Pen Computing

CSEP 510
Lecture 6, February 12, 2004
Richard Anderson

Announcements

Outline
- Pen computing
- Stylus based input
- Tablet PC
- Inking
- Recognition

Pen Computing
- Wide range of devices fall under pen computing
- Classification by size
  - Hand held
  - Tablet
  - Large display

Key features
- Stylus for continuous input
- Direct manipulation with the display

History of Pen Computing
- Sketchpad (1963)
- GRiD GRIDPad (1989)
- GO Pen Point (1991)
- Microsoft Windows for Pen Computing (1992)
- Apple Newton (1993)
Handwriting test

- Write the following phrase in journal:
  The quick brown fox jumped over the lazy dog
- Convert to text
  - Edit -> Select All
  - Actions -> Convert Handwriting to Text...
- For each word, record status
  - correct, correct?, incorrect?, incorrect

Text Entry

Reco

Stylus Input

- Specialized alphabets
- Issues
  - Ease of recognition (accuracy)
  - Ease of learning
  - Speed
- Human readability and archiving are not issues
  - Write-only

Graffiti (Palm)

More
Graffiti
- Mostly single stroke
- Close to standard alphabet (learnability)
- Write only
- Location written for additional meaning

Quikwrite [Perlin, NYU]

Homework assignment
- Implement quikwriting

Flow Menu
- Use movement through octants for control information

Interaction with direct manipulation

Cirrus [Mankoff, GaTech]
What are the limits to input speed

- Key stroke rates
- Pointing rates (from Fitts’ law)
- Information rate

Information theory

- Binary code for \{a, b, c, d\}
  - 00, 01, 10, 11
  - bad = 010011
- Binary code for \{a, b, c, d\}
  - 0, 10, 110, 111
  - bad = 100111
- Letter probabilities
  - 0.5, 0.25, 0.125, 0.125

Information content

- Information content
  - Average length of optimally compressed text
- Content of English
  - 1 bit per character
  - Keyboards inefficient by a factor of 6
- Faster typing
  - Input in optimal code
- Fitts’ law measurements
  - Information rate of pointing – 14 bits per second
  - Input rate – 170 words per minute
  - With just one finger!

Dashier

- Pen based selection (as cirrus)
- Dynamically change layout
- Prediction of next character

Mode Problem

- Cognitive difficulties in remembering / keeping track of modes
  - Which mode?
  - Remapping operations
  - Retaining mode across context switch
- But modes are very useful
  - Efficient use of limited input controls
- Not all modes are the same
  - Shift key vs. Caps Lock
  - Mouse move vs. mouse drag
  - Pen color

Pen mode issues

- Advantages of pen
  - Single, well understood device
  - Low cognitive requirements for use
  - High precision
  - Ideal for two dimensional curve input
- Disadvantages
  - Limited number of distinct operations
  - Continuous operations make discrete actions more difficult
  - Conflict between input and control
Pen mode options
- Pen buttons
- Separate pens
  - Eraser
  - Multiple colored pens for smart board
- Short term modes
  - Distinct writing areas

Implicit modes
- Ink recognition for modes
- Requires overloading some writing
- Scratch out gesture
  - Inferred mode (Saund – Lank)
    - Infer meaning of operations
    - Soft prompt on ambiguous strokes

Tablet PC
- Basic slate hardware
  - Full fledged PC
  - Hi DPI Screen
- Portable
  - Light weight, 2 to 5 pounds
  - Wireless

Tablet vs. Laptop
- Convertible form factor
- Dual purpose device
- Many activities are best done with a keyboard
- Compromises to support both slate and laptop use

Digitizers
- Resistive digitizers
  - Respond to pressure
- Electromagnetic digitizer
  - Detect pen at distance
  - Hardware below LCD screen
  - Can detect rotation and tilt
- Digitizer distortion

Display Hardware
- Protective cover for LCD
  - Avoid psychedelic color blooming
  - Damage to LCD
- Glass covering
  - Hard surface
  - Slippery
  - Thickness
Parallax
- Difference between real and apparent location due to viewing angles

Alignment problems
- Stone ink thought to be valuable
- What are chunky ballpens!
- Hygienic Error - Why???
- Time to Archive True (?)
- Dye it ReJoin!
- Cool in removable - length!

Pens
- Highly personal item
- Close to ballpoint in size and weight (?)

Pen as a Mouse
Positioning
- Pen positioning can be arduous work if UI requires targeting all over the display
  - Mouse can move the cursor far without much arm/hand movement, but the pen requires a lot of arm/hand movement
  - Menus and toolbars are typically at the top of a window; editing often occurs mid-way or toward the bottom
  - Lots of physical arm/hand movement results – a real pain for users
    - More local UI is desirable (e.g. context menus)

Pen as a Mouse
Clicking
- Clicking with a pen is tough
  - Legacy applications typically assume during a click the mouse doesn't move
  - Pen taps are more like little strokes or stabs because of pen skidding and high-precision digitizers
  - Detecting the difference between tap and a drag is an interesting problem!
- Double-clicking is even tougher
  - Quick motion means sloppier result
- Right-clicking is even tougher

Pen as a Mouse
Targeting guidelines
- Cursor feedback
  - Bigger, easily-targeted controls
  - Generous tap, double-click, and hover tolerances
  - Keep related objects in proximity
Hover
- Hovering still with a pen is tough
  - While in-air, control of a pen is considerably worse than when the pen is touching a surface
  - Mouse is intrinsically static, pen is not
  - Many applications typically assume cursor must be perfectly still for e.g. tooltips to appear
  - Software help needed to "smooth" hover location of cursor

Target and selection
- Buttons, Targets, Splitters, Scroll bars need to be bigger
- Digitizer inaccuracy
- Parallax
- Pen movement
- Mobility
- Pen slippage
- Obstructions

Pen UI Examples

Obstructions and handedness
- Hand blocks the screen
- Accommodate left and right handedness
  - Menu direction
  - Context menus
- Difficulties at the edge of the screen

Screen orientation
- Landscape vs. Portrait mode
- Surprisingly big difference in feel of applications
- Tablet PC requires rapid orientations switch
- Many standard desktop apps not designed for portrait mode

Digital Ink
- Writing needs to be real time
  - Inking must not lag
- Ink should look smooth
  - No "jaggies" -> antialiased
  - No straight lines -> curve-fitted
- Use pen pressure information
  - Vary stroke width (more pressure means wider stroke)
- Support pen tips
  - Round/ballpoint vs. rectangular/ highlighter
Under the hood

Ink Representation

- Stroke representation
  - Array of ink packets
  - Ink packets in HIMETRIC Coordinates
  - Sampled ~150 times per second
  - Ink may have pressure and other properties
  - Stroke information – color, pen nib, transparency
- Ink Represntation
  - Array of strokes

Interpolation

Sample code

```csharp
void BroadcastStroke(Stroke stroke)
{
    Ink.Ink ink = new Ink.Ink();
    Ink.Ink.Strokes strokes = stroke.Ink.CreateStrokes(new int[] {stroke.ID});
    ink.AddStrokesAtRectangle(strokes, strokes.GetBoundingBox());
    SendObject(new IncomingStroke(ink));
}

private void ReceiveStroke(IncomingStroke incomingStroke)
{
    Ink.Ink.Scribble s = GetOverlay().InkScribble;
    Strokes strokes = incoming.Stroke.Ink.Strokes;
    lock(this)
    {
        s.Ink.AddStrokesAtRectangle(strokes, strokes.GetBoundingBox());
    }
}
```

Hello World

Form Code

```csharp
private InkCollector inkCollector;
private Recognizers recognizers;
public Form1()
{
    InitializeComponent();
    this.inkCollector = new InkCollector(this.Handle);
    this.recognizers = new Recognizers();
}
```
Event Code

```csharp
private void ClearInk()
{
    this.inkCollector.Enabled = false;
    this.inkCollector.Ink = new Ink();
    this.Invalidate();
    this.inkCollector.Enabled = true;
}
private void OnReco(object sender, System.EventArgs e)
{
    this.textBox1.Text = this.inkCollector.Ink.Strokes.ToString();
}
```

Limitations of digital ink

- Not real enough!
- Uniform, solid color
- Fails to show layering, speed, timing . .

Inking examples

```
Entropy(S) \equiv -p_0 \log_2 p_0 - p_1 \log_2 p_1 \\
```

Enhanced Ink

```
Hello World
```

Boundary added to ink

```
Hello World! \times \times
```

Fading ink

```
Hello World
Hello World
Hello World
```
Gestures
- Gestures need precise tuning
  - Trade-off between accidental activation vs. not getting when wanted
  - False activations are annoying and distracting to the task!
- Guideline
  - Use gestures guardedly
  - Error on the side of having "zero" incidence of false activation
  - Non-destructive consequences are better

Inking regions
- Users have an initial expectation that they can write digital ink anywhere
- Guideline
  - Communicate clearly where users can ink
  - Ink enabled controls should be self-evident

Recognition
- Handwriting recognition is highly variable by person
  - Errors are expected
  - Perceived good or bad handwriting effects expectation of accuracy
- Guideline
  - Be realistic about recognition accuracy rate, don’t rely heavily on it for authoring

Handwriting results

<table>
<thead>
<tr>
<th></th>
<th>quick</th>
<th>brown</th>
<th>fox</th>
<th>jumped</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Correct?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incorrect?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incorrect</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tablet PC Usability Observations
- Buttons
  - Already covered
- Toshiba symbol commander
- Compaq / HP Q-Menu
- Auto rotate
- Toshiba button misfires by putting pen in holder
- Compaq pen failure – leaky pen