SketchGraph: Gestural Data Input for Mobile Tablet Devices

Jacquelyn Martino, Paul Matchen, Harold Ossher, Rachel Bellamy, Cal Swart
IBM Research
19 Skyline Drive
Hawthorne, NY 10532
{jmartino,matchen,ossher,rachel,cals}@us.ibm.com

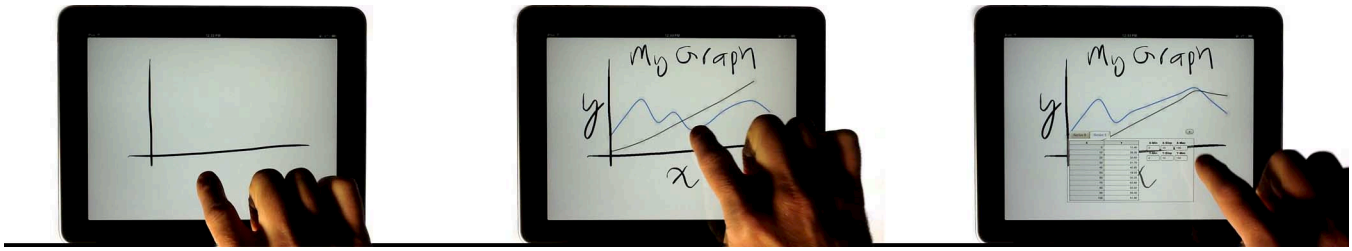

Figure 1. *User draws a recognized graph gesture (left), strokes curves and labels (middle), gestures to get the underlying data table (right).*

1 Introduction

As tablets become ever more powerful and popular, people want to use them broadly, including for business applications like spreadsheet data graphing. Tablets are better suited to informal exploration through sketching, however, than to inputting data into a spreadsheet. It would be much more appealing, and suitable to the medium, to sketch a graph as if you were drawing on a napkin. We describe an early prototype to support a gestural, graphical interface for inputting and updating graph data that is as easy as drawing a few strokes. With it, users can focus on exploring their domain, rather than on the mechanics of data entry.

2 Exposition

In our *hand-drawn computing* project [Bellamy et al. 2011] we investigate user interfaces (UI) for natural, hand-drawn problem solving using digital tablets. Sketching, and the visual thinking it enables [Tversky et al. 2003], are powerful tools for problem solving, especially during the early, exploratory stages. Its effectiveness largely depends on enabling users to "stay in the flow" [Csikszentmihalyi 1990], focused on their free-flowing, problem-solving activities without being distracted by the medium. At the same time, however, we have design goals similar to LaViola [2007] in that we want to leverage the power of digital media, building models implied by sketch marks as appropriate and elevating the "paper" plane to a dynamic computational space. This has the potential to replace the familiar disconnect between early exploration and later, detailed problem solving, design, and modeling with a smooth transition eased by shared information.

Our talk describes early research in the area of numeric exploration on tablets. For many domains, problem–solving involves "playing with data" to understand relationships and trends. A "flowy" way to work is to sketch graphs, but the graphs remain entirely informal, with no actual data behind them. Alternatively, one can enter data into a table or spreadsheet, and produce graphs from them. This has the opposite problem: entering data breaks the flow when one's goal is to produce a graph that shows a desired relationship or trend. We want the best of both worlds.

SketchGraph allows the user to sketch a graph on a tablet, using a few simple gestures (Fig. 1). The strokes making up the axes are recognized as such, and subsequent strokes within the upper-right quadrant are then interpreted as curves. A collection of points on each stroke is computed and tabulated automatically. A smooth curve is drawn through these points, with the points highlighted. The graph can be made up of many curves, each of which is interpreted as distinct data series, distinguished by color. The user can label the axes as well as the graph as a whole.

The user can manipulate the curves, moving the identified points around at will, with quick, automatic data interpolation and underlying data table updates. Dragging a curve from its starting point and off the edge of the graph deletes it. A "C" gesture clears the entire graph area. The user can thus manipulate graphs on the tablet in a flowy way, even more so than on a whiteboard (which does not support point movement with automatic interpolation).

In addition, the user can, at any time, make a down stroke to see the underlying table of numbers. The data series for each curve is shown in the same color as the curve itself. The user can now go back and forth, examining and changing either the numbers in the table or the positions of points on the graph, at will. As the user changes one representation, the system automatically updates the other, so both the visual and numeric effects are manifest.

We envision the user staying "in the flow" throughout their entire interaction with numerical data, supported through gestures such as circling to select, and pinching and stretching to shrink and expand axes. Our sketch–like gestural interactions are in contrast with OmniGraphSketcher's (www.omnigroup.com) palette approach to drawing and manipulating graphs. In their case, a traditional WIMP–like interface breaks user flow by demanding additional interaction overhead.

Our prototype is an HTML5 application, initially targeting the iPad. We use a number of underlying technologies to enable our UI design investigations. The $1 Unistroke Recognizer recognizes strokes as a graph or as individual characters (http://depts. washington.edu/aimgroup/proj/dollar/). To extend the recognized template to multiple strokes, we added a timer to trigger the recognizer after a delay in drawing a stroke sequence and pieced the strokes together into a single stroke for recognition. KineticJS provides an object and event model that allows interactivity on an otherwise static canvas element (http://www.kineticjs.com/). To determine Bézier control points for pleasing curves through a set of points we refer to Rob Spenser (http://scaledinnovation.com/).

References

BELLAMY, R.K.E., et al., Sketching tools for ideation: NIER track. In Proceedings of ICSE. 2011, 808-811.

CSIKSZENTMIHALYI, M. 1990. Flow: The Psychology of Optimal Experience, New York: Harper and Row.

LAVIOLA, J. J. Jr.. 2007. Mathematical sketching. In ACM SIGGRAPH 2007 courses (SIGGRAPH '07). ACM, New York, NY, USA,Article 10

TVERSKY, B. et al. (2003). Sketches for design and design of sketches. In Ugo Lindemann (Ed), Human behavior in design: Individuals, teams, tools. Pp. 79-86. Berlin: Springer.D.

SIGGRAPH 2012, Los Angeles, California, August 5 – 9, 2012.
ISBN 978-1-4503-1435-0/12/0008