

GIẢI PHÁP NHẬP DỮ LIỆU TIẾNG VIỆT VIẾT TAY TRÊN TABLET PC

A SOLUTION FOR INPUTTING VIETNAMESE HANDWRITTEN TEXT ON TABLET PC

Nguyễn Đăng Quang, Đinh Công Doan

Đại Học Sư Phạm Kỹ Thuật TP.HCM

ABSTRACT

TabletPCs are quite familiar nowadays. Vietnamese handwriting recognition on Tablet PC however has not been supported so far. We developed a solution to input Vietnamese handwritten text on TabletPC by using the Microsoft Windows TabletPC API. The solution was proposed to replace the default TabletPC Input Panel (TIP) whenever users want to enter Vietnamese text on slate TabletPC or convertible TabletPC in slate mode.

I. GIỚI THIỆU

TabletPC (máy tính bảng điều khiển bằng bút) được giới thiệu từ năm 2001 bởi Bill Gates và Microsoft là công ty dẫn đầu trong lĩnh vực này. Phiên bản Windows XP for TabletPC là Hệ điều hành dành riêng cho Tablet PC được tích hợp các tính năng nhập liệu viết tay và ra lệnh bằng giọng nói. Trong phiên bản Windows Vista, tính năng nhận dạng chữ viết tay hoàn thiện hơn so với XP và đã cho kết quả nhận dạng với độ chính xác rất cao. Tính năng này lại càng tốt hơn nữa trên Windows 7 khi mà công nghệ giao tiếp người - máy hiện nay ngày càng phát triển.

Số lượng ngôn ngữ giao diện hỗ trợ bởi Windows 7 hiện nay đã lên đến 36. Tuy nhiên số ngôn ngữ được hỗ trợ nhận dạng chữ viết tay chỉ có 13 bao gồm English (US), English (UK), Japanese, Chinese (Traditional), Chinese (simplified), Korean, French, German, Italian, Dutch, Portugese (Brazil), Spanish và Swedish. Các gói ngôn ngữ này chỉ có thể cài đặt trên phiên bản Ultimate và vẫn chưa có tiếng Việt.

Sản phẩm vnOnline 1.0 – chương trình nhận dạng chữ viết tay on-line tiếng việt trên Tablet PC của Viện CNTT Hà Nội là sản phẩm duy nhất có giá trị thương mại tính đến thời điểm này nhưng không được phổ biến rộng rãi và chỉ hoạt động trên Windows XP.

Xuất phát từ nhu cầu nhập tiếng Việt viết tay trên các máy tính TabletPC ở Slate mode (chế độ

không có bàn phím), nhóm tác giả đã nghiên cứu thư viện lập trình Tablet PC SDK của Microsoft để xây dựng một bảng nhập liệu tiếng Việt viết tay. Đề tài không đề xuất giải thuật nhận dạng chữ viết tay mà chủ yếu tìm hiểu và khai thác thư viện nhận dạng chữ viết sẵn có. Bảng nhập liệu cho phép người sử dụng viết tay tiếng Việt theo kiểu gõ Telex từ bàn phím; ngoài ra để hỗ trợ cho người nhập, chương trình có thiết kế thêm một bàn phím ảo để nhập nhanh tiếng Việt bằng bút. Chương trình còn kết hợp tính năng nhắc từ ghép để tăng tốc độ nhập liệu.

II. Ý TƯỞNG THIẾT KẾ

Để thuận tiện cho việc nhập dữ liệu, bảng nhập liệu được thiết kế trong một cửa sổ luôn luôn nằm trên tất cả các cửa sổ khác trên màn hình.

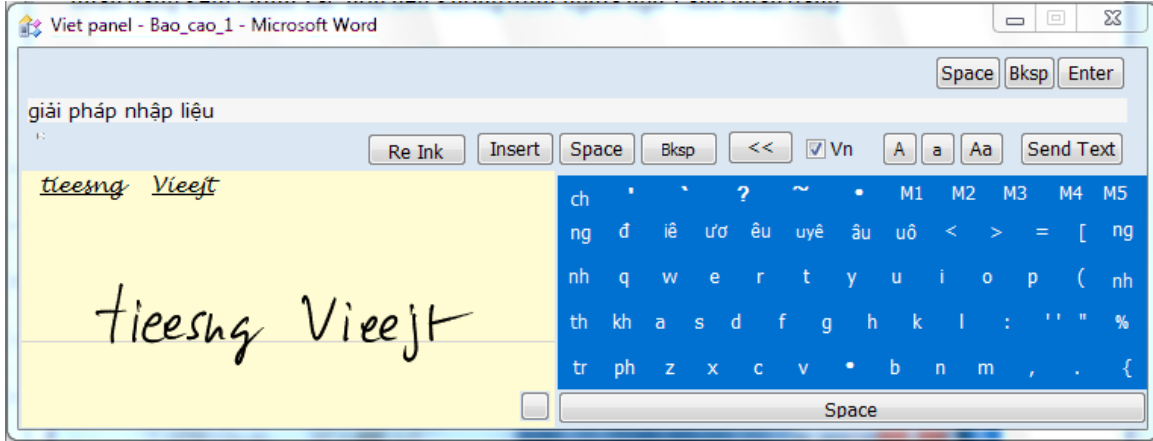
Kích thước của sổ cố định, có độ rộng nhỏ hơn độ phân giải nhỏ nhất phổ biến của các loại màn hình TabletPC để bảng nhập liệu luôn nằm gọn trong màn hình bất kể khi màn hình được xoay theo hướng dọc (Landscape) hoặc hướng ngang (Portrait).

Các độ phân giải màn hình phổ biến hiện nay: 1024x768, 1280x800. Vì vậy, độ rộng của sổ bảng nhập liệu được chọn là 768.

Phần nhập dữ liệu được chia thành 2 vùng:

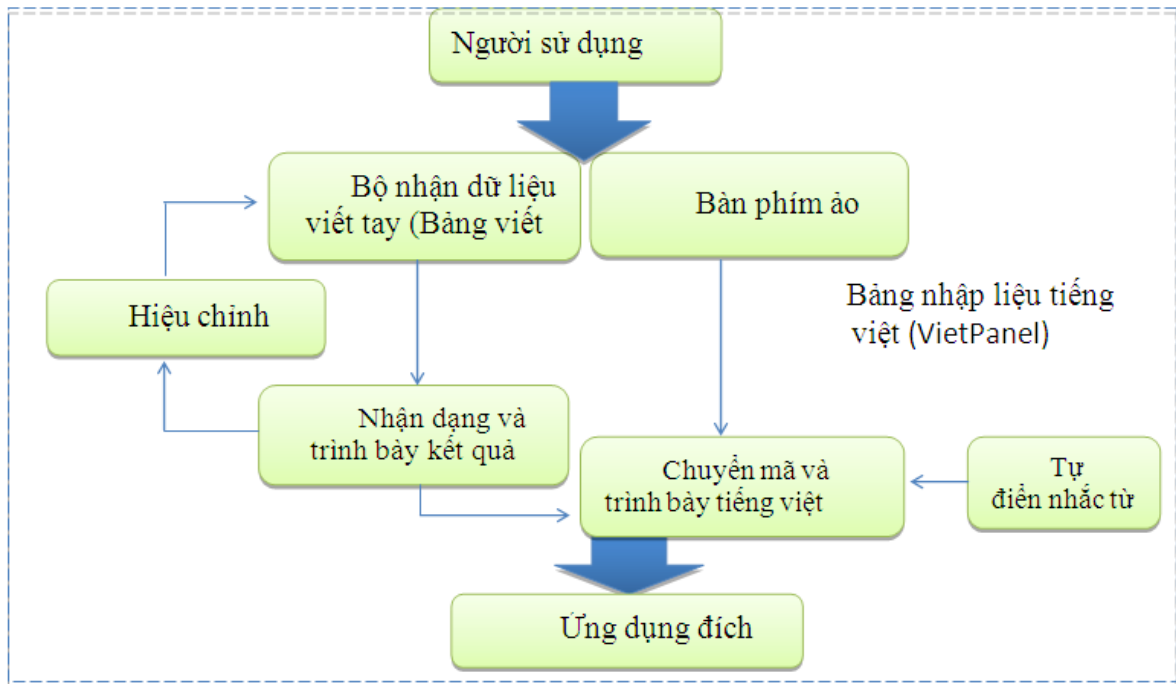
Vùng thu nhận nét viết tay (bên trái) và bàn phím ảo bên phải. Thiết kế này giúp người sử dụng có thể phối hợp viết và bấm nút phím. Bảng nhập liệu có thể mở rộng nếu người sử dụng có thói quen viết.

Số hàng phím cũng được cân nhắc sao cho đảm bảo tính hiệu quả khi nhập liệu, tính thẩm mỹ, cân đối với bảng viết bên trái và chiều cao của bảng nhập liệu không chiếm diện tích quá lớn trên màn hình. Giao diện của giải pháp như ở hình 1.



Hình 1: Giao diện của giải pháp

III. SƠ ĐỒ CHỨC NĂNG



Hình 2: Sơ đồ chức năng của giải pháp nhập dữ liệu tiếng việt

IV. THU NHẬN NÉT VIẾT VÀ NHẬN DẠNG

1. Cơ bản về TabletPC API

TabletPC API được định nghĩa trong TabletPC SDK cung cấp một thư viện lớp hoàn chỉnh để thu nhận tín hiệu từ bút (Ink Collector), quản lý nét viết (Ink management) và nhận dạng (Ink Recognizer).

Hoạt động thu nhận dữ liệu từ bút được thực hiện thông qua các đối tượng InkCollector hoặc InkOverlay. InkCollector thu nhận nét viết (Ink Strokes) từ một cửa sổ trên màn hình và chuyển dữ liệu vào đối tượng Ink. Ink Strokes sau đó được xử lý hoặc chuyển cho bộ nhận dạng (recognizer) để nhận dạng.

InkOverlay thường sử dụng trong các ứng dụng không nhận dạng, đó là các loại ứng dụng sử dụng Ink để vẽ hoặc ghi chú. Loại ứng dụng này thường chỉ quan tâm đến cỡ, dạng, màu sắc và vị trí của Ink.

Ink được thu nhận từ digitizer và được xử lý tương tự như dữ liệu thu nhận từ mouse. Lớp Ink được tự động khởi tạo khi một đối tượng Ink Collector hoặc InkOverlay được tạo. Trong Ink chứa một tập các nét mực (Strokes). Stroke được định nghĩa là một tập dữ liệu được thu nhận khi xảy ra một chuỗi liên tiếp sự kiện pen-down, pen-move và pen-up.

Dữ liệu Stroke chứa một tập các packet. Mỗi Packet là một tập dữ liệu mà thiết bị Tablet gửi cho ứng dụng tại mỗi điểm lấy mẫu. Dữ liệu này gồm tọa độ, độ ấn của, góc nghiêng của viết ... Các thông tin này chứa trong thuộc tính PacketDescription của đối tượng Stroke.

```
Renderer renderer = new Renderer();
```

```
Graphics panelGraphics = pictureBox1.CreateGraphics();
```

Xác định kích thước vùng viết theo InkSpace

```
Point sizeAsPoint = new Point(pictureBox1.Width, pictureBox1.Height);
```

```
renderer.PixelToInkSpace(panelGraphics, ref sizeAsPoint);
```

Xác định kích thước vùng vẽ theo InkSpace

Ink space sử dụng hệ tọa độ HIMETRIC. Trong hệ tọa độ này, mỗi đơn vị bằng với 1 đơn vị HIMETRIC. Mỗi đơn vị HIMETRIC tương đương 0.01 mm. Hệ tọa độ này có gốc (0,0) nằm ở góc trên bên trái màn hình, hướng trục x từ trái sang phải và hướng trục y từ trên xuống dưới.

Renderer là đối tượng trình bày nét mực trên màn hình hoặc thiết bị ngoài như máy in bằng các phương thức Draw hoặc DrawStroke.

RecognizerContext là đối tượng nhận dạng theo ngữ cảnh. RecognizerContext nhận dữ liệu vào là Ink Strokes, trả kết quả về qua đối tượng RecognitionResult. Guide, Factoids và Wordlist là các tham số định nghĩa ngữ cảnh giúp bộ nhận dạng cho kết quả chính xác hơn.

Kết quả nhận dạng trả về trong đối tượng RecognitionResult gồm một chuỗi có độ tin cậy cao nhất TopString và một bộ giá trị lựa chọn khác chứa trong đối tượng RecognitionAlternates

Bộ nhận dạng được kích hoạt bằng một trong 2 phương thức: đồng bộ hoặc không đồng bộ. Nhận dạng đồng bộ thực hiện bằng cách gọi phương thức Recognize, nhận dạng không đồng bộ được gọi bằng cách kích hoạt tiến trình nền BackgroundRecognizeWithAlternates.

2. Thiết kế bảng viết tay sử dụng Tablet API

Bảng nhập liệu viết tay được xây dựng từ đối tượng PictureBox. InkCollector được khởi tạo và gắn với đối tượng này để thu nhận tất cả nét viết của người sử dụng và chuyển cho bộ nhận dạng.

Vùng thu nhận nét viết cũng là Ink Renderer của ứng dụng. Renderer được khởi tạo.

```

Point marginsAsPoint = new Point(margin, bottomMargin);
renderer.PixelToInkSpace(panelGraphics, ref marginsAsPoint);
int marginInkSpace = marginsAsPoint.X;
int bottomMarginInkSpace = marginsAsPoint.Y;

```

Xác định vị trí đường midline trên vùng viết

```
int midlineHeight = (sizeAsPoint.Y - marginInkSpace - bottomMarginInkSpace) / 3;
```

Khởi tạo Ink Collector

```
myInkCollector = new InkCollector(pictureBox1.Handle);
```

Gán sự kiện khi có thay đổi nét viết và kích hoạt InkCollector

```
myInkCollector.Enabled = true;
myInkCollector.Stroke += Stroke_Event;
```

Giải pháp sử dụng phương pháp nhận dạng không đồng bộ: Hoạt động nhận dạng được chạy ở tiến trình nền – nhận dạng ngay kết quả mỗi khi có nét viết mới bằng cách sử dụng phương thức BackgroundRecognizeWithAlternates của RecognizerContext:

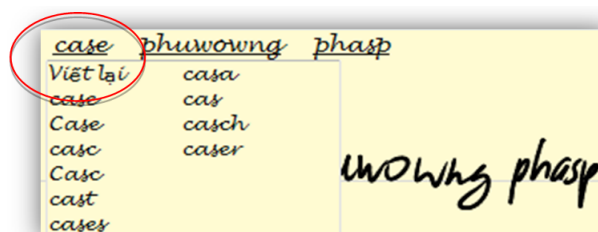
```
private void Stroke_Event(object sender, InkCollectorStrokeEventArgs e)
{
    theStrokes.Add(e.Stroke);
    theRecognizerConText.BackgroundRecognizeWithAlternates();
}

```

Đôi khi kết quả chứa trong TopString không phải là kết quả mong muốn, giải pháp của đề tài là nạp tất cả các lựa chọn khác vào trong một danh sách để người viết có thể lựa chọn hiệu chỉnh khi click tại từ (Hình 3)

```
listAlternates.Items.Clear();
foreach (RecognitionAlternate theAlternate in theRecognitionAlternates)
{
    listAlternates.Items.Add(theAlternate.ToString());
}

```



Hình 3: Click tại kết quả nhận dạng để lựa giá trị khác

3. Trình bày kết quả nhận dạng

Kết quả nhận dạng được thể hiện tức thời sau mỗi nét viết và trình bày ở phần trên của bảng viết tay (Hình 3). Tất cả các theo tác hiệu chỉnh, xóa, chèn được xử lý tại các đối tượng trên vùng này.

4. Xử lý từ nhận dạng sai

Kết quả nhận dạng sai thường do một trong 2 nguyên nhân (i) Từ nhận dạng chưa có trong từ điển, (ii) Nét viết không rõ ràng.

Cách xử lý của TabletPC Input Panel (TIP) mặc định (i) Sửa lại từ viết sai ở mức ký tự hoặc (ii) Xóa nét viết sai rồi viết lại.

Cách giải quyết của TIP mặc định rất tự nhiên, phù hợp với thói quen khi viết của người sử dụng. Tuy nhiên, thực tế cho thấy cách làm trên không hiệu quả vì việc hiệu chỉnh nét rất mất thời gian chưa kể đến kết quả không chính xác.

Giải pháp của đề tài là xóa hẳn từ viết sai rồi viết lại vào vị trí xóa. Lựa chọn xóa được thực hiện bằng cách click tại từ muốn xóa rồi chọn “Viết lại” trong danh sách các lựa chọn (Hình 3).

Để có thể xóa nét mực tương ứng với từ muốn xóa, thực hiện giải thuật tìm kiếm nhóm nét tương ứng với từ chọn, chứa vào đối tượng Strokes, sau đó xóa bằng phương thức Remove

```
RecognitionAlternate selectedAlternate =  
    (RecognitionAlternate)this.listAlternates.Items[1];  
Strokes deleteStrokes = selectedAlternate.Strokes;  
theRecognizerConText.Strokes.Remove(deleteStrokes);
```

5. Nâng cao độ chính xác kết quả nhận dạng

Độ chính xác của kết quả trong giải pháp được tăng cường bằng cách thiết lập các thuộc tính sau của RecognizerContext:

Guide: Xác định kích thước vùng nhận dạng bằng cách định nghĩa giá trị phù hợp cho RecognizerGuide của RecognizerContext. Vùng nhận dạng được định nghĩa có 1 dòng, 1 cột

Từ điển nhận dạng: Đây là thuộc tính quan trọng nhất quyết định độ chính xác của kết quả

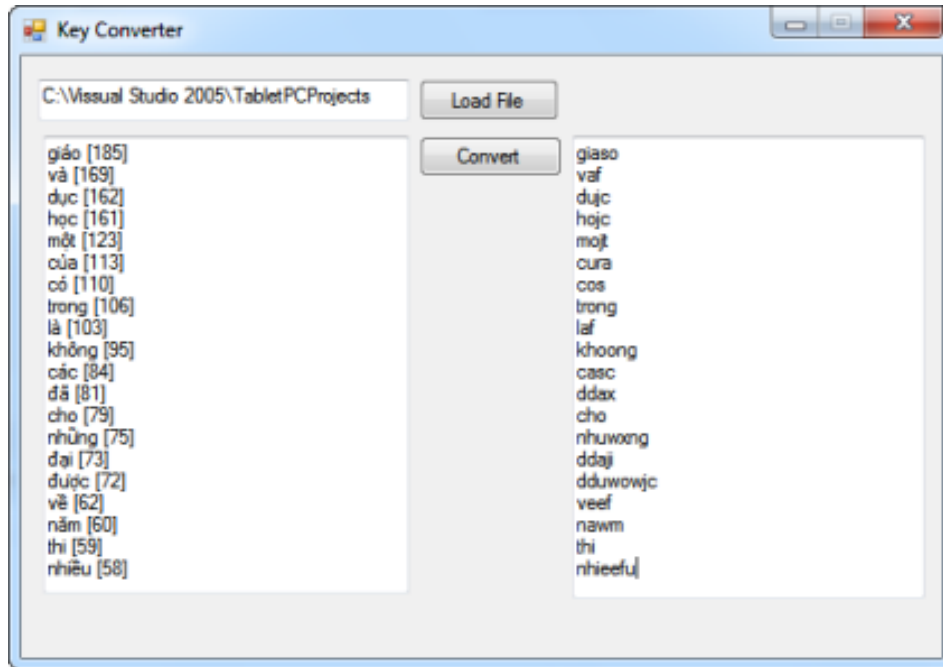
nhận dạng trong giải pháp. Kết quả nhận dạng chính xác sẽ giúp người viết có thể nhập liệu nhanh hơn vì không phải dừng lại để hiệu chỉnh. Do từ tiếng việt khi viết theo qui tắc gõ dấu Telex không có trong từ điển mặc định nên ngay cả đối với bảng nhập liệu có sẵn của Windows (Tablet PC Input Panel – TIP) việc viết đúng được một từ không có trong từ điển là hết sức khó khăn và mất nhiều thời gian. Từ điển nhận dạng được nạp vào Recognizer qua thuộc tính WordList:

```
WordList vnWords = new WordList();  
String[] myString = {“dd”, “cao”, “case”, “cawn”, “chuw”, “uwow”, “iee”,  
“duduwowjc”, “THCS”, “GS.TS”, “GD-DDT”, “TP.HCM”, “MSSQL”, “SQL”, “mySQL”,  
“mawst”, “VNDD”};  
foreach (String s in myString) {  
    vnWords.Add(s);  
}  
theRecognizerConText.WordList = myWords;
```

Bộ tự điển nhận dạng sử dụng trong đề tài được xây dựng từ một văn bản tổng hợp thuộc các lĩnh vực giáo dục, khoa học, y tế có 11770 từ.

Qua một thuật giải phân loại, số từ trong văn bản được xác định có 1427 từ tiếng việt. Danh sách

từ sau đó được chuyển sang từ viết theo cách gõ tiếng việt Telex (Hình 5) và nạp vào WordList. Tự điển từ được lưu trữ dạng văn bản, mỗi từ trên 1 dòng để thuận tiện cho việc bổ sung các từ mới, nhất là các từ viết tắt hoặc từ chuyên ngành.



Hình 5: Chuyển từ tiếng việt sang từ viết theo kiểu gõ Telex

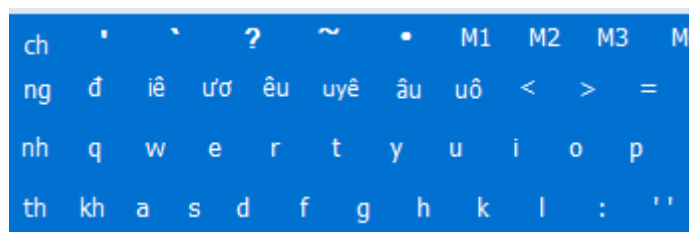
Bằng cách làm như trên, có thể xây dựng tự điển từng loại lĩnh vực và nạp vào ứng dụng cho phù hợp với nhu cầu nhận dạng.

V. THIẾT KẾ BÀN PHÍM ẢO

Dùng nhập dữ liệu bằng bút trong trường hợp bộ nhận dạng không thể nhận dạng được nét viết. Bàn phím ảo được thiết kế tương tự như bàn phím chuẩn đối với ký tự. Tuy nhiên, để tăng tốc độ nhập, bàn phím được thiết kế lại như sau:

- Thêm một số phím đại diện cho các cụm nguyên âm trong tiếng việt như iê, ươ, uă, uâ, oă, uô, uyê, âu, êu.

- Thêm một số phím đại diện cho các phụ âm ghép đầu từ: nh, ch, kh, th, tr, ng, ph, đ. Trong nhóm phụ âm ghép này, các cụm phụ âm nh và ng có thể xuất hiện ở đầu và cuối từ nên được bố trí ở cả 2 phía của bàn phím (Hình 6).
- Các phím gõ dấu tiếng việt: sắc, huyền, hỏi, ngã, nặng được bố trí ở hàng trên cùng. Trong quá trình thử nghiệm, nhận thấy phím nặng đặt ở dòng trên sẽ khó khăn hơn khi bấm bằng bút do phải di chuyển ngược tay lên phía trên. Vì vậy, bàn phím được bố trí thêm một phím nặng nữa ở dòng phím dưới cùng (Hình 6).



Hình 6: Vị trí các phím ng, nh và dấu nặng

- Các bộ ký hiệu ngoặc đôi: [], { }, (), “ ”, ‘ ’ chỉ được đại diện bằng ký tự mở [, { , (, “ , ‘ . ký tự đóng tương ứng sẽ được chèn tự động sau dấu mở và con trỏ chèn sẽ được lập trình để nằm giữa 2 dấu ngoặc nhằm thuận tiện cho việc chèn nội dung nhập ngay sau đó.
- Nhóm phím tắt M1, M2, M3, M4, M5: giúp chèn nhanh những từ dài hoặc lặp lại nhiều khi nhập, ví dụ như: “những”, “người”, “trường”, “phương” ... Nhóm nút này được lập trình để người sử dụng có thể định nghĩa tùy thuộc vào nội dung loại văn bản sẽ nhập hoặc tùy thuộc vào thói quen của người nhập.
- Nhóm phím soạn thảo
 - Re Ink: Xóa nét viết, viết lại.
 - Insert: Chèn kết quả nhận dạng vào khung nội dung nhập
 - Space: Chèn một khoảng trắng tại con trỏ vào khung nội dung nhập
 - Bksp: Xóa một ký tự trước con trỏ trong khung nội dung nhập
 - <<: Xóa về phía trước 1 từ trong khung nội dung nhập
 - A: Chuyển thành chữ hoa
 - A: Chuyển thành chữ thường

- Aa: Chuyển chữ đầu từ thành hoa
- Nhóm phím thao tác trên ứng dụng
 - Send Text: Chuyển nội dung nhập vào ứng dụng
 - Nhóm phím hiệu chỉnh trực tiếp nội dung trên ứng dụng: Space (Khoảng trắng), Bksp (xóa ký tự trước) và Enter (xuống dòng).

Việc bố trí các nhóm phím iê, ươ, uă, uâ, oă, uô, uyê, âu, êu và nh, ch, kh, th, tr, ng, ph, đ sẽ dựa trên kết quả thống kê số lần xuất hiện trong một số văn bản được chọn làm mẫu.

Nhóm thực hiện đã thống kê 2 lần:

- Lần 1 thống kê trên các văn bản thuộc lĩnh vực giáo dục có 10974 từ
- Lần 1 thống kê trên các văn bản tổng hợp thuộc các lĩnh vực giáo dục, khoa học, y tế có 11770 từ.

Tổng hợp kết quả thống kê

Dữ liệu vào:

| | Lần 1 | Lần 2 |
|------------|-------|-------|
| Tổng số từ | 10974 | 11770 |
| Số loại từ | 1355 | 1880 |

Kết quả thống kê:

| Cụm nguyên âm | Lần 1 | Lần 2 |
|---------------|-------|-------|
| iê | 24 | 22 |
| ươ | 10 | 13 |
| êu | 9 | 13 |
| uyê | 7 | 9 |
| âu | 5 | 7 |
| uô | 3 | 6 |
| uâ | 1 | 3 |
| uă | 1 | 1 |
| oă | 1 | 1 |

| Cụm phụ âm | Lần 1 | Lần 2 |
|------------|-------|-------|
| ng | 208 | 285 |
| đ | 100 | 144 |
| ch | 93 | 116 |
| nh | 92 | 126 |
| th | 88 | 99 |
| tr | 58 | 73 |
| kh | 37 | 58 |
| ph | 36 | 50 |
| | | |

Bảng 1: Thống kê mức độ sử dụng các cụm phím phụ âm và cụm nguyên âm

Kết quả thống kê cho thấy số lần sử dụng của các nhóm phím không khác nhau nhiều:

- Nhóm cụm nguyên âm: iê > ươ > êu > uyê > âu > uô
- Nhóm cụm phụ âm: ng > ch > đ > nh > th > kh > tr > ph

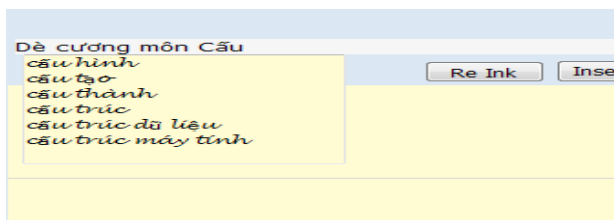
Suy ra:

- Vị trí của các phím cụm nguyên âm sẽ được xếp từ trái sang phải theo mức độ thường xuyên sử dụng giảm dần.
- Vị trí của các phím cụm phụ âm sẽ được xếp từ trên xuống theo mức độ thường xuyên sử dụng giảm dần.

Ngoài ra, phím Space cũng được thiết kế rộng để dễ bấm bằng bút (Hình 6).

VI. TỰ ĐIỂN NHẮC TỪ TIẾNG VIỆT

Nhằm mục đích tăng tốc độ nhập liệu, bảng nhập liệu sử dụng một bộ tự điển nhắc từ ghép tiếng Việt. Khi người gõ khoảng trắng sau từ nhập, nếu có kết quả tìm kiếm trong tự điển, một danh sách sẽ xuất hiện cho người viết chọn lựa (Hình 7).



Hình 7: Nhắc từ ghép tiếng Việt

Bộ tự điển nhắc từ có khoảng 11,000 từ ghép tiếng Việt phổ thông unicode, được cung cấp bởi tác giả Hồ Ngọc Đức [2]. Tự điển được lưu dạng Text, mỗi cụm từ trên một dòng. Trong quá trình sử dụng, người dùng có thể tùy ý bổ sung hoặc điều chỉnh ví dụ như nạp các cụm từ chuyên ngành hoặc những từ ghép mới chưa có trong tự điển.

Nhằm tăng tốc độ tìm kiếm và lưu trữ hiệu quả, tự điển từ được lưu trữ trong cấu trúc dữ liệu SortedDictionary dựa trên cây AVL cung cấp bởi

Maksim Goleta [3].

Chuyển Đổi Sang Tiếng Việt

Tiếng Việt có dấu được tạo ra bằng cách sử dụng thư viện gõ dấu tiếng Việt Vietpad cung cấp tại địa chỉ <http://vietpad.sourceforge.net>

Để sử dụng Vietpad trong chương trình, khởi tạo bộ xử lý phím trên đối tượng sẽ nhận kết quả (txtResults) rồi đăng ký xử lý sự kiện khi có phím gõ (OnKeyPress)

```
VietKeyHandler keyHandler = new
VietKeyHandler(txtResults);
```

```
txtResults.KeyPress += keyHandler.
OnKeyPress;
```

Chọn phương pháp gõ ký tự Telex

```
VietKeyHandler.InputMethod =
InputMethods.Telex;
```

```
VietKeyHandler.VietModeEnabled = true;
```

Kết quả nhận dạng được chuyển đổi sang tiếng Việt có dấu bằng cách gửi từng ký tự vào TextBox kết quả (txtResults) sử dụng hàm SendWait để giả lập thao tác gõ phím.

VII. ĐÁNH GIÁ

Hiệu quả nhập liệu của giải pháp được đánh giá bằng cả 2 phương pháp: lý thuyết và thực nghiệm trên cả 2 phương thức nhập: viết tay và bấm phím.

Đối với bảng viết tay: Thực nghiệm đếm số nét viết (Stroke) trung bình trên 3 đoạn văn bản có trung bình 30 từ. Số nét viết khi viết bình thường ít hơn số nét viết khi viết bằng giải pháp khoảng 12% (132 so với 148 nét).

Đối với bàn phím ảo: Hiệu quả của bàn phím ảo được đánh giá bằng tỷ lệ số lần bấm phím trên bàn phím của TabletPC Input Panel (TIP) so với số lần bấm trên bàn phím của giải pháp (VietPanel). Bảng 2 so sánh số lần bấm trên các cụm phụ âm và nguyên âm được rút gọn.

| | VietPanel | Thường | TIP | | VietPanel | TIP |
|------|-----------|--------|-----|-----|-----------|-----|
| (1) | (2) | (3) | (4) | (5) | (6) | (7) |
| iê | 1 | iee | 3 | ch | 1 | 2 |
| ươ | 1 | uwow | 4 | ng | 1 | 2 |
| êu | 1 | eeu | 3 | đ | 1 | 2 |
| uyê | 1 | uyee | 4 | nh | 1 | 2 |
| âu | 1 | aau | 3 | th | 1 | 2 |
| uô | 1 | uoo | 3 | kh | 1 | 2 |
| | | | | tr | 1 | 2 |
| | | | | ph | 1 | 2 |
| Cộng | 6 | | 20 | | 8 | 16 |

Bảng 2: Số phím bấm trên bàn phím ảo của giải pháp so với bàn phím ảo của TIP

Nếu xét trên văn bản sử dụng thống kê khi thiết kế bàn phím ảo (phần V), chỉ xét riêng các loại chuỗi ký tự cần nhập ở cột (1) và cột (5). Số lần bấm phím được thống kê trong bảng 3 và 4.

| | Lần 1 | Lần 2 | VietPanel | TIP | Lần 1 | | Lần 2 | | Trung bình | |
|-----|-------|-------|-----------|-----|-----------|-----|-----------|-----|------------|-------|
| | | | | | VietPanel | TIP | VietPanel | TIP | VietPanel | TIP |
| iê | 24 | 22 | 1 | 3 | 24 | 72 | 22 | 66 | 23 | 69 |
| ươ | 10 | 13 | 1 | 4 | 10 | 40 | 13 | 52 | 11.5 | 46 |
| êu | 9 | 13 | 1 | 3 | 9 | 27 | 13 | 39 | 11 | 33 |
| uyê | 7 | 9 | 1 | 4 | 7 | 28 | 9 | 36 | 8 | 32 |
| âu | 5 | 7 | 1 | 3 | 5 | 15 | 7 | 21 | 6 | 18 |
| uô | 3 | 6 | 1 | 3 | 3 | 9 | 6 | 18 | 4.5 | 13.5 |
| uâ | 1 | 3 | 1 | 3 | 1 | 3 | 3 | 9 | 2 | 6 |
| uã | 1 | 1 | 1 | 3 | 1 | 3 | 1 | 3 | 1 | 3 |
| oã | 1 | 1 | 1 | 3 | 1 | 3 | 1 | 3 | 1 | 3 |
| | | | | | 61 | 200 | 75 | 247 | 68 | 223.5 |

Bảng 3: Số lần bấm phím trên TIP so với VietPanel đối với nội dung ở cột 1

| | Lần 1 | Lần 2 | VietPanel | TIP | Lần 1 | | Lần 2 | | Trung bình | |
|----|-------|-------|-----------|-----|-----------|------|-----------|------|------------|------|
| | | | | | VietPanel | TIP | VietPanel | TIP | VietPanel | TIP |
| ch | 208 | 285 | 1 | 2 | 208 | 416 | 285 | 570 | 246.5 | 493 |
| ng | 100 | 144 | 1 | 2 | 100 | 200 | 144 | 288 | 122 | 244 |
| đ | 93 | 116 | 1 | 2 | 93 | 186 | 116 | 232 | 104.5 | 209 |
| nh | 92 | 126 | 1 | 2 | 92 | 184 | 126 | 252 | 109 | 218 |
| th | 88 | 99 | 1 | 2 | 88 | 176 | 99 | 198 | 93.5 | 187 |
| kh | 58 | 73 | 1 | 2 | 58 | 116 | 73 | 146 | 65.5 | 131 |
| tr | 37 | 58 | 1 | 2 | 37 | 74 | 58 | 116 | 47.5 | 95 |
| ph | 36 | 50 | 1 | 2 | 36 | 72 | 50 | 100 | 43 | 86 |
| | | | | | 712 | 1424 | 951 | 1902 | 831.5 | 1663 |

Bảng 4: Số lần bấm phím trên TIP so với VietPanel đối với nội dung ở cột 1

Tổng hợp kết quả từ bảng 3 và 4, số lần bấm phím trên TIP trung bình là $223.5 + 1663 = 1886.5$ lần so với $68 + 831.5 = 899.5$ lần của VietPanel, giảm đi khoảng phân nửa số lần bấm phím (*).

Để đánh giá bằng thực nghiệm, nhóm sử dụng một đoạn văn 26 từ. Kết quả thực nghiệm trình bày trong bảng 5

| Stt | Phương thức nhập | Lần 1 | Lần 2 | Lần 3 | Lần 4 | Lần 5 | Trung bình |
|-----|--|-------|-------|-------|-------|-------|------------|
| 1 | Gõ bằng bàn phím | 70 | 43 | 40 | 42 | 41 | 47.2 |
| 2 | Nhập bằng bàn phím của TIP + unikey | 130 | 100 | 79 | 77 | 78 | 92.8 |
| 3 | Nhập bằng bàn phím của VietPanel (không sử dụng nhắc từ) | 147 | 101 | 104 | 93 | 94 | 107.8 |
| 4 | Nhập bằng viết (không nhắc từ) | 250 | 140 | 118 | 125 | 101 | 146.8 |
| 5 | Nhập bằng viết, có sử dụng tự điển nhắc từ | 120 | 93 | 88 | 80 | 75 | 91.2 |

Bảng 5: Thử nghiệm nhập dữ liệu trên VietPanel (đơn vị tính: giây)

Một số nhận xét từ số liệu ở bảng 5:

Thời gian nhập liệu bằng bàn phím của VietPanel không nhanh hơn rõ rệt so với nhập bằng bàn phím của TIP và Unikey, điều này có vẻ không phù hợp với kết luận (*). Tuy nhiên tốc độ chậm khi nhập bằng VietPanel được giải thích bằng thói quen sử dụng bàn phím. Khi đã làm quen với bàn phím VietPanel, thời gian nhập sẽ nhanh hơn nhiều.

Thời gian nhập bằng viết không nhanh hơn đáng kể so với nhập bằng bàn phím. Tuy nhiên cảm giác nhập thoải mái hơn khi phải bấm bằng bút trên bàn phím.

Khuyến nghị :

Thử nghiệm cho thấy không nên viết liên tục nhiều từ vì bộ nhận dạng phải nhận dạng lại từ đầu mỗi khi có nét thay đổi, người viết phải chờ cho đến khi bộ nhận dạng hoạt động xong mới có thể viết nội dung kế tiếp.

Nên viết mỗi lần tối đa 4 từ và khai thác từ điển nhắc từ sẽ cải thiện tốc độ nhập (trường hợp 5).

VIII. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết quả của giải pháp là bảng nhập liệu viết tay bằng bút có thể phối hợp phương thức nhập bằng cách viết tay hoặc bấm bút trên bàn phím

ảo. Ứng dụng đã được sử dụng trong công việc hằng ngày khi cần nhập dữ liệu trên Tablet PC ở Slate mode. Khi đã làm quen với việc sử dụng bút cũng như bàn phím ảo, người sử dụng có thể phối hợp cả 2 phương thức để nhập liệu một cách thoải mái.

Hạn chế của giải pháp:

- Người sử dụng phải mất thời gian để làm quen với cách viết và bàn phím ảo,
- Số nét viết khi viết tăng khá nhiều so với cách viết có dấu bình thường,

Hướng phát triển

- Khai thác thư viện InkAnalyzer thay cho InkCollector, RecognizerContext để cải thiện tốc độ nhận dạng, cho phép người sử dụng viết nhanh hơn và viết liên tục nhiều từ hơn.
- Định nghĩa một tập Gesture tương tự phương pháp viết tốc ký để tăng tốc độ viết tay.

TÀI LIỆU THAM KHẢO

- [1] Tài liệu TabletPC SDK của Microsoft <http://msdn.microsoft.com/enus/library/ms837816.aspx>
- [2] Tự điển từ ghép tiếng Việt của Hồ Ngọc Đức <http://www.informatik.uni-leipzig>.

de/~duc/software/misc/wordlist.html.

[3] Mã nguồn của Maksim Goleta <http://goletas.com/csharp-collections> .

[4] L.R.B. Schomaker, “From Handwriting Analysis to Pen-Computer Applications”,

Electronic and Communicaiton Eng. Pp. 93-102, June 1998.

[5] M.K. Brown and S. Ganapathy, “Preprocessing Techniques for Cursive Script Word Recognition”, Pattern Recognition, Vol.16, pp. 447-458, 1983.