Pen-Based User Interfaces

Lecture #1: Introduction
Fall 2009
Joseph J. LaViola Jr.

Instructor

Professor – Joseph J. LaViola Jr.
Email – jjl@cs.ucf.edu
Office Hours – Tues. 4:00pm – 5:30pm
Wed. 6:00pm – 7:00pm
Office is Harris 321

Website will have all required info
www.eecs.ucf.edu/courses/cap6105/fall09/


**Class Goals**

- Provide foundation for pen-based user interface research and development
- Learn to critique research papers
- Speaking and presentation skills
- Start of master’s projects and PhD dissertations
- Possible publications
  - Advanced Visual Interfaces 2010
  - Sketch-based Interfaces and Modeling 2010
  - User Interface Software and Technology 2010
  - SIGGRAPH 2010

**Reference Books**

*WPF Recipes in C# 2008: A Problem-Solution Approach*

Sam Noble, Sam Beurton, and Allen Jones

Apress
Grading

Assignment 1  10%
Assignment 2  10%
Assignment 3  10%
Assignment 4  10%
Paper discussion 5%
Paper presentations 5%
Final Project  50%

Final Projects

- Encourage 2 person teams
- Must have research component
- Everyone must write and get approved a project proposal
- Final Project write up required
- DEMO DAY!!!! – December 14, 2009
- More on Wednesday – August 26th
Class Structure (see syllabus for details)

- Lectures
  - fundamentals of pen computing
  - sketch-based interfaces
- Paper discussions
  - 3 or 4 papers
  - students lead discussion
- Student paper presentation
  - 25 minute presentation
- Final project update sessions

Tools

- Tablet PC lab – Harris 208
  - will meet there sometimes
  - 12 HP Tablet PCs
    - 1.83 GHz Dual Core
    - 2GB memory
    - Windows XP Tablet PC Edition
  - key access to room
- Visual Studio 2008
  - C#
  - Windows Presentation Foundation
  - starPad SDK
starPad SDK

www.starpad.org

Collaboration and Late Policy

- Collaboration encouraged
  - do your own work on assignments
  - cheating = BAD!!!
- All assignments must be handed in on time
  - assignments – by 11:59pm on due date
Sketching and Gestures

- What is Sketching?
  - to make a hasty or undetailed drawing or painting often made as a preliminary study (dictionary)
- What is a Gesture?
  - the act of moving the limbs or body as an expression of thought or emphasis (dictionary)
    - not focusing on this type of gesture
    - interested in 2D pen, finger, and mouse-based gestures
- Gestures are like simple sketches

Pen-Based Interfaces

- Interaction stylus (2D) or finger
- Strokes for the computer to interpret
  - commands (gestural UI)
  - drawings
  - symbols, words, mathematics
- Mimic pencil and paper
- Inference and ambiguity

(ChemPad 2007)
**Historical Perspective (60s and 70s)**

- **SketchPad** (Sutherland 1963)
- **Architecture-By-Yourself** (Weinzapfel & Negroponte 1976)
- **HUNCH** (Herot 1976)
- **Logic Diagrams** (Sutherland 1966)
- **Math Reco** (Anderson 1967)

**Historical Perspective (80s and 90s)**

- **Wang Freestyle** (1988)
- **GRIDPad** (1989)
- **PenWindows** (1991)
- **GO +PenPoint** (1991)
- **Slate** (1992)
- **Anoto** (1999)
- **Newton** (1993)
- **Palm Pilot’s Graffiti** (1994)
- **CrossPad** (1999)
RIP – (adapted from Bill Buxton)

Freestyle +
Grid +
Pen for Windows +
GO +
Slate +
Newton +
Crosspad +
Anoto (1999)

Today

- Much improved hardware support
  - Tablet PC
  - Digitizers
    - Wacom Cintiq
    - Smartboard
- Much improved software support
  - Tablet SDK
    - handwriting recognition
    - speech recognition
  - character recognizers
- Better recognition algorithms
  - machine learning (use those cycles!)
Moving Forward

- Multi-touch is now the rage
  - large screens
  - laptops (Windows 7)
  - iPhone
- Interesting questions
  - killer app?
  - what is multi-touch good for?
  - how to integrate pen and multi-touch together?

A Sketch Input Continuum

Ambiguity level refers to sketch interpretation difficulty and domain generality
Pen-Based Applications

- 2D/3D Graphics
- UI Prototyping
- Animation
- Note Taking/Annotation
- Symbol/Word/Math Recognition
- Mathematical Sketching
- Etc…

Conceptual 2D Design

Electronic Cocktail Napkin
Intelligent Sketching Paper

(Gross 1994)
Character and Mathematical Expression Recognition

\[\lambda(k,l) = \sqrt{(4k+l)^2 + (4k+l)^2}\]
\[A_{(k,l)}(x,g) = \frac{\cos((2k+1)\pi x) \sin((2k+1)\pi g)}{(2k+1)(2k+1)}\]
\[u(x,y,t) = \frac{1}{\pi} \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} A_{(k,l)}(x,y) e^{-\lambda^{k+l}(x,y) t}\]
\[x(t) = 2 \sin(t^2)\]

3D Modeling

SKETCH
(Zeleznik et al. 1996)
TEDDY
(Igarashi et al. 1999)
Parameterized Object Sketching
(Yang et al. 2005)
Musical Score Creation

Music NotePad (Forsberg et al. 1998)

User Interface Prototyping

SILK (Landay and Myers 1995)
Simulation

Sim-U-Sketch
(Kara and Stahovich 2004)

VibroSketch
(Kara et al. 2004)

Mathematical Sketching

(LaViola and Zeleznik 2004)
Electronic Whiteboard Systems

Tivoli
(Pedersen et al. 1993)

Flatland
(Mynatt et al. 1999)

Animation

Motion Doodles
(Thorne et al. 2004)
Pen UI Resources (1)

- Siggraph 2007 course notes
- EG Workshop on Sketch-Based Interfaces and Modeling
- Sketch-based interface project web pages
- Microsoft Center for Research on Pen-Centric Computing website
  - [http://graphics.cs.brown.edu/research/pcc/home.html](http://graphics.cs.brown.edu/research/pcc/home.html)
- Various other conferences (UIST, CHI, SIGGRAPH)
- Check course website for links

Pen UI Resources (2)

*Sketch Understanding*

Papers from 2002 AAAI Spring Symposium
Randall Davis, James Landay, and Tom Stahovich, *Program Cochairs*
Technical Report SS-02-08
Published by The AAAI Press, Menlo Park, California
see [http://www.aaai.org/Library/Symposia/Spring/ss02-08.php](http://www.aaai.org/Library/Symposia/Spring/ss02-08.php)

*Making Pen-Based Interaction Intelligent and Natural*

Papers from the 2004 AAAI Fall Symposium
Randall Davis, James Landay, Tom Stahovich, Rob Miller, and Eric Saund *Program Cochairs*
Technical Report FS-04-06
Published by The AAAI Press, Menlo Park, California
see [http://www.aaai.org/Library/Symposia/Fall/fs04-06.php](http://www.aaai.org/Library/Symposia/Fall/fs04-06.php)
Why Sketches and Gestures?

- Mimic pencil and paper
  - direct and natural for many tasks
  - familiar affordances
- Powerful and expressive
  - more freedom
  - can be faster
  - non-WIMP

Key Issues – Recognition, Resolving Ambiguity, and Self-Disclosure

- Recognition
  - need to understand sketch components
- Ambiguity
  - deal with multiple interpretations
- Self-Disclosure
  - invisible interface (mostly gestural commands)

www.ueda.nl/earth/development.html
Recognition

resolving ambiguity

- Difficult problem
- Focal point of research
- Many approaches
  - limiting the domain
  - underlying rules and knowledge
  - suggestive interface
Self-Disclosure

How do we interact with this application?  What are the commands?

Where do I begin?

How many commands are there?

Pen-Based Interface Dataflow

Raw Stroke Data → Preprocessing → Segmentation

Sketch Understanding → Ink Parsing → Classification → Feature Extraction And Analysis

Make Inferences
Representing Data

- Points and strokes
  \[ s = p_1, p_2, \ldots, p_n \]
  
  where
  \[ p_i = (x_i, y_i, t_i), \quad 1 \leq i \leq n \]
  
  \[ S = s_1 s_2 \ldots s_m \]

- Image
  - pixel matrix
  - not as popular

Preprocessing

- Often required to clean raw data
- Filtering and Smoothing
- Stroke Invariance
  - scale
  - position
  - orientation
- Dehooking

Representing Data

\[ (x_m, y_m, t_m) \]

Image

- Pixel matrix
- Not as popular

Preprocessing

- Often required to clean raw data
- Filtering and Smoothing
- Stroke Invariance
  - Scale
  - Position
  - Orientation
- Dehooking

Diagram: Normal view of stroke

Diagram: Zoomed in view of stroke showing unwanted cusps and self-intersections
Segmentation

- Determine which strokes go together
- Determine which strokes should be apart
- Can be done in real-time or in batch
- Often uses proximity and timing information

\[ y = \frac{1}{2} x^2 \]
\[ y = x^2 e^{-\frac{1}{2} t} \]

Feature Extraction and Analysis

- Want to distinguish sketch components from one another
- Good features are critical
- Extract important information
  - geometrical, statistical, contextual
- Examples include
  - arc length, histograms, cusps, aspect ratio
  - self-intersections, stroke area, etc…
Classification

- Use features as input to a classification algorithm
  - recognize sketch components and gestures
- Can be simple as an FSA
- Commonly use machine learning algorithms
  - linear classifiers, neural networks, HMMs, SVMs
  - AdaBoost, K-means classifiers, etc...
- Algorithm choice dependent on problem

Sketch Parsing

- Often recognition of strokes is insufficient
  - except for gestures
- Require an understanding of spatial relationships
  - good examples are mathematical expressions
- Higher level classifications
  - is it a word or a drawing?

www.engr.ucr.edu/~stahov/research/acsparc.htm
Making Inferences

- Sketches are often insufficient for understanding
  - can be under- or over-constrained
- Can infer based on
  - context
  - domain knowledge
  - domain restrictions
  - stroke location

Sketch Understanding

- Understanding a sketch/recognizing a gesture is only half the battle
- What do we do with it?

VibroSketch: Sketch Understanding
(Kara, Gennari, Stahovich 2004)

Kirchhoff's Pen (de Silva et al. 2007)
Next Class – Discussion

- Final Project Ideas
- Readings