1. Introduction

Palm devices and Pocket PCs are tablet-like devices that have achieved a high level of success in the market place. They are small, specialized and designed to be carried around in your pocket. Tablet PCs, on the other hand, are general-purpose computers, with sensitive screens designed to interact with an accompanied pen. They run on the same processors as a laptop, have large hard drives, and have as much memory as any other computer [6].

Tablet PCs are hybrids of handheld devices, laptops and other information tools. They are powered by special tablet PC versions of operating systems. The paper focuses on the Microsoft Windows XP Tablet PC Edition and its related software. The Windows XP Tablet PC is a superset of Windows XP; therefore all applications that can run on a regular PC can also run on the Tablet PC. This includes anything from MS Office to the applications we write ourselves.

The current Tablet PC tools offered by Microsoft include Input Panel, Office OneNote, Windows Journal, Sticky Notes, and the Education Pack. Input Panel dynamically converts handwriting input by a user to text. OneNote is a tool to enhance note taking: users can copy notes from other sources or insert documents from other Office programs such as Word and PowerPoint. Windows
Journal is a basic note-taking tool that is used to capture handwritten notes and drawings, convert handwritten notes to text, import graphics files, and share notes with others. The Sticky Notes tool is used to write and store short notes, phone numbers, and other reminders. Sticky notes can be placed directly on the desktop as quick reminders or in MS Word to add electronic comments or reminders to documents [11]. The latest tool for the academic environment is the Education Pack which includes Ink Flash Cards and Equation Writer as the main programs [10]. With Ink Flash Cards, a student can create two-sided question-and-answer cards to test their knowledge. Equation Writer helps users handwrite a math equation and convert it to text with the touch of a pen; much more efficient than using Equation in Word.

Academic staff members at universities around the world have developed Tablet PC-based applications in their academic environments. Typical application includes lecture presentations, teaching computer science and software engineering courses and providing peer-review comments. For lecture presentations, the Classroom Presenter system [2] has been developed to enhance an instructors’ flexibility in giving presentations, by allowing easy navigation through a presentation and the ability to write on the presentation itself. K. Mock presented his experience in teaching computer science and software engineering courses with a tablet PC [1, 9]. He used the Tablet PC in the classroom as a digital whiteboard, by connecting it to a data projector. Outside of the classroom, the Tablet PC is also a useful tool for grading assignments, creating lecture material, and capturing meeting notes. Pérez-Quiñones and S. Turner performed an informal study on peer-review comments using the Tablet PC, by observing how students used the pen on a Tablet-PC to provide feedback on an object-oriented design created by other students [14]. Peer-review has been found to be a good way to help students learn from each other and to increase their understanding of coding and design issues.

In this paper, we review the current applications provided by Microsoft and also those developed by professors at universities for academic purposes. We then present the Tablet PC applications developed at our university, using the Visual Studio.NET software package. With extra Windows controls designed for Tablet PCs, we can develop pen-based programs for teaching purposes as well as handwriting recognition and signature authentication for research purposes.
2. Tablet PC Tools

In this section we present some of the most commonly used tools designed for Tablet PCs: Input Panel, Office OneNote, Windows Journal, Sticky Notes, and the Education Pack. These tools are provided by Microsoft [10, 11].

The Input Panel tool dynamically converts handwriting to text. There are three modes for input: Writing Pad, Character Pad and On-screen Keyboard. Figure 1 displays the Input Panel using Writing Pad to convert handwriting input to text. The Input Panel also helps the user make corrections quickly and easily before inserting text.

![Input Panel](image1.png)

Figure 1. Handwriting-to-text conversion using the Input Panel tool in Writing Pad mode.

OneNote is a note taking tool. It enables users to copy notes from a Microsoft Windows Mobile-based Pocket PC or Smartphone, record video notes, capture screen clippings, import meeting details from Outlook, and insert documents from other Office programs such as Word or PowerPoint. Interestingly, OneNote files can be saved as MHTML files, which other users can view with Microsoft Internet Explorer version 5.0 or later.

![OneNote and Windows Journal](image2.png)

Figure 2. Pages in Office OneNote and Windows Journal, respectively.

Windows Journal is a basic note-taking tool that is used to capture handwritten notes and drawings, convert handwritten notes to text, import graphics files, and share notes with others. Windows Journal can be used as a powerful presentation enhancement, by making presentations more interactive with use of the pen, digital ink, and highlighter tools. Similar to Office OneNote,
Journal files can be shared with non-Tablet PC users by exporting them as MHTML files.

*Sticky Notes* tool is used to write and store short notes, phone numbers, and other reminders. Sticky Notes can be placed directly on the desktop as quick reminders or used in MS Word to add electronic comments or reminders to documents. They are easy to use and quick to access. Users can also use a microphone to record information. A sticky note can contain both writing and recordings.

The latest tool for an academic environment is *Education Pack*. It includes five programs that can help users get organized and study effectively. The most interesting programs are Ink Flash Cards and Equation Writer. With Ink Flash Cards, a student can create two-sided question-and-answer cards to test their knowledge. Equation Writer helps users handwrite a math equation and convert it to text with the touch of a pen; much quicker than using Equation in Word. The following is an expression generated by Equation Writer:

\[
\alpha \sum_{\beta}^{1} \int_{0}^{1} e^{-x}dx
\]

![Equation Writer example](image.png)

Figure 3. Equation Writer is used to generate mathematic expressions and equations.

### 3. Applications at Our University

In this section we present some applications of Tablet PCs, for teaching and research at our university, the University of Canberra, Australia. We use the Tablet PC for marking and grading student assignments and exercises and adding peer-review comments to drafts of research papers and theses. We also developed our own Tablet PC signature recognition system. Students can develop Tablet PC applications as a semester-long project using MS Visual Studio .NET. With the support of MS Visual Studio .NET, ink-enabled applications can easily be created. There are two ink-based controls for Windows applications which are InkEdit and InkPicture. The InkEdit control gets handwriting input and converts it into text. The InkPicture control displays handwriting inputs as they looked when created. Other controls on the Windows
form can be ink-enabled using the InkOverlay class. The InkCollector class could be used to recognize the pen when it moves over the textbox and open up a larger input area when the pen hovers within its range. This control can automatically recognize the handwritten text and paste it into the textbox when the pen moves away [6].

3.1. Marking, Grading and Peer-Review Comments

In order to use the Tablet PC for marking and grading, students are required to submit their assignments, exercises and reports electronically via email or a Web page such as WebCT. Electronic submission makes it easy for instructors to check the submission dates for assignments. Marking, grading and providing comments to the submissions is fast and clear with the Tablet PC. Instructors can also keep a copy of marked submissions without the hassle of having to photocopy hard copies.

Reports, presentations and research papers written amongst a group of authors require several draft revisions. It is fast, clear and concise for authors to use a Tablet PC to provide comments and suggestions on the same draft. Figure 4 presents an example of using the Tablet PC to suggest comments on a research paper draft.

Figure 4. Comments are given on a research paper draft.

3.2. Handwriting Recognition Applications

The proposed handwriting recognition applications for teaching purpose include implementation of crosswords and puzzles. These applications were developed using MS Visual Studio .NET.

There are currently some applications for solving crosswords and puzzles, especially Sudoku puzzle [20] available on the Internet. However these applications require a user to use mouse and keyboard to solve those crosswords and puzzles. The Tablet PC provides a very convenient way to the user to solve those crosswords and Sudoku puzzle displayed on the Tablet PC screen. The user is able to use pen to handwrite letters or digits to cells in a crossword or in a
Sudoku puzzle. The cells in these applications are created with the use of the InkEdit control available in MS Visual Studio.NET. The handwritings will then be converted to text. Figure 5 shows a Sudoku puzzle application for the Tablet PC. The handwritten digits in the first three cells are converted to digits after a preset recognition timeout.

![Sudoku puzzle](image)

Figure 5. A Sudoku puzzle for Tablet PC. The first three cells are entered the handwritten digits of 4, 2 and 8, respectively. After a preset recognition timeout, the handwritten words are converted to text.

### 3.3. Signature Recognition System

We have developed a signature recognition system for the Tablet PC. The system consists of three subsystems which are enrolment, identification and verification as shown in Figure 6.
New users register to the system using the enrolment subsystem. For a new user registration, a Windows form is provided for the user to enter a username and two copies of his/her signature. The system extracts features from the entered signatures and builds a signature model using a modeling technique such as vector quantization, Gaussian mixture modeling or hidden Markov modeling. Signature models are stored in XML (Extensible Markup Language) format.

Vector quantization (VQ), regarded as the simplest modeling method, was applied to this system. The fuzzy c-means (FCM) clustering method [4] was used to create signature models known as vector quantization codebooks. The method is summarized as follows.

Let \( X = \{x_1, x_2, ..., x_T\} \) be a set of \( T \) feature vectors. Fuzzy clustering known as unsupervised learning in \( X \) is a fuzzy partitioning of \( X \) into \( C \) fuzzy subsets or \( C \) clusters where \( 1 < C < T \). The most important requirement is to find a suitable measure of clusters, referred to as a fuzzy clustering criterion. Objective function methods allow the most precise formulation of the fuzzy clustering criterion. The most well known objective function for fuzzy clustering in \( X \) is the least-squares functional, that is, the infinite family of FCM functionals, generalized from the classical within-groups sum of squared error function

\[
J_m(U, \mu; X) = \sum_{i=1}^{C} \sum_{t=1}^{T} u_{it}^m d_{it}^2
\]

(1)

where \( U = \{u_{it}\} \) is a fuzzy \( c \)-partition of \( X \), each \( u_{it} \) represents the degree of vector \( x_t \) belonging to the \( i \)th cluster and is called the fuzzy membership function. For \( 1 \leq i \leq C \) and \( 1 \leq t \leq T \), we have
\[
0 \leq u_{it} \leq 1, \quad \sum_{i=1}^{C} u_{it} = 1, \quad \text{and} \quad 0 < \sum_{i=1}^{T} u_{it} < T
\]  

(2)

\[m \geq 1\] is a weighting exponent on each fuzzy membership \(u_{it}\) and denotes the degree of fuzziness; \(\mu = (\mu_1, \ldots, \mu_C)\) are cluster centers and, \(d_{it}\) is the distance in the A norm (\(A\) is any positive definite matrix) from \(x_i\) to \(\mu_j\), known as a measure of dissimilarity

\[
d_{it}^2 = \|x_i - \mu_j\|^A = (x_i - \mu_j)'A(x_i - \mu_j)
\]  

(3)

The basic idea in the FCM is to minimize \(J_m\) over the variables \(U\) and \(\mu\), on the assumption that matrices \(U\) that are part of optimal pairs for \(J_m\) identify good partitions of the data. Minimizing the fuzzy objective function \(J_m\) in (1) gives

\[
u_{it} = \left[ \sum_{k=1}^{C} \left( \frac{d_{it}}{d_{kt}} \right)^{m-1} \right]^{-1}
\]  

(4)

\[
\mu_j = \frac{\sum_{i=1}^{T} u_{it}^m x_i}{\sum_{i=1}^{T} u_{it}^m}
\]  

(5)

The FCM algorithm is known as the fuzzy vector quantization (FVQ) algorithm in pattern recognition and is used to train codebooks in the VQ approach.

After registration, the user can log on to the system using his/her registered signature for identification or verification.

For identification purpose, the user enters his/her signature on a identification form displayed after clicking a button.
The unknown signature $X$ will be scaled and extracted features to calculate $N$ likelihood functions $P(X \mid \lambda_i)$, $i = 1, \ldots, N$. The highest likelihood value is chosen and the unknown signature is identified as the signature of the corresponding user

\[
\text{Choose the signature model } \lambda_{i*} \text{ if } i^* = \arg \max P(X \mid \lambda_i).
\]

For verification purpose, the user’s signature plays the role of a password. The user logs on to the system using his/her registered username and signature, via a verified login form. The entered signature will be compared with the signature model whose identity is claimed. If the match is above a given threshold, the identity claim is accepted. The verification method is summarized as follows.

In a statistical approach, the verification problem is usually formulated as a problem of statistical hypothesis testing. For an input signature $X$ and a claimed identity $S$, the task is to determine if $X$ was signed by the claimed writer $S$.

Assuming that $X$ contains signature features from only one writer, the signature verification task can be regarded as a basic hypothesis test between the null hypothesis $H_0$: $X$ is from the claimed writer $S$ against the alternative hypothesis $H$: $X$ is not from the claimed writer $S$. According to Neyman-Pearson Lemma, if the probabilities of both the hypotheses are known exactly, the optimum test to decide between these two hypotheses is a likelihood ratio test given by

\[
S(X) = \frac{P(X \mid H_0)}{P(X \mid H)} \begin{cases} > \theta & \text{accept } H_0 \\ \leq \theta & \text{reject } H_0 \end{cases}
\]

where $\theta$ is a predefined decision threshold and $S(X)$ is referred to as the similarity score of the input signature $X$. This approach provides a good theoretical formulation to signature verification.
However, in practical verification problems, it is impossible to obtain the exact probability density functions for either the null hypothesis or the alternative hypothesis. A parametric form of the distribution under each hypothesis is assumed to estimate these probability density functions. Let $\lambda_c$ be the claimed signature model and $\lambda$ be a model representing all other possible signatures, i.e. impostors. Let $P(X | \lambda_c)$ and $P(X | \lambda)$ be the likelihood functions of the claimed writer and impostors, respectively. The similarity score is calculated as follows:

$$S(X) = \begin{cases} 0 & \text{if } P(X | \lambda) \leq \theta \\ \frac{P(X | \lambda_c)}{P(X | \lambda)} & \text{if } P(X | \lambda) > \theta \end{cases}$$

For the verification stage, the impostor model was approximated by a background signature model. We used the same training data to build the claimed signature model and background signature model [16, 17]. We trained these 2 VQ models for each signature with codebook sizes of 16 and 32 codewords, respectively. The 16 codeword model was used as the background model and the 32 codeword model was used as the claimed signature model. The background modeling method is useful for practical and portable devices such as palm-top computers, since a higher demand for computation and memory requirements may not be desirable for such applications [16].

For evaluation, the signature verification system was tested by more than 100 signatures in two classes and in a Science Festival. The verification equal
error rate result was 17.5% for using a computer mouse to produce signatures and about 9.5% for using the Tablet pen. It is hard to produce good signatures using a computer mouse; therefore the verification equal error rate was higher than that using the Tablet pen. However, those are very promising results as the VQ modeling technique was used. A better version using Markov modeling technique is implementing to achieve a lower verification equal error rate.

4. Conclusion

In this paper, we have presented Tablet PC-based applications supplied by Microsoft and developed by academic staff at universities around the world. We have also presented applications at our university. With the use of MS Visual Studio.NET, we have developed a signature verification system using fuzzy vector quantization technique for modeling and new background modeling technique for verification. More applications are in development at other organizations and universities.

Acknowledgements

The first author would like to acknowledge the support of the University of Canberra Multidisciplinary and Divisional Research Grants.

References