactively.

### 2.7.1 Current Work

Work in this area is particularly sparse. Saund et al. [61] describe a system that works with document images. In their system, bitmap images of documents are imported, then computer vision algorithms generate some sort of structure from the data which the user can interact with. Moran et al. [47] take this concept a step further and allow objects to be imported along with a definition of their behaviour within their Informal Interface.

As far as exporting information is concerned, both primitive diagram components [23] and handwritten script [52] can be recognized, though correction may be required. From these basic elements, higher level, domain specific, structures can be formed [24].

## 2.8 Summary

In this chapter, we have introduced and explored the concept of Informal Interaction. The ideas embodied in Informal Interaction are taken from and have inspired research into pen-based interfaces. Despite this, Informal Interaction has remained a very loosely defined concept, incorporating ideas from a number of different existing research areas.

We have taken the concept of Informal Interaction and sought a comprehensive and precise definition. We have formed and defined a structural framework of an Informal Interface, based on the underlying values of Informal Interaction and the literature available on pen-based interfaces. We then used this framework to review the literature associated Informal Interaction is a cohesive fashion.

The next chapter takes the application of this framework a step further, in using it to isolate the research topics which have been glossed over or have yet to be addressed. In this way we seek to identify the "principal barriers to an intuitive interface to draft and edit documents on a pen-based computer".

# Chapter 3

# Analysis of Potential Work Areas

This chapter identifies a number of different issues to be overcome in the implementation of an Informal Interface. It concludes by identifying those that represent the "principal barriers" to implementation.

## 3.1 Introduction

The aim of this thesis is to show how the principal barriers to an intuitive interface to draft and edit documents on a pen-based computer can be overcome.

In the previous chapter we put forward a conceptual model of an intuitive interface. Published literature on pen-based interfaces was then reviewed in the context of our model.

In this chapter we return to our conceptual model and use it to isolate the "principal barriers" to implementing the interface. Once the principal barriers have been isolated, it will then be possible to show how they may be addressed.

The approach taken to isolate the principal barriers is to take each component layer in turn and identify research areas that may require further investigation. Finally, these findings are analyzed to determine those areas that represent the "principal barriers". The four component layers of our Informal Interface are shown again for reference in Figure 3.1 towards the end of this chapter.

## 3.2 Data Layer

The Data layer encapsulates the digital ink data type and includes the implementation of: an abstract data structure, graphical representations, and a file format. For the most part, work at this level is just a case of implementing and integrating existing algorithms and techniques.

Work concerning the abstract data structure appears to be fairly mature. Moran et al. [47] have published work on incorporating "domain objects" into pen-based interfaces. Hong and Landay have demonstrated the handling of basic ink and other graphical objects with their SATIN toolkit [27].

Work concerning graphical representations however, seem to have received less attention. Although there is work published on ink rendering in artistic domains, there is none concerning digital ink in pen-based interfaces. Here topics such as legibility start to become an issue as users will now want to *read* their own handwriting instead of getting the computer to recognize it. Legibility problems with handwritten notes on PDAs have already been reported [13, 14], slowing reading and writing times. This issue should not be overlooked.

Further work could also continue on topics concerning the digital ink file format, including defining extensions to InkML[1] to encode non-ink objects and structural information.

## 3.3 Structural Layer

The purpose of the Structural layer is to apply different structural algorithms to the abstract data stored at the Data layer. This includes algorithms to identify possible structures as well as algorithms to govern the behaviour of those structures.

Work concerning the identification of structures is split into work on diagrams and handwritten script. Work with diagrams appears to be fairly well developed, particularly through the work of Gross and Do [24]. Work with handwritten script appears to be far more limited.

The area of stroke clustering, to combine pen-strokes into word groups, seems to have been overshadowed by research into handwriting character recognition.

[1] http://www.w3.org/TR/InkML/

Handwriting recognition does require this functionality, but can also use the recognition process to feed back and inform clustering process. This strategy is not perfect as it would never be possible to have a complete language vocabulary stored, or be language independant. There appears to be a requirement for geometric solutions to the clustering problem as has already been evidenced in the literature reviewed. Chiu and Wilcox [10] have demonstrated just such an algorithm. It seems, from the published work, to be adequate for manipulating "blocks" of handwritten script, but their is no indication on its accuracy in identifying individual words.

Perfect groupings may not be required, since interaction and selection techniques can improve and correct classification. Potential future work should look both at the accuracy of geometric algorithms, as well as the practical implications of limited clustering accuracy on the ease of completing editing tasks.

As far as work concerning grammars to describe the behaviour of ink-stroke structures is concerned, Li et al. [36], building on the work of Chiu and Wilcox, have demonstrated a grammar to facilitate interaction with handwritten notes. The definition and use of grammars to facilitate interaction in different domains could also be investigated.

## 3.4 Interaction Layer

The Interaction layer provides the interface to the user. Its functionality falls broadly into two different areas: interface components and interaction mediation. "Interface components" describe visualization techniques and interface interaction methods such as pie-menus or pen-gestures. "Interaction mediation" describes the techniques used to resolve the intended meaning of ambiguous commands. The latter is particularly pertinent to the implementation of "modeless" interfaces where there is no explicit specification of when input is to be considered as data or a command.

Work concerning interface components will no doubt continue as metaphors are developed and pen-based interfaces are used in different domains. The development of such components does not appear to be a pressing need.

The initial potential problem for a modeless system, will be to differentiate between data strokes and interaction (command) strokes. Determining the mode of interaction. Gross and Do [24] solve this problem by recognizing every stroke as a 'glyph'. A glyph may be ambiguous, or a member of an 'interaction' set, or an 'element' set. The two sets do not intersect. While delivering a modeless interface, this solution may inhibit freedom of expression.

Although a modeless interface is desirable, there are alternatives. Some interfaces implement a modal, draw–edit, interface [62], and others an automatically progressing modal (draw→edit→draw) interface [28].

Research in the area of interaction mediation seemed fairly disparate until work was published on a taxonomy of mediation by Mankoff et al. [44]. Their work provides a foundation on which to investigate interaction mediation further. The shortcoming of their classification is not to include *mode* mediation, which would probably be required for a modeless interface.

Reliable automatic mediation is the ideal solution to all types of ambiguity during experience capture, as this would not distract users from their primary task. Automatic mediation on its own will not be sufficient, even if perfect mediation were possible users may change their minds. Methods to explicitly correct classifications will be needed from time to time. This would be through repetition or selection.

Of particular interest is context mediation (segmentation mediation according to Gross and Do), as multiple interpretations of context (different possible groupings of ink, or segmentation of the page) may be possible. For instance if a command is issued on a group of ink which may represent both handwritten script or a diagram, then if the command is recognized as being specific to either one type or the other, then the correct context (script or diagram) can be selected. Once the context is known, the correct grouping can be set, and the command executed on the correct perceptual entity. If this is not the case, as it may not be in some situations, then other solutions will have to be investigated and employed.

## 3.5 Translation Layer

The Translation layer allows information to be imported from, and exported to, other applications. This could be done either by file import/export or some sort of

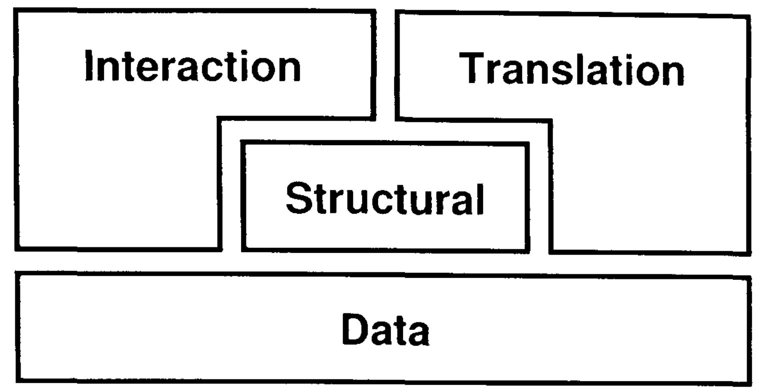



Figure 3.1: Informal Interface Component Layers

'live' interprocess communication.

Work in this layer could be progressed by developing an extensible translation framework, so that new data formats can be added as required. For importing information, the framework will have to have a consistent way of breaking up a document into its basic parts to be represented in the Data layer, and of storing their initial structural relationships in the Structural layer. For exporting information, the framework will have to recognize and encode basic objects, along with the correct structural information for the desired output. This could initially be piloted using an SVG[2] file which would be able to contain recognized text and diagram components, as well as being able to store unrecognized objects as raster graphics or stroke paths. An export filter should also be able to employ recognition mediation techniques, and have some method to specify the interpretation required.

Translation functionality could be built into an Informal Interface, or implemented independently if translation is not required within a particular Informal Interface application.

## 3.6 Analysis

We have taken each component layer in our conceptual model of an Informal Interface (see Figure 3.1) and identified topics for future research. These are as

[2]http://www.w3.org/TR/SVG/

follows:

**At the Data layer** we identified work to do concerning both a digital file format for Informal Interfaces, and work concerning the graphical representation of digital ink on-screen, particularly with reference to legibility. While work is already underway with digital ink file formats, nothing has been found concerning digital ink rendering and legibility.

**At the Structural layer** we identified that work detailing the clustering of ink-strokes into handwritten words was rather vague concerning the accuracy of, and constants used in, the algorithm presented. We also identified the study of ink grammars as a potential topic for more exploration.

**At the Interaction layer** we identified work concerning interaction mediation. In particular the automatic mediation of mode and context, as well as methods to interact with this mediation process when required are all topics that can be researched further.

**At the Translation layer** we identified the need to investigate many issues concerning import and export processes. This included the capability of preserving the structural information contained within different files and investigating techniques to allow new file types to be integrated easily.

The nature of our conceptual model indicated that functionality of the upper layers rests on the lower layers. The most pressing and important shortcomings would appear to be at the foundational layers: the Data layer and Structural layer. When the issues at these layers have been addressed it will be possible to build systems to investigate the issues at the higher layers.

At the Data layer, the most pressing issue would seem to be that of digital ink legibility. If an Informal Interface is uncomfortable to use because of poor ink legibility, then users may choose not to use it at all.

At the Structural layer it seems wise to investigate the efficacy of stroke clustering algorithms. The investigation of different structural grammars may be done later as Informal Interfaces are applied to different domains.

The principal barriers to an intuitive interface to draft and edit documents on a pen-based computer are then likely to be the legibility of handwritten digital ink, and the quality of clustering pen-strokes into handwritten words without employing character recognition. After addressing these issues, barriers at the Interaction and Translation layers can be fully investigated.

The remainder of this thesis investigates these two issues.

# Chapter 4

# Digital Ink Legibility

This chapter we explore the topic of digital ink legibility, undertaking two experiments to investigate the effect of rendering resolution.

In the previous chapter, we employed our conceptual model of an Informal Interface to identify two principal barriers to implementing an intuitive interface to draft and edit documents on a pen-based computer. The first of these barriers concerned the legibility of digital ink. In this chapter we explore the issue of digital ink legibility. We pilot two methods to measure legibility, examining the effect that rendering resolution has on digital ink. We establish that digital ink legibility need not present a barrier to implementing an informal interface. We also examine whether or not the topic of digital ink legibility remains consistently within our structural model.

To do this, we first review the problem digital ink legibility presents, then establish our hypotheses. We look at related work concerned with the measurement of legibility and present the different rendering algorithms we will examine. Experiments addressing our hypotheses are then presented and discussed. Finally we draw conclusions from our work. Parts of this work have already been published [8, 9].

## 4.1 Introduction

We found that a number of recent research projects [13, 37, 63, 80] argue that preserving handwritten input is, in certain situations, more preferable to recognition. They have all demonstrated Informal Interface applications which highlight this. As pen-based computers, such as Tablet PCs, become more widespread we may see similar applications become mainstream products.

The only project that did undertake an extensive user trial of an application using handwritten script [13] reported problems with legibility. This was discovered to slow the reading and writing process. It could be argued that the project used devices with a low screen resolution–which caused poor legibility–and that devices with higher resolution screens will not suffer from this problem. This does not determine however, what an acceptable resolution might be.

We would like to know if current hardware and software technology is sufficient to display handwriting on a computer in a way that does not cause undue stress or discomfort to the user as they read or write. As we will see later, improving legibility should decrease the mental stress and fatigue in reading [73].

## 4.2 Hypotheses

As we cannot increase the hardware resolutions of commercially available screens, we have decided to look at software techniques, such as antialiasing, to enhance the perceived screen resolution.

Specifically, We have chosen to examine the effects of horizontal resolution enhancement on legibility. When reading handwriting, humans rely on identifying: vertical down-strokes; crossings; and points of high curvature, particularly at the beginning and end of each word [64]. We chose horizontal resolution enhancement, since this will improve the definition of predominantly vertical components. We have implemented a pen-trace rendering algorithm, ClearPen, which exploits sub-pixel addressing on LCD displays. This gives a perceived three-fold resolution enhancement in the direction of the scan-lines of the LCD display.

By measuring the legibility of handwritten script on a computer screen at different perceived resolutions, we will be able to see what, if any, effect perceived resolution has on legibility.

To examine the effect of horizontal resolution enhancement on legibility we have two hypotheses which are addressed in two separate experiments. These are:

1. increasing the perceived horizontal rendering resolution of a handwritten word, displayed on a computer screen, enhances its legibility.

2. A user will perceive the effect of horizontal resolution enhancement as beneficial, and will prefer reading script rendered in this manner.

## 4.3 Related Work

We need to measure the legibility of handwritten script. This has traditionally only been done in educational fields, by comparison against a set of graded samples. We however require measurements that are more objective and more descriptive.

### 4.3.1 Legibility

Legibility is the term which describes the effect of the spatial aspects of a text on its readability. Legibility is affected by a number of different graphical properties. These include but are not limited to:

1. Letter shape and word form;

2. Spacing between letters, words, and lines;

3. Line length and letter size;

4. Contrast of words against a page.

Tinker, in his 1964 book, *The Legibility of Print* [73] states that:

> "Optimal legibility of print, therefore, is achieved by a typographical arrangement in which shapes of letters and other symbols, characteristic word forms, and all other typographical factors such as type size, line width, leading, etc., are coordinated to produce comfortable vision and easy and rapid reading with comprehension."

In essence, improving the legibility of a text will decrease the strain and fatigue of a reader, as well as facilitate efficient and accurate reading.

### 4.3.2 Measuring Legibility

There are a number of approaches to measuring legibility. These include:

1. Speed of reading a passage [4];
2. Speed of a search task [48];
3. Word recognition rate [72, 83];
4. Oculomotor measurements of eye fatigue [75];
5. Subjective preferences [4, 75, 83].

The use of speed of reading measures, although preferred for measuring the legibility of print [73], has not established significant differences in the legibility of typefaces on a computer screen [4, 75].

We investigated the measurement of legibility through the speed of reading ourselves by carrying out an experiment in a similar style to Boyarski et al. [4]. We employed a published speed-of-reading test which involved volunteers marking a target word (a word out of context) in each paragraph they read. The speed of reading depends on comprehension as well as time taken, since if a sentence is not understood there is no guarantee that it was read. Volunteers were given 7 minutes to read as many paragraphs as they could, with their speed of reading calculated as the number of paragraphs attempted less the number of incorrect paragraphs.

Our experiment found no difference in the speed of reading between different rendering resolutions. There was however a difference in the recognition rate, the proportion of correctly identified target words. We suppose this to be due to the fact that, unlike printed words, isolated handwritten words are not 100% legible, and that context aids recognition. Words that are out of context are therefore very difficult to identify with certainty. This leads us to focus on word recognition as suitable measure of legibility for handwritten words.

Measures that have been employed successfully to measure legibility include: the speed of a search task; the word recognition rate; and subjective preferences. To the best of our knowledge, the only experiment that has employed a search task as a legibility measure, required volunteers to locate random 4-letter strings among 100 similar distractors [48]. We decided that this method was unsuitable for our needs since trying to read arbitrary combinations of letters written by hand is a very uncommon activity. This would likely depend on different types of visual perception than normal reading.

Established research shows that humans perceive word forms more readily than they do individual letters or non-words [54, 56]. Coupled with the fact that letters within handwritten words are often joined as part of a single unit, and that individual words are free from any contextual clues as to their identity, word recognition rate would appear to be a good indicator of the legibility of handwritten script.

### Word Recognition

Word recognition has been used successfully to measure differences in legibility, including measurement of the effect of resolution on the legibility of typed text [83]. Word recognition is commonly measured after displaying a word for a few hundred milliseconds [72], although there is some evidence that differences in word recognition can be measured without the use of such short exposures [54, 83].

Tinker [73] notes that the short exposure method is particularly useful for measuring the relative legibility of different printed symbols. Since we have established: that recognition rate shows differences in legibility; that handwritten words are read as units; and that the method has been used to measure the legibility of printed material on screen of letters, words, and headlines; we are confident that measuring recognition rate through a short exposure method is a good indicator of the legibility of handwritten script displayed on a computer screen.

To determine an ideal exposure time we carried out a small experiment which measured the recognition rate of individual handwritten words shown for different short durations. Six volunteers each saw 90 common words shown for dura-

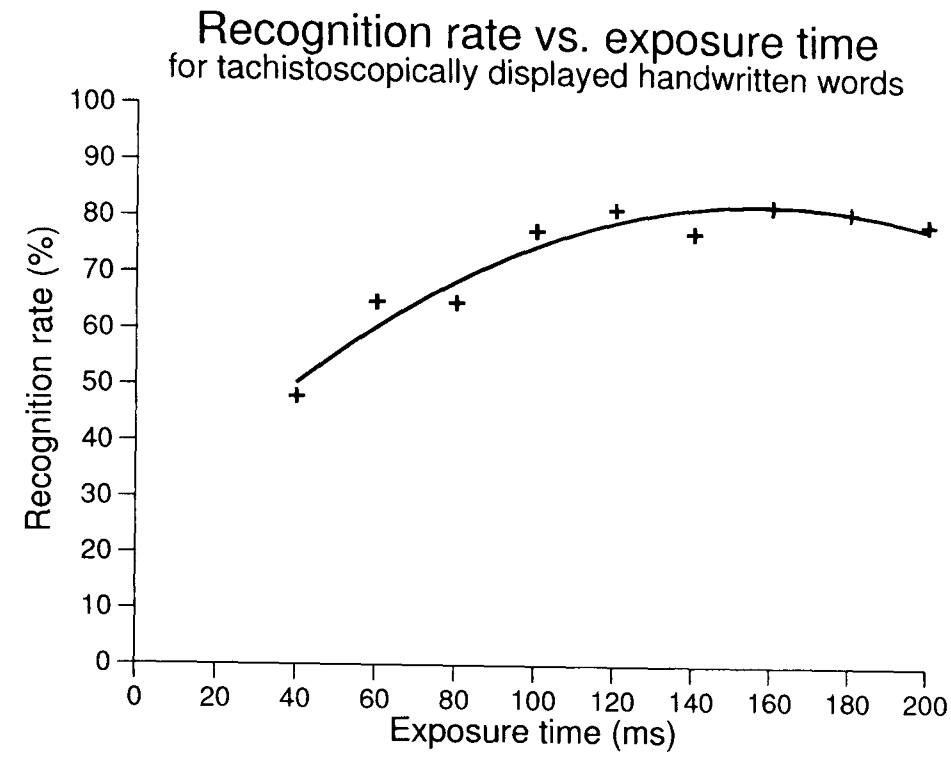



Figure 4.1: Recognition Rate against Exposure Time

tions between 40 and 200 milliseconds. The results, shown in Figure 4.1, reveal a recognition rate of handwritten words without any contextual information of about 80%. This recognition rate starts to deteriorate at an exposure time of 100–120ms. This is in agreement with Timmers et al. [72] who state that a stimulus presentation time of 100ms, being shorter than the visual reaction time, allows one single fixation and precludes any systematic influence on the experimental results.

### Subjective Preferences

Subjective preferences, where volunteers answer questionnaires [4, 75, 83] or rank display modes in order of preference [75] also seem to consistently reveal differences in reader preference when comparing different modes of display or fonts.

Subjective preference questionnaires have employed Likert scales with either 5 or 9 points. Not all questions established significant differences. Experimenters have focussed on aspects of the clarity of text [4], and on indicators of physical strain on the reader [75]. Experiments have used, on average, around 20 volunteers.

Particular care must be taken when using subjective measures. Firstly responses to questions will depend on the volunteers' interpretation of the meaning. If certain questions are too ambiguous, volunteers may respond in reference to different factors. Secondly, care should be taken when drawing conclusions from small numbers of volunteers, particularly if they are all drawn from similar backgrounds.

Overall, measurements of subjective preferences complement objective measures, but on their own the results should be treated with caution.

## 4.4 Rendering Algorithms

In our experiments, handwritten data was rendered on-screen using three different rendering methods: ClearPen (high resolution); antialiased pen (medium resolution); and pixel pen (low resolution).

When rendering a pen trace the handwritten data samples, collected using a pen digitizer, are first processed through a pen model. This models the volume of ink flowing from the pen nib to the page. The model produces line segments, describing the path and intensity of the pen trace. Each rendering method takes line segments and renders them on-screen.

### 4.4.1 Pen Model

Our pen model algorithm is an observational model, similar to the work of Sousa and Buchanan [66], except that we are modeling a fountain pen alone rather than pencil, paper, and other artistic materials.

The pen model algorithm operates by modeling the volume of ink flowing from the pen nib to the page. The ink volume is represented by an "intensity" value. The more ink, the higher the intensity of the pen trace.

The volume of ink deposited on the page depends both on the speed the pen is moving at (Figure 4.2), and the pressure applied to the tip (Figure 4.3). The pen model generates a series of consecutive line segments describing the path of each pen stroke. Each line segment consists of two triplets detailing the x-coordinate, y-coordinate, and an intensity value, at either end of the segment.

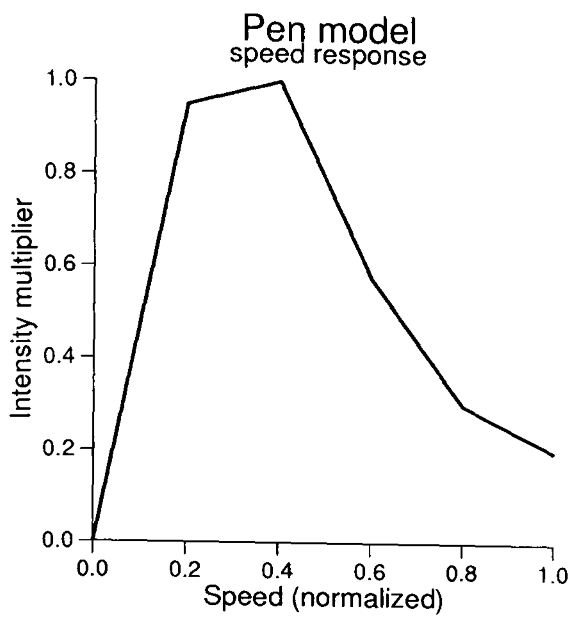



Figure 4.2: Pen Model: Speed Response

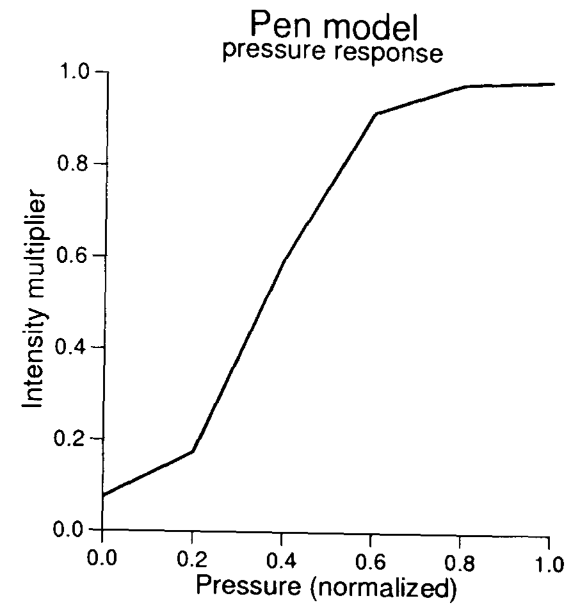



Figure 4.3: Pen Model: Pressure Response

### 4.4.2 Pixel Pen

The pixel pen simply converts the line segment coordinates to screen coordinates using a nearest neighbor division. A one-pixel-wide line is drawn between the two end points. The colour of each pixel along the line is determined by interpolating the intensity values along the line segment, and converting them to a colour. The more intense the pixel, the darker the pixel colour.

### 4.4.3 Antialiased Pen

The antialiased pen and ClearPen algorithms are more complicated, but similar to each other. Instead of being rendered directly onto the screen, each line segment is rendered onto a grid at nine times the display resolution. As in the case of the pixel pen, the intensity values are interpolated along the line segment.

Instead of filling a one-point-wide line with intensity values, each grid point becomes the centre of a "tip filter". The tip filter is a two-dimensional filter representing a hemi-ellipsoidal pen tip. At each point along the line the intensity value is dissipated over the area covered by the filter (Figure 4.4).

The shape used for the tip filter is that of a hemi-ellipsoid. An ellipsoid is a quadratic surface given by the formula:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

Where $a$, $b$, and $c$ are the $x$, $y$, and $z$ dimensions of the surface. The volume of an ellipsoid, $V$, is given by the formula:

$$V = \frac{4}{3}\pi abc$$

In our tip filter, $a = b = 6$, and $c = 1$. The dissipation value $z_{(x,y)}$ is normalized by half the ellipsoid volume, so that the hemi-ellipsoid has a total volume of 1.

The filtered line segment is then mapped onto the "intensity grid". The intensity grid is three times the display resolution. It is populated by summing each square of nine intensity values from the first grid into the corresponding cell of the intensity grid.

The process of translating the values stored in the intensity grid to screen pixels is where the ClearPen and antialiasing algorithms differ. The antialiasing algorithm averages each square of nine intensity values, and converts them to a shade of grey representing their intensity. That colour is then rendered onto the corresponding screen pixel.

### 4.4.4 ClearPen

ClearPen rendering recognizes that on an LCD panel, scan-lines are composed of individually addressable colour component pixels (sub-pixels) in an ordered sequence, usually red–green–blue (Figure 4.5). Each screen pixel is formed from a triplet of adjacent sub-pixels.

The assignment of sub-pixels to pixels is static, however as each sub-pixel is individually addressable, any three adjacent sub-pixels can be combined to give the appearance of a full pixel. This technique allows us to position "perceptual pixels" at three times the normal precision of the LCD display.

In antialiased rendering, squares of nine intensity values are averaged and converted to shades of grey. In ClearPen rendering, columns of three intensity values

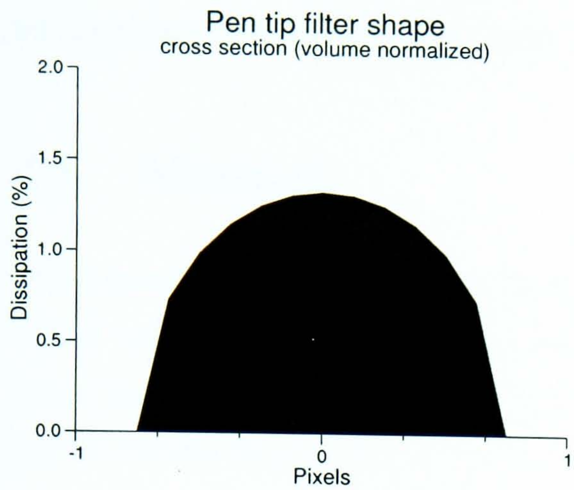



Figure 4.4: Pen Tip Filter Shape

Figure 4.5: An LCD Panel Scan-Line

Each scan-line on an LCD display is composed of coloured subpixels

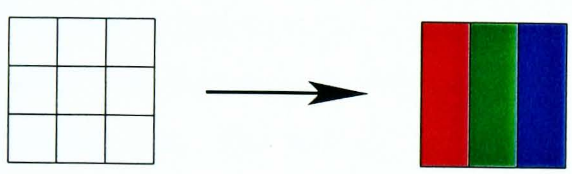


Figure 4.6: Intensity to Pixel Mapping

Each square on the intensity grid is mapped to a colour component sub-pixel. The columns are averaged before being mapped onto the corresponding sub-pixel.

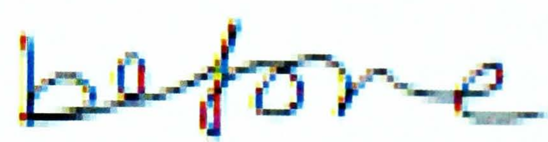


Figure 4.7: ClearPen Rendering

are averaged and converted into sub-pixel colour components (Figure 4.6).

The result of ClearPen rendering can be seen in Figure 4.7. The sub-pixel order is red–green–blue. Of particular interest are the vertical pen strokes. The screen pixels lit on their left-hand side (those on the left side of a stroke, as black is "unlit") appear yellow-brown since the (right-most) blue component is not lit. Likewise, those pixels lit on their right-hand side, on the right side of each vertical stroke, appear turquoise-blue since the (left-most) red component is not lit. When viewed at normal magnification the adjacent colour components combine to form a black pixel in between the actual screen pixels.

The sub-pixel technique is common knowledge, but has not been applied to handwriting before, only type fonts[1].

# 4.5 Method

The experiments commenced with the collection of samples of handwritten script. These were rendered using the methods described previously. The experiment room and testing software were prepared, volunteers recruited, and experiments run to test the hypotheses.

## 4.5.1 Handwritten Material

To reduce variability in script style, handwriting samples were collected from a single writer. The samples were copied by the writer from typed sheets. The writer was instructed to write with the knowledge that other people would have to read what they had written. Every sample was written on paper using a Wacom Intuos Inking Pen. The writing paper was attached to a Wacom Intuos digitizer tablet which sampled the pen movement at around 94Hz. The information sampled includes:

- Movement data, at 100 points per millimetre.
- Pressure data, at 1024 levels.
- A time stamp.
- Pen tilt data.

These data were logged in files and were then rendered to produce the required images of the handwriting on-screen.

### Experiment 1

For the first experiment, measuring the recognition rate of individual handwritten words using a short exposure, 199 words were collected. These are the most

[1] http://grc.com/cleartype.htm

Figure 4.8: "across" Rendered with Five Different Pixel Widths

common six and seven letter words in the English language as listed in the LOB corpus [26].

To simulate different rendering resolutions, our volunteers read from a distance around 3 times the usual reading distance from the monitor. Word images were created using pixels from 1 to 3 times the real screen pixel size. The coarsest resolution (3 times the normal screen pixel size) then represented the normal screen resolution, and the other resolutions represented enhancements.

To create equivalent images of identical words at different horizontal resolutions, the pen data coordinates for each word were mathematically scaled to $\frac{1}{1}$, $\frac{1}{1.5}$, $\frac{1}{2}$, $\frac{1}{2.5}$, and $\frac{1}{3}$ of their original horizontal dimensions. These data were then rendered into word images using the antialiased rendering method. An antialiased rendering method was used since using the ClearPen algorithm in this fashion would have caused unlit pixel components. Finally the word images were graphically scaled back to their original size. This produced words with pixel widths of 0.30, 0.45, 0.60, 0.75, 0.90mm respectively. The result can be seen in Figure 4.8. On average the rendered words are 75 pixels in length and 23 pixels high.

According to Tinker, brightness contrast between print and paper has an effect on legibility. This is confirmed by Timmers et al. [72] who found that, when presented parafoveally (at a visual angle of $\pm 1.5°$), the legibility of words decreased significantly with decreasing contrast. Words presented foveally (0° visual angle) were much less affected by decreasing contrast (Figure 4.9).

Although we presented our words foveally, we took care to preserve the contrast of the word images against the screen background.

The experiment used 100 words, 5 groups of 20 drawn from the 199 samples. Each of the word groups (A–E) consisted of 10 six-letter words and 10 seven-letter words. Within each group, all words were rendered at the same resolution.

There were also 5 volunteer groups (1–5). Each group saw the 100 words in a different random order. The rendering resolution used for the word groups shown

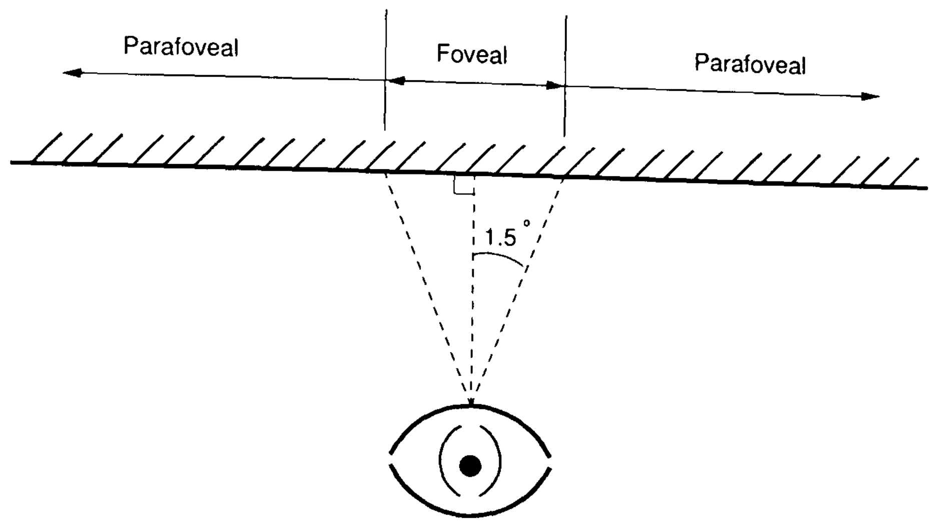



Figure 4.9: Visual Angle

Table 4.1: Pixel Widths (mm) for each Volunteer Group

| | | Word Group | | | | |
|---|---|---|---|---|---|---|
| | | A | B | C | D | E |
| | 1 | 0.30 | 0.45 | 0.60 | 0.75 | 0.90 |
| **Volunteer** | 2 | 0.90 | 0.30 | 0.45 | 0.60 | 0.75 |
| **Group** | 3 | 0.75 | 0.90 | 0.30 | 0.45 | 0.60 |
| | 4 | 0.60 | 0.75 | 0.90 | 0.30 | 0.45 |
| | 5 | 0.45 | 0.60 | 0.75 | 0.90 | 0.30 |

to each of the volunteer groups was varied so that, although each volunteer saw the same 100 words, no group saw the same word *image* as any other group. Over the five volunteer groups all 100 words were displayed in each of the 5 resolutions. This is illustrated in Table 4.1.

**Experiment 2**

The second experiment tested volunteers' preference for reading script rendered with an enhanced horizontal rendering resolution. Three samples of script were rendered using pixelated, antialiased, and ClearPen rendering methods. The script collected consisted of twenty random items selected from the *Tinker Speed of*

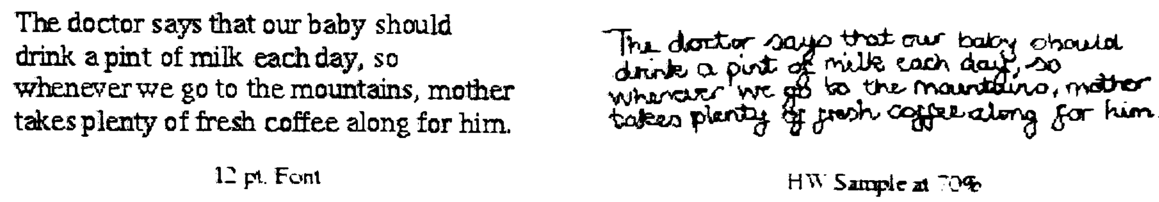



Figure 4.10: Equivalent Typeface and Handwriting Sizes

*Reading Test*[2] [74]. Ten items were collected on each page. Test items were copied into boxes on the paper form to constrain the size of the handwriting, and also keep the items clearly distinct. Three columns of script, each consisting of five Tinker test items were selected for the experiment.

For this second experiment, volunteers were re-allocated into six volunteer groups. Each column of test items was presented to the volunteers as a single page. Volunteers in each group saw the 3 pages in the same order. The rendering method used to render each page was varied between groups, so that each group saw the rendering methods in one of the 6 different orders.

The Tinker Test has been used in previous research on the legibility of text [4]. Although designed to measure speed of reading, we used the test simply to engage our volunteers with the handwritten script so they could answer a questionnaire.

The handwriting was presented at 70% of its original size. This scaling allowed us to render the script at a size comparable to a font size of 12 points (Figure 4.10). This size was chosen as a large number of documents are presented at this point size.

Satisfactory results will indicate that handwriting may be read on pages with the same amount of information as pages containing typeface. Previous work has indicated that "Documents containing 10 point text or larger are quite readable" [63] on a pen tablet with a 1024×768 display resolution.

[2]The Tinker Speed of Reading Test provided courtesy of the University of Minnesota Press: ©1947, 1955 by Miles A. Tinker. All rights reserved. Published by the University of Minnesota Press, Minneapolis.

### 4.5.2 Equipment

The experiments were conducted in a closed room with no natural light to control for illumination. Software was written for both experiments and run on a 750MHz Intel Pentium III PC running Mandrake Linux. A Wacom PL-500, a digitizer tablet with an integrated LCD display, was used as the computer screen. This has a 1024×768 display resolution, and a 0.3mm dot pitch.

In the screening task and first experiment, volunteers were seated so that their heads were approximately 180cm away from the screen. They were asked to keep their back straight against the back of the chair so as to keep the distance between their eyes and the screen constant.

During the second experiment, volunteers were required to mark on the tablet with a pen so were permitted to position the pen tablet however they liked. The majority of volunteers left the tablet in its original position, upright on the desk, at a viewing distance of around 45cm. This was the easiest angle for the volunteers to read the script at, as the illumination of the screen dropped off rapidly as the viewing position moved away from being normal to the screen.

The arrangement of the room during the experiments is shown in Figure 4.11.

### 4.5.3 Volunteers

21 volunteers from the University of Hertfordshire administration staff were used in the experiments. Administration staff were chosen as we assumed that they are more likely to work with other people's handwriting than any other type of staff.

All volunteers were screened for visual acuity. This was done by asking the volunteers to complete a word identification task, using the tachistoscopic display program. Volunteers were asked to wear their glasses or contact lenses if they usually did so.

The experimental set up was identical to that in Experiment 1, described below. Volunteers were asked to identify 10 words, each displayed for 200ms. Volunteers with less than a 60% recognition rate did not participate in the first experiment. As the second experiment was less visually demanding, with only an informal constraint on viewing distance, all volunteers participated in it. The screening process also served as a familiarization task for the first experiment.

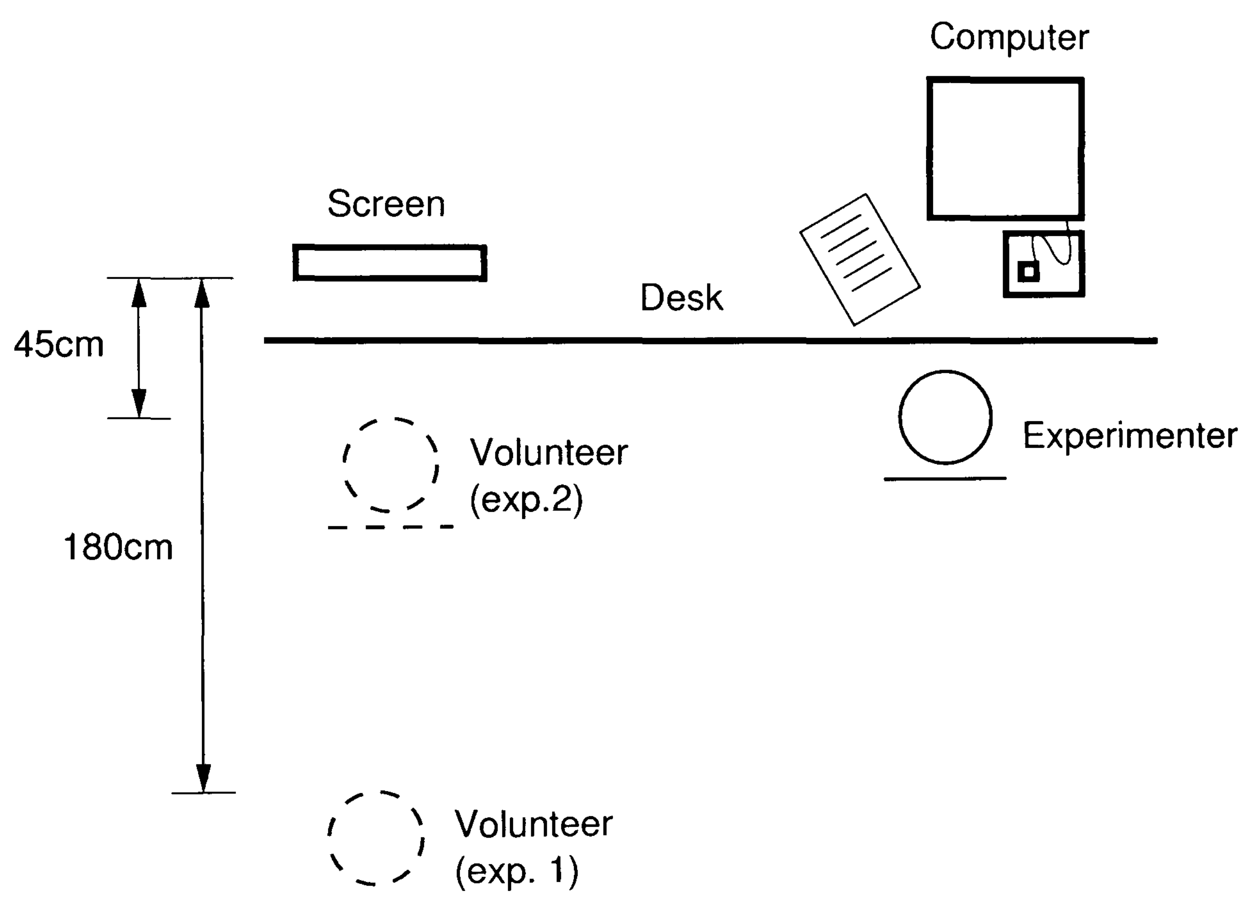



Figure 4.11: Experiment Room Layout

17 volunteers completed the first experiment, 18 the second.

### 4.5.4 Word Recognition

Upon entering the room, volunteers were asked to sit on a chair 180cm across from the computer screen. They were read instructions describing the overall experiment, and the screening process. A copy of all instructions can be found in Appendix C. They were encouraged to ask any questions that they may have had. After successfully completing the screening/familiarization task, volunteers were assigned in turn to one of the five volunteer groups. They were given a keyboard to rest in their lap, and shown the pre/post exposure field in the centre of the screen. They were asked to start when they were ready.

Upon pressing the space-bar on the keyboard the first exposure commenced. After a small random delay between 0.5–1.5s a word was displayed. This technique required the volunteer to focus their attention on the screen, as they could not anticipate exactly when the word would be displayed. The volunteer then spoke the word they thought they had seen. Each volunteer saw 100 words, displayed for 120ms each. Before the volunteer could proceed to the next word, their response was marked by the experimenter using a mouse click. The left-hand button marked a correct response and the right-hand button an incorrect response. Mark systems using 3 buttons to record 'near misses' or 'total refusals' were trialled, but were found to be too difficult to use reliably.

The results of each experiment were logged to a text file recording: the word; the exposure time; the delay before exposure; and the mark recorded by the experimenter. The experiment took around 5 minutes for each volunteer to complete.

### 4.5.5 Subjective Factors

After a short break, each volunteer commenced the second experiment. The volunteer was positioned at the desk directly in front of the screen. The volunteers were given a second set of instructions, explaining the procedure and nature of the experiment, including examples.

After being read the instructions and asking any questions, each volunteer was asked to complete a familiarization task. The familiarization task consisted of

5 Tinker Test items presented on the screen in a 12 point typeface. As in the preliminary drill of the Tinker Test [74], volunteers were asked to cross through the word that spoils the meaning of each item. Volunteers were asked to "work for speed and accuracy, that is work rapidly but do not make mistakes".

Volunteers were asked to complete the 3 pages of 5 test items, each time reminded of the instructions to "work for speed and accuracy". They were told that the computer would record their responses and the amount of time it took them to complete the task. After completing each page, volunteers were asked to complete a short questionnaire.

The questionnaire was based on QUIS 7.0[3] from the University of Maryland, adapted to evaluate reading handwritten script. Our questionnaire consisted of 3 question groups testing: the overall experience of reading the script; the ease of reading; and the clarity of the words displayed. Each group consisted of 4 or 5 questions with a negative to positive response scale of 5 points. In the first group, all 5 questions were taken from the QUIS. The 4 questions in the second group were completely new, based around terms commonly used to describe legibility. In the third group, the first 2 questions were adapted from the QUIS and the second two were new. A copy of the questionnaire can be seen in Appendix B

Finally, after answering the questionnaire for the third time, volunteers were asked to rate the rendering methods in order of preference, from 1 to 3 with 1 as their favorite. As an aide-memoir, the volunteers were shown a screen with a sample of each rendering method side by side in the order they had originally seen them.

## 4.6 Results

Twenty-one volunteers took part in the experiment. One volunteer did not have English as their first language, so she was dropped. Of the remaining twenty volunteers, three failed the initial screening task so did not complete the first part of the experiment. Two volunteers did not complete the last page of the questionnaire, so they were dropped from the second part of the experiment.

[3]Questionnaire for User Interaction Satisfaction

Table 4.2: Pixel Width against Recognition Rate

| **Pixel Width (mm)** | **Mean Recognition Rate** (SD) |
|---|---|
| 0.30 | 82% (19%) |
| 0.45 | 84% (14%) |
| 0.60 | 81% (17%) |
| 0.75 | 71% (20%) |
| 0.90 | 59% (16%) |

Table 4.3: Summary of Mean User Preferences

| **Rendering Method** | **Overall Reactions** | **Ease of Reading** | **Clarity of Words** | **Preference** |
|---|---|---|---|---|
| ClearPen | 3.12 | 3.43 | 3.56 | 1.28 |
| Antialiased Pen | 3.11 | 3.39 | 3.47 | 1.72 |
| Pixel Pen | 2.54 | 2.74 | 2.86 | 3.00 |

Thus in total there were 17 volunteers for the first part of the experiment, and 18 volunteers for the second.

The results for the first part of the experiment are summarized in Table 4.2, "Pixel width against recognition rate". For the second part of the experiment, the component questions have been averaged over each question group. The results are summarized in Table 4.3, "Summary of mean user preferences". Table 4.3 also contains the volunteers' rank-order preference of rendering method. While the responses to the QUIS questions were ranked 1–5 with 5 as the most preferable, the rank-order data has a range of 1–3 with 1 as the most preferable.

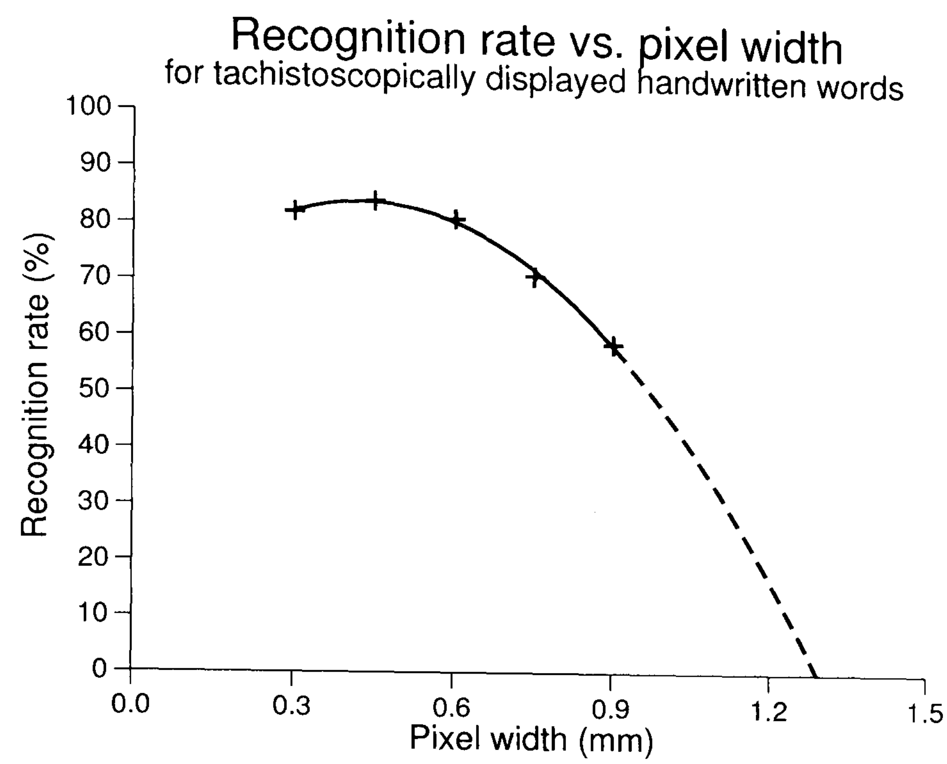



Figure 4.12: Recognition Rate against Pixel Width

## 4.7 Analysis

In order to test the significance of any differences in the means of the data from the tachistoscopic presentation experiment (Table 4.2), an Analysis of Variance (ANOVA) was performed.

The result of the ANOVA, $F(4, 80) = 5.481, p < .001$, suggests that the differences in performance in word recognition observed in the experiment could be ascribed to the effect of the independent variable, pixel width.

Post-hoc comparisons were made. These showed that the mean performance in the 0.90mm condition was significantly worse than performance in the 0.30–0.60mm conditions. To test for a relationship between pixel width and recognition rate, a Pearson's Product Moment Correlation was performed on the data in Table 4.2. The results of this analysis, $r = 0.353, p < 0.001, N = 85$, suggest that there is a relationship between pixel width and recognition rate. This is illustrated in Figure 4.12.

The questionnaire response means in Table 4.3 for the ClearPen and antialiased rendering methods all fall above the response scale mid-point (3). The responses

Table 4.4: Subjective Factors: Rank Order Correlation

| SPEARMAN'S RHO | | $N = 54$ |
|---|---|---|
| **Question** | $\rho$ | **Sig. (1-tailed)** |
| Overall Reading | .370 | .003 |
| Ease of Reading | .303 | .013 |
| Clarity of Words | .339 | .006 |
| Preference | $-.861$ | .000 |

for the pixel pen rendering method consistently fall below this point. This would suggest, that in general the volunteers reacted positively to the higher resolution rendering methods, and negatively towards the pixel pen method.

The data from the subjective preferences questionnaire and the preference rank order data, were further analyzed using Spearman's Rank Order Correlation. These results are summarized in Table 4.4.

The responses to the subjective factors questions are significant at the $p < .05$ level, and the preference rating at the $p < .001$ level.

## 4.8 Discussion

Both experiments confirm the hypotheses they set out to address. Increasing horizontal rendering resolution increases both the legibility of, and preference for working with, handwritten script.

### 4.8.1 Legibility

The first experiment addressed the hypothesis:

> "Improving the horizontal rendering resolution of a handwritten word, displayed on a computer screen, enhances its legibility."

The result of the first experiment affirms this hypothesis. If a linear relationship between pixel size and recognition rate is accepted, then any improvement in horizontal rendering resolution will enhance the legibility of handwritten script.

Figure 4.12 however does not show a linear relationship. The curve fitted is preferred for two reasons. Firstly it confirms the $\approx 80\%$ plateau in the recognition rate that was identified during our investigation into exposure time (Figure 4.1). Secondly, it indicates that the recognition rate falls off quite sharply in contrast to a linear relationship. This is in agreement with an informal observation during the planning of the experiment, where it was noted that words rendered with a pixel width greater than 0.9mm were practically illegible.

The results therefore indicate that legibility of handwritten script can be improved by increasing the horizontal rendering resolution if recognition is below the optimum plateau.

We predict that different styles of handwriting will have different recognition rate thresholds. However, assuming that the recognition plateau will always be reached around the same resolution, the results from this experiment can be used to estimate the optimal screen resolution for a screen to display handwriting.

Figure 4.12 shows the recognition rate threshold being reached at a pixel width of around 0.60mm. At the viewing distance of 1800mm, the 0.60mm pixels occupy around $0.019^\circ$ of the visual field. At a normal viewing distance of around 450mm, $0.019^\circ$ of the visual field would translate into a pixel width of around 0.15mm. This is equivalent to a display resolution of around 170 dpi. This suggests that 170 dpi LCD screens may be sufficient for displaying handwritten script.

### 4.8.2 Preference

The second experiment addressed the hypothesis:

> "A user will perceive the effect of horizontal resolution enhancement as beneficial, and will prefer reading script rendered in this manner."

Table 4.4 clearly shows that the volunteers in the experiment perceived a difference in the three different rendering methods, and that their preference followed increasing resolution.

Even though antialiased script may, on cursory observation, appear similar to script rendered in ClearPen, this experiment has shown an appreciable difference.

Referring again to Figure 4.12, the equivalent pixel widths for ClearPen and the pixel pen, read at 450mm, would be 0.4mm and 1.2mm respectively. The ClearPen result falls comfortably within the recognition rate plateau. Although we are not sure of the equivalent resolution of the antialiased pen, the shape of the graph shows why the antialiased rendering method was rated positively. Even small gains in resolution over that of the pixelated rendering method will significantly improve legibility.

## 4.9 Conclusions

Reading handwriting on a computer is as feasible as it is on paper and need not be hampered by poor script legibility. The results of the two experiments are mutually supportive. The legibility of handwritten script displayed on an LCD computer screen is improved by increasing the horizontal rendering resolution. People are able to perceive this improvement and prefer reading more legible script.

A screen resolution of 170 dpi was proposed as being sufficient for reading handwriting. This value is dependent on the assumption that the recognition rate plateau will always be reached at the same resolution, independent of the style of handwriting (which will certainly affect the *level* of the plateau). The generality of this result is yet unproven as handwriting from only one writer was used in the experiment.

The ClearPen rendering method is capable of improving the legibility of handwritten script displayed on an LCD screen. However, it does have a number of limitations. Firstly, script must be written along the direction of a scan-line. Informal observation has shown that legibility is not greatly impacted by vertical resolution enhancement. Secondly, the technique involves sacrificing colour for resolution. In applications where colour or freedom of orientation are important, ClearPen may not be suitable.

These experiments have pioneered the objective measurement of the legibility of handwritten script. They have also helped to generalize the findings of Wright et al. [83] by confirming that resolution affects legibility.

We have also confirmed that the legibility of digital ink can be explored within the data layer of our structural framework of an Informal Interface, without straying into the other layers.

## 4.10 Further Work

The most significant shortcoming of this work is that the findings are not based on a large group of writers. There is plenty of scope for running similar trials with many different types of script. This could extend beyond Western scripts.

Although we have established that perceived horizontal rendering resolution does indeed affect script legibility, and that volunteers prefer interacting with script rendered with greater legibility, we have not made a direct quantitative measurement comparing different rendering methods in a "real world" task. This could be investigated.

Taking a broader outlook, the legibility measurement technique could be applied to other types of experiment. This experiment addressed the effect of resolution on legibility. Further experiments could look at enhancing the legibility of script in other ways, such as applying geometric transforms to emphasize features pertinent to human recognition.

Finally, work could be undertaken to discover whether or not resolution enhancement has any effect on ease of use when writing onto a screen.

# Chapter 5

# Digital Ink Clustering

This chapter addresses the second principal barrier to implementing a pen-based informal interface: structuring digital ink. Specifically, this chapter presents pilot work to explore digital ink clustering. Instead of concentrating on clustering techniques, this chapter examines digital ink strokes: The correlation of their various geometric properties to their membership of clusters representing handwritten words.

## 5.1 Introduction

To structure and manipulate a page of handwritten ink words on a computer, will require that individual ink strokes are clustered together in groups representing the individual words. Research suggests that this is the ideal level at which to interact with handwritten script: Humans perceive word forms more readily than they do individual letters or non-words [54, 56]; a single stroke will often represent many letters whereas a word will be composed of one or more whole strokes; and other research projects also work at a word level [10, 15, 27, 36, 37, 45, 61].

Digital ink clustering research started in the study of on-line handwriting recognition[1]. In handwriting recognition it is often necessary to cluster strokes into letter or word groups before the strokes are analyzed by the recognition al-

[1]On-line recognition refers to algorithms that process pen-tip coordinate data, as opposed to off-line algorithms which process images of handwriting

gorithm. Initially this problem was solved by imposing some constraints on the user, such as writing letters within boxes or pausing for a timeout after writing each word [70]. Later work looked at the segmentation of *unconstrained* on-line handwriting and used the time and distance between strokes to determine word groups [18, 32, 41]. More recently, handwriting recognition has used a feedback process where recognition results can correct an initial segmentation [52].

Although this approach appears to have been adopted with Microsoft's Tablet PC software, it may not be practical to implement this in all situations. Such an approach requires a sophisticated recognition algorithm and a large dictionary of words. Such dictionaries cannot be exhaustively complete or language independant.

Geometric approaches to stroke grouping have been presented recently using temporal and spatial measures to calculate distances in a hierarchical agglomerative clustering algorithm [36, 61]. This work is all based on a paper by Chiu and Wilcox [10], which we will examine shortly.

No published work quotes how accurately their "word-entities" correspond to actual words, nor do they detail how their algorithm constants are determined, neither does any algorithm detail the use of any metric, apart from temporal and spatial measures of the gaps between strokes.

## 5.2 Hypotheses

We believe that accurate digital ink clustering is a principal barrier to the implementation of an intuitive interface to draft and edit documents on a pen-based computer. Consequently we contend that the geometric information in handwriting is sufficient to reliably group ink strokes into words. In this chapter we are focussed specifically on determining the level of information contained within the ink strokes, as opposed to the analysis of any particular clustering technique. Our contention then gives rise to three questions which we will answer in turn.

Firstly, what is an "acceptable" level of errors for any clustering algorithm to be considered "sufficient" or "reliable"? For our work presented here, we will assume that rate to be very low, perhaps less than 1%. To pin-point an "acceptable" value is a non-trivial task requiring extensive user trials, and is out of the scope of our current investigation. We will however address the measurement of errors.

Secondly, what is the general accuracy of geometric clustering algorithms? What sort of general accuracies can we expect when grouping strokes based on temporal and spatial measures of the gaps between them? Having determined this, we will be able to gauge the magnitude of the problem to be overcome.

Finally, how might improvements be made to geometric clustering algorithms if their accuracy is significantly less that what is required? Which geometric factors can inform stroke clustering, and are these factors dependent on handwriting style?

These last two questions lead us to propose three hypotheses:

1. The general accuracy of geometric clustering algorithms primarily based on temporal and spatial gap measures is not sufficient to reliably group ink strokes into words.

2. In addition to temporal and spatial measures of gap distance, there are a number of other measures that can contribute information to geometric clustering algorithms.

3. Writing styles can be statistically characterized and this information used to predict the relative importance of different measures contributing information to the clustering algorithm.

The following two sections detail related work, and data collection and preparation respectively. The three chapters after that then address each hypothesis in turn. That is followed by a discussion and conclusions from the work carried out.

## 5.3 Related Work

Work concerning the geometric clustering of unconstrained online handwritten words started in the late 80s in the field of handwriting recognition.

Initial approaches to stroke grouping used only the spatial gaps between strokes. Mandler [41] simply grouped ink strokes into "symbols" if the shortest distance between any two strokes was below a certain threshold. The process described in his paper does not seem to be focussed on grouping words from a large body of script, more individual characters from a small number of strokes. Kurtsberg and Tapper [32] also used the shortest distance between any two strokes, however their threshold values are reported to be different between the 'x' and 'y' dimensions. Neither paper reports the value of thresholds used, or the accuracy of the segmentation.

Fox and Tappert [18] report a system that groups cursive words using spatial, temporal, and "language dependent" information. As in previous work, the spatial information is the smallest distance between strokes. The temporal information, the time gap between strokes. The "language dependent" information appears to refer to the observation that, for the English language, one word is usually completed before the next is started, and that diacritical marks are generally added last of all. Again no thresholds or accuracies are detailed.

Recent approaches to grouping ink strokes into words use selection driven methods, such as hierarchical agglomerative clustering around a selected stroke [10], or recursively clustering intersecting strokes through a selection stroke [15]. While this would seem to fit in with the Informal Interaction paradigm, only recognizing when required, problems can arise if the user expects any dynamic behaviour such as words that re-flow around cut and paste operations.

Non-selection driven methods include our own work, lookinig at the applicaiton of Neural Networks to the grouping of strokes [69], and work (at a graphical, rather than geometric, level) by Saund et al. [61], who use both proximity measures and hierarchical agglomerative clustering to form "proto-words". These basic structures are then assembled into extended structures, most often corresponding to lines of text. They do not report any details of the exact process stating that it is "continuously in flux as we attempt to improve its performance".

Li et al. [36] describe a method to structure freeform handwritten notes. Their fundamental structural element is a "strokeblock", a cluster of pen strokes. strokeblocks can be created by a clustering algorithm or selection gestures. They report using the same minimum distance function as Chiu and Wilcox [10], and group strokes in strokeblocks if the minimum distance is below a certain threshold.

Chiu and Wilcox, in their paper *"A Dynamic Grouping Technique for Ink and Audio Notes"* [10], demonstrate the application of an ink distance measure and

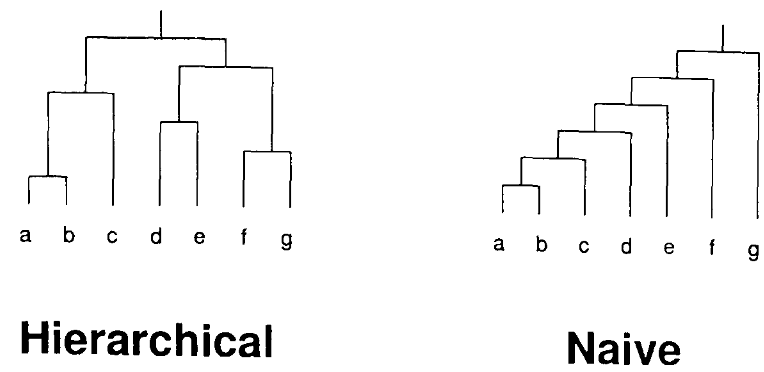



Figure 5.1: Agglomerative Clustering

hierarchical agglomerative clustering. Hierarchical agglomerative clustering follows the following procedure:

1. Calculate the minimum distance between all clusters (each single stroke is initially a one element cluster).

2. Merge the two closest clusters to form a new cluster.

3. Repeat this process until there is only one cluster containing all strokes.

This contrasts with naive hierarchical clustering which will always merge the closest stroke to the *initial* cluster as can be seen in Figure 5.1. Chiu and Wilcox report that the grouping levels within the hierarchical agglomerative clustering "correspond roughly to words, lines, or paragraphs of text".

Chiu and Wilcox's distance algorithm is a weighted sum of the time and space between two strokes. When calculating the distance between two clusters, the minimum distance between member strokes from each group is used. The time distance is the time between the end of one stroke and the start of another, this value will not exceed a given threshold. The space distance is the weighted sum of the 'x' and 'y' separations of stroke bounding boxes. The weighting is set to make horizontal grouping more likely than vertical grouping since European script is usually written in horizontal lines. If the bounding boxes overlap in either their 'x' or 'y' projection, that component will be zero, to avoid negative distances.

In summary, hierarchical agglomerative clustering, with a maximum distance threshold with which to cut the clustering tree, seems to be the conventional method for forming "word entity" clusters. Space and time are the preferred measures of distance between strokes. To use in practice, algorithms require certain constants, Chiu and Wilcox's distance algorithm employs three constants. No published paper has been found that details what these constants are, or how they may be calculated. Neither has any paper been found that quotes the accuracy of the correspondence between their "word entity" clusters and the actual written words.

## 5.4 Data Collection

We are examining the correlation between the geometric properties of strokes on a page, and their grouping into words. To do this we collected a number of different online handwriting samples, annotated the data with word division information, and calculated a number of geometric properties for the strokes and gaps between the strokes.

To simplify the analysis of the handwritten strokes we made one major assumption, that words are composed of strokes written consecutively in time. This follows Fox and Tappert's observation that "English ... writers tend to complete a word before beginning the next" [18]. This allowed us to examine gaps only between temporally adjacent strokes as opposed to all gaps between all strokes. In this respect we have departed from the accepted hierarchical agglomerative clustering approach. This allows us to explore a large number of different handwriting features over a small number of gaps instead of examining a small number of features over a large number of gaps.

Based on our assumption that one word is completed before the next is started, our investigation can be thought of as identifying the gap between temporally consecutive strokes as either a "within-word" gap or a "between-word" gap. To do this we analyse each gap, measuring both features describing the gap and features describing the strokes on either side of the gap.

Table 5.1: Source Text Statistics

| | |
|---|---|
| **Readability** | |
| FOG Level | 18.0 |
| Flesch Reading Ease | 53.6 |
| Flesch-Kincaid Grade Level | 7.5 |
| Passive Sentences (%) | 0.0 |
| **Averages** | |
| Words per Sentence | 22.0 |
| Characters per Word | 5.1 |
| **Counts** | |
| Words | 176.0 |
| Characters | 899.0 |
| Sentences | 8.0 |

### 5.4.1 Source Text

The text used was taken from the BBC News web site[2], as a source of contemporary English language. The sample used was on the subject of pensions and its readability statistics are shown in Table 5.1.

The FOG Level and Flesch Reading Ease readability scores [3, 71] indicate that the level of the text is probably above the "standard writing level" (readily understood by 12–14 year-olds). It is perhaps of A-Level standard or above. The Flesch-Kincaid Grade Level puts the text at the "standard writing level", but this test can be skewed by certain writing styles.

### 5.4.2 Collection Procedure

Volunteers copied the same news article, via dictation, onto a page of 8mm lined paper. The paper was fastened with tape on a Wacom Intuos digitizer tablet and the volunteers wrote with a Wacom Intuos Inking Pen. The information was sampled at around 94Hz and included movement data, pressure data, and time data. This information was recorded in a text file as integer data (see Figure 5.2).

[2] http://news.bbc.co.uk/

| x-pos. | y-pos. | x-tilt | y-tilt | pressure | buttons | time (s) | time ($\mu$s) |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Figure 5.2: Data File Format

The dictation took place in a quiet office environment, volunteers were observed by the orator and the dictation proceeded at the speed of the writer. Volunteers were only told that a sample of handwriting was required. Before the dictation commenced the volunteers were allowed to use the pen and paper until they felt comfortable with using them to write.

Six samples were selected to represent a broad range of handwriting styles. Of these samples, three were written by female writers, three by male writers. All writers were right handed. The size of each sample is similar due to using 8mm lined paper in the collection process. Samples of the volunteers handwriting can be seen in Figure 5.3.

### 5.4.3 Data Annotation

Once the samples were collected, the word groups were annotated. This was done using a program that "played back" the handwriting data file a stroke at a time. The end of each word was then marked with a key press. At the end of this process, a tag-file was generated which details the limits of the words and the gaps between them. Its format can be seen in Figure 5.4. The time stamps and sample numbers correspond to the sample information in the original data file. Records in the tag file are marked "SPCE" and "WORD", for space and word data respectively. The comment field was not assigned, but included for future extensions.

Occasionally the assumption that writers complete a word before beginning the next did not hold. On these occasions extra word tags were introduced so that a group of strokes tagged as a word never included gap that should be tagged as a space. This is illustrated in Figure 5.5 where, against our assumption, the diacritical marks in "this" are made after the word "is" is written.

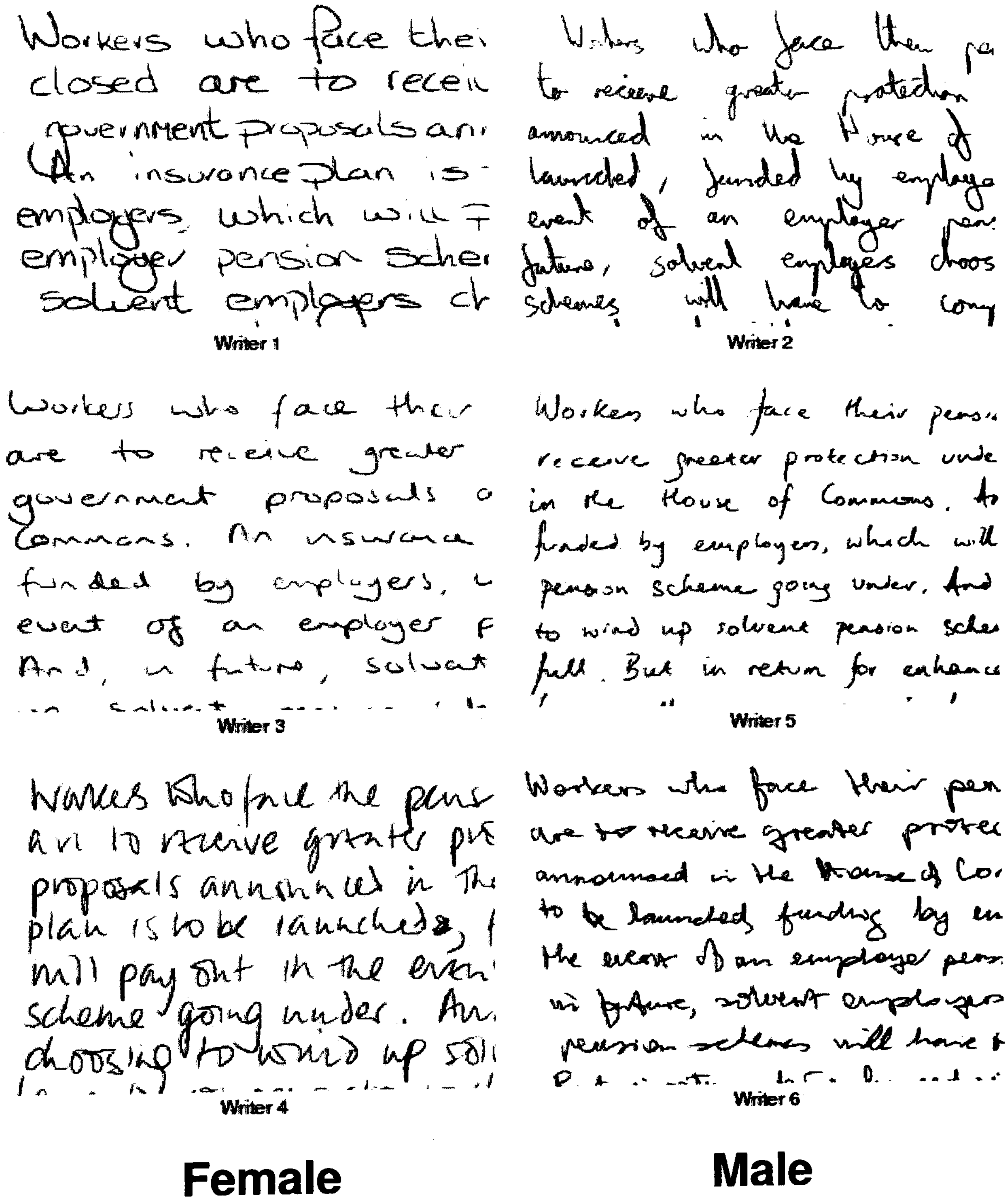



Figure 5.3: Styles of Handwriting Collected

| initial time (s) | initial time ($\mu$s) | end time (s) | end time ($\mu$s) | initial sample num. | end sample num. | tag | comment |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Figure 5.4: Tag File Format

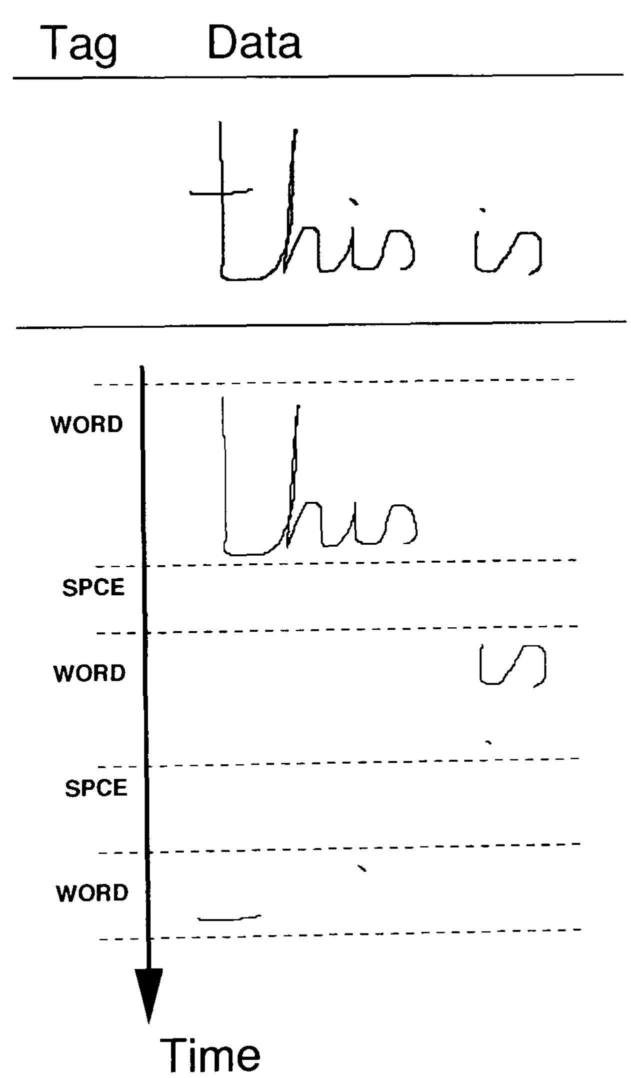



Figure 5.5: Tagging Unexpected Sequences

| | | | | | |
|---|---|---|---|---|---|
| gap id | stroke num. (**n-1**) | stroke num. (**n**) | 25 stroke features (stroke **n-1**) | 14 gap features | 25 stroke features (stroke **n**) |
| gap id | stroke num. (**n**) | stroke num. (**n+1**) | 25 stroke features (stroke **n**) | 14 gap feature | 25 stroke features (stroke **n+1**) |
| | | | | | |

Figure 5.6: Gap Feature File Format

## 5.5 Gap Features

The focus of this work is on the identity of gaps between temporally consecutive strokes: whether they are between-word gaps, or within-word gaps. Consequently, the focus of our analysis is on the information describing gaps. We calculated 64 features describing each gap, 25 features each for the preceding and following strokes, and 14 features for the gap itself. The features were determined by a program which read the data and tag files to generate a feature file that logged the features and identity of the gap they described. The format of this file is illustrated in Figure 5.6.

Of the 64 features that describe each gap, 30 are distinct: 16 describe only strokes; 5 only gaps; and 9 features are common to both strokes and gaps. The first 13 features we used came from Rubine's work on gesture recognition [59]. Rubine defines 13 features to characterize single stroke gestures. These include angles, lengths, time, and speed. We added a further 17 features which include more angle and length measurements, acceleration, and pressure measurements.

The 16 stroke features are as follows:

1. (From Rubine) The cosine of the initial angle of a stroke taken between sample points 0 and 2. Feature number: 1.

2. (From Rubine) The sine of the initial angle of a stroke taken between sample points 0 and 2. Feature number: 2.

3. (From Rubine) The total stroke length. Feature number: 8.

4. (From Rubine) The total angle traversed over the stroke, taken as the sum of the angles turned through between each sample point. Feature number: 9.

5. (From Rubine) The sum of absolute angles traversed. Feature number: 10.

6. (From Rubine) The sum of the squared angles traversed. Feature number: 11.

7. (From Rubine) The maxium speed squared. Feature number: 12.

8. The initial angle of the stroke, taken between sample points 0 and 2. Feature number: 14.

9. The final angle of the stroke, taken between sample points N and N-2. Feature number: 15.

10. The x-balance. The average of all the x-sample values, less the half-way value between the minimum and maximum x-sample values. Feature number: 16.

11. The y-balance. Calculated as for the x-balance. Feature number: 17.

12. The total acceleration. Feature number: 18.

13. The total acceleration squared. Feature number: 19.

14. The average pressure. Feature number: 20.

15. The sum of change in pressure along the stroke. Feature number: 26.

16. The sum of change in pressure squared along the stroke. Feature number: 27.

The 5 gap features are calculated on the difference between the values for the strokes preceding and following the gap of interest. These features are:

1. The distance between the centre-of-gravities (CoGs). The $CoG_{(x,y)}$ is calculated as the average of all the x and y sample values. Feature number: 24.

2. The displacement is the same as the distance between CoGs except that the value will be negative if the x-value of the following stoke CoG is less than the x-value of the preceding stroke CoG. Feature number: 25.

3. The distance between CoGs including pressure, i.e. $CoG_{(x,y,z)}$ where z represents the average pressure. Feature number: 28.

4. The angle between the end of the preceding stroke and the start of the following stroke. Feature number: 29.

5. The minimum distance between all sample points in the preceding and following strokes. Feature number: 30.

The 9 common features are possible because when the pen moves within 5mm of the tablet surface, information is still recorded. There is no guarantee that the entire path of the pen between strokes will be recorded, but there will be information detailing the pen position at the beginning and end of the gap at least. The 9 common gap and stroke features are:

1. (From Rubine) The length of the stroke (or gap) bounding box (BB) diagonal. Feature number: 3.

2. (From Rubine) The angle of the BB diagonal. Feature number: 4.

3. (From Rubine) The distance between first and last point of the stroke or gap. Feature number: 5.

4. (From Rubine) The cosine between the first and last point of the stroke or gap. Feature number: 6.

5. (From Rubine) The sine between the first and last point of the stroke or gap. Feature number: 7.

6. (From Rubine) The duration of stroke or gap. Feature number: 13.

7. The total x-displacement of the stroke or gap. Feature number: 21.

8. The total y-displacement of the stroke or gap. Feature number: 22.

9. The number of sample points in the stroke or gap. Feature number: 23.

Overall, these measures can be described as: gap features describing angles, lengths, and time; and stroke features describing angles, lengths, time, speed, acceleration, pressure, and balance. The balance describes the distance of the centre of gravity of a stroke (the 'x' and 'y' average of all sample points in a stroke) from the centre of its bounding box. Some gap features incorporate stroke data, such as the gap length measure "distance between centres of gravity", which is a measure of the distance between the centres of gravity of the strokes either side of a gap.

A number of features are related and can give similar or identical values in certain situations. For example, the length of the "bounding box diagonal", and the "distance between first and last sample points", will usually be the same distance for a gap (pen movement is logged even when the pen-tip is off the tablet surface), but different for a stroke, unless it is a straight line.

## 5.6 A Clustering Algorithm

In our first hypothesis, we stated that:

> "The general accuracy of geometric clustering algorithms primarily based on temporal and spatial gap measures is not sufficient to reliably group ink strokes into words."

The purpose of this section is to determine a rough accuracy for a geometric clustering algorithm principally employing temporal and spatial gap measures. This figure can then be compared with our estimated "acceptable" level to evaluate our hypothesis.

The approach taken is to identify features which exhibit a good discrimination between the two gap types (not surprisingly temporal and spatial gap measures) then, remove outlying data points to reduce the variance in that feature and minimize possible systematic errors. After this the gaps are classified based on a spatial gap distance measure (temporal measures are implicit as our data set is ordered in time, and stroke clustering is constrained to serial groups).

### 5.6.1 Data Cleaning

In preparation for stroke clustering, outlying data points in the important gap features were removed from the data sets describing each sample of handwriting. The points removed represented such things as diacritical marks and line breaks. The strategy employed to do this for each writer was as follows:

1. Perform t-tests on the 64 features to identify the features that are significant in determining the identity of the gap they are associated with.
2. Plot a histogram of a significant feature to see if there are any outlying or overlapping data points between the two gap types.
3. Highlight the strokes causing these outlying data points in the original handwriting sample and examine for any similarities.
4. Propose and test functions to filter these points from the data set.
5. When a suitable function has been found, filter the corresponding samples from the data set, making any recalculations that may be necessary.
6. Repeat the procedure for another significant feature, until no more outlying features can be removed.

This process is now described in detail, showing examples from writer 4.

**Diacritic Removal**

Steps were undertaken to improve the "gap distance" feature separation between the two gap types. This was done by removing diacritic strokes from the data set. For the purposes of this work, "diacritics" are *any* small marks added to letters, such as the dot on an 'i' or the cross-bar of a 't'. The usefulness of diacritic removal is language specific, relying on the fact that diacritics are usually added just before starting the next word. The process can also help overcome some instances of the tagging problem illustrated in Figure 5.5.

We employed two procedures to remove diacritics, the first that removed short strokes preceding a large gap, the second that removed very short strokes. This was done by filtering the gap feature file (Figure 5.6) for the appropriate features. In the case of writer 4, for the first procedure we looked for strokes which were less than 6mm followed by gaps that were greater than 3.1mm, for the second procedure we removed all strokes shorter than 1mm.

Most constants were determined for each individual writer through observation of each feature histogram. The histograms for stroke length and gap distance, for writer 4, are shown in Figures 5.7 and 5.8 respectively. In general constants were selected specifically for each writer. When this was done values which could possibly be determined algorithmically were used. For instance, in Figure 5.8, a minimum seems to occur at 3.1mm between two peaks centered at about 1.5mm and 5.1mm. This was the value selected for gap distance. Stroke length was an exception to this and values of 6mm and 1mm were used for all handwriting samples. These values were chosen by inspection of all handwriting samples: 6mm to match all diacritics; 1mm to match only diacritics. For the purposes of this experiment, this approach proved satisfactory. To select these constants specifically for each writer would have introduced unnecessary complexity.

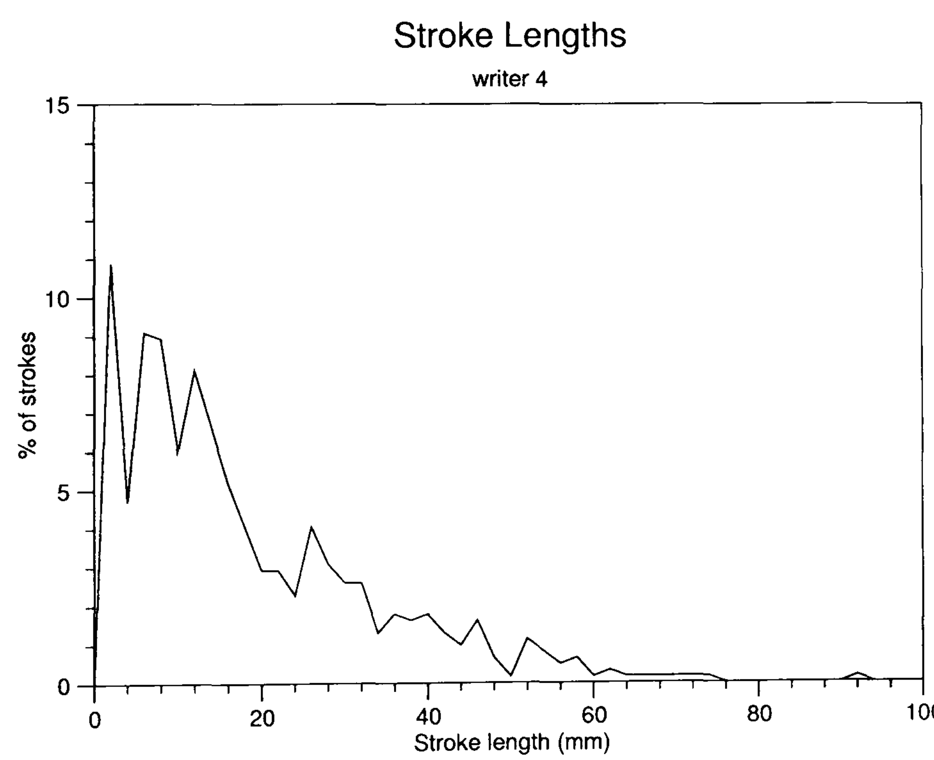



Figure 5.7: Stroke Length Histogram (Raw Data)

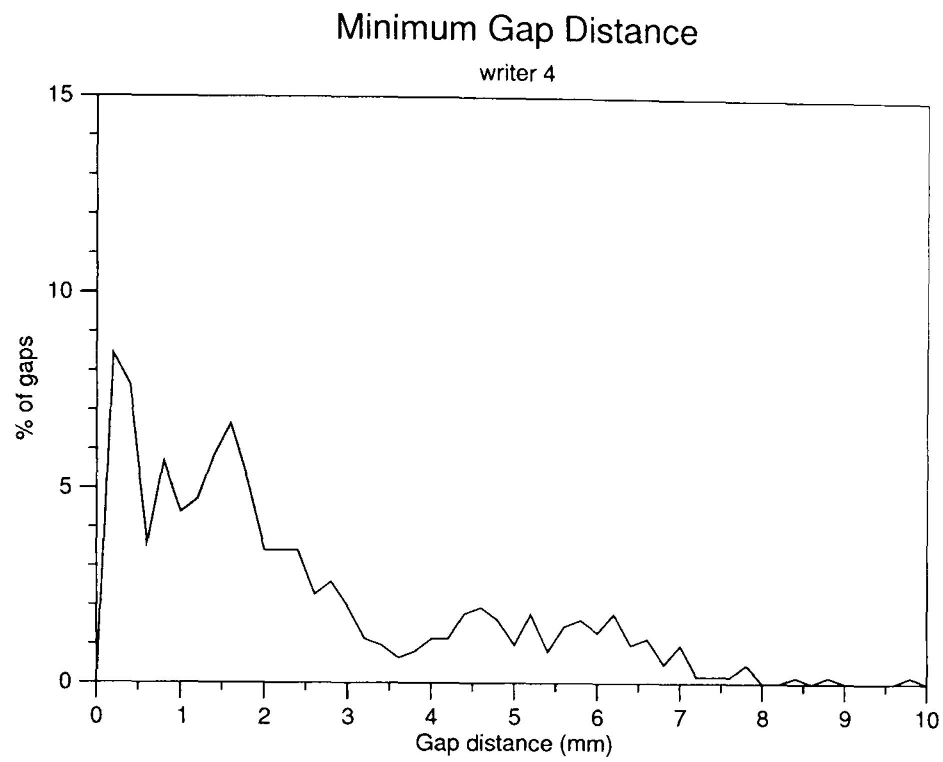



Figure 5.8: Minimum Gap Distance Histogram (Raw Data)

Gaps are removed from the "Gap Feature" file, simply by deleting the line that describes them and their surrounding strokes. When strokes are removed, as in the procedures to remove diacritic marks, a certain amount of recalculation is required. First of all, reference to the target stroke is removed from the Gap Feature file. This is done by removing the two lines that refer to the target stroke and replacing them with a line describing the preceding and following strokes, this process is illustrated in Figure 5.9. Secondly, the gap information in the replacement line is recalculated as if the removed stroke was not there. This is trivial for gap distance and angle features, as they can be calculated in the same way as before stroke removal. This is not the case for gap time features, since taking the difference between stroke time-stamps will yield a gap time incorporating the time taken to write the removed stroke. This is not desirable since it can make the gap artificially large.

The problem of calculating new gap times after stroke removal is illustrated in Figure 5.10. Here stroke 's2' is to be removed. The time gap between the remaining strokes, 's1' and 's3' is designated by 'tc'. There are many different

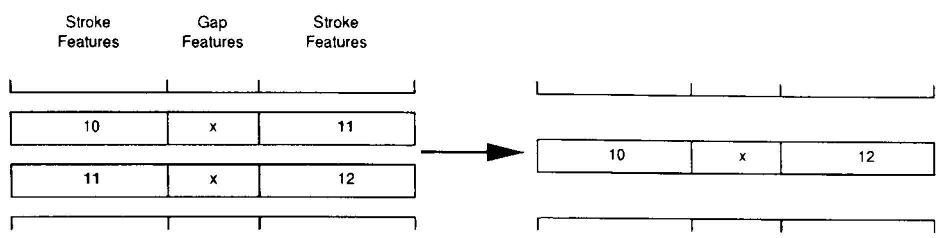



Figure 5.9: Stroke Removal from the Gap Feature File

ways to calculate 'tc'. We chose to calculate 'tc' as $tc = t2 + tb - ta$. This compensates for the backward movement associated with adding diacritics after the body of the word is completed [18]. The disadvantage of this approach is that it can introduce negative gap times.

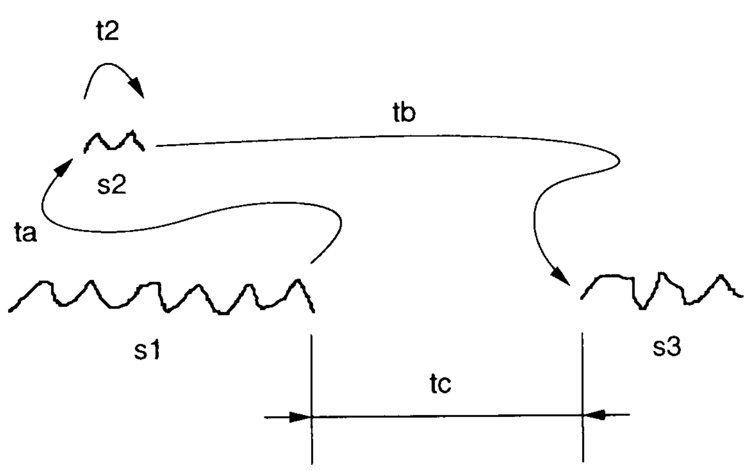



Figure 5.10: Calculating New Gap Time Features

### Gap Classification

Further data cleaning concerned the classification and removal of gap data. Initially very large gaps corresponding to outlying points on a "gap distance" histogram were removed. These can be seen on the right-hand side of Figure 5.11 and correspond primarily to line breaks. We can see that most gap distances are less than 14mm. In the case of writer 4, we removed all gaps larger than 18mm, immediately after the second consecutive empty division on a histogram showing 2mm increments.

This method is not perfect, and can remove some within-word gaps. This

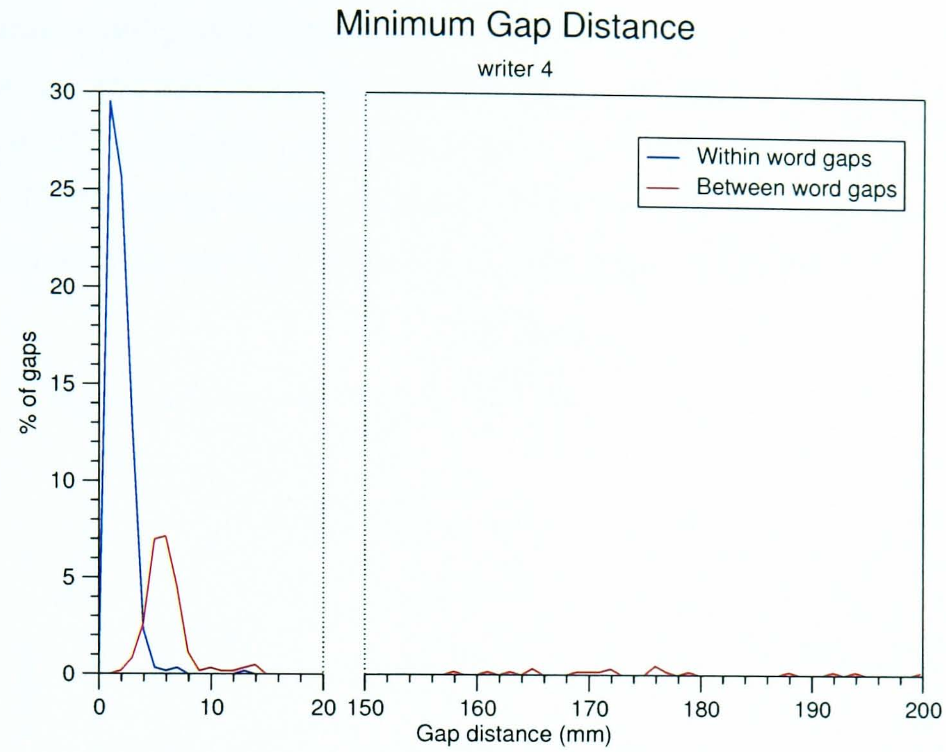



Figure 5.11: Minimum Gap Distance Histogram

occurred exclusively with writer 4 and is depicted in Figure 5.12. Here the writer has written "£65,000,000", adding the pound sign after the digits as a diacritic would be added. Since the pound sign is too large to be classified as a diacritic and be removed its addition has caused us to misclassify this gap. It would be possible to prevent such occurences by changing the relevant algorithm but this was not done since the error occurred so infrequently.

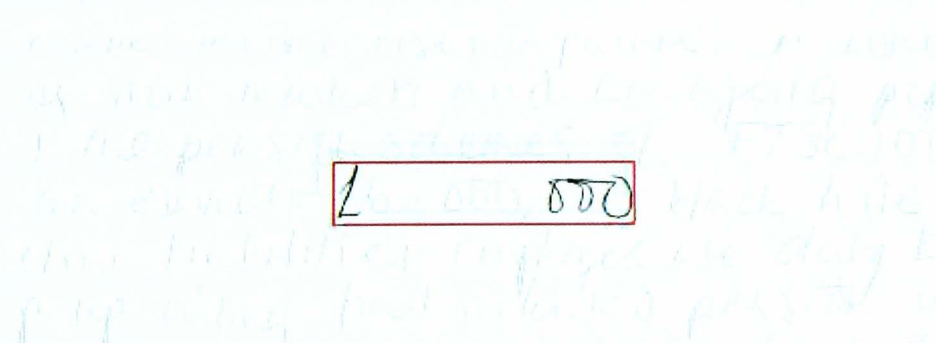


Figure 5.12: A Large Within-Word Gap

Finally, gaps were removed to improve the separation of temporal gap measures. within-word gaps are generally short in time and space, whereas between-word gaps are generally long in time and space. Figure 5.13 shows the gap time distribution. The majority of within-word gaps occur between 0 and 0.1s. The majority of between-word gaps between 0.2 and 0.3s. There is an area of overlap between 0.1 and 0.2 seconds. This was the same across all writers.

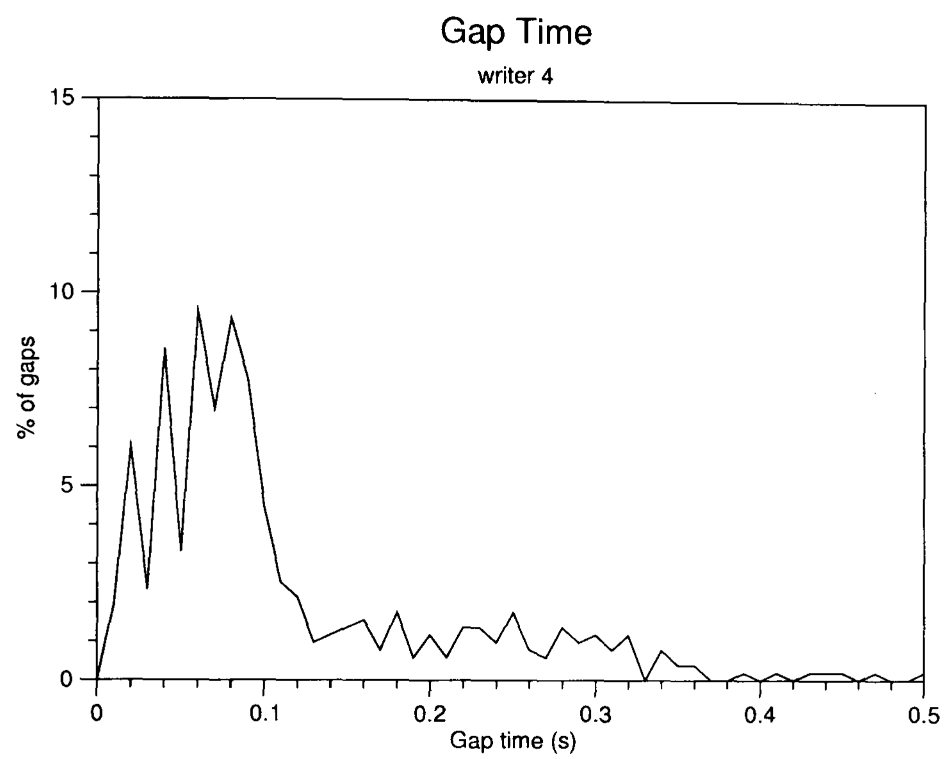



Figure 5.13: Gap Time Histogram (Partially Cleaned Data)

We removed between-word gaps with a duration of less than 0.2s and within-word gaps with a duration of greater than 0.1s. We used gap distance to discriminate between the two groups. Referring back to Figure 5.8 and the first diacritic removal procedure, we divided the 'minimum gap' data into two groups. Gaps less than 3.1mm and gaps greater than 3.1mm, roughly corresponding to within and between-word gaps. We took the means and standard deviations of both groups, added the standard deviation to the mean of the within-word gaps (2.4mm), and subtracted the standard deviation from the mean of the between-word gaps (3.6mm).

Combining these features we were able to remove outlying gaps in the time dimension. Firstly very quick but wide between-word gaps, and secondly very slow but thin within-word gaps. The effect of this can be seen in Figures 5.14 and 5.15 which show the time and distance gap measures before and after the removal of the outlying time gaps.

### 5.6.2 Analysis of Cleaning Process

The outcome of the data cleaning for each writer is summarized in Table 5.2. The average number of words in each data set is 177 (average number of between-word gaps + 1), one more than the actual number of words in the source data set. This may be due to introducing extra words during dictation, inserting spurious marks, or completing words after starting the following word. On average the data cleaning process removes 30% of the gap data. Of these removed points, only 1.5% are misclassified.

Table 5.2: Data Cleaning Summary

| | Writer | | | | | |
|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** |
| **Summary** | | | | | | |
| Original Num. of Gaps | 1007 | 387 | 771 | 616 | 612 | 374 |
| Num. of Between Gaps | 175 | 178 | 177 | 176 | 173 | 176 |
| Tot. Records Removed | 246 | 202 | 245 | 179 | 134 | 143 |
| Tot. Incorrectly Removed | (6) | (1) | (2) | (1) | (0) | (7) |
| **Individual Stages** | | | | | | |
| Diacritics Before Gap | 102 | 131 | 82 | 45 | 68 | 60 |
| Incorrect | (4) | (0) | (1) | (0) | (0) | (0) |
| Very Short Diacritics | 7 | 8 | 39 | 35 | 31 | 6 |
| Incorrect | (0) | (0) | (0) | (0) | (0) | (0) |
| Very Long Gaps | 23 | 18 | 23 | 22 | 15 | 18 |
| Incorrect | (0) | (0) | (0) | (1) | (0) | (0) |
| Quick and Wide Gaps | 20 | 19 | 20 | 30 | 9 | 30 |
| Incorrect | (1) | (0) | (1) | (0) | (0) | (4) |
| Slow and Thin Gaps | 94 | 26 | 81 | 47 | 11 | 29 |
| Incorrect | (1) | (1) | (0) | (0) | (0) | (3) |

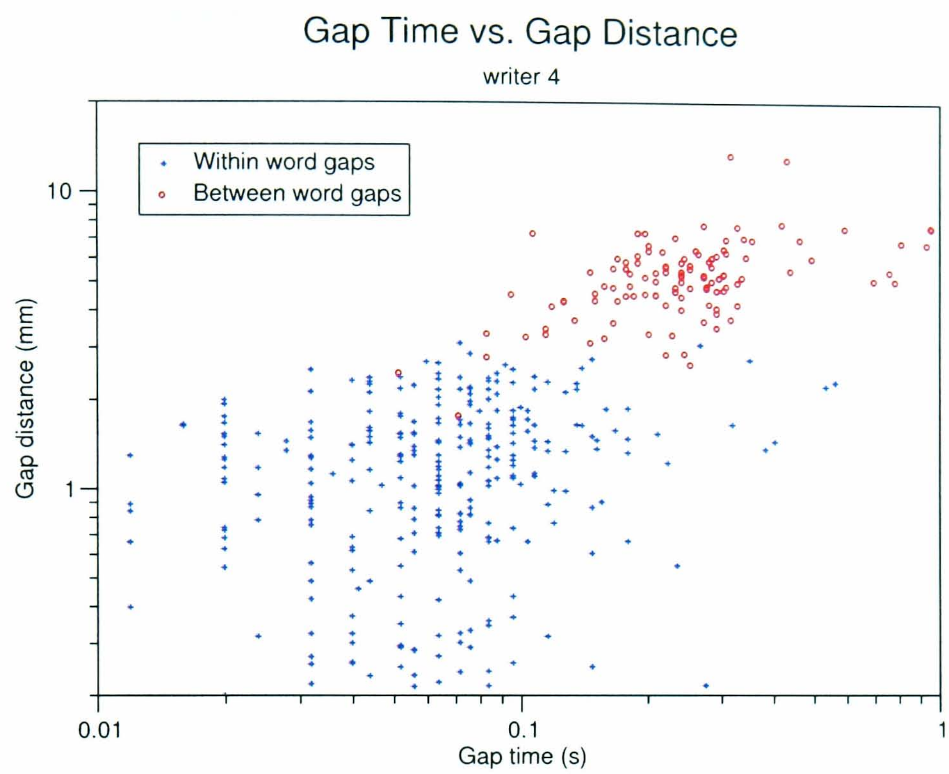



Figure 5.14: Time and Distance Gap Data Before Removing Outliers

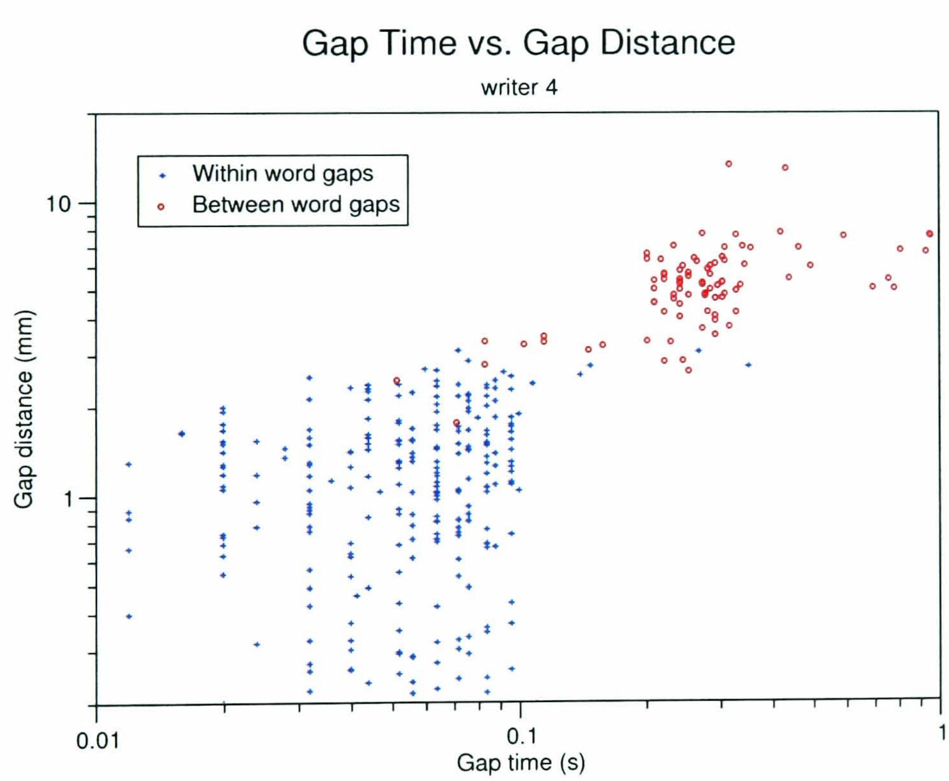



Figure 5.15: Time and Distance Gap Data After Removing Outliers

The incorrectly removed data points represent a number of anomalies within the data. In the case of "Diacritics Before Gap" the five misclassifications are 4 short 'i' strokes separated from their diacritic, which was added immediately by writer 1, and 1 short 'l' stroke in the middle of a word with an appreciable gap on either side. The single "Very Long Gap" misclassified in the writer 4 data set has already been discussed (see Figure 5.12).

The "quick and wide gaps" misclassifications relate to: for writer 1, an 's' added to the end of a word after they had gone back and added a diacritic at the beginning of the word; for writer 3, a quick and wide gap created within a word by the incorrect removal of a short 'l' stroke mistaken for a diacritic; and for writer 6, 4 particularly large within-word gaps. The misclassifications of the "slow and thin gaps" are caused exclusively by writers leaving very little space between words.

Idiosyncrasies in style can produce problems in data cleaning, such as writer 1's tendency to occasionally write 'i' characters with the dot longer than the down stroke, and writer 6's tendency to write some within-word gaps very much larger than their between-word gaps, and vice-versa.

### 5.6.3 Clustering Accuracy

To illustrate the effect of the cleaning process, the gap data for each of the writers were inspected before and after the cleaning process. To do this we split the gap data into two groups simply on whether the 'minimum gap distance' was above or below the 'minimum gap distance' constant used in the first and last data cleaning stages. As well as this split on cleaned and uncleaned data, we also incorporated the gaps classified during the cleaning process in the cleaned data set.

Instead of looking at the percentage of gaps correctly or incorrectly classified, we looked at the number of words spoilt by incorrect classifications. We believe word accuracy to be a more suitable measure than gap accuracy since we assume that interaction happens at the word level. For instance, 100 words written in a cursive style (few strokes per word) with 20 gap misclassifications may have a gap accuracy of 10%. The same 100 words written in a print style (many strokes per word) with 20 gap misclassifications may have a gap accuracy of 5%, though both styles may in fact have the same number of words spoilt by the misclassifications.

The results are presented in Table 5.3.

Table 5.3: Word Clustering Results

| | Writer 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Text Statistics** | | | | | | |
| Number of Words Written | 174 | 179 | 177 | 177 | 173 | 177 |
| Gaps Separated at (mm) | 2.5 | 3.5 | 4.0 | 3.1 | 2.5 | 2.2 |
| **Before Cleaning** | | | | | | |
| Words Split | 49 | 25 | 8 | 15 | 9 | 24 |
| Words Joined | 8 | 4 | 7 | 6 | 4 | 10 |
| Words Spoilt (%) | 32.8 | 16.2 | 8.5 | 11.9 | 7.5 | 19.2 |
| **After Cleaning** | | | | | | |
| Words Split | 7 | 2 | 1 | 2 | 1 | 8 |
| Words Joined | 10 | 5 | 6 | 6 | 4 | 11 |
| Words Spoilt (%) | 9.8 | 3.9 | 4.0 | 4.5 | 2.9 | 10.7 |

## 5.7 Independent Feature Analysis

In our second hypothesis, we stated that:

> "In addition to temporal and spatial measures of gap distance, there are a number of other measures that can contribute information to geometric clustering algorithms."

The purpose of this section is to investigate the contribution individual features may make to gap classification, and thus stroke clustering. The approach taken was to identify six significant and semantically independent features. These features were then tested for correlations between them across writers.

Features which correlate strongly with each other are likely not to contribute any extra information to gap identity. Thus features without strong correlations to other features could be exploited to resolve ambiguous gap identities.

### 5.7.1 Feature Selection

We identified the most promising six gap features for classifying the gap identity. The features were selected by ranking each of the significant gap features ($p < 0.001$) for each writer in order of increasing $p$ value from a students t-test. This produced lists of about 30 features for each writer. The results for the t-test on the cleaned data from writer 4, are shown in Table 5.4. The table only contains the p-values.

Table 5.4: T-Test Results for Writer 4

| **Previous Stroke** | | **Gap** | | **Following Stroke** | |
|---|---|---|---|---|---|
| Feat. Num. | P-value | Feat. Num. | P-value | Feat. Num. | P-value |
| 1 | 7.07E-001 | 3 | 2.08E-017 | 1 | 3.10E-002 |
| 2 | 7.36E-002 | 4 | 1.98E-032 | 2 | 3.01E-007 |
| 3 | 1.85E-005 | 5 | 4.35E-052 | 3 | 4.43E-009 |
| 4 | 3.25E-002 | 6 | 3.02E-061 | 4 | 4.51E-002 |
| 5 | 9.76E-003 | 7 | 1.57E-014 | 5 | 7.82E-006 |
| 6 | 5.64E-002 | 13 | 8.65E-014 | 6 | 1.70E-001 |
| 7 | 2.21E-003 | 21 | 1.83E-066 | 7 | 4.77E-006 |
| 8 | 6.52E-004 | 22 | 3.19E-002 | 8 | 8.82E-008 |
| 9 | 2.83E-003 | 23 | 5.54E-022 | 9 | 6.89E-001 |
| 10 | 9.71E-003 | 24 | 1.11E-048 | 10 | 6.13E-006 |
| 11 | 1.70E-001 | 25 | 1.09E-049 | 11 | 5.80E-006 |
| 12 | 3.72E-003 | 28 | 5.73E-001 | 12 | 5.45E-012 |
| 13 | 4.12E-003 | 29 | 4.05E-002 | 13 | 5.07E-006 |
| 14 | 3.97E-002 | 30 | 5.07E-064 | 14 | 6.54E-007 |
| 15 | 4.98E-002 | | | 15 | 1.45E-001 |
| 16 | 7.62E-001 | | | 16 | 1.02E-003 |
| 17 | 7.43E-001 | | | 17 | 3.12E-006 |
| 18 | 4.39E-001 | | | 18 | 9.74E-006 |
| 19 | 1.61E-001 | | | 19 | 3.50E-011 |
| 20 | 2.91E-010 | | | 20 | 4.67E-001 |
| 21 | 5.96E-002 | | | 21 | 1.64E-001 |
| 22 | 1.28E-001 | | | 22 | 2.06E-006 |
| 23 | 4.17E-003 | | | 23 | 5.08E-006 |
| 26 | 4.77E-007 | | | 26 | 1.13E-001 |
| 27 | 2.31E-006 | | | 27 | 8.59E-001 |

Table 5.5: Most Promising Features Among All Writers

| Rank | Feature |
|---|---|
| 4 | g30 |
| 5 | g25 |
| 6 | g21, g24 |
| 8 | g5 |
| 9 | g3 |
| 16 | g23 |
| 17 | g6 |
| 18 | g13 |
| 22 | g4 |
| 28 | f19 |

'g' represents a gap feature, 'f' a following stroke feature.

By juxtaposing each of the six lists, and then comparing them, common significant features were identified and recorded after they had appeared in all six lists. This process returned 11 significant features (Table 5.5); however after examining their identities, it was found that they only represented 4 independent concepts: gap distance, gap time, gap angle, and the acceleration in the following stroke. The process was repeated to incorporate features appearing in any 5 of the 6 feature lists. This returned a further 3 features consisting of two measures of previous stroke length and a measure of previous stroke pressure. One feature from each group was selected by choosing those which occurred first across all writers. The features finally selected were:

**Minimum Gap Distance:** The minimum distance found when measuring the distance between each of the preceding stroke sample points, and each of the following stroke sample points.

**Gap Time:** The duration of the gap. The difference in time stamps from the first sample point in the gap, to the last.

**Gap Angle:** The cosine of the angle between the first gap sample to the last gap sample.

**Following Stroke Acceleration:** Total acceleration squared. The sum of $(d^2s/dt^2)^2$ across all sample points in the stroke.

**Preceding Stroke Average Pressure:** The average pressure across all the sample points in the stroke.

**Preceding Stroke Length:** The total length along the path of the stroke. The sum of the Euclidean distances between consecutive sample points.

### 5.7.2 Statistical Analysis

To test for relationships between variables we used Pearson's Product Moment Correlation. Our analyses investigated the correlations between the different selected features in both the within-word and between-word cases for each writer. After isolating the correlating features for each gap type and writer, we looked for correlations common to the majority of writers. We were able to establish three strong relationships between particular gap features where correlations were significant at the 0.01 or 0.05 levels (2-tailed) across 11 of the 12 cases (between and within gaps for 6 writers).

Searching for these correlations serves two purposes. Firstly if different features commonly occur together, then the amount of gap data can be reduced by only measuring one feature, or taking a product of the correlated features. Secondly, knowledge of the correlations may help improve gap classification algorithms.

### 5.7.3 Observations

Our analyses revealed strong positive correlations across gap types and writers for:

1. Preceding Stroke Length and Preceding Stroke Average Pressure: The longer the stroke the higher the pressure.

2. Gap Distance and Gap Angle: The further the distance between strokes, the higher the cosine of the angle between the end of the preceding and start of the following strokes.

3. Gap Distance and Gap Time: The further the distance between strokes, the longer the pause between them.

The correlations between Gap Distance, Angle, and Time have been improved or uncovered by the cleaning process. The removal of diacritics and line-ends entails that the large between-word gaps only have small positive or negative gap angles, and thus a large cosine. The removal of "quick and wide" gaps and "slow and thin" gaps improves the positive correlation between gap distance and time. The correlation between stroke length and average pressure would seem a little less self-evident. Although if we assume that the longer a stroke is, the more it progresses along the x-axis, this result would appear to be in agreement with observations made by Wann and Nimmo-Smith [78]. They confirmed the results of a previous study that "observed a steady increase in pressure as a writing task progressed along the x (horizontal) axis".

There was also a positive correlation between Gap Angle and Gap Time. This correlation was only significant across the between-word gaps of three of the six writers. This lends weight to the second and third correlations above. Further experimentation with a larger set of writers may reveal more correlations.

### 5.7.4 Implications

The stronger correlations, while initially surprising, have logical explanations largely associated with the cleaning process. The implication of these correlations is that there is little point in measuring the strongly correlated features in the hopes of resolving gap-type ambiguities. This allows us to reduce the volume of gap data using either a single measure or the product of related features. While gap distance measures seem to be the most useful for classifying gap type, features such as stroke length should be explored in resolving gap-type ambiguities.

## 5.8 Style Analysis

In our third hypothesis we stated that:

> "Writing styles can be statistically characterized and this information used to predict the relative importance of different measures contributing information to the clustering algorithm."

The purpose of this section is to demonstrate writer characterization and explore the relationship of this with the average feature values, selected in the previous section, for each writer. This information could help build a word gap recognition or generation model which incorporates writing style.

### 5.8.1 Writer Characterization

We consulted literature for information concerning the characterization of handwriting. Handwriting style characterization is often used in handwriting recognition to select a recognition algorithm appropriate to the style of writing. Consequently, contemporary work is closely associated with the recognition algorithms used, such as Hidden Markov Models [68] or Self Organizing Maps [77], and has no direct application without the use of neural networks.

Leroy [35] was also concerned with improving handwriting recognition algorithms and looked at a number of factors including word height, number of strokes per word, and the space between words. Another common idea in this field is the concept of cursivity: A scale stretching from hand printed words through to cursive writing. This could be measured by the average number of strokes per word, as in Leroy's work.

Of most interest is work by Peeples and Retzlaff [51] who, through the analysis of 25 handwriting characteristics, proposed three major factors in determining handwriting style: Heights; widths, and angles. We used this work to guide our selection of measurements.

To characterize our writers, we chose 5 statistics calculated from the original stroke data: average between-word gap; average within-word gap; average number of strokes per word; average stroke length; and average pen pressure. We also chose measurements of height, width, and angle, from Peeples and Retzlaff [51]. The last three measures were made by hand due to the combination of two main reasons. Firstly, due to the small number of different handwriting styles, this work will only indicate potential measurements to catagorize style by. Secondly, the time required to implement an algorithm to calculate the measures was estimated to be too great in comparison with the accuracy an algorithmic method may have provided. The measures were made on copies of the original handwritten pages enlarged to 200%.

The manual measures made for each writer were: For height, the average height on the vertical axis of the first 10 'l' characters; for width, the average width on the horizontal axis of the first 10 3-letter words; and for angle, the average angle of the first 10 'l' characters with 90° representing a perfectly vertical character with no slant. In addition to these 8 characteristics we also calculated an average 1-character width and a "roundness" ratio, the average character height over the average character width. The characteristics are shown in Table 5.6. The values measured automatically by software were calculated on the cleaned data, apart from the average number of strokes per word which was calculated on uncleaned data.

### 5.8.2 Statistical Analysis

To test for relationships between variables (shown in Table 5.6) we used Pearson's Product Moment Correlation. This set of analyses investigated the correlations between the individual writer characterization features and the selected average gap feature values across all writers for both within-word and between-word gaps. We examined the strong and weak correlations between each writer characterization feature and gap feature.

### 5.8.3 Observations

Our analyses revealed a number of expected and unexpected strong and weak correlations. We present them here in order of writer characterization feature:

**Between Word Gap:** As expected, this writer characteristic had a significant correlation with the same gap feature. It also correlated weakly with 'average gap time' and negatively with 'average stroke pressure', against the averages from both gap types. The first correlation shows that as the average between-word gap increases, so do the durations of those gaps. This is

Table 5.6: Writer Characterization

| | Writer 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Measured by Software** | | | | | | |
| Between Word Gap (mm) | 5.77 | 8.04 | 7.51 | 5.71 | 4.95 | 4.36 |
| (s.d.) | (2.28) | (3.15) | (2.14) | (2.06) | (1.41) | (1.66) |
| Within Word Gap (mm) | 0.84 | 0.98 | 1.15 | 1.06 | 0.84 | 0.80 |
| (s.d.) | (1.18) | (1.21) | (0.76) | (0.74) | (0.53) | (0.84) |
| Strokes per Word | 5.73 | 2.17 | 4.34 | 3.49 | 3.52 | 2.12 |
| (s.d.) | (2.86) | (1.16) | (2.43) | (2.08) | (2.00) | (1.46) |
| Stroke Length (mm) | 11.54 | 31.41 | 8.90 | 18.74 | 10.99 | 24.72 |
| (s.d.) | (8.17) | (22.08) | (5.71) | (13.90) | (8.21) | (17.54) |
| Pressure | 567.40 | 634.15 | 512.12 | 521.38 | 645.48 | 769.08 |
| (s.d.) | (72.55) | (99.22) | (60.90) | (79.20) | (102.74) | (106.54) |
| **Measured by Hand** | | | | | | |
| ‘l’ Slant Angle (deg.) | 93 | 96 | 90 | 84 | 71 | 87 |
| (s.d.) | (9) | (5) | (4) | (4) | (4) | (18) |
| ‘l’ Height (mm) | 4.9 | 5.8 | 3.3 | 5.0 | 3.7 | 4.0 |
| (s.d.) | (1.0) | (0.8) | (0.7) | (0.5) | (0.5) | (0.8) |
| 3-Character Width (mm) | 13.8 | 9.1 | 11.2 | 11.1 | 8.1 | 9.7 |
| (s.d.) | (3.1) | (1.2) | (1.3) | (1.7) | (1.2) | (1.5) |
| Roundness (calculated) | 1.1 | 1.9 | 0.9 | 1.4 | 1.4 | 1.2 |

in agreement with the results from the independent feature analysis in the previous section. The correlation suggests that *the pen speed over gaps is independant of the gap size.*

The latter correlation indicates that as pen pressure increases between-word gaps decrease (1-tailed: stroke pressure before within gaps, $r = -0.573, p = 0.117, N = 6$; stroke pressure before between gaps, $r = -0.590, p = 0.109, N = 6$). This suggests that *writers who write their strokes closer together generally press harder on the pen.*

**Within Word Gap:** As with the between-word gap, the within-word gap measures correlated between the writer characteristic and gap feature measures. Again, as with the between-word gap, the within-word gap correlated negatively with stroke pressure, this time significant at the $p < 0.05$ level for stroke pressures preceding both within and between gaps.

A weaker correlation was also evident. In line with a correlation from the independent feature analysis in the previous section, as within-word gaps increase, so does the within-word gap angle cosine (1-tailed: $r = 0.707, p = 0.058, N = 6$). Which suggests that *as strokes are written further apart, the angle between their end points decreases.* This does not correlate with the between-word gap cosine.

**Strokes per Word:** This characteristic correlates with stroke length and stroke pressure in both within-word and between-word gap cases. The stroke length correlation suggests that the greater number of strokes per word, the shorter the average stroke length (1-tailed: before within gaps, $r = -0.767, p = 0.038, N = 6$; before between gaps, $r = -0.804, p = 0.027, N = 6$). *Since a writer who writes many strokes per word will naturally use shorter strokes than if they used fewer stokes per word. This suggests a disjoint writing style.*

The stroke pressure correlation suggests that the greater number of strokes per word, the lower the average stroke pressure (1-tailed: before within gaps, $r = -0.767, p = 0.038, N = 6$; before between gaps, $r = -0.804, p = 0.027, N = 6$). *One can imagine a writer who writes many strokes per word* *would do so quickly with a light pressure on the pen as it is constantly lifted and repositioned. It may also be that pen pressure is increased when executing curved strokes and that writing styles using a high number of strokes per word do not produce many curved strokes.*

**Stroke Length:** As expected, stroke length correlated between the writer characteristic and gap feature measures. A strong positive correlation also exists between stroke length and average stroke acceleration, $p < 0.01$ for both within and between gap features. This was due to a methodical mistake in the way that stroke acceleration is calculated. Being the sum of the acceleration squared between each sample point, the longer the stroke the higher the acceleration value will tend to be. The acceleration value should have been divided by the stroke length. Since the stroke length and acceleration measures describe the preceding and following strokes respectively they did not correlate in the first group of analyses, however since we are now looking at averages across all gap measurements they do.

**Pressure:** As expected, stroke pressure correlated between the writer characteristic and gap feature measures. There was, as previously covered, a negative correlation between pressure and gap distance (1-tailed: before within gaps, $r = -0.756, p = 0.041, N = 6$; before between gaps, $r = -0.521, p = 0.145, N = 6$). There were also correlations between average stroke pressure and the between-word gap time indicating the higher the stroke pressure, the shorter the between-word gap time (2-tailed: $r = -0.708, p = 0.116, N = 6$). *This may be indicative of a style that employs high pen pressure and a regular writing rhythm.*

**Character Slant Angle:** This writer characteristic made two potential correlations. Firstly, as slant angle increased, within-word gap time increased (1-tailed: $r = 0.748, p = 0.044, N = 6$). *This may indicate that the higher the slant angle, the longer it took the writer to position the pen for the next stroke.*

Secondly, as slant angle increased the between-word gap distance decreased (1-tailed: $r = 0.600, p = 0.104, N = 6$). *This would be commensurate with* *the view that high slant angles reduce the distance between words, as the initial capitals or ascenders of words "lean" into the previous word.*

**Character Height:** This characteristic correlated with stroke length, stroke acceleration, and gap angle. As previously covered stroke length and acceleration are themselves strongly correlated. The correlation between character height and stroke length may indicate that longer strokes entail higher letters (1-tailed: strokes before within gaps $r = 0.646, p = 0.083, N = 6$, strokes before between gaps $r = 0.664, p = 0.075, N = 6$).

The negative correlation between character height and gap angle cosine indicates the higher the characters, the larger the gap angle between the stroke end points (1-tailed: strokes before within gaps $r = -0.578, p = 0.115, N = 6$; strokes before between gaps $r = -0.631, p = 0.094, N = 6$). *This will follow as strokes commonly end near the baseline, and start at the ascender height.*

**Character Width:** This characteristic correlates negatively with the gap angle of between word gaps and positively with the gap time of within word gaps. These are weak indications, for which no explanations can be proposed, as such there may be no correlations.

**Character Roundness:** This characteristic correlates positively with stroke length and stroke acceleration (which are strongly correlated with each other). The taller a character becomes in relation to its width (an increase in roundness) the longer the average stroke lengths (1-tailed: strokes before within gaps, $r = 0.700, p = 0.061, N = 6$; strokes before between gaps, $r = 0.832, p = 0.020, N = 6$). *This correlation indicated that tall thin characters require longer strokes than short fat characters.*

### 5.8.4 Commentary

There were six strong correlations found, three positive and three negative. The positive correlations were between: average between-word gap and gap time; average stroke length and character height; and average stroke length and character roundness. The negative correlations were between: average stroke pressure and gap distances; average number of strokes per word and stroke length; and average number of strokes per word and stroke pressure.

From these correlation we can see that the average stroke length, and the average stroke pressure may be good candidates for style characterization features to inform an algorithm classifying gap type. Average stroke pressure could indicate how wide gaps may be expected to be. Average stroke length may give an indication of the number of strokes to be expected in any word. There may also be further features that could contribute to gap classification that would become apparent using a larger sample of handwriting styles.

## 5.9 Conclusions

Our original contention was that the geometric information in handwriting is sufficient to reliably group ink strokes into words. In the work presented we have focussed on determining the extent to which geometric information can inform an algorithm grouping digital ink strokes into words. This approach left the question of quantifying how accurate a clustering algorithm need be to be considered "sufficient" or "reliable", and instead concentrated on determining an estimate of clustering accuracy for geometric methods, and on ways to improve this figure.

### 5.9.1 Discussion

Our initial experiment undertook to estimate a general accuracy for ink clustering algorithms using geometric information. This was done by "cleaning" the ink data of information that may adversely affect the accuracy of a simple grouping algorithm, and then grouping strokes based on the spatial distance of the gaps between them. The process of cleaning was successful, and reduced the error rate of the clustering algorithm by 50% or more for each writer (refer to Table 5.3). The cleaning process was however also a source of some errors. The process often contributed at least 1% of the error rate figure (almost 30% of the total errors in some cases), and in the worst case contributed 4% of the error rate figure (or 37% of the total errors).

It may well be beneficial to work at reducing this level of errors in the cleaning process, in addition to looking to refine the subsequent clustering criteria. Although alterations to the cleaning process may well affect the correlations between the clustering features.

The subsequent experiment, to analyse the independence of potential features to inform clustering algorithms, showed all measures of the gap between two strokes to be strongly correlated. These were thought to be uncovered and enhanced by the cleaning process and left measures of stroke length, pressure, and acceleration to be investigated. The first two features were strongly correlated with each other, and the final feature shown to correlate strongly with stroke length in the final experiment, due to the erroneous way the acceleration was measured. This work indicates that stroke-related features, such as stroke length may well be able to contribute to stroke clustering algorithms. As this sort of analysis is fairly time consuming, furthering this work should probably only commence if required after determining an acceptable clustering accuracy and making improvements to the existing cleaning and clustering stages.

The final experiment set out to characterize handwriting styles and relate this to the potential stroke clustering features. This indicated that both stroke length and pressure may contribute useful information towards grouping algorithms. Although stroke pressure and length were strongly correlated in the previous experiment, they exhibited different correlations in this. More experiments will be required to determine how these features may be used in gap classification.

The work presented has provided a starting point from which to refine and evaluate further algorithms and techniques to group digital ink strokes into hand written words.

### 5.9.2 Findings

We have made a number of findings, specifically these were:

1. To devise and employ a measure of accuracy for stroke clustering algorithms.

2. To confirm the primacy of gap time and distance measures in determining

stroke clusters.

3. To propose and investigate further features to improve stroke clustering algorithms, such as stroke length and pressure.

4. To show how handwriting style may affect the use of stroke clustering features.

### 5.9.3 Further Work

Further work to assess the efficacy of geometric stroke clustering algorithms can continue on three major fronts. Initially, work should be undertaken to quantify an "acceptable" error rate. This should be followed by extending the investigations presented in this chapter, specifically with a larger set of handwriting samples. Finally the resulting "serial" grouping method should be compared against "hierarchical agglomerative clustering" methods, and whether or not the two approaches can inform and improve each other.

Work to quantify an "acceptable" error rate could take a number of script samples from a number of writers, each segmented by hand with errors randomly inserted to produce a known error rate. A large group of volunteers could then be asked to complete a task manipulating the script. Task completion time and subjective factors could then be used to assess how acceptability varies with clustering error rate.

Work to extend the current investigations should proceed in each of the three areas presented. Firstly the data cleaning process should be reviewed, error rates reduced, and the automatic calculation of the feature values used be implemented. Secondly, more potential clustering features should be investigated. A larger body of script samples may reveal more style specific features. Finally work concerning writer characterization should be furthered to potentially develop a model of clustering based on writing style. Different styles could be classed based on the relative importance of the different clustering features determined in the previous feature analysis. Writer characterization could then be explored to see if these classes of writer can be determined from a combination of simple characterization features.

The final area of further work should commence by implementing a hierarchical agglomerative clustering algorithm and measuring its accuracy. Work could then be undertaken to investigate the most appropriate solution for a stroke clustering algorithm in an intuitive interface to draft and edit documents on a pen-based computer.

# Chapter 6

# Conclusions

This chapter presents the conclusions to the thesis. The chapter commences with a summary of the work done, followed by the major findings and a discussion of the work undertaken. It concludes by mapping out directions for future work.

## 6.1 Summary

This thesis set out to find a way forward in the development of pen-based interfaces. To do this we proposed a structural model for applications designed to interact with digital ink based on the concept of Informal Interaction. We demonstrated how this model can be used: Firstly to integrate and analyse the broad range of published work related to pen-based interfaces; secondly to identify areas that have little or no published research addressing the topics the area represents. We piloted two research projects addressing two issues highlighted by our model, from two separate parts of the model. This work was undertaken to increase confidence in the usefulness of the model by demonstrating that each research topic still remained within the bounds of the model layer in which it was first identified.

### 6.1.1 Interface Model

We proposed a structural model, inspired by the concept of Informal Interaction, for applications designed to manipulate digital ink. We defined each layer of the model and showed how literature associated with pen-based interfaces fits within this framework.

We then demonstrated the usefulness of the model by using it to identify pertinent research topics that had yet to be addressed. Due to the layered structure of our model, we were able to identify those topics on which the 'usefulness' of the interface rested.

From the topics that we identified we chose to address the two principal barriers to implementing our digital ink interface and piloted experiments to address them. The work from these two experiments not only adds to the body of literature concerned with pen-based interfaces, but also contributes to the integrity of our model.

### 6.1.2 Digital Ink Legibility

The topic of digital ink legibility was identified in the Data layer of our interface model. This layer was described as encapsulating all internal and external representations of the digital ink data type and other basic objects an Informal Interface may be required to manipulate.

We found that the topic of digital ink legibility had not previously been addressed in literature. We deemed this topic to be important since we found that poor legibility can adversely affect the usability of pen-based interfaces.

We devised methods to objectively and subjectively measure the legibility of handwritten script. These methods were then piloted in experiments that varied the horizontal rendering resolution of handwritten words displayed on a computer screen. The experiments demonstrated that the methods were sufficient to reveal significant differences in legibility between the different resolutions.

This work has contributed methods of measuring digital ink legibility to the field of digital ink interaction research. These methods overcome specific problems faced in the measurement of legibility: the use of relatively small script-sample sizes; a short duration experiment; and effects due to the differences between handwritten script and typed text.

### 6.1.3 Digital Ink Clustering

The topic of digital ink clustering was identified within the Structural layer of our interface model. The layer was described as encapsulating all algorithms to structure ink strokes and to describe the logical behaviour of those structures.

We identified a shortcoming in research to group ink strokes into handwritten words. A facility essential to enable the structuring and manipulation of digital ink in meaningful units. We focussed on geometric based algorithms as opposed to recognition based ones.

Our work investigating digital ink clustering has contributed a number of findings to the field of digital ink interaction research. We have devised and employed a measure of accuracy for stroke clustering algorithms. We have confirmed the primacy of gap time and distance measures in determining word groups. We have also proposed and investigated further potential features to improve grouping accuracy, and shown how the importance of these may be linked to particular styles of handwriting.

## 6.2 Findings

This thesis set out to address the question:

> "How can the principal barriers to an intuitive interface to draft and edit documents on a pen-based computer be overcome."

The answer this thesis proposes is:

> "The principal barriers to an intuitive interface to draft and edit documents on a pen-based computer can be overcome by proposing a structural model of such an interface, using this to identify the principal barriers to its implementation and addressing those issues in turn."

The principal contribution to knowledge made by this work was to propose and demonstrate the usefulness of a model of an intuitive interface for drafting and editing documents. Other contributions included: a method to objectively measure the legibility of digital ink; and an analysis of the relation of ink stroke features to word grouping.

Some of these results have already been published: basic details of the intuitive interface model [6, 7]; investigation into the legibility of digital ink [8, 9]; and investigation into word grouping [69].

## 6.3 Discussion

The findings of this thesis rest on a number of assumptions. We will now take time to look at some of these, in the light of the work undertaken, and comment on the scope of the valid application of the work.

### 6.3.1 Interface Model

Our work on an interface model set out to define a unifying framework in which published literature on pen-based interfaces can be placed and understood as a whole. This was done by looking at the natural strengths and weaknesses associated with using a pen, examining literature on pen-based interfaces, and envisaging an interface that exploits those features. This conceptual interface was then decomposed into broad layers describing the different types of functionality associated with the interface. Finally the published literature was placed in its associated layer.

Our interface model is quite similar to the architecture of the "Back of an Envelope" (BoE) system [24]. As in our Informal Interface the BoE also implements three main functional units, and a "Right Tool Right Time Manager" which interfaces geometric structures with external applications, as does our Translation layer. The three main functional units, Electronic Cocktail Napkin, Drawing, and Geometric Processing, correspond to our Interaction, Data, and Structural layers respectively.

As in our Informal Interface model, user input goes into the Electronic Cocktail Napkin (Interaction layer) and feedback is displayed through the Drawing (Data layer). The BoE and our Informal Interface are in fact almost congruent. There are however three major departures between these architectures. Firstly the central component of each is different. The Back of an Envelope is centred around the Drawing unit, into which the other units interface, whereas our Informal Interface is centred around the Structural layer. Secondly the Informal Interface possesses a layered structure. This is not apparent in the BoE. Finally the BoE also divides structural recognition across two units, the Electronic Cocktail Napkin, which recognizes basic elements and configurations, and the Geometric Processing Unit, which recognizes higher level structures such as 3D sketches and emergent shapes. This functionality is contained solely within the Structural layer in the Informal Interface model.

It is interesting to see the differences in two very similar approaches to implementing a pen-based interface. The BoE architecture seems to have evolved through the design and implementation of a particular interface. Although we have not implemented an Informal Interface from our model, the similarities we have exposed lend weight to its usefulness and the potential that it describes something that can be implemented.

The overall usefulness of our model should also be considered. It was designed around published literature and used to organize it. We have also used the model to isolate research topics. Finally, the model is intended to be a structural model to guide the implementation of an Informal Interface. Although the model seems useful in identifying research topics, it may be that its structure needs to be more fine-grained to be of any real use, since it only acts as an aid to thinking logically about the requirements of Informal Interfaces. To assess its usefulness the remaining issues identified as "to be addressed" could be compared to those encountered when actually implementing the interface. This process may improve the design and granularity of the model.

### 6.3.2 Digital Ink Legibility

Our work on digital ink legibility set out to find whether or not current hardware and software technologies are sufficient to display digital ink legibly. To answer this question, we defined "legibility", proposed methods to measure legibility, and used these to methods to compare samples representing different display hardware and software technologies. This work raises certain questions, in particular related to: Whether or not legibility is an important issue; the correctness of our definition of legibility; and our choice of techniques to measure legibility.

Informal observation of the latest tablet PCs suggests that digital ink legibility is no longer an issue for these devices. They employ high resolution screens and use antialiasing rendering methods. That said legibility measurement may still be required for other devices using digital ink, for instance: mobile phones; PDAs; and low-budget tablet computers. This issue could be addressed conclusively by looking at the legibility of different resolution displays, rendering methods, and handwriting styles, quantifying the relationship between them.

Our definition of legibility was taken from a publication dating back to 1964 [73]. We need to consider whether or not this is still valid. Tinker was concerned with the legibility of typeface on paper, whereas we are concerned with handwritten script on an illuminated screen. Tinker is frequently referenced in publications addressing legibility, yet our problem domain is now quite far removed from his. There is little guarantee that his findings will hold in our domain.

Most important to consider here is the idea that legibility is bound with the method of measurement, which brings us to our final point for consideration. We measured recognition rate and preference as indicators of legibility. Both followed increasing rendering resolution. There are, however, other methods used in literature. Although we chose to measure recognition rate for reasons of practicality, it is worth considering the alternatives and how each relates to the concept of legibility. In particular, with reference to our problem domain, rather than that in which they were originally employed.

We found five possible approaches to measuring legibility. We trialled and eliminated a speed of reading method, because it did not work with handwritten script. This does not entail that speed of reading methods are not suitable for measuring the legibility of handwritten script since different methodologies could be used, such as a comprehension test. For the sake of acceptance and applicability it is probably best to use a measurement associated with completing a "real world" task i.e. a task that is, or reflects, the actual use of the interface, such as a reading or writing exercise. Our use of a short exposure technique did not follow this ethos, however recognition rate could be measured without short exposure, for instance by having subjects read a list of words displayed on screen for as long as needed. In this respect, search tasks are perhaps the most difficult tasks to conceive tests that reflect actual real world tasks. They may also suffer from the same problem encountered during our initial speed of reading experiment due to the intrinsic imperfect legibility of handwritten words. The use of oculomotor measurements of eye fatigue, inversely correlated with legibility, seems to be ideal, since any arbitrary task can be set using eye fatigue measurements taken before and after the task.

As well as considering the different techniques used to measure legibility, we should also consider the way the chosen technique is employed. A conscious decision was made to use administration staff as it was assumed that they are more likely to work with other people's handwriting than any other type of staff. This was done since we anticipate that people who will interact with a computer using handwriting will be familiar with, and confident in, working with handwriting.

In general, the selection of volunteers for our study is comparable with published literature. The five related studies we examined [4, 48, 72, 75, 83] used a similar number of subjects ($av.17.6, s.d.8.6$) and all appear to have been undertaken within universities or associated research centres. Two studies did not mention the source of their volunteers [48, 83], one of the studies selected volunteers from among undergraduate students [75], and the final two [4, 72] selected from staff or graduate students.

It is important to consider how representative the selection of volunteers is of a predicted user population since this may affect the applicability of the results obtained. Future devices stemming from this research may well have a wide user base and as such, any experiment associated with their development should aim to employ a diverse range of volunteers.

### 6.3.3 Digital Ink Clustering

Our work on digital ink clustering set out to find whether or not geometric grouping methods are sufficient for grouping digital ink strokes into words, which can be used as a basic structural element within an Informal Interface.

To answer this question, we need to know both the accuracy limits of geometric grouping methods, and what level an "acceptable" error rate is. We did not address this second component of the answer, assuming an acceptable error rate to be less than 1%. If the acceptable error rate were to be higher, this may reduce the need to improve grouping algorithms. This is certainly a topic that should be addressed as this work progresses.

That said, even if the average error rate was below some "acceptable" value, there may be a number of different handwriting styles that generate an unacceptable level of grouping errors. Our work demonstrated quite a high level of variability in grouping error rate, even when the constants of our algorithm were selected for each individual writer. In this case, we could say that if the acceptable error rate is found to be in the vicinity of our own average error rate ($\approx$6%) then work looking at improving the grouping algorithm for different styles of writing will still remain a relevant research topic.

A further point that rises from this work is our use of an algorithm grouping strokes occurring serially in time, whereas other important work in this field employs hierarchical agglomerative clustering (HAC) techniques. The serial method was chosen principally because of the number of parameters (stroke or gap features) being analyzed. By restricting grouping to consecutive strokes, we reduced the amount of data we had to generate and analyse. This restriction implicitly incorporated temporal distance measures into our grouping algorithm. With a HAC method, spatial and temporal measures may not be as strongly correlated as we found them to be.

Due to the different methods used, we cannot assume a direct correspondence between our measures of grouping error rate, and those that may be found using HAC methods. Any relevant features we may find in strokes that indicate adjacent gap identity may not be immediately implemented using HAC methods, though may give some indication of how distance calculations may be improved. Finally, there may also be issues to be considered if writer characterization information is to be translated between the two different grouping approaches. Ultimately, the two approaches should be compared side by side, once the requirements of grouping algorithms are quantified, and the most appropriate method selected.

### 6.3.4 Methodology

We set out to examine how the principal barriers to an intuitive interface to draft and edit documents on a pen-based computer can be overcome. The methodology we employed was to propose a model of such an intuitive interface based on literature, then review the literature in the context of the model. In doing so we identified a number of barriers to implementing the interface, those area which had not previously been addressed. We used the position of the identified topics in the model, their layer and layer functionality, to identify the principal barriers. We then went on to demonstrate how these barriers may be overcome.

The question we now wish to address is whether or not this was the most suitable strategy to adopt to answer our question? There are a number of points to examine: Firstly, whether or not we, in fact, identified an intuitive interface; secondly, whether or not our model is capable of highlighting the principal barriers; and finally, if the work undertaken actually overcame these barriers.

The first issue concerns whether or not we identified an intuitive interface. This is unfortunately very difficult to assess. Our design process, looking at the natural strengths of pen and paper, as well as other literature in the area, coupled with the fact our model corresponds to that of the "Back of an Envelope" project suggest that we are on the right lines, yet the only way of quantifying whether or not an interface is truly intuitive is to implement and test it.

The second issue questions the capability of the model in highlighting the principal barriers. Due to the division and specification of functionality, our model focusses attention on relatively low-level component-concepts. It is certainly suitable for identifying gaps within its component layers. It is also capable of representing a rough hierarchy of these concepts such that, as with our work on ink structuring, although higher level concepts such as ink grammars may have been addressed, those topics on which others rest are revealed. We cannot claim to have found *all* barriers, however within the confines of our assuptions we have certainly addressed the principal barriers.

The final issue asks whether or not the work we undertook addressed the barriers we identified. In both topic areas, we proposed and piloted approaches to address the issue we identified. While neither issue was addressed conclusively, we did demonstrate processes through which they can be addressed and provided evidence to that effect.

Overall, we can conclude that the methodology led us to identify a likely intuitive interface, and provided a structure in which published literature can be understood. This led us to identify and address two fundamental low-level issues, principal barriers to intuitiveness within the proposed interface.

## 6.4 Further Work

We propose that work in this area should now progress by implementing an Informal Interface from the model that we have presented. This will allow one to address a number of the issues raised in the previous section. The resulting interface will enable user experiments to analyse intuitiveness in the use of the interface and examine those issues that fall outside the scope of our interface model.

Finally, if required, work could continue in further addressing the issues already explored and those identified in the model. This will include: comprehensively addressing the topic of digital ink legibility; further exploring the issue of digital ink clustering; and tackling topics such as interfaces for ink-entity selection in different domains.

# Bibliography


[1] Benjamin B. Bederson and James D. Hollan. Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In *UIST 1994, Symposium on User Interface Software and Technology*, pages 17–26. ACM, 1994.

[2] Benjamin B. Bederson and James D. Hollan. Pad++: A Zoomable Graphical Interface System. In *CHI 1995 Conference Companion, Conference on Human Factors in Computing Systems*, pages 23–24. ACM SIGCHI, 1995.

[3] J. R. Bormuth. Cloze test readability, Critical Reference Scores. *Journal of Educational Measurement*, 5(3):189–196, 1968.

[4] Dan Boyarski, Christine Neuwirth, Jodi Forlizzi, and Susan Harkness Regli. A Study of Fonts Designed for Screen Display. In *CHI 1998 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 87–94. ACM SIGCHI, 1998.

[5] Vannevar Bush. As We May Think. *The Atlantic Monthly*, 176(1):101–108, 1945.

[6] Timothy S. Butler. Pen-Based Digital-Document Technology. In Benjamin Zayas and Claudia Gama, editors, *Information Technologies and Knowledge Construction: bringing together the best of two worlds*, pages 3–5. University of Sussex at Brighton, 2001.

[7] Timothy S. Butler. Informal Interaction. *apendig*, (4):12–14, 2002.

[8] Timothy S. Butler. ClearPen: Improving the legibility of handwriting. In *CHI 2003 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 678–679. ACM SIGCHI, 2003.

[9] Timothy S. Butler, Trevor Barker, and Alex Shafarenko. Improving the legibility of Digital Ink. *In preparation for: ACM Transactions on Computer Human Interaction.*

[10] Patrick Chiu and Lynn Wilcox. A Dynamic Grouping Technique for Ink and Audio Notes. In *UIST 1998, Symposium on User Interface Software and Technology*, pages 195–202. ACM, 1998.

[11] Sitt Sen Chok and Kim Marriott. Automatic Construction of Intelligent Diagram Editors. In *UIST 1998, Symposium on User Interface Software and Technology*, pages 185–194. ACM, 1998.

[12] Christian Heide Damm, Klaus Marius Hansen, and Michael Thomsen. Tool Support for Cooperative Object-Oriented Design: Gesture Based Modelling on an Electronic Whiteboard. In *CHI 2000 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 518–525. ACM SIGCHI, 2000.

[13] Richard C. Davis, James A. Landay, Victor Chen, Jonathan Huang, Rebecca B. Lee, Frances C. Li, James Lin, Charles B. Morrey, Ben Schleimer, Morgan N. Price, and Bill N. Schilit. NotePals: Lightweight Note Sharing by the Group, for the Group. In *CHI 1999 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 338–345. ACM SIGCHI, 1999.

[14] Richard C. Davis, James Lin, Jason A. Brotherton, James A. Landay, Morgan N. Price, and Bill N. Schilit. A Framework for Sharing Handwritten Notes. In *UIST 1998, Symposium on User Interface Software and Technology*, pages 119–120. ACM, 1998.

[15] Laurent Denoue, Patrick Chiu, and Tohru Fuse. Shared Freeform Input for Note Taking across Devices. In *CHI 2003 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 710–711. ACM SIGCHI, 2003.

[16] Yasuo Endo, Shinji Akimichi, and Murray Milne. The Context-Based Graphic Input System: T-Board. In G. Salvendy and M. J. Smith, editors, *Advances in Human Factors/Ergonomics, Human-Computer Interaction: Software & Hardware Interfaces*, volume 19B, pages 1004–1009. Elsevier Science Publishers, 1993.

[17] Claudie Faure and Zi Xiong Wang. Automatic perception of the structure of handwritten mathematical expressions. In R. Plamondon and C. G. Leedham, editors, *Computer Processing of Handwriting*, pages 337–361. World Scientific Publishing Co., 1990.

[18] A. S. Fox and C. C. Tappert. On-line external word segmentation for handwriting recognition. In *Proceedings of the Third International Symposium on Handwriting and Computer Applications*, pages 53–55, 1987.

[19] David Frohlich and Richard Hull. The usability of scribble matching. In *CHI 1996 Conference Companion, Conference on Human Factors in Computing Systems*, pages 189–190. ACM SIGCHI, 1996.

[20] David Goldberg and Aaron Goodisman. Stylus User Interfaces for Manipulating Text. In *UIST 1991, Symposium on User Interface Software and Technology*, pages 127–135. ACM, 1991.

[21] John D. Gould and Josiane Salaun. Behavioral Experiments on Handmarkings. In *CHI + GI 1987 Conference Proceedings, Conference on Human Factors in Computing Systems and Graphics Interface*, pages 175–181. ACM, 1987.

[22] Saul Greenberg. A Fisheye Text Editor for Relaxed-WYSIWIS Groupware. In *CHI 1996 Conference Companion, Conference on Human Factors in Computing Systems*, pages 212–213. ACM SIGCHI, 1996.

[23] Mark D. Gross and Ellen Yi-Luen Do. Ambiguous Intentions: A Paper-like Interface for Creative Design. In *UIST 1996, Symposium on User Interface Software and Technology*, pages 183–192. ACM, 1996.

[24] Mark D. Gross and Ellen Yi-Luen Do. Drawing on the Back of an Envelope: a framework for interacting with application programs by freehand drawing. *Computers & Graphics*, 24:835–849, 2000.

[25] Gary Hardock, Gordon Kurtenbach, and William Buxton. A marking based interface for collaborative writing. In *UIST 1993, Symposium on User Interface Software and Technology*, pages 259–266. ACM, 1993.

[26] Knut Hofland and Stig Johansson. *Word Frequencies in British and American English*. Longman, 1982.

[27] Jason I. Hong and James A. Landay. SATIN: A Toolkit for Informal Ink-based Applications. In *UIST 2000, Symposium on User Interface Software and Technology*, pages 63–72. ACM, 2000.

[28] Takeo Igarashi, Satoshi Matsuoka, Sachiko Kawachiya, and Hidehiko Tanaka. Interactive Beautification: A Technique for Rapid Geometric Design. In *UIST 1997, Symposium on User Interface Software and Technology*, pages 105–114. ACM, 1997.

[29] Alan Kay and Adele Goldberg. Personal Dynamic Media. *Computer*, 10(3):31–41, 1977.

[30] A. Kosmala and G. Rigoll. Recognition of on-line handwritten formulas. In *Proceedings of the Sixth International Workshop on Frontiers in Handwriting Recognition*, pages 219–228, 1998.

[31] Gordon Kurtenbach and William Buxton. Issues in combining marking and direct manipulation techniques. In *UIST 1991, Symposium on User Interface Software and Technology*, pages 137–144. ACM, 1991.

[32] J. M. Kurtzberg and C. C. Tappert. Segmentation procedure for handwritten symbols and words. *IBM Technical Disclosure Bulletin*, 25(7B):3848–3852, December 1982.

[33] James A. Landay and Brad A. Meyers. Sketching Interfaces: Toward More Human Interface Design. *Computer*, 34(3):56–64, March 2001.

[34] Jintae Lee. Diffusion rendering of black ink paintings using new paper and ink models. *Computers & Graphics*, 25:295–308, 2001.

[35] Annick Leroy. Correlation Between Handwriting Characteristics. In M. L. Simner and C. G. Leedham, editors, *Handwriting and Drawing Research: Basic and Applied Issues*, pages 403–417. IOS Press, 1996.

[36] Yang Li, Zhiwei Guan, Hongan Wang, Guozhong Dai, and Xiangshi Ren. Structuralizing Freeform Notes by Implicit Sketch Understanding. In *AAAI Spring Symposium on Sketch Understanding*, 2002.

[37] James Lin, Mark W. Newman, Jason I. Hong, and James A. Landay. DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. In *CHI 2000 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 510–517. ACM SIGCHI, 2000.

[38] A. Chris Long, James A. Landay, Lawrence A. Rowe, and Joseph Michiels. Visual Similarity of Pen Gestures. In *CHI 2000 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 360–367. ACM SIGCHI, 2000.

[39] Allan Christian Long, James A. Landay, and Lawrence A. Rowe. Implications For a Gesture Design Tool. In *CHI 1999 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 40–47. ACM SIGCHI, 1999.

[40] Kimiyoshi Machii, Hidehiro Fukushima, and Masaki Nakagawa. On-line Text/Drawings Segmentation of Handwritten Patterns. In *Proceedings of the Second International Conference on Document Analysis and Recognition. 1993*, pages 710–713, 1993.

[41] Eberhard Mandler. Advanced preprocessing technique for on-line script recognition of nonconnected symbols. In *Proceedings of the Third International Symposium on Handwriting and Computer Applications*, pages 64–66, 1987.

[42] Jennifer Mankoff, Gregory D. Abowd, and Scott E. Hudson. OOPS: A Toolkit Supporting Mediation Techniques for Resolving Ambiguity in Recognition-Based Interfaces. *Computers & Graphics*, 24(6):819–834, 2000.

[43] Jennifer Mankoff, Scott E. Hudson, and Gregory D. Abowd. Interaction techniques for ambiguity resolution in recognition-based interfaces. In *UIST 2000, Symposium on User Interface Software and Technology*, pages 11–20. ACM, 2000.

[44] Jennifer Mankoff, Scott E. Hudson, and Gregory D. Abowd. Providing Integrated Toolkit-Level Support for Ambiguity in Recognition-Based Interfaces. In *CHI 2000 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 368–375. ACM SIGCHI, 2000.

[45] Thomas P. Moran, Patrick Chiu, and William van Melle. Pen-Based Interaction Techniques For Organizing Material on an Electronic Whiteboard. In *UIST 1997, Symposium on User Interface Software and Technology*, pages 45–54. ACM, 1997.

[46] Thomas P. Moran, Patrick Chiu, William van Melle, and Gordon Kurtenbac. Implicit structures for pen-based systems within a freeform interaction paradigm. In *CHI 1995, Conference Proceedings on Human Factors in Computing Systems*, pages 487–494. ACM SIGCHI, 1995.

[47] Thomas P. Moran, William van Melle, and Patrick Chiu. Spatial Interpretation of Domain Objects Integrated into a Freeform Electronic Whiteboard. In *UIST 1998, Symposium on User Interface Software and Technology*, pages 175–184. ACM, 1998.

[48] R. A. Morris, R. D. Hersch, and A. Coimbra. Legibility of Condensed Perceptually-Tuned Grayscale Fonts. In Roger D. Hersch, Jacques André, and Heather Brown, editors, *Electronic Publishing, Artistic Imaging, and Digital Typography*, number 1375 in Lecture Notes in Computer Science, pages 281–293. Springer-Verlag, 1998.

[49] Elizabeth D. Mynatt, Takeo Igarashi, W. Keith Edwards, and Anthony LaMarca. Flatland: New Dimensions in Office Whiteboards. In *CHI 1999 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 346–353. ACM SIGCHI, 1999.

[50] Elin Rønby Pedersen, Kim McCall, Thomas P. Moran, and Frank G. Halasz. Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings. In *INTERCHI 1993, Conference Proceedings on Human Factors in Computing Systems*, pages 391–398. ACM SIGCHI, 1993.

[51] E. Edward Peeples and Paul D. Retzlaff. A Component Analysis of Handwriting. *The Journal of General Psychology*, 118(4):369–374, 1991.

[52] Réjean Plamondon and Sargur N. Srihari. On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, January 2000.

[53] Alex Poon, Karon Weber, and Todd Cass. Scribbler: A Tool for Searching Digital Ink. In *CHI 1995 Conference Companion, Conference on Human Factors in Computing Systems*, pages 252–253. ACM SIGCHI, 1995.

[54] William Prinzmetal. The word-superiority effect does not require a T-scope. *Perception & Psychophysics*, 51(5):473–484, 1992.

[55] Eugene H. Ratzlaff. Inter-line distance estimation and text line extraction for unconstrained online handwriting. In L. R. B. Schomaker and L. G. Vuurpijl, editors, *Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition*, pages 33–42, 2000.

[56] G. M. Reicher. Perceptual recognition as a function of meaningfulness of stimulus material. *Journal of Experimental Psychology*, 81:275–280, 1969.

[57] Hélène Richy and Guy Lorette. On-line Correction of Web Pages. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99*, pages 581–584, 1999.

[58] George G. Robertson and Jock D. Mackinlay. The document lens. In *UIST 1993, Symposium on User Interface Software and Technology*, pages 101–108. ACM, 1993.

[59] Dean Rubine. Specifying Gestures by Example. *Computer Graphics*, 25(4):329–337, 1991.

[60] Manojit Sarkar, Scott S. Snibbe, Oren J. Tversky, and Steven P. Reiss. Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens. In *UIST 1993, Symposium on User Interface Software and Technology*, pages 81–91. ACM, 1993.

[61] Eric Saund, James Mahoney, David Fleet, Dan Larner, and Edward Lank. Perceptual Organization as a Foundation for Intelligent Sketch Editing. In *AAAI Spring Symposium on Sketch Understanding*, 2002.

[62] Eric Saund and Thomas P. Moran. A Perceptually-Supported Sketch Editor. In *UIST 1994, Symposium on User Interface Software and Technology*, pages 175–184. ACM, 1994.

[63] Bill N. Schilit, Gene Golovchinsky, and Morgan N. Price. Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations. In *CHI 1998 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 249–256. ACM SIGCHI, 1998.

[64] Lambert Schomaker and Eliane Segers. Finding features used in the human reading of cursive handwriting. *International Journal on Document Analysis and Recognition*, (2):13–18, 1999.

[65] Michael Shilman, Hanna Pasula, Stuart Russell, and Richard Newton. Statistical Visual Language Models for Ink Parsing. In *AAAI Spring Symposium on Sketch Understanding*, 2002.

[66] Mario Costa Sousa and John W. Buchanan. Observational Models of Graphite Pencil Materials. *Computer Graphics Forum*, 19(1):27–49, 2000.

[67] M. Stolpmann and D. Roller. Sketching Editor for Engineering Design. In G. Salvendy and M. J. Smith, editors, *Advances in Human Factors/Ergonomics, Human-Computer Interaction: Software & Hardware Interfaces*, volume 19B, pages 609–614. Elsevier Science Publishers, 1993.

[68] Jayashree Subrahmonia. Similarity measures for writer clustering. In L. R. B. Schomaker and L. G. Vuurpijl, editors, *Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition*, pages 541–546, 2000.

[69] Yi Sun, Timothy S. Butler, Alex Shafarenko, Rod Adams, and Neil Davey. Segmenting Handwritten Text Using Supervised Classification Techniques. In *Submitted to the International Joint Conference on Neural Networks*, 2004.

[70] Charles C. Tappert, Ching Y. Suen, and Toru Wakahara. The State of the Art in On-Line Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8):787–808, August 1990.

[71] W. L. Taylor. Cloze procedure, A new tool for measuring readability. *Journalists' Quarterly*, 30:415–433, 1953.

[72] H. Timmers, F. L. van Nes, and F. J. J. Blommaert. Visual word recognition as a function of contrast. In E. Grandjean and E. Vigliani, editors, *Ergonomic Aspects of Visual Display Terminals, Proceedings of the International Workshop*, pages 115–120. Taylor & Francis Ltd, March 1980.

[73] M. A. Tinker. *Legibility of Print*. Iowa State University Press, 1964.

[74] Miles A. Tinker. *Tinker Speed of Reading Test*. University of Minnesota Press, Minneapolis, 1947, 1955.

[75] Richard A. Tyrrell, Thad B. Pasquale, Thomas Aten, and Ellie L. Francis. Empirical Evaluation of User Responses to Reading Text Rendered Using ClearType™ Technologies. In *SID '01 Digest*, pages 1205–1207. Society for Information Display, 2001.

[76] Shingo Uchihashi and Lynn Wilcox. Automatic Index Creation for Handwritten Notes. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3453–3456, 1999.

[77] Louis Vuurpiijl and Lambert Schomaker. Coarse writing-style clustering based on simple stroke-related features. In *Proceedings of the Fith International Workshop on Frontiers in Handwriting Recognition*, pages 29–32, 1996.

[78] John Wann and Ian Nimmo-Smith. The control of pen pressure in handwriting: A subtle point. *Human Movement Science*, 10:223–246, 1991.

[79] Lisa K. Welbourn and R. J. Whitrow. A gesture based text and diagram editor. In R. Plamondon and C. G. Leedham, editors, *Computer Processing of Handwriting*, pages 221–234. World Scientific Publishing Co., 1990.

[80] Lynn D. Wilcox, Bill N. Schilit, and Nitin Sawhney. Dynomite: A Dynamically Organized Ink and Audio Notebook. In *CHI 1997 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 186–193. ACM SIGCHI, 1997.

[81] Catherine G. Wolf and Palmer Morrel-Samuels. The use of hand-drawn gestures for text editing. *International Journal of Man-Machine Studies*, (27):91–102, 1987.

[82] Catherine G. Wolf, James R. Rhyne, and Hamed A. Ellozy. The paper-like interface. In G. Salvendy and M. J. Smith, editors, *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, pages 494–501. Elsevier Science Publishers, 1989.

[83] S. L. Wright, I. L. Bailey, K-M. Tuan, and R. T. Wacker. Resolution and Legibility: A Comparison of TFT-LCDs and CRTs. In *SID '99 Digest*, pages 346–349. Society for Information Display, 1999.

# Appendix A

# Glossary

**Ambiguity:** A concept peculiar to *Recognition-Based Interfaces*. Ambiguity describes the idea that any input (command or data) may have multiple possible interpretations, and is thus ambiguous.

**Clustering:** A term describing a basic structuring process within an *Informal Interface*. Ink *Strokes* may be clustered into words, words may be clustered into paragraphs.

**Diacritic:** A small mark added to a letter. For instance the dot on an 'i' or 'j', or the cross-bar on an 'f' or 't'.

**Digital Ink:** The path, displayed on screen, described by a pen as it is moved over the screen of a pen-based computer. Digital ink *Strokes* form the basic data type within an *Informal Interface*.

**Formal Interaction:** A term describing interaction with a computer interface that enables interaction with a formally structured document, such as a diagram editor or word processor. These interfaces facilitate structured editing.

**Free-Form Interaction:** A term describing interaction with a computer interface that enables interaction with a document very little structure, such as a paint program or simple text editor. These interfaces facilitate freedom of expression.

**Gesture:** A method of issuing commands in a pen-based interface. A gesture is a specially shaped *Stroke* that specifies a command (from its shape) and operands (from its location).

**Informal Interaction:** A term describing interaction with a computer application, specifically a pen-based interface, that enables both freedom of expression and structured editing.

**Informal Interface:** A *Recognition-Based Interface* that implements *Informal Interaction.*

**Legibility:** The graphical component of readability. Readability refers to how easy a text is to read. This will be affected by fluency and legibility. Fluency refers to concepts like how well written the text is, reading level, and complexity. Legibility describes how the text is presented including factors like letter size, line length, and colours.

**Perceptual Structure:** A concept peculiar to *Informal Interfaces.* Perceptual structure is the structure a user perceives in their free-form input. One of the goals of an *Informal Interface* is to recognize these structures, so users can manipulate them.

**Recognition-Based Interface:** Any interface that employs some form of recognition on input commands or data.

**Rendering Resolution:** The granularity of an image. All image producing devices have a maximum resolution governed by physical constraints. For instance, the size of pixels on a computer screen, or the size of dots produced by a printer. Rendering resolution may be enhanced in excess of native device levels by antialiasing. This employs different levels of gray, or varying intensities of LCD sub-pixels, to represent partially filled pixels or dots.

**Segmentation:** An alternative term describing *Clustering.* For instance, a page of *Strokes* could be segmented into words, or *Strokes* on a page could be clustered into words. In this thesis the term *Clustering* is preferred.

**Stroke:** The term referring to the basic unit of *Digital Ink*. In a pen-based interface a stroke is data that describes the path of the pen from when it is placed on the screen, to when it is lifted off.

# Appendix B

# Subjective Factors Questionnaire

The appendix shows a copy of the questionnaire used to measure the subjective preferences of volunteers, in response to changes in the method used to render handwritten script.

The questions were duplicated for each of the three samples of handwriting each volunteer saw.

## Overall Reactions

Please circle the numbers which most appropriately reflect your impressions about the handwriting you have just read.
Not Applicable = NA.

| | | | | | |
|---|---|---|---|---|---|
| 2.1 | I found *reading* the handwriting was: | terrible | | wonderful | |
| | | | 1 2 3 4 5 | | NA |
| 2.2 | | frustrating | | satisfying | |
| | | | 1 2 3 4 5 | | NA |
| 2.3 | | dull | | stimulating | |
| | | | 1 2 3 4 5 | | NA |
| 2.4 | | difficult | | easy | |
| | | | 1 2 3 4 5 | | NA |
| 2.5 | | rigid | | flexible | |
| | | | 1 2 3 4 5 | | NA |

## Specific Reactions

Please circle the numbers which most appropriately reflect your impressions about the handwriting you have just read.
Not Applicable = NA.

| | | | | | |
|---|---|---|---|---|---|
| 3.1 | Ease of reading | strenuous | 1 2 3 4 5 | comfortable | NA |
| | 3.1.1 Ease of word perception | difficult | 1 2 3 4 5 | automatic | NA |
| | 3.1.2 Speed of word perception | slow | 1 2 3 4 5 | fast | NA |
| | 3.1.3 Demand of reading | tiring | 1 2 3 4 5 | effortless | NA |
| 3.2 | Words on the page | hard to read | 1 2 3 4 5 | easy to read | NA |
| | 3.2.1 Pen trace | fuzzy | 1 2 3 4 5 | sharp | NA |
| | 3.2.2 Clearness of the words | blurred | 1 2 3 4 5 | distinct | NA |
| | 3.2.3 Recognizability of the words | unrecognizable | 1 2 3 4 5 | recognizable | NA |

# Appendix C

# Experiment Instructions

The following two sections detail the instructions given to the experimenter and the volunteers, for the experiment to measure the legibility of handwritten script, respectively.

## C.1 Experimenter Instructions

Welcome the participant. Tell them they are free to ask any questions they like, except when performing the experiment. Check that:

1. They are University administrative staff
2. They have normal eyesight (with correction if necessary)
3. Ask them to fill in a consent form – assign an anonymity number (AN) – record this on the group sheet

Make sure they are comfortable in the chair, and have no problem with the keyboard. Give them the BRIEFING to read and check they understand it, go over the more important points: Interface; and Response.

### C.1.1 Start the Screening

```
% cd ~/tim/experiment2/screening/
% screening.sh AN
```

Check the participant is clear about what they are expected to do, then commence the screening. The screening script will return a score. If the score is less than 6, proceed to Part 2.

A mouse button must be pressed before the subject can see the first slide. Do this when you are ready for them to start. After the subject says each word, mark it as follows:

| **Response** | **Mark** |
|---|---|
| Correct | Mouse Button 1 (left) |
| Wrong | Mouse Button 3 (right) |

### C.1.2 Part 1

Assign the participant to a Part 1 group (N1) and record this on the sheet. Give the subject a chance to ask any questions they may have. Then proceed immediately with the tachistoscopic display:

```
% cd ~/tim/experiment2/part1/groupN1/
% part1.sh AN
```

### C.1.3 Part 2

Give the subject a short break while you move the chair to the desk, and flatten the monitor. Assign the subject to a Part 2 group (N2) and record this on the sheet.

Complete the first page of the questionnaire with the subject.

Refer the subject again to the Briefing, and give them the Familiarization sheet. Check their understanding is OK, then ask them to proceed with the exercise. Tell them to tell you when they have finished, and that you will press the 'space' bar to stop the timer.

```
% cd ~/tim/experiment2/part2/groupN2/
% part2.sh AN
```

After this, check the participant is happy with the equipment and procedure. Then proceed with each slide. After each slide, ask the participant to complete a page of the questionnaire.

### C.1.4 Finally

Show the participant the final slide and ask them to rate the three samples in order of preference. With their favorite sample first. Record this on the Preference sheet.

### C.1.5 Then

Answer any further questions the participant my have, and thank them for their time.

## C.2 Volunteer Briefing

You are taking part in an experiment to assess the legibility of handwritten script, displayed on a computer screen. There are two parts to this experiment, preceded by a Screening exercise to determine your suitability for the first part.

In the Screening exercise and first part, you will read individual words which will be flashed on the computer screen for a short period of time. In the second part of the experiment, you will be timed as you read a number of short passages. You will then answer a short questionnaire after each passage.

If you have any questions, you can ask them before and after the experiment.

### C.2.1 Screening

Please sit on the chair in front of the screen, making sure at all times that your back is firmly against the backrest. This will keep your head at a known distance from the screen. If you usually wear glasses for reading, please do so. Focus on the square in the centre of the screen. The words will appear here. When you are ready to see a word, press the space-bar on the keyboard. After a short time (about 1s), a word will flash on the screen. Please speak the word you see. If you're not sure of the word, guess. After you have spoken the word and are ready to see the next word, please press the space-bar. You will see a total of 10 words.

### C.2.2 Part 1

This part is exactly the same as the screening task, except that the words will be displayed for a slightly shorter period. As in the screening, please make sure at all times that your back is firmly against the backrest. Please speak the word you see. If you're not sure of the word, guess. After you have spoken the word and are ready to see the next word, please press the space-bar. You will see a total of 100 words.

### C.2.3 Part 2

This part is a speed of reading test. Before you start the test, you will be able to complete some practice examples, which will be given to you (on paper and on the computer). You will do the test on the computer with a pen that allows you to write on the computer screen. There will be three pages of hand writing for you to read, as fast as possible. After you have read each page you will then answer a short questionnaire. The questionnaire will not be timed.

### C.2.4 Finally

Finally, you will be asked to rank three samples of handwritten script in order of preference.

# Pen-Based Digital-Document Technology

*Timothy S. Butler*

Science and Technology Research Centre,
University of Hertfordshire, Hatfield, AL10 9AB, UK
http://homepages.feis.herts.ac.uk/~bt7al/
t.s.butler@herts.ac.uk

## Abstract

This paper highlights the ways in which current computer interfaces impair the capture and refinement of knowledge. It defines an interaction paradigm "informal interaction", and proposes that pen-based interfaces designed around the concept of informal interaction have the potential to overcome the current impediments. Finally, it identifies specific areas of research required to develop informal interaction pen-based interfaces.

## Background

Knowledge work is dominated by the use of paper especially when capturing, organizing, and refining information [6][8]. Recent studies show that despite the recognized benefits of using a computer such as prototyping, document duplication, and error checking, computer literate professionals still prefer to capture and refine their ideas on paper. There is a gulf between the way humans express and manipulate ideas, and how computer interfaces require users to structure and interact with data [4].

An author's primary task is to express ideas. Computer interfaces require authors to explicitly define and structure their knowledge as they expressed it. These demands are premature and cognitively demanding, focussing the author on the interaction, not the idea. Consequently, many authors reject computer use and return to pen and paper.

Pen and paper constitute a mature and familiar interface for the capture and refinement of knowledge. Analyzing and identifying specific reasons for the use of paper will guide the development of computer interfaces that overcome current interface restrictions [8]. Introducing computer assistance to traditional pen and paper tasks can increase the efficiency of these processes [5].

The resemblance of pen-based computers to pen and paper suggests that these devices are ideal for implementing paper-like interfaces. Unfortunately pen-based computers currently fall a long way short of this goal. Pen-based computers commonly employ traditional WIMP-style interfaces, use handwriting recognition or on screen keyboards to replace typing, and follow the pen tip round the screen to simulate a mouse. The resulting interface is often cumbersome. Handwriting is slower than typing, recognition is far from perfect and requires frequent mediation, and a user's hand may conceal on-screen information when pointing with the pen. These interfaces also ignore the natural strengths of the pen: precision control; immediacy of expression; and direct manipulation.

Successful pen-based interfaces will be significantly different from familiar desktop computer interfaces. They will exploit the natural characteristics of the pen. Special attention must be given to *when* and *how* such interfaces will be used, or they will remain subordinate to paper.

## Initial Findings

The creative phases, of capturing and structuring information and ideas, are not supported by commercial computer interfaces [4][9]. The capture of information is characterized by sketching and note-making activities, whether assimilating new information or transcribing mental knowledge [1][6]. The representations generated embody the most important concepts of the information and delay the specification of explanatory detail. Before an author invests time in producing a detailed description of his knowledge, he will try out, explore, and restructure his initial ideas and arguments. Commitment to a particular expression of his knowledge may come late in the authoring process.

We require computer interfaces that allow us to express and capture our ideas immediately. We need the ability to build on these ideas incrementally, reworking, restructuring, and reinterpreting them throughout the authoring process. We want to add detail and define formalisms step by step until we have captured and presented our knowledge in a refined and ordered way. This is the concept of "informal interaction".

Informal interaction was first defined by Moran et al. in 1997 [7]. Informal interaction is characterized by a modeless interface which combines freedom of expression with structured editing capabilities, and does not overtly engage the user in recognition mediation. Informal interaction occupies a conceptual void between free-form and formal interaction. Free-form interaction covers applications like a simple paint package that handles user input only on the level of a coloured bitmap of pixels, it may have many painting tools but their use has no semantic effect. Applications such as word processors are representative of formal interaction. Words are all members of a language set, and have other attributes and relationships to each other, such as titles, lists, and numbered paragraphs.

Informal interaction is founded on the assumption that during the phases of capturing and refining knowledge, an author can perform the tasks they want to with only a small set of general formalisms over the information they have expressed. As an author interacts with their creation, they can add more detail, and define and redefine their own formalisms. As the level of formalism increases, so does the logical merit of the information. Eventually a point is reached where a creation can be automatically imported into a formal application. This concept has been recently demonstrated in work by Gross and Do [3].

There appears to be a consensus in literature that the pen is the most natural tool for informal interaction, including both diagramming [2][3][5][6][7], and writing tasks [1]. Research has established pen-based informal interaction as a sound concept. This has taken place within the confines of specific applications. However there remain a number of challenges to be tackled to create a cohesive system that implements the informal interaction paradigm. These include, but are not limited to:

- Identifying and implementing basic sets of formalisms;
- Providing unobtrusive mechanisms to define new formalisms;
- Facilitating the transitions between levels of formality;
- Providing feedback to the user on the machine's current interpretation.

## Future Work

Future work will be focussed on informal interaction with handwritten text. This will enhance diagram based applications, handling labels and annotations, and enable meaningful interaction with handwritten documents. Work already underway has assumed a notebook metaphor to guide the identification of basic formalisms. The word has been isolated as the basic handwritten textual entity. Implicit functions include moving, entering, and deleting words. The process of grouping words will define new formalisms over the data and mechanisms for doing this will be investigated.

It has been noted that text written directly onto a computer screen can be difficult to read. Handwriting recognition is not an option here since mediating the recognition process will distract the author and working with handwriting is inline with the informal nature of the interaction paradigm. Consequently work is progressing in the beatification of handwritten text. Experiments will provide a quantitative evaluation of algorithms to improve handwritten text legibility. Word segmentation algorithms will also be evaluated. Feedback to facilitate interaction may be given by applying legibility enhancement to segmented words.

Finally, working with large numbers of handwritten documents will require solutions to problems such as indexing, searching, and sorting. Algorithms that search handwritten text currently rely on searching an entire corpus for best-matches, or on recognizing words and then searching alphabetically. Both approaches are impractical on large amounts of handwritten data. Experiments are planned to evaluate algorithms that classify and match handwritten words. This will allow standard sorting and searching techniques to be employed on handwritten data.

## References


[1] Patrick Chiu and Lynn Wilcox. A Dynamic Grouping Technique for Ink and Audio Notes. In *UIST 1998, Symposium on User Interface Software and Technology*, pages 195–202. ACM, 1998.

[2] Christian Heide Damm, Klaus Marius Hansen, and Michael Thomsen. Tool Support for Cooperative Object-Oriented Design: Gesture Based Modelling on an Electronic Whiteboard. In *CHI 2000 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 518–525. ACM SIGCHI, 2000.

[3] Mark D. Gross and Ellen Yi-Luen Do. Drawing on the Back of an Envelope: a

framework for interacting with application programs by freehand drawing. *Computers & Graphics*, 24:835–849, 2000.

[4] Frank M. Shipman III and Catherine C. Marshall. Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representaions in Interactive Systems. *Computer Supported Cooperative Work*, 8:333–352, 1999.

[5] James A. Landay and Brad A. Meyers. Sketching Interfaces: Toward More Human Interface Design. *Computer*, 34(3):56–64, March 2001.

[6] James Lin, Mark W. Newman, Jason I. Hong, and James A. Landay. DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. In *CHI 2000 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 510–517. ACM SIGCHI, 2000.

[7] Thomas P. Moran, Patrick Chiu, and William van Melle. Pen-Based Interaction Techniques For Organizing Material on an Electronic Whiteboard. In *UIST 1997, Symposium on User Interface Software and Technology*, pages 45–54. ACM, 1997.

[8] Abigail Sellen and Richard Harper. Paper as an Analytic Resource for the Design of New Technologies. In *CHI 1997 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 319–326. ACM SIGCHI, 1997.

[9] I. M. Verstijnen, R Stuyver, J. M. Hennessey, C. C. van Leeuwen, and R. Hamel. Considerations for Electronic Idea-Creation Tools. In *CHI 1996 Conference Companion, Conference on Human Factors in Computing Systems*, pages 197–198. ACM SIGCHI, 1996.

# Informal Interaction

An interaction paradigm for the digital pen

*Timothy S. Butler*

Science and Technology Research Centre,
University of Hertfordshire, Hatfield, AL10 9AB, UK
http://homepages.feis.herts.ac.uk/~bt7al/
t.s.butler@herts.ac.uk

**Abstract**

This paper highlights the ways in which current computer interfaces impair the capture and refinement of knowledge. It defines an interaction paradigm "informal interaction", and proposes that pen-based interfaces designed around the concept of informal interaction have the potential to overcome the current impediments. Finally, it identifies specific areas of research required to develop informal interaction pen-based interfaces.

## Background

Knowledge work is dominated by the use of paper especially when capturing, organizing, and refining information [6, 8]. Recent studies show that despite the recognized benefits of using a computer such as prototyping, document duplication, and error checking, computer literate professionals still prefer to capture and refine their ideas on paper. There is a gulf between the way humans express and manipulate ideas, and how computer interfaces require users to structure and interact with data [4].

An author's primary task is to express ideas. Computer interfaces require authors to explicitly define and structure their knowledge as they express it. The structuring is in terms of producing a spacial arrangement of units laid out on a two-dimensional sheet, either as a sequence (e.g. a sequence of words forming the lines and paragraphs of a document) or a spatial arrangement (e.g. a block diagram) or a combination of these. Such demands are premature and cognitively distractive, as they focus the author on externalising his ideas, rather than merely capturing them. Consequently, many authors reject computer use and return to pen and paper.

Pen and paper constitute a mature and familiar interface for the capture and refinement of knowledge. Analyzing and identifying specific reasons for the use of paper will guide the development of computer interfaces that overcome current interface restrictions [8]. Introducing computer assistance to traditional pen and paper tasks can increase the efficiency of these processes [5].

The resemblance of state-of-the-art pen-based computing to pen and paper as a tool of recording and interacting with knowledge is very superficial, but it still suggests that these devices can present paper-like interfaces to the user. The problem is that pen-based computers commonly employ traditional WIMP-style interfaces, use handwriting recognition or on-screen keyboards to replace typing, and follow the pen tip round the screen to simulate a mouse. The resulting interface is often cumbersome. Handwriting is slower than typing, recognition is far from perfect and requires frequent mediation, and a user's hand may conceal on-screen information when pointing with the pen. These interfaces also ignore the natural strengths of the pen: precision control, immediacy of expression and direct positioning.

Successful pen-based interfaces will be significantly different from familiar desktop computer interfaces. They will exploit the natural characteristics of the pen. When designing these interfaces, attention must be given to *when* and *how* such interfaces will be used, or they will remain subordinate to paper.

### Informal Interaction

The creative phases, of capturing and structuring information and ideas, are not supported by commercial computer interfaces [4, 9]. The capture of information is characterized by sketching and note-making activities, whether assimilating new information or transcribing mental knowledge [1, 6]. The representations generated embody the most important concepts of the information and delay the specification of explanatory detail. Before an author invests time in producing a detailed description of his knowledge, he will try out, explore, and restructure his initial ideas and arguments. Commitment to a particular expression of his knowledge may come late in the authoring process.

We require computer interfaces that allow us to express and capture our ideas immediately. We need the ability to build on these ideas incrementally, reworking, restructuring, and reinterpreting them throughout the authoring process. We want to add detail and define formalisms step by step until we have captured and presented our knowledge in a refined and orderly fashion. This is the concept of "informal interaction".

Informal interaction was first defined by Moran et al. in 1997 [7]. Informal interaction is characterized by a modeless interface which combines freedom of expression with structured editing capabilities, and does not overtly engage the user in recognition mediation. Informal interaction occupies a conceptual void between free-form and formal interaction. Free-form interaction covers applications like a simple paint package that handles user input only on the level of a coloured bitmap of pixels, it may have many painting tools but their use has no semantic effect in the sense that the computer perceives the canvas as one unstructured bitmap at all times. Applications such as word processors are representative of formal interaction. The image on the screen is a mere graphic representation of an underlying formal structure, which is a list of words with their relative positioning and formal attributes, such as font, size, information content, etc. Several "views" are supported by the wordprocessing software on the same underlying structure, and it is that structure that the user is interacting with, rather than its graphical image. The latter is there to serve as a perceptual conduit, which facilitates the user's mental connection with the underlying structure manipulated by the computer.

Informal interaction is founded on the assumption that during the phases of capturing and refining knowledge, an author can perform the tasks they want to with only a small set of general formalisms over the information they have expressed. As an author interacts with their creation, they can add more detail, and define and redefine their own formalisms. As the level of formalism increases, so does the logical merit of the information. Eventually a point is reached where a creation can be automatically imported into a formal application. This concept has been recently demonstrated in work by Gross and Do [3].

There appears to be a consensus in literature that the pen is the most natural tool for informal interaction, including both diagraming [2, 3, 5, 6, 7], and writing tasks [1]. Research has established pen-based informal interaction as a sound concept. This has taken place within the confines of specific applications. However there remain a number of challenges to be tackled to create a cohesive system that implements the informal interaction paradigm. There are four areas of development that need to be addressed to do this:

1. To develop the digital ink data type. This may include: pen response models – so an author feels like they are using a real pen; ink rendering algorithms – so the image on a computer screen looks like a real pen; and a well defined data format so that digital ink can be stored and distributed.

2. To identify and implement basic sets of formalisms that can be applied by a computer to handwritten data. This will allow authors to interact with ink in a meaningful way. As interaction progresses formalisms can be assigned to, and unassigned from, the data. In this way the system builds up a model of the author's interpretation of the ink.

3. To provide feedback to the author of the formalisms that have been assigned, and to enable him to correct them if he wishes. This must be done in a non-intrusive way, and must not detract from the primary task of information capture.

4. To enable automatic transitions between partially structured, partially formalized, handwritten information and data in existing formal computer applications.

**Current Work**

Current work is focussed on informal interaction with handwritten script. This will enhance diagram based applications, handling labels and annotations, and enable meaningful interaction with handwritten documents.

Work is already underway to develop the digital ink data type. The effect of pen models and rendering methods on the legibility of handwritten script is being evaluated. This will also lead to an estimation of the ideal screen resolution for working with handwritten script.

Some basic formalisms have also been identified. The word has been isolated as the base handwritten script entity. Initial work is concerned with segmenting pages of script into groups of ink-strokes representing individual words. These elements will facilitate interaction with ink and allow further higher level formalisms to be derived.

Finally, working with large numbers of handwritten documents will require solutions to problems such as indexing, searching, and sorting. Algorithms that search handwritten text currently rely on searching an entire corpus for best-matches, or on recognizing words and then searching alphabetically. Both approaches are impractical on large amounts of handwritten data. Experiments are planned to evaluate algorithms that classify and match handwritten words. This will allow standard sorting and searching techniques to be employed on handwritten data.

## References


[1] Patrick Chiu and Lynn Wilcox. A Dynamic Grouping Technique for Ink and Audio Notes. In *UIST 1998, Symposium on User Interface Software and Technology*, pages 195–202. ACM, 1998.

[2] Christian Heide Damm, Klaus Marius Hansen, and Michael Thomsen. Tool Support for Cooperative Object-Oriented Design: Gesture Based Modelling on an Electronic Whiteboard. In *CHI 2000 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 518–525. ACM SIGCHI, 2000.

[3] Mark D. Gross and Ellen Yi-Luen Do. Drawing on the Back of an Envelope: a framework for interacting with application programs by freehand drawing. *Computers & Graphics*, 24:835–849, 2000.

[4] Frank M. Shipman III and Catherine C. Marshall. Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representaions in Interactive Systems. *Computer Supported Cooperative Work*, 8:333–352, 1999.

[5] James A. Landay and Brad A. Meyers. Sketching Interfaces: Toward More Human Interface Design. *Computer*, 34(3):56–64, March 2001.

[6] James Lin, Mark W. Newman, Jason I. Hong, and James A. Landay. DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. In *CHI 2000 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 510–517. ACM SIGCHI, 2000.

[7] Thomas P. Moran, Patrick Chiu, and William van Melle. Pen-Based Interaction Techniques For Organizing Material on an Electronic Whiteboard. In *UIST 1997, Symposium on User Interface Software and Technology*, pages 45–54. ACM, 1997.

[8] Abigail Sellen and Richard Harper. Paper as an Analytic Resource for the Design of New Technologies. In *CHI 1997 Conference Proceedings, Conference*

*on Human Factors in Computing Systems*, pages 319–326. ACM SIGCHI, 1997.

[9] I. M. Verstijnen, R Stuyver, J. M. Hennessey, C. C. van Leeuwen, and R. Hamel. Considerations for Electronic Idea-Creation Tools. In *CHI 1996 Conference Companion, Conference on Human Factors in Computing Systems*, pages 197–198. ACM SIGCHI, 1996.

# ClearPen: Improving the Legibility of Handwriting

**Timothy S. Butler**
University of Hertfordshire
Department of Computer Science
Hatfield, AL10 9AB, UK
+44 1707 286362
t.s.butler@herts.ac.uk

## ABSTRACT
We describe the application of a pen model, and sub-pixel addressing (ClearPen), to render handwriting on an LCD display. This technique is shown to improve the legibility of handwriting. ClearPen can increase the viability of working with handwriting on a computer. This has direct significance to TabletPC applications such as note taking or annotating documents.

### Keywords
Handwriting, legibility, pen model, sub-pixel addressing

## INTRODUCTION
Traditionally, pen-based computing applications have implemented a strategy of recognizing handwritten script and displaying this input as typed text. This style of interface can be cumbersome and difficult to use. A number of recent research projects[1, 2, 3, 6] have argued that preserving handwritten input is, in certain situations, more preferable to recognition. They have all demonstrated applications which highlight this. As pen-based computers, such as TabletPCs, become more widespread[1] we are likely to see similar applications become mainstream products.

Handwriting is an ideal mode of communication for numerous tasks. Tasks that are traditionally performed with pen and paper. These tasks are typically characterized by capturing loosely structured ideas, immediately expressing thoughts, or forming representations of important concepts without specifying intricate details. For example: note taking[1, 6]; annotating slides or documents[3]; or design work[2].

Despite the benefits computer assistance can bring to pen-based tasks, the general acceptance of such applications is hampered by many things. Not least, the legibility of handwritten script on a computer screen. Computer screens that are too small with insufficient pixel resolution result in script that is either barely legible, or too large to warrant reading any significant amount.

[1] http://www.microsoft.com/windowsxp/tabletpc/


*CHI 2003*, April 5-10, 2003, Ft. Lauderdale, Florida, USA.
ACM 1-58113-637-4/03/0004

In this light we have chosen to investigate the effect of horizontal resolution enhancement on the legibility of handwritten script. Horizontal enhancement was chosen because it is likely that the definition of salient features in handwriting[4] will be improved by horizontal resolution enhancement.

## METHOD
Resolution enhancement in one direction can be achieved by exploiting sub-pixel addressing on LCD displays, which are typically used in TabletPCs. Sub-pixel addressing gives a three-fold resolution enhancement in the direction of the scan-lines of the LCD display. This technique is common knowledge, but has not been applied to handwriting before, only type fonts[2].

As well as employing sub-pixel addressing techniques, we have implemented a pen model that mimics the characteristics of real pens. These algorithms are used together to produce handwritten script on a computer screen, similar to its appearance on paper. We call our enhancements ClearPen.

### ClearPen Model
Our pen model algorithm is model based on observation, similar to the work of Sousa and Buchanan[5], except that we are modeling a fountain pen alone rather than pencil, paper, and other artistic materials.

The pen model algorithm operates at a geometric level by modeling the volume of ink flowing from the pen nib to the page. The ink volume is represented by an "intensity" value. The more ink, the higher the intensity of the pen trace.

The volume of ink deposited on the page depends both on the speed the pen is moving at, and the pressure applied to the tip. The pen model generates a series of consecutive line segments describing the path of each pen stroke at a resolution far higher than that of the computer screen.

### ClearPen Rendering
ClearPen rendering recognizes that on an LCD panel, scan-lines are composed of individually addressable colour component pixels (sub-pixels) in an ordered sequence, usually red–green–blue. Each screen pixel is formed from a triplet of adjacent sub-pixels.

[2] http://grc.com/cleartype.htm

The assignment of sub-pixels to pixels is static, however as each sub-pixel is individually addressable, any three adjacent sub-pixels can be combined to give the appearance of a full pixel. This technique allows us to position "perceptual pixels" at three times the normal precision of the LCD display.

Instead of being rendered directly onto the screen, each line segment is rendered onto a grid at nine times the display resolution. The intensity values are interpolated along the line segment. Each grid point along the line segment becomes the centre of a "Tip-Filter". The Tip-Filter is a two-dimensional filter representing a hemi-ellipsoidal pen tip. At each point along the line the intensity value is dissipated over the area covered by the filter. The filtered line segment is then mapped onto an "Intensity Grid". An Intensity Grid is three times the display screen size. It is populated by summing each square of nine intensity values from the first grid into the corresponding cell of the Intensity Grid.

Finally, columns of three intensity values are averaged and converted into sub-pixel colour components. Each group of three sub-pixels forms one coloured pixel. That colour is finally rendered onto the corresponding screen pixel.

The result of ClearPen rendering is a sharp image of a fine line pen strokes. When viewed at a normal distance the colour components in adjacent pixels combine to form a smooth black line. The ClearPen model and rendering is simple enough to process and render handwriting in real time, on a 750MHz Pentium III PC, with no discernible lag.

## EVALUATION

Two experiments were conducted. The first experiment assessed the relationship between horizontal rendering resolution and legibility, by measuring the recognition rate of individual tachistoscopically displayed words. The second experiment assessed user preference for reading ClearPen against two alternative rendering methods, anti-aliasing and a "nearest pixel" plot, using a questionnaire. All three methods used the same pen model.

The first experiment showed a strong correlation between pixel width and recognition rate ($F(4, 80) = 5.481, p = .001$). A quadratic relationship ($p < .001$) depicted a recognition threshold of $\approx 80\%$, reached at an equivalent of 170 dpi viewed from 450mm, which dropped off rapidly as screen resolution decreased. This was in agreement with formal and informal observations. The responses in the second experiment, analysed using Spearman's Rank Order Correlation, suggest that the subjects perceived a difference in the three different rendering methods ($p < .05$), and that their preference followed increasing resolution ($p < .001$).

## CONCLUSION

ClearPen is capable of improving the legibility of handwritten script displayed on an LCD screen. The technique improved the horizontal resolution of the 85 dpi display used in the experiment to around 250 dpi, well within the recognition rate threshold. This enhancement increases the viability of reading handwritten script on a computer, including both TabletPCs ($\approx$ 120 dpi) and Handheld PCs ($\approx$ 100 dpi).

ClearPen does however have a number of limitations. Firstly, script must be written along the direction of a scan-line. Informal observation has shown that legibility is not greatly impacted by vertical resolution enhancement. Secondly, the technique involves sacrificing colour for resolution. In applications where colour or freedom of orientation are important, ClearPen may not be suitable.

The legibility of handwritten script displayed on an LCD computer screen is improved by ClearPen. People are able to perceive, and prefer, this improvement. Reading handwriting on a computer is as feasible as it is on paper and need not be hampered by poor script legibility.

## REFERENCES


1. R. C. Davis, J. A. Landay, V. Chen, J. Huang, R. B. Lee, F. C. Li, J. Lin, C. B. Morrey, B. Schleimer, M. N. Price, and B. N. Schilit. NotePals: Lightweight Note Sharing by the Group, for the Group. In *CHI 1999 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 338–345. ACM SIGCHI, 1999.

2. J. Lin, M. W. Newman, J. I. Hong, and J. A. Landay. DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. In *CHI 2000 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 510–517. ACM SIGCHI, 2000.

3. B. N. Schilit, G. Golovchinsky, and M. N. Price. Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations. In *CHI 1998 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 249–256. ACM SIGCHI, 1998.

4. L. Schomaker and E. Segers. Finding features used in the human reading of cursive handwriting. *International Journal on Document Analysis and Recognition*, (2):13–18, 1999.

5. M. C. Sousa and J. W. Buchanan. Observational Models of Graphite Pencil Materials. *Computer Graphics Forum*, 19(1):27–49, 2000.

6. L. D. Wilcox, B. N. Schilit, and N. Sawhney. Dynomite: A Dynamically Organized Ink and Audio Notebook. In *CHI 1997 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 186–193. ACM SIGCHI, 1997.

# Improving the Legibility of Digital Ink

TIMOTHY S. BUTLER, TREVOR BARKER,
and ALEX SHAFARENKO
University of Hertfordshire

**Abstract**

The widespread adoption of handwriting-based interfaces, now made feasible by TabletPCs, will depend on their usability as well as their utility. We present two experiments which examine the effect of horizontal rendering resolution on the legibility of digital ink. We start by describing a pen model and rendering method which mimics the behavior of real-world pens to produce an authentic-looking pen-trace on screen. We then analyze published work related to assessing the legibility of handwritten script. From this evidence we propose and implement two methods for measuring the legibility of digital ink. The first method measures the human recognition rate of isolated words displayed tachistoscopically. The second method assesses the reader response, to a number of subjective factors, after reading short passages of script from the computer screen. The results of the experiments show that increasing horizontal rendering resolution increases both the legibility of, and preference for working with, handwritten script. A computer screen resolution of 170 dpi is proposed as being optimal for reading handwriting.

## 1 Introduction

This paper describes two experiments that examine the effect that rendering resolution has on the legibility of digital ink. This builds on previous work which used a speed of reading test [2] and an experiment employing tachistoscopic display [3].

Traditionally, pen-based computing applications have implemented a strategy of recognizing handwritten script and displaying this input as typed text. This style of interface can be cumbersome and difficult to use. A number of recent research projects [4, 6, 10, 17] have argued that preserving handwritten input is, in certain situations, more preferable to recognition. They have all demonstrated applications which highlight this. As pen-based computers, such as TabletPCs, become more widespread we are likely to see similar applications become mainstream products.

In this light, we are researching techniques to enhance interaction with digital ink. In particular we are examining how rendering methods, the processes used to describe a pen-trace on screen, can enhance the legibility of handwritten script. Legibility enhancement should increase a user's preference to work with handwritten script.

We have chosen to examine the effects of horizontal resolution enhancement on legibility. When reading handwriting, humans rely on identifying: vertical down-strokes; crossings; and points of high curvature, particularly at the beginning and end of each word [11]. We chose horizontal resolution enhancement, rather than vertical, on the assumption that the definition of these components will be improved more by horizontal enhancement. We have implemented a pen-trace rendering algorithm, ClearPen, which exploits sub-pixel addressing on LCD displays. This gives a three-fold resolution enhancement in the direction of the scan-lines of the LCD display.

# 2 Hypotheses

To examine the effect of horizontal resolution enhancement on legibility we have two hypotheses which are addressed in two separate experiments. These are:

1. Improving the horizontal rendering resolution of a handwritten word, displayed on a computer screen, enhances its legibility.

2. A user will perceive the effect of horizontal resolution enhancement as beneficial, and will prefer reading script rendered in this manner.

# 3 Rendering Algorithms

In our experiments, handwritten data was rendered on-screen using three different rendering methods: ClearPen (high resolution); anti-aliased pen (medium resolution); and pixel pen (low resolution).

When rendering a pen trace the handwritten data samples, collected using a pen digitizer, are first processed through a pen model. This models the volume of ink flowing from the pen nib to the page. The model produces line segments, describing the path and intensity of the pen trace. Each rendering method takes line segments and renders them on-screen.

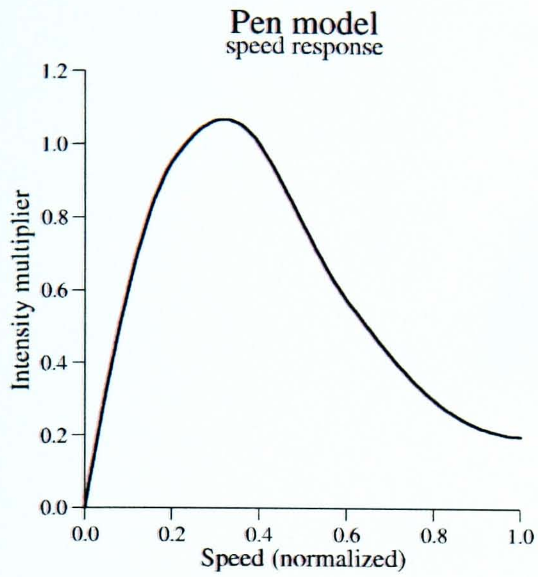



Figure 1: Pen model - speed response

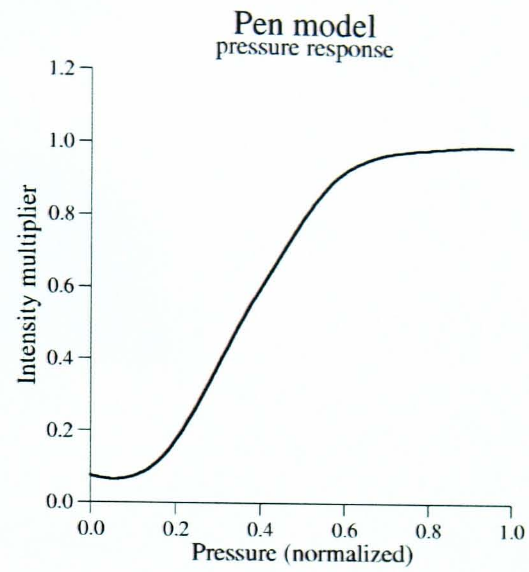



Figure 2: Pen model - pressure response

## 3.1 Pen Model

Our pen model algorithm is an observational model, similar to the work of Sousa and Buchanan [12], except that we are modeling a fountain pen alone rather than pencil, paper, and other artistic materials.

The pen model algorithm operates at a geometric level by modeling the volume of ink flowing from the pen nib to the page. The ink volume is represented by an "intensity" value. The more ink, the higher the intensity of the pen trace.

The volume of ink deposited on the page depends both on the speed the pen is moving at (Figure 1), and the pressure applied to the tip (Figure 2). The pen model generates a series of consecutive line segments describing the path of each pen stroke. Each line segment consists of two triplets detailing the x-coordinate, y-coordinate, and an intensity value, at either end of the segment.

## 3.2 Pixel Pen

The pixel pen simply converts the line segment coordinates to screen coordinates using a nearest neighbor division. A one-pixel-wide line is drawn between the two end points. The colour of each pixel along the line is determined by interpolating the intensity values along the line segment, and converting them to a colour. The more intense the pixel, the darker the pixel

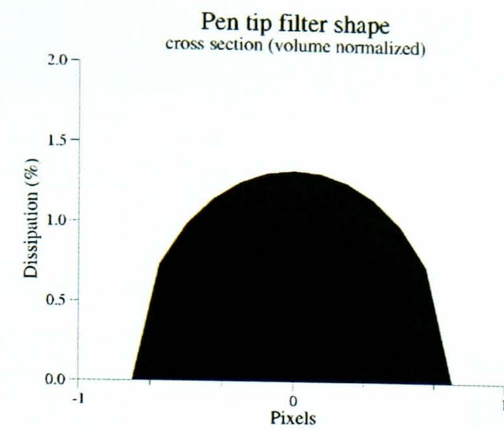



Figure 3: Pen tip filter shape

colour.

### 3.3 Anti-aliased Pen

The anti-aliased pen and ClearPen algorithms are more complicated, but similar to each other. Instead of being rendered directly onto the screen, each line segment is rendered onto a grid at nine times the display resolution. As in the case of the pixel pen, the intensity values are interpolated along the line segment.

Instead of filling a one-point-wide line with intensity values, each grid point becomes the centre of a "Tip-Filter". The Tip-Filter is a two-dimensional filter representing a hemi-ellipsoidal pen tip. At each point along the line the intensity value is dissipated over the area covered by the filter (Figure 3). The filtered line segment is then mapped onto the "Intensity Grid". The Intensity Grid is three times the display resolution. It is populated by summing each square of nine intensity values from the first grid into the corresponding cell of the Intensity Grid.

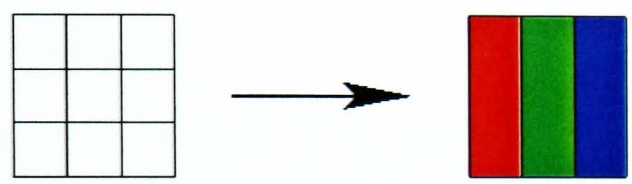
The process of translating the values stored in the Intensity Grid to screen pixels is where the ClearPen and antialiasing algorithms differ. The antialiasing algorithm averages each square of nine intensity values, and converts them to a shade of grey representing their intensity. That colour is then rendered onto the corresponding screen pixel.

### 3.4 ClearPen

ClearPen rendering recognizes that on an LCD panel, scan-lines are composed of individually addressable colour component pixels (sub-pixels) in an ordered

Each scan-line on an LCD display is composed of coloured subpixels.

Figure 4: An LCD panel scan-line



Each square on the Intensity Grid is mapped to a colour component sub-pixel. The columns are averaged before being mapped onto the corresponding sub-pixel.

Figure 5: Intensity to pixel mapping

sequence, usually red–green–blue (Figure 4). Each screen pixel is formed from a triplet of adjacent sub-pixels.

The assignment of sub-pixels to pixels is static, however as each sub-pixel is individually addressable, any three adjacent sub-pixels can be combined to give the appearance of a full pixel. This technique allows us to position "perceptual pixels" at three times the normal precision of the LCD display.

In anti-aliased rendering, squares of nine intensity values are averaged and converted to shades of grey. In ClearPen rendering, columns of three intensity values are averaged and converted into sub-pixel colour components (Figure 5).

The result of ClearPen rendering can be seen in Figure 6. The sub-pixel order is red–green–blue. Of particular interest are the vertical pen strokes. The screen pixels lit on their left-hand side (those on the left side of a stroke, as black is "unlit") appear yellow-brown since the (right-most) blue component is not lit. Likewise, those pixels lit on their right-hand side, on the right side of each vertical stroke, appear turquoise-blue since the (left-most) red component is not lit. When viewed at normal magnification the adjacent colour components combine to form a black pixel in between the actual screen pixels.

The sub-pixel technique is common knowledge, but has not been applied to handwriting before, only type fonts[1].

[1] http://grc.com/cleartype.htm

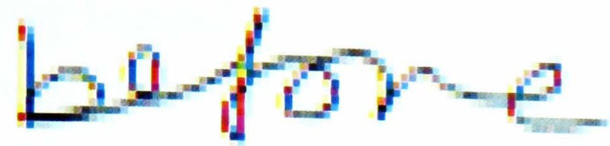


Figure 6: ClearPen rendering

# 4 Related Work

We need to measure the legibility of handwritten script. This has traditionally only been done in educational fields, by comparison against a set of graded samples. We however require measurements that are more objective and more descriptive.

## 4.1 Legibility

Legibility is the term which describes the effect of the spatial aspects of a text on its readability. Legibility is affected by a number of different graphical properties. These include but are not limited to:

1. Letter shape and word form;
2. Spacing between letters, words, and lines;
3. Line length and letter size;
4. Contrast of words against a page.

Tinker, in his 1964 book, *The Legibility of Print* [14] states that:

> "Optimal legibility of print, therefore, is achieved by a typographical arrangement in which shapes of letters and other symbols, characteristic word forms, and all other typographical factors such as type size, line width, leading, etc., are coordinated to produce comfortable vision and easy and rapid reading with comprehension."

In essence, improving the legibility of a text will decrease the strain and fatigue of a reader, as well as facilitate efficient and accurate reading.

## 4.2 Measuring Legibility

There are a number of approaches to measuring legibility. These include:

1. Speed of reading a passage [1];

2. Speed of a search task [7];

3. Word recognition rate [13, 18];

4. Oculomotor measurements of eye fatigue [16];

5. Subjective preferences [1, 16, 18].

The use of speed of reading measures, although preferred for measuring the legibility of print [14], has not established significant differences in the legibility of typefaces on a computer screen [1, 16]. Our own pilot of this method to measure the legibility of on-screen handwritten script also failed to detect any significant difference [2].

Measures that have been employed successfully include: The speed of a search task; the word recognition rate; and subjective preferences.

To the best of our knowledge, the only experiment that has employed a search task as a legibility measure, required volunteers to locate random 4-letter strings among 100 similar distractors [7]. We decided that this method was unsuitable for our needs since, obviously, trying to read arbitrary combinations of letters written by hand is a very uncommon activity. This would likely depend on different types of visual perception than normal reading.

Established research shows that humans perceive word forms more readily than they do individual letters or non-words [8, 9]. Coupled with the fact that letters within handwritten words are often joined as part of a single unit, and that individual words are free from any contextual clues as to their identity, word recognition rate would appear to be a good indicator of legibility.

### 4.2.1 Word Recognition

Word recognition has been used successfully to measure differences in legibility, including measurement of the effect of resolution on the legibility of typed text [18]. Word recognition is commonly measured through tachistoscopic (short) exposure [13] although there is some evidence that differences in word recognition can be measured without the use of a tachistoscope [8, 18]. We found in a previous experiment [2] that during a timed reading task, volunteers achieved higher recognition rates with higher rendering resolutions, although there was no significant difference in their overall reading speeds.

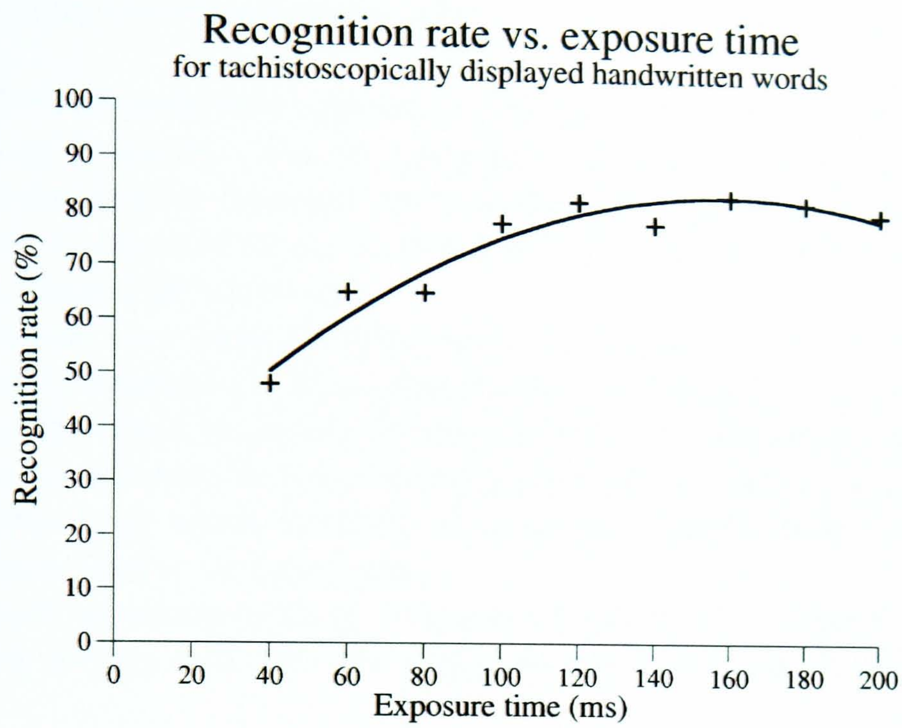



Figure 7: Recognition rate against exposure time

Tinker [14] notes that the short exposure method is particularly useful for measuring the relative legibility of different printed symbols. Since we have established: That recognition rate shows differences in legibility; that handwritten words are read as units; and that the method has been used to measure the legibility of printed material on screen of letters, words, and headlines; we are confident that measuring recognition rate through tachistoscopic exposure is a good indicator of the legibility of handwritten script displayed on a computer screen.

Our previous work [3], using a single author, revealed a recognition rate of handwritten words without any contextual information of about 80%. This recognition rate starts to deteriorate at an exposure time of 100–120ms (Figure 7). This is in agreement with Timmers et al. [13] who state that a stimulus presentation time of 100ms, being shorter than the visual reaction time, allows one single fixation and precludes any systematic influence on the experimental results.

### 4.2.2 Subjective Preferences

Subjective preferences, where volunteers answer questionnaires [1, 16, 18] or rank display modes in order of preference [16] also seem to consistently reveal differences in reader preference when comparing different modes of display or fonts.

Subjective preference questionnaires have employed Likert scales with either 5 or 9 points. Not all questions established significant differences. Experimenters have focussed on aspects of the clarity of text [1], and on indicators of physical strain on the reader [16]. Experiments have used, on average, around 20 volunteers.

Particular care must be taken when using subjective measures. Firstly responses to questions will depend on the volunteers' interpretation of the meaning. If certain questions are too ambiguous, volunteers may respond in reference to different factors. Secondly, care should be taken when drawing conclusions from small numbers of volunteers, particularly if they are all drawn from similar backgrounds.

Overall, measurements of subjective preferences complement objective measures, but on their own the results should be treated with caution.

# 5 Method

The experiments commenced with the collection of samples of handwritten script. These were rendered using the methods described previously. The experiment room and testing software were prepared, volunteers recruited, and experiments run to test the hypotheses.

## 5.1 Handwritten Material

To reduce variability in script style, handwriting samples were collected from a single writer. The samples were copied by the author from typed sheets. The author was instructed to write with the knowledge that other people would have to read what they had written. Every sample was written on paper using a Wacom Intuos Inking Pen. The writing paper was attached to a Wacom Intuos digitizer tablet which sampled the pen movement at around 94Hz. The information sampled includes:

- Movement data, at 100 points per millimetre.
- Pressure data, at 1024 levels.
- A time stamp.
- Pen tilt data.

These data were logged in files and were then rendered to produce the required images of the handwriting on-screen.

Figure 8: "across" rendered with five different pixel widths

### 5.1.1 Experiment 1

For the first experiment, measuring the recognition rate of individual handwritten words in tachistoscopic display, 199 individual words were collected. These are the most common six and seven letter words in the English language as listed in the LOB corpus [5].

To create equivalent images of identical words at different horizontal resolutions, the pen data coordinates for each word were mathematically scaled to $\frac{1}{1}$, $\frac{1}{1.5}$, $\frac{1}{2}$, $\frac{1}{2.5}$, and $\frac{1}{3}$ of their original horizontal dimensions. These data were then rendered into word images using the anti-aliased rendering method. Finally the word images were graphically scaled back to their original size. This produced words with pixel widths of 0.30, 0.45, 0.60, 0.75, 0.90mm respectively. The result can be seen in Figure 8. On average the rendered words are 75 pixels in length and 23 pixels high.

According to Tinker, brightness contrast between print and paper has an affect on legibility. This is confirmed by Timmers et al. [13] who found that, when presented parafoveally (at a visual angle of $\pm 1.5°$), the legibility of words decreased significantly with decreasing contrast. Words presented foveally ($0°$ visual angle) were much less affected by decreasing contrast.

Although we presented our words foveally, we took care to preserve the contrast of the word images against the screen background.

The experiment used 100 words, 5 groups of 20 drawn from the 199 samples. Each of the word groups (A–E) consisted of 10 six-letter words and 10 seven-letter words. Within each group, all words were rendered at the same resolution.

There were also 5 volunteer groups (1–5). Each group saw the 100 words in a different random order. The rendering resolution used for the word groups shown to each each of the volunteer groups was varied so that, although each volunteer saw the same 100 words, no group saw the same word *image* as any other group. Over the five volunteer groups all 100 words were displayed in each of the 5 resolutions. This is illustrated in Table 1.

| | | Word Group | | | | |
|---|---|---|---|---|---|---|
| | | A | B | C | D | E |
| | 1 | 0.30 | 0.45 | 0.60 | 0.75 | 0.90 |
| **Volunteer** | 2 | 0.90 | 0.30 | 0.45 | 0.60 | 0.75 |
| **Group** | 3 | 0.75 | 0.90 | 0.30 | 0.45 | 0.60 |
| | 4 | 0.60 | 0.75 | 0.90 | 0.30 | 0.45 |
| | 5 | 0.45 | 0.60 | 0.75 | 0.90 | 0.30 |

Table 1: Pixel widths (mm) for each volunteer group

#### 5.1.2 Experiment 2

The second experiment tested volunteers' preference for reading script rendered with an enhanced horizontal rendering resolution. Three samples of script were rendered using pixelated, anti-aliased, and ClearPen rendering methods. The script collected consisted of twenty random items selected from the *Tinker Speed of Reading Test*[2] [15]. Ten items were collected on each page. Test items were copied into boxes on the paper form to constrain the size of the handwriting, and also keep the items clearly distinct. Three columns of script, each consisting of five Tinker test items were selected for the experiment.

For this second experiment, volunteers were re-allocated into 6 volunteer groups. Each column of test items was presented to the volunteers as a single page. Volunteers in each group saw the 3 pages in the same order. The rendering method used to render each page was varied between groups, so that each group saw the rendering methods in one of the 6 different orders.

The Tinker Test has been used in previous research on the legibility of text [1, 2]. Although designed to measure speed of reading, we used the test simply to engage our volunteers with the handwritten script so they could answer a questionnaire.

The handwriting was presented at 70% of its original size. This scaling allowed us to render the script at a size comparable to a font size of 12 points (Figure 9). This size was chosen as a large number of documents are presented at this point size.

Satisfactory results will indicate that handwriting may be read on pages

[2]The Tinker Speed of Reading Test provided courtesy of the University of Minnesota Press: ©1947, 1955 by Miles A. Tinker. All rights reserved. Published by the University of Minnesota Press, Minneapolis.

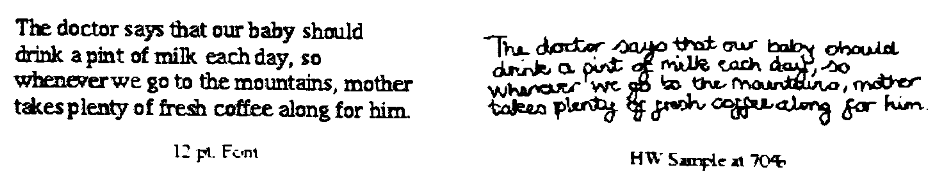



Figure 9: Equivalent typeface and handwriting sizes

with the same amount of information as pages containing typeface. Previous work has indicated that "Documents containing 10 point text or larger are quite readable" [10] on a pen tablet with a 1024×768 display resolution.

## 5.2 Equipment

The experiments were conducted in a closed room with no natural light to control for illumination. Software was written for both experiments and run on a 750MHz Intel Pentium III PC running Mandrake Linux. A Wacom PL-500, a digitizer tablet with an integrated LCD display, was used as the computer screen. This has a 1024×768 display resolution, and a 0.3mm dot pitch.

In the screening task and first experiment, volunteers were seated so that their heads were approximately 180cm away from the screen. They were asked to keep their back straight against the back of the chair so as to keep the distance between their eyes and the screen constant.

During the second experiment, volunteers were required to mark on the tablet with a pen so were permitted to position the pen tablet however they liked. The majority of volunteers left the tablet in its original position, upright on the desk, at a viewing distance of around 45cm. This was the easiest angle for the volunteers to read the script at, as the illumination of the screen dropped off rapidly as the viewing position moved away from being normal to the screen.

## 5.3 Volunteers

21 volunteers from the University of Hertfordshire administration staff were used in the experiments. Administration staff were chosen as we assumed that they are more likely to work with other people's handwriting than any other type of staff.

All volunteers were screened for visual acuity. This was done by asking the volunteers to complete a word identification task, using the tachistoscopic display program. Volunteers were asked to wear their glasses or contact lenses if they usually did so.

The experimental set up was identical to that in Experiment 1, described below. Volunteers were asked to identify 10 words, each displayed for 200ms. Volunteers with less than a 60% recognition rate did not participate in the first experiment. As the second experiment was less visually demanding, with only an informal constraint on viewing distance, all volunteers participated in it. The screening process also served as a familiarization task for the first experiment.

17 volunteers completed the first experiment, 18 the second.

## 5.4 Word Recognition

Upon entering the room, volunteers were asked to sit on a chair 180cm across from the computer screen. They were read instructions describing the overall experiment, and the screening process. They were encouraged to ask any questions that they may have had. After successfully completing the screening/familiarization task, volunteers were assigned in turn to one of the five volunteer groups. They were given a keyboard to rest in their lap, and shown the pre/post exposure field in the centre of the screen. They were asked to start when they were ready.

Upon pressing the space-bar on the keyboard the first exposure commenced. After a small random delay between 0.5–1.5s a word was displayed. This technique required the volunteer to focus their attention on the screen, as they could not anticipate exactly when the word would be displayed. The volunteer then spoke the word they thought they had seen. Each volunteer saw 100 words, displayed for 120ms each. Before the volunteer could proceed to the next word, their response was marked by the experimenter using a mouse click. The left-hand button marked a correct response and the right-hand button an incorrect response. Mark systems using 3 buttons to record 'near misses' or 'total refusals' were trialled, but were found to be too difficult to use reliably.

The results of each experiment were logged to a text file recording: the word; the exposure time; the delay before exposure; and the mark recorded by the experimenter. The experiment took around 5 minutes for each volunteer to complete.

## 5.5 Subjective Factors

After a short break, each volunteer commenced the second experiment. The volunteer was positioned at the desk directly in front of the screen. The volunteers were given a second set of instructions, explaining the procedure and nature of the experiment, including examples.

After being read the instructions and asking any questions, each volunteer was asked to complete a familiarization task. The familiarization task consisted of 5 Tinker Test items presented on the screen in a 12 point typeface. As in the preliminary drill of the Tinker Test [15], volunteers were asked to cross through the word that spoils the meaning of each item. Volunteers were asked to "work for speed and accuracy, that is work rapidly but do not make mistakes".

Volunteers were asked to complete the 3 pages of 5 test items, each time reminded of the instructions to "work for speed and accuracy". They were told that the computer would record their responses and the amount of time it took them to complete the task. After completing each page, volunteers were asked to complete a short questionnaire.

The questionnaire was based on QUIS 7.0[3] from the University of Maryland, adapted to evaluate reading handwritten script. Our questionnaire consisted of 3 question groups testing: the overall experience of reading the script; the ease of reading; and the clarity of the words displayed. Each group consisted of 4 or 5 questions with a negative to positive response scale of 5 points. In the first group, all 5 questions were taken from the QUIS. The 4 questions in the second group were completely new, based around terms commonly used to describe legibility. In the third group, the first 2 questions were adapted from the QUIS and the second two were new.

Finally, after answering the questionnaire for the third time, volunteers were asked to rate the rendering methods in order of preference, from 1 to 3 with 1 as their favorite. As an aide-memoir, the volunteers were shown a screen with a sample of each rendering method side by side in the order they had originally seen them.

# 6 Results

Twenty-one volunteers took part in the experiment. One volunteer did not have English as their first language, so she was dropped. Of the remaining twenty volunteers, three failed the initial screening task so did not complete the first part of the experiment. Two volunteers did not complete the last page of the questionnaire, so they were dropped from the second part of the experiment.

Thus in total there were 17 volunteers for the first part of the experiment, and 18 volunteers for the second.

[3]Questionnaire for User Interaction Satisfaction

| Pixel Width (mm) | Mean Recognition Rate (SD) |
|---|---|
| 0.30 | 82% (19%) |
| 0.45 | 84% (14%) |
| 0.60 | 81% (17%) |
| 0.75 | 71% (20%) |
| 0.90 | 59% (16%) |

Table 2: Pixel width against recognition rate

| Rendering Method | Overall Reactions | Ease of Reading | Clarity of Words | Preference |
|---|---|---|---|---|
| ClearPen | 3.12 | 3.43 | 3.56 | 1.28 |
| Anti-aliased Pen | 3.11 | 3.39 | 3.47 | 1.72 |
| Pixel Pen | 2.54 | 2.74 | 2.86 | 3.00 |

Table 3: Summary of mean user preferences

The results for the first part of the experiment are summarized in Table 2, "Pixel width against recognition rate". For the second part of the experiment, the component questions have been averaged over each question group. The results are summarized in Table 3, "Summary of mean user preferences". Table 3 also contains the volunteers' rank-order preference of rendering method. While the responses to the QUIS questions were ranked 1–5 with 5 as the most preferable, the rank-order data has a range of 1–3 with 1 as the most preferable.

# 7 Analysis

In order to test the significance of any differences in the means of the data from the tachistoscopic presentation experiment (Table 2), an Analysis of Variance (ANOVA) was performed.

The result of the ANOVA, $F(4, 80) = 5.481, p < .001$, suggests that the differences in performance in word recognition observed in the experiment could be ascribed to the effect of the independent variable, pixel width.

| Pearson's Correlation | | $N = 85$ |
| --- | --- | --- |
| **Relationship** | **Correlation** | **Sig. (2-tailed)** |
| Pixel Size against Recognition Rate | .353 | .001 |

Table 4: Experiment 1, Pearson PM correlation

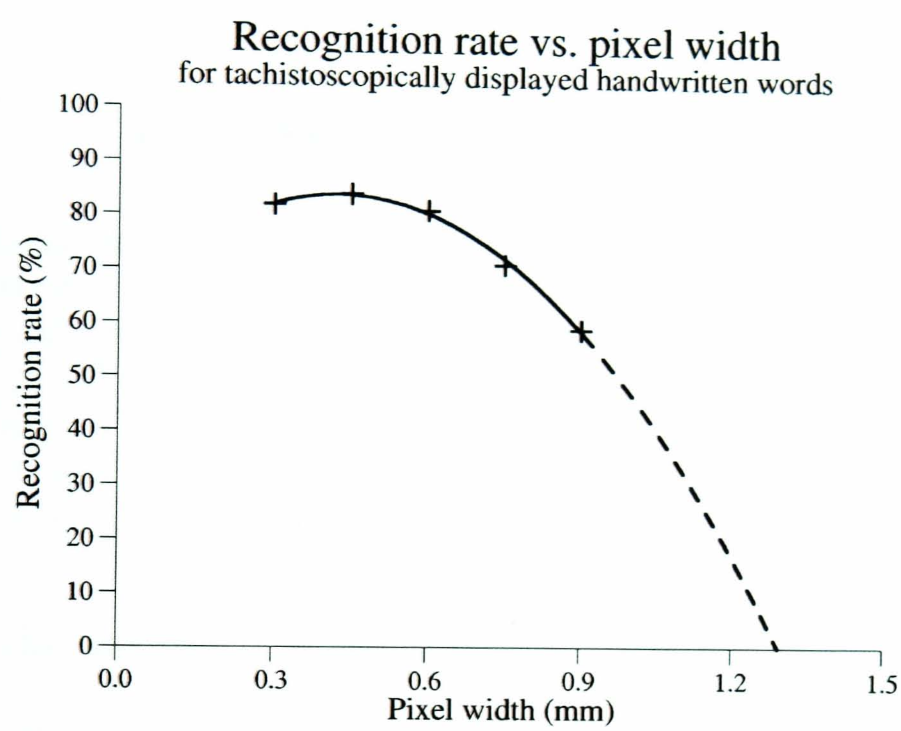



Figure 10: Recognition rate against pixel width

Post-hoc comparisons were made. These showed that the mean performance in the 0.90mm condition was significantly worse than performance in the 0.30–0.60mm conditions. To test for a relationship between pixel width and recognition rate, a Pearson's Product Moment Correlation was performed on the data in Table 2. The results of this analysis are shown in Table 4. The value of $p < .001$ (Table 4) suggests that there is a relationship between pixel width and recognition rate. This is illustrated in Figure 10.

The questionnaire response means in Table 3 for the ClearPen and anti-aliased rendering methods all fall above the response scale mid-point (3). The responses for the pixel pen rendering method consistently fall below this point. This would suggest, that in general the volunteers reacted positively

| SPEARMAN'S RHO | | $N = 54$ |
|---|---|---|
| **Question** | $\rho$ | **Sig. (1-tailed)** |
| Overall Reading | .370 | .003 |
| Ease of Reading | .303 | .013 |
| Clarity of Words | .339 | .006 |
| Preference | −.861 | .000 |

Table 5: Experiment 2, rank order correlation

to the higher resolution rendering methods, and negatively towards the pixel pen method.

The data from the subjective preferences questionnaire and the preference rank order data, were further analyzed using Spearman's Rank Order Correlation. These results are summarized in Table 5.

The responses to the subjective factors questions are significant at the $p < .05$ level, and the preference rating at the $p < .001$ level.

# 8 Discussion

Both experiments confirm the hypotheses they set out to address. Increasing horizontal rendering resolution increases both the legibility of, and preference for working with, handwritten script.

## 8.1 Legibility

The first experiment addressed the hypothesis:

> Improving the horizontal rendering resolution of a handwritten word, displayed on a computer screen, enhances its legibility.

The result of the first experiment affirms this hypothesis. If a linear relationship between pixel size and recognition rate is accepted, then any improvement in horizontal rendering resolution will enhance the legibility of handwritten script.

Figure 10 however does not show a linear relationship. The curve fitted is preferred for two reasons. Firstly it depicts the $\approx 80\%$ threshold in the recognition rate that was identified in previous work (Figure 7). Secondly, it indicates that the recognition rate falls off quite sharply in contrast to a linear relationship. This is commensurate with an informal observation during the planning of the experiment, where it was noted that words rendered with a pixel width greater than 0.9mm were practically illegible.

The results therefore indicate that legibility of handwritten script can be improved by increasing the horizontal rendering resolution, but that there is a limit to the effect that this will have.

We predict that different styles of handwriting will have different recognition rate thresholds. However, assuming that the threshold will always be reached at the same resolution, the results from this experiment can be used to estimate the optimal screen resolution for a screen to display handwriting.

Figure 10 shows the recognition rate threshold being reached at a pixel width of around 0.60mm. At the viewing distance of 1800mm, the 0.60mm pixels occupy around 0.019° of the visual field. At a normal viewing distance of around 450mm, 0.019° of the visual field would translate into a pixel width of around 0.15mm. This is equivalent to a display resolution of around 170 dpi.

This suggests that 170 dpi LCD screens may be ideal for displaying handwritten script. For lower resolution screens, such as the 85 dpi screen used in the experiment, the ClearPen algorithm adequately enhances the horizontal resolution.

## 8.2 Preference

The second experiment addressed the hypothesis:

> A user will perceive the effect of horizontal resolution enhancement as beneficial, and will prefer reading script rendered in this manner

Table 5 clearly shows that the volunteers in the experiment perceived a difference in the three different rendering methods, and that their preference followed increasing resolution.

Even though anti-aliased script may, on cursory observation, appear similar to script rendered in ClearPen, this experiment has shown an appreciable difference.

Referring again to Figure 10, the equivalent pixel widths for ClearPen and the pixel pen, read at 450mm, would be 0.4mm and 1.2mm respectively. The ClearPen result falls comfortably within the recognition rate threshold. Although we are not sure of the equivalent resolution of the anti-aliased pen, the shape of the graph shows why the anti-aliased rendering method was rated positively. Even small gains in resolution over that of the pixelated rendering method will significantly improve legibility.

# 9 Conclusion

Reading handwriting on a computer is as feasible as it is on paper and need not be hampered by poor script legibility. The results of the two experiments are mutually supportive. The legibility of handwritten script displayed on an LCD computer screen is improved by increasing the horizontal rendering resolution. People are able to perceive this improvement and prefer reading more legible script.

A screen resolution of 170 dpi was proposed as being optimal for reading handwriting. This value is dependent on an assumption that the recognition rate threshold will always be reached at the same resolution, independent of the style of handwriting (which will certainly affect the *level* of the threshold). The generality of this result is yet unproven as handwriting from only one author was used in the experiment.

The ClearPen rendering method is capable of improving the legibility of handwritten script displayed on an LCD screen. However, it does have a number of limitations. Firstly, script must be written along the direction of a scan-line. Informal observation has shown that legibility is not greatly impacted by vertical resolution enhancement. Secondly, the technique involves sacrificing colour for resolution. In applications where colour or freedom of orientation are important, ClearPen may not be suitable.

These experiments have pioneered the objective measurement of the legibility of handwritten script. They have also helped to generalize the findings of Wright et al. [18] in confirming that resolution affects legibility.

# 10 Further Work

Although we have established that horizontal rendering resolution does indeed affect script legibility, and that volunteers prefer interacting with script rendered at higher horizontal resolutions, we have not made direct quantitative measurement comparing different rendering methods in a "real world" task.

Further work could address the difference in the legibility between ClearPen and an anti-aliasing rendering method. This could be done through a speed of reading with comprehension test. The findings of this work are not based on a large group of authors. There is also plenty of scope for running similar trials with many different types of script. This could extend beyond Western scripts.

Taking a more broad outlook, the legibility measurement technique could be applied to other types of experiment. This experiment addressed the effect of resolution on legibility. Further experiments could look at enhancing the legibility of script in other ways, such as applying geometric transforms to emphasize features pertinent to human recognition.

Finally, work could be undertaken to discover whether or not resolution enhancement has any effect on ease of use when writing onto a screen.

# References

[1] Dan Boyarski, Christine Neuwirth, Jodi Forlizzi, and Susan Harkness Regli. A Study of Fonts Designed for Screen Display. In *CHI 1998 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 87–94. ACM SIGCHI, 1998.

[2] Timothy S. Butler. The Effect of Rendering on the Legibility of Digital Ink. Technical report, University of Hertfordshire, May 2002.

[3] Timothy S. Butler. The Recognition Rate of Handwritten Words in Tachistoscopic Display. Technical report, University of Hertfordshire, September 2002.

[4] Richard C. Davis, James A. Landay, Victor Chen, Jonathan Huang, Rebecca B. Lee, Frances C. Li, James Lin, Charles B. Morrey, Ben Schleimer, Morgan N. Price, and Bill N. Schilit. NotePals: Lightweight Note Sharing by the Group, for the Group. In *CHI 1999 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 338–345. ACM SIGCHI, 1999.

[5] Knut Hofland and Stig Johansson. *Word Frequencies in British and American English.* Longman, 1982.

[6] James Lin, Mark W. Newman, Jason I. Hong, and James A. Landay. DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. In *CHI 2000 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 510–517. ACM SIGCHI, 2000.

[7] R. A. Morris, R. D. Hersch, and A. Coimbra. Legibility of Condensed Perceptually-Tuned Grayscale Fonts. In Roger D. Hersch, Jacques

André, and Heather Brown, editors, *Electronic Publishing, Artistic Imaging, and Digital Typography*, number 1375 in Lecture Notes in Computer Science, pages 281–293. Springer-Verlag, 1998.

[8] William Prinzmetal. The word-superiority effect does not require a T-scope. *Perception & Psychophysics*, 51(5):473–484, 1992.

[9] G. M. Reicher. Perceptual recognition as a function of meaningfulness of stimulus material. *Journal of Experimental Psychology*, 81:275–280, 1969.

[10] Bill N. Schilit, Gene Golovchinsky, and Morgan N. Price. Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations. In *CHI 1998 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 249–256. ACM SIGCHI, 1998.

[11] Lambert Schomaker and Eliane Segers. Finding features used in the human reading of cursive handwriting. *International Journal on Document Analysis and Recognition*, (2):13–18, 1999.

[12] Mario Costa Sousa and John W. Buchanan. Observational Models of Graphite Pencil Materials. *Computer Graphics Forum*, 19(1):27–49, 2000.

[13] H. Timmers, F. L. van Nes, and F. J. J. Blommaert. Visual word recognition as a function of contrast. In E. Grandjean and E. Vigliani, editors, *Ergonomic Aspects of Visual Display Terminals, Proceedings of the International Workshop*, pages 115–120. Taylor & Francis Ltd, March 1980.

[14] M. A. Tinker. *Legibility of Print.* Iowa State University Press, 1964.

[15] Miles A. Tinker. *Tinker Speed of Reading Test.* University of Minnesota Press, Minneapolis, 1947, 1955.

[16] Richard A. Tyrrell, Thad B. Pasquale, Thomas Aten, and Ellie L. Francis. Empirical Evaluation of User Responses to Reading Text Rendered Using ClearType™ Technologies. In *SID '01 Digest*, pages 1205–1207. Society for Information Display, 2001.

[17] Lynn D. Wilcox, Bill N. Schilit, and Nitin Sawhney. Dynomite: A Dynamically Organized Ink and Audio Notebook. In *CHI 1997 Conference Proceedings, Conference on Human Factors in Computing Systems*, pages 186–193. ACM SIGCHI, 1997.

[18] S. L. Wright, I. L. Bailey, K-M. Tuan, and R. T. Wacker. Resolution and Legibility: A Comparison of TFT-LCDs and CRTs. In *SID '99 Digest*, pages 346–349. Society for Information Display, 1999.

# Segmenting Handwritten Text Using Supervised Classification Techniques

Yi Sun*, Tim Butler†, Alex Shafarenko‡, Rod Adams§, Neil Davey¶
Department of Computer Science
Faulty of Engineering and Information Sciences
University of Hertfordshire
College Lane, Hatfield
Hertfordshire AL10 9AB
*Y.2.Sun@herts.ac.uk
†t.s.butler@herts.ac.uk
‡a.shaferenko@herts.ac.uk
§r.g.adams@herts.ac.uk
¶N.Davey@herts.ac.uk

*Abstract*— **Recent work on extracting features of gaps in handwritten text allows a classification into *inter-word* and *intra-word* classes using suitable classification techniques. In this paper, we apply 5 different supervised classification algorithms from the machine learning field on both the original dataset and a dataset with the best features selected using mutual information. The classifiers are compared by employing McNemar's test. We find that SVMs and MLPs outperform the other classifiers and that preprocessing to select features works well.**

## I. Introduction

In this paper, we address the problem of identifying word boundarys in handwritten text: a process known as word segmentation. We make use of a selection of contemporary classification algorithms, such as multi-layer perceptrons, support vector machines, and Gaussian mixture models.

Surprisingly, little attention has been paid to the word segmentation problem by the neural net community. Nevertheless, recent work on extracting features of gaps between pieces of handwritten text allows for attaining segmented words by classifying gaps to *inter-word* and *intra-word* classes directly [3]. In this paper we try to find a suitable classifier to automatically segment so-called *digital ink*, i.e. graphically enhanced fragments of pen trace representing handwritten words, shapes and symbols of the sort that usually appear on paper when real ink is used for writing. Further details about the problem domain can be found in the next section. The previous work was done by using statistical methods to classify gaps into two classes based on one significant feature, named *river*, which is described in more detail in the following section. Each stroke involves an array of time-stamped sample points. However, as indicated in [12], exceptions are commonplace because of flourishes in writing styles with leading and trailing ligatures in handwriting. It is important to consider other possible features, as combinations of variables can provide significant information which is not available in any of the individual variables separately. The task is therefore to propose a classifier which can make as few errors as possible, based solely on the set of features.

In this work, we test 5 different supervised classification learning algorithms from the machine learning field to categorise gaps. We are also interested in selecting the most significant features. Since there is a proportion of gaps which can be classified with 99 percent accuracy in terms of the value of *river* directly, we apply these classification techniques for those patterns which cannot be judged easily by the feature *river*.

We expound the problem domain in the next section. In section III, we introduce the datasets used in this paper. We explain how we select a subset of features in terms of mutual information. In addition, a fuzzy dataset is obtained by using thresholds of *river*. Section IV briefly lists the classifiers used in our experiments and gives all the experimental results. We analyse the classification results by applying McNemar's test as well. The paper ends in section V with a discussion.

## II. Problem Domain

Despite the widespread use of office computers, handwriting has been and remains an important mode of capturing and annotating textual information. Computer-assisted handwriting is an increasingly important part of the general interface between the electronic media and the business world. Indeed, apart from the niche market of Personal Digital Assistants (including mainly smart phones and palmtop PCs), where the use of pen input devices is motivated primarily by their greater compactness, the mainstream computing technology now includes so called Tablet PCs. A tablet PC is a portable computer with a sensitive screen and a digital stylus, which is used as the main, or even the only, input device. The OS of a tablet PC is augmented with components that can handle *digital ink*. It is important to understand the difference between the digital ink and character-recognition interfaces. While the latter is merely a form of machine intelligence capable of recognising letters of an alphabet so that a keyboard can be replaced by an equivalent, but more compact, tablet and pen, the digital ink represents a separate form of input. It persists in documents as long as desirable for the author or/and readers. More importantly though, it is *processed* in its native form, i.e. as a graphical object. Words may be inserted, deleted or replaced at will without first being converted into a semantically focused form, such as an ASCII string. Such a conversion may happen eventually, when the final copy is produced.

There is therefore a fine balance for digital ink applications, namely one between the graphical form and semantic substance. One would like to benefit from the immediacy of pen input, its highly informal nature and potentially unlimited alphabet of letters, features and symbols, while at the same time having the computer penetrate the *structure* of the ink to the extent that it is necessary to be able to edit distinct parts of it. The depth of such penetration needs to be no more than superficial, down to a level of large self-contained units, such as lines and words, where the structuring is fairly well (albeit informally) defined. On the other hand, if no analysis is done of the ink input, then it is not really treated as handwriting, but as a general freehand graphical input. Consequently computer assistance (in the form of automatic placement, formatting and linkage with the rest of the document environment) would be very limited.

In this paper we focus on one level of the semantic penetration of pen input: the level of words. By 'word' we mean a group of pen strokes that have lexical significance, i.e. one that represents a word in a human language or a distinct symbol that can be used as a word. We wish to automatically segment digital ink represented as a *sampled pen trace* into word fragments purely on the basis of spatiotemporal relations between consecutive strokes, ignoring any meaning that may be represented by each such stroke. This has been a known problem in handwriting recognition research as well, although in this area of technology, word segmentation is seen merely as a precursor to full character recognition. In their recent comprehensive survey of handwriting recognition research, Plamondon and Srihari state that "prior to any recognition the acquired data is generally preprocessed to... segment the signal into meaningful units" [12].

The history of word segmentation research is delimited by the survey [12] and the one 10 years earlier [16], which is also referenced in [12]. The significant achievements reported in [16] for this area are confined to straightforward geometric segmentation using convex shells [8] with some consideration given to stroke timing. It is noteworthy that these early proposals have not been developed any further as is evidenced by [12]. One can only speculate about the reason why no further progress has been reported. Our experience shows that simple segmentation methods are prone to error due to an individual writer's idiosyncrasies as well as the fact that these methods fail to capture more subtle structural and temporal signals which would strengthen the basis for segmentation. More recent work is attempting to improve structure recognition by introducing hierarchical agglomerative clustering, see [14], [9] in a broader context of automatic structural analysis of handwritten document. These in our opinion are interesting approaches, though they are susceptible to writing idiosyncrasies while being insensitive to any recurrent features of the language (or symbolic system) used by the writer.

The variability of one's writing style as well as the inherent diversity of writers would strongly advocate an adaptive solution. The solution would not be confined to any specific *ad hoc* metric of the pen trace as the basis of segmentation, but would accommodate a reasonably large set of these metric, taking into account both prime features (such as the size and duration of inter-stroke gaps) as well as any secondary ones which may be significant. Such features are still proposed on the basis of their plausibility, without much formal basis or a priori evidence. However, we have been guided by [13] where a thorough geometric and temporal classification was provided for a pen gesture recogniser. To give an idea of the sort of features that were being used there, we illustrate some of them in figure 1. It presents a single pen stroke with its bounding box. The features $x$ and $y$ as shown give the dimensions of the bounding box and the angle $\alpha$ is linked with its aspect ratio. The distance $s$ is between the end points of the stroke, and $\beta$ is the angle between the line connecting those points and the vertical. Finally, if $\theta_i$ is the angle between two consecutive pen segments of the stroke, $i$ and $i+1$ then one can use the feature

$$\sigma = \sum_{i=1}^{n-1} \theta_i$$

as a measure of curvature. The proposed features were not all purely geometric; there were a few related to the time interval of the stroke and the speed of the pen tip. Note that most of these features are inapplicable to inter-stroke gaps, but some still make sense, e.g., $x$, $y$ $\beta$, etc. We have introduced a gap feature which has proven especially useful for our purposes. We call it *river width* or *river* for short, following Fox and Tappert [8]. The river of a gap is the shortest distance between two consecutive strokes, i.e. the length of the shortest chord drawn between pen position samples from neighbouring strokes, as shown in figure 2. Two rivers are indicated there by double-headed arrows.

We have expanded the set proposed therein by our own form factors, see [3] for each pen stroke. The pen trace has thus been abstracted to a sequence of stroke and gap, where each gap is represented by 14 variables. In this work, we are interested in classifying gaps, so we ignore the strokes. A human reader has annotated the gaps in our experimental traces as either intra-word or inter-word by recognising the words in the language. Thus the task is to search for a classification method which can produce the same annotations with as few errors as possible.

## III. The Description of The Datasets

### A. Gaps Datasets

In this paper, we present experimental results on the gap datasets. The *original* gap dataset includes 2482 data points labeled by *inter-word* and 4980 *intra-word*. In the experiments, 2/3 of the data points from the dataset are used for training,

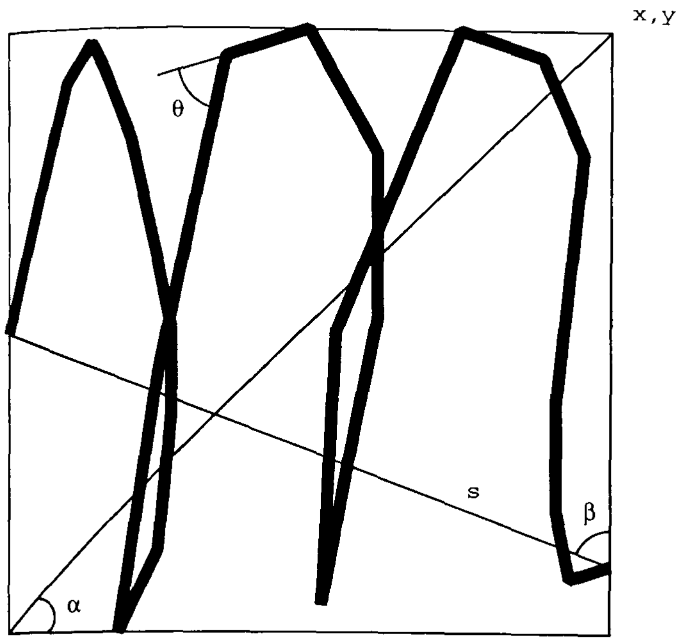



Fig. 1. An illustration: the sort of features of a single pen stroke with its bounding box.

Fig. 2. An illustration: two rivers of gaps are shown by double-headed arrows.

while 1/3 for test. We do experiments with all 14 features and *reduced* features involving the 8 most significant to the classification found by analysing mutual information as discussed in section III-B.

## B. Feature Extraction by Using Mutual Information

The features associated with gaps are reduced by employing mutual information. The mutual information of two variables is a measure of the common information shared between them [10]. In this work, the two variables are the class variable $c$ and the feature variable $x$. $c$ may take one of two values and $x$ one of 14 features. The bigger the value of the mutual information, the more common information is shared. If two variables are independent, their mutual information is zero. An advanced treatment of feature extraction using mutual information maximization can be found in [5].

Assuming data points are generated from $C$ classes (In this paper, $C = 2$). Mutual information, denoted by MI, is given by [4]

$$MI = H(c) - H(c|x), \tag{1}$$

where $H(c)$ is the entropy of the classes prior probability $P(c_i)$ given by

$$H(c) = -\sum_{i=1}^{C} P(c_i) \log P(c_i). \tag{2}$$

and $H(c|x)$ is conditional entropy having the form, as follows

$$H(c|x) = -\sum_{i=1}^{C} P(c_i, x) \log P(c_i|x), \tag{3}$$

where $P(c_i, x)$ are the joint probability distributions, and $P(c_i|x)$ are posterior probabilities. Equation (3) can be further written as

$$H(c|x) = -\sum_{i=1}^{C} P(c_i) \int p(x|c_i) \log P(c_i|x)\, dx. \tag{4}$$

Note that $\int p(x|c_i) \log P(c_i|x) dx$ is the expectation of $\log P(c_i|x)$ given the probability density $p(x|c_i)$.

Empirically the conditional entropy $H(c|x)$, which is based on the probability density function of the variable $x$, can be approximated as follows, when considering two classes:

$$\begin{aligned} H(c|x) \approx\ & -\frac{1}{N_1} P(c_1) \sum_{k=1}^{N_1} \log P(c_1|x^k) \\ & -\frac{1}{N_2} P(c_2) \sum_{l=1}^{N_2} \log P(c_2|x^l), \end{aligned} \tag{5}$$

where $x^k$ and $x^l$ denote the feature values given that the data points are generated from two densities $p(x|c_1)$ and $p(x|c_2)$, respectively. $N_1$ and $N_2$ are number of samples from the two distributions, respectively.

To compute (5), a sufficient number of data points: $N_1$ plus $N_2$, are required, sampled from the two estimated distributions. We employ two Gaussian mixture models to model the distributions of the gaps data collected from the class *inter-word* denoted by $c_1$ and *intra-word* denoted by $c_2$. The *expectation-maximisation* (EM) algorithm [6] is used for finding parameters of each model. A mixture distribution having $M$ components (in this work, $M = 5$) can be calculated using:

$$p(x|c_i) = \sum_{j=1}^{M} p(x|j, c_i) P(j), \tag{6}$$

where $P(j)$ are mixing coefficients and

$$p(x|j, c_i) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left\{ -\frac{1}{2\sigma_j^2}(x - \mu_j)^2 \right\}, \tag{7}$$

where $\mu_j$ and $\beta_j$ are mean and variance of each component $j$ respectively. More details about Gaussian mixture models can be found in [1]. Then 500,000 data points were sampled from these two distributions. Finally, the posterior probability can be computed using Bayes' theorem

$$P(c_i|x) = \frac{P(c_i)p(x|c_i)}{\sum_{i=1}^{C} P(c_i)p(x|c_i)}. \tag{8}$$

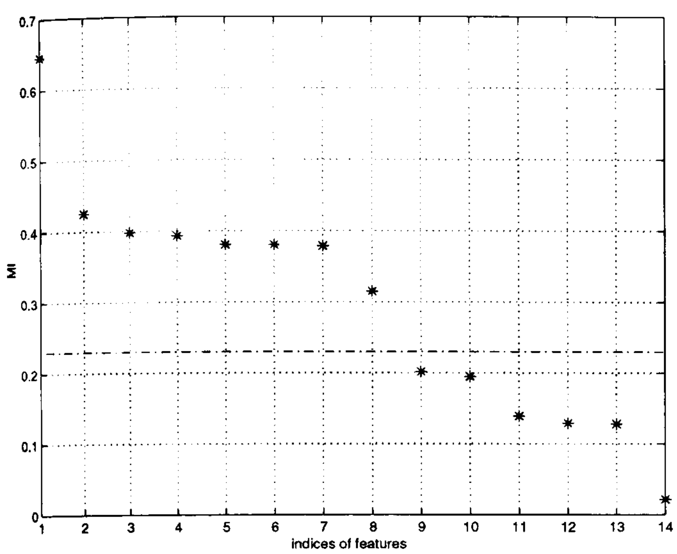



Fig. 3. Mutual information of class variable and each feature of gaps: each value is shown as a star sign. The background are dashed lines that are major grid lines to the current axes. The horizontal dash-dot line denotes the cut off value.

Figure 3 shows the mutual information of each feature with the class variable sorted by their values. As shown, there is a reasonable "jump" from the ninth value to the eighth. We ignore those features indexed from 9 to 14. Thus 8 features with mutual information values more than 0.3, a subset of whole features, can be obtained.

### C. *The Fuzzy Dataset with Thresholds*

As seen in Figure 3, there is one feature which is the most significant to classification, named *river*. Since it measures the shortest distance between samples in adjacent strokes, gaps between words usually have a larger value than gaps within words. One can expect to benefit from this variable as much as possible, though exceptions often occur with variety in writing styles as mentioned in the introduction section. Two boundaries of the values of *river* can be determined as displayed in figure 4. In this figure, the river values increase from left to right. Boundary 1 specifies a *river* value, on the left of which one can ensure that the probability that the gap belongs to class *intra-word* is not less than 99 percent; while boundary 2 specifies another value of *river*, on the right of which the probability that the gap belongs to class *inter-word* is not less than 99 percent. Then the whole dataset is filtered by means of these two thresholds. In this way, a sub-dataset called *fuzzy*, whose values of the *river* feature are within these two boundaries, is obtained. This subset therefore consists of 3361 gaps that cannot easily be classified by the *river* feature.

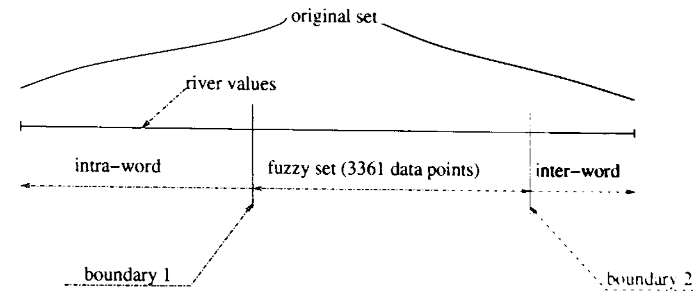



Fig. 4. A diagram: explaining how the fuzzy dataset is generated with two boundaries.

## IV. Experimental Results with Supervised Classifiers

### A. *Supervised Classifiers*

In this section, we first list the supervised classifiers used in our experiments. Readers who are interested in those classification techniques can follow the references to learn more.

- Logistic discrimination analysis (LDA) [1];
- K-nearest neighbor classification (KNN) [11];
- Guassian mixture model (GMM) [1];
- Multi-layer perceptron (MLP) using scaled conjugate gradients algorithm [1];
- Support vector machine (SVM) using Gaussian kernel [15].

Parameters of each class-condition density were estimated from the training dataset in the GMM. For the MLP, a two-layer architecture was set up, since it has been proved for classification task that the MLP with sigmoidal activation function and two layers of weights can approximate any decision boundary to arbitrary accuracy [2].

### B. *Experiments*

Experiments were performed on both the original dataset and the fuzzy dataset with all 14 features and the selected set of 8 features. The user-chosen parameters for each classifier were selected by cross-validation, where the training set was divided into 10 partitions. 9 partitions were used to train the model and the other one was used as a validation set. The SVM experiments were completed using LIBSVM, which is available from the URL
`http://www.csie.ntu.edu.tw/~cjlin/libsvm.`
The others were implemented using the NETLAB toolbox, which is available from the URL
`http://www.ncrg.aston.ac.uk/netlab/.`

In Table I, we present all the user-chosen parameters attained by using cross-validation.

### C. *Classification Results*

Classification results for each test dataset with different supervised classifiers are displayed in Table II. The accuracy

TABLE I

USER-CHOSEN PARAMETERS FROM CROSS-VALIDATION. $K$ DENOTES THE NUMBER OF NEIGHBOURS; $nc1$ AND $nc2$ ARE THE NUMBER OF GAUSSIAN MODELS IN EACH MIXTURE; $j$ SIGNIFIES THE NUMBER OF HIDDEN UNITS IN THE MLP; $A$ IS THE UPPER BOUND OF COEFFICIENTS $\alpha_i$ IN THE SVM; AND $\sigma$ IS WIDTH OF RADIAL BASIS FUNCTION.

| | KNN ($K$) | GMM ($nc1$, $nc2$) | MLP ($j$) | SVM ($A$, $\sigma^2$) |
|---|---|---|---|---|
| fuzzy 8 | 9 | 6, 6 | 8 | 25, 0.16 |
| fuzzy 14 | 9 | 6, 4 | 5 | 20, 0.1 |
| orig. 8 | 5 | 8, 9 | 15 | 25, 0.25 |
| orig. 14 | 5 | 9, 9 | 5 | 5, 0.16 |

is defined as the number of correct classified patterns over the number of total patterns in the test set. The results to the GMM and MLP shown in Table II are average of 10 repetitions with different random initial conditions.

TABLE II

RESULTS ON GAP DATASETS: ACCURATE RATE %

| | LDA | KNN | GMM | MLP | SVM |
|---|---|---|---|---|---|
| fuzzy 8 | 86.0 | 90.1 | 86.6 | 92.1 | 92.2 |
| fuzzy 14 | 87.5 | 89.2 | 84.5 | 91.5 | 92.5 |
| orig. 8 | 92.7 | 93.8 | 92.0 | 95.3 | 95.8 |
| orig. 14 | 93.2 | 93.8 | 90.4 | 96.1 | 96.2 |

Table II shows that using the reduced features as found by mutual information one can obtain a result as good as using all 14 features when employing the KNN, MLP and SVM. In addition, it also suggests that the MLP and SVM provide more accurate classification than the LDA, KNN and GMM classifiers. Interestingly, one can work on the fuzzy dataset and still achieve comparable results. The values given in the first two rows of Table II for the dataset are the accurate rate for just the fuzzy gaps. Since the rest of original dataset has already been classified with 99% accuracy, the classification for the whole dataset achieved by this quicker method can be calculated. For instance, the SVM classifier gives a full classification rate for the whole dataset, when processing a dataset involving 3361 fuzzy gaps among all 7462 gaps, as follows,

$$\frac{3361}{7462} \times 92.5\% + \frac{7462 - 3361}{7462} \times 99\% = 96.1\%.$$

### D. Statistical Test for Comparing Supervised Classification Learning Algorithms

Our primary goal is to choose the best learning algorithm for recognising the two class gaps. Looking at Table II, it can be seen that there is no big difference between the MLP and SVM algorithms. As addressed in [7], McNemar's test can be used for determining whether one learning algorithm is better than another on a special task with acceptable the probability of incorrectly detecting a difference when no difference exists. Thus we apply McNemar's test for comparing these two algorithms. In addition, we provide results of McNemar's test on the KNN and SVM as a comparison.

We first calculate the contingency table assuming there are two algorithms $I$ and $II$, illustrated in Table III [7], where

TABLE III

$2 \times 2$ CONTINGENCY TABLE

| | |
|---|---|
| $n_{00}$ | $n_{01}$ |
| $n_{10}$ | $n_{11}$ |

$n_{00}$ is number of samples misclassified by both algorithms; $n_{01}$ number of samples misclassified by algorithm $I$ but not $II$; $n_{10}$ number of samples misclassified by algorithm $II$ but not $I$; $n_{11}$ are correctly classified by both algorithms.

McNemar's test has a chi-square distribution with 1 degree of freedom [7]. Quantity $\chi^2$ is computed as follows:

$$\chi^2 = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}}. \tag{9}$$

The null hypothesis assumes that the performance of two different learning algorithms is the same, i.e. $n_{10} = n_{01}$. The $P$-value from a chi-square value is computed with McNemar's test. Since small $P$-values suggest that the null hypothesis is unlikely to be true, we may reject the null hypothesis if the probability that $\chi^2 \geq 3.84$ is less than 0.05 [7].

TABLE IV

RESULTS OF MCNEMAR'S TEST FOR COMPARING THE MLP WITH THE SVM AND THE KNN WITH THE SVM ALGORITHMS.

| | mlp-svm | | knn-svm | |
|---|---|---|---|---|
| dataset | $\chi^2$ | $P$-value | $\chi^2$ | $P$-value |
| fuzzy 8 | 0.77 | 0.38 | 7.22 | 0.0072 |
| fuzzy 14 | 1.47 | 0.23 | 13.28 | 0.0003 |
| orig. 8 | 1.80 | 0.18 | 22.52 | 0.0001 |
| orig. 14 | 0.62 | 0.43 | 31.65 | 0.0001 |

Table IV displays results for comparing the MLP with the SVM and the KNN with the SVM algorithms. The $\chi^2$ for the MLP and SVM is an average calculated over the 10 runs. Looking at the third column, since all $P$-values are greater than 0.05, we cannot reject the null hypothesis which suggests that applying the MLP and SVM learning algorithms to construct classifiers for this application can achieve the same classification results. In addition, since the SVM outperforms the LDA, KNN and GMM, as seen in Table II, one can expect that the $P$-value should be smaller than 0.05 when comparing them with the SVM. This is illustrated in the last column where the KNN and SVM classifier results are compared.

## V. DISCUSSION

In this paper, we apply a variety of contemporary classification algorithms to the word segmentation problem. We report classification results obtained by using 5 different supervised classifiers: LDA, KNN, GMM, MLP and SVM. The various classifiers are compared by McNemar's test. The results show the best result can be achieved by using non-linear classification techniques: the MLP and SVM algorithms. Mutual information is employed to select the most significant subset of features.

The results show the smaller set of features characterises the data as well as the full set. One of the features allows for 99% correct classification of roughly half the data by simple thresholding. Removing these data points leaves a reduced dataset which can then be classified using the more sophisticated non-linear techniques. The results show that this work well and it is faster than using the full dataset.

## REFERENCES


[1] C. M. Bishop, *Neural Networks for Pattern Recognition*, New York: Oxford University Press, 1995.

[2] E. K. Blum and L. K. Li, "Approximation theory and feedforward networks," *Neural Networks*, vol. 4, no. 4, pp. 511–515, 1991.

[3] T. Butler, *Department of computer Science*. Ph.D. thesis, University of Hertfordshire, Department of Computer Science, University of Hertfordshire, Hertfordshire, UK.

[4] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, Inc., 1991.

[5] K. Torkkola, "Feature extraction by non-parametric mutual information maximization", *Journal of machine learning Reseach*, vol. 3, pp. 1415-1438, 2003.

[6] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *J. Roy. Stat. Soc. B*, vol. 39, pp. 1–38, 1977.

[7] T. G. Dietterich, "Approximate statistical tests for comparing supervised classifi cation learning algorithms", *Neural Computation*, vol. 10, no. 7, pp. 1895-1923, 1998.

[8] A. S. Fox and C. C. Tappert, "On-line external word segmentation for handwriting recognition", *Proceedings of the Third International Symposium on Handwriting and Computer Applications*, 1987, pp. 53–55.

[9] Y. Li, Z. Guan, H. G. Wang, G. Z. Dai and X. S. Ren, "Structuraizing freeform notes by implicit sketch understanding", *AAAI Spring Symposium on Sketch Understanding*, 2002.

[10] D. Michie, D. J. Spiegelhalter and C. C. Taylor, (Eds), *Machine Learning, Neural and Statistical Classifi cation*, Ellis Horwood, 1994.

[11] I. T. Nabney, *Netlab Algorithm for Pattern Recognition*, Springer, 2001.

[12] R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: a comprehensive survey", *IEEE Transactions on Pattern Analysis and machine Intelligence*, V 22, no. 1, pp. 63-84, 2000.

[13] D. Rubine, "Specifying gestures by example", *Computer Graphics*, vol. 25, no. 4, pp. 329-337, 1991.

[14] E. Saund, J. Mahoney, D. Fleet, D. Larner and E. Lank, "Perceptual organization as a foundation for intelligent sketch editing", *AAAI Spring Symposium on Sketch Understanding*, 2002.

[15] B. Scholköpf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press, 2002.

[16] C. C. Tappert, C. Y. Suen and T. Wakahara, "The State of the art in on-line handwriting recognition", *IEEE Transactions on Pattern Analysis and machine Intelligence*, vol. 12, no. 8, pp. 787-808, 1990.