Creating Models for Learning and Recognizing Multitouch Gestures

Google Summer of Code Project Proposal

Sashikanth Damaraju

Abstract

This project is aimed at creating a language / framework independent Gesture Recognition toolkit that takes OSC messages formatted with TUIO specification as input and outputs recognized gestures via OSC protocol. I will use the gesture recognition toolkit AMELiA to describe models specifically for the domain of multitouch gestures. This project will enable multitouch application developers to easily define a gesture and utilize it within their application, creating more engaging experiences.

Project Proposal

This project is aimed at creating a language / framework independent Gesture Recognition toolkit that takes OSC messages formatted with TUIO specification as input and outputs recognized gestures via OSC protocol. The problem with gestures such as the pinch-to-zoom, rotate are hard-coded and require more coding to add new gestures into the system. Apart from this, there are toolkits that go one step ahead to learn and recognize single point stroke gestures based on their shape, like the nui wave to bring up a menu. Although this is very useful for application developers to use, it is still limited by not using multiple points of input that a multitouch surface allows.

This project will focus on gestures of two types, non-parameterized gestures and parameterized gestures. Non-parameterized gestures, as the name suggests, map to actions that do not require any quantified parameters for their corresponding action, such as a "switch to fullscreen" or "Delete element". Parameterized gestures, on the other hand, once recognized will follow with a stream of live parameters, such as a "Change hue(/saturation/value/alpha) of an element" allowing control over parameters of that action. Recognized gestures will be outputed as OSC messages with optional parameters (such as x,y coords of where the gestures is performed), so applications can contextualize the operation of the gesture.

This project will build an architecture for future multitouch applications to use natural gestures that are decided by the application developer, and can be learnt with minimal training samples. The tools and communication channels used are cross-platform, language independant. Much like the TUIO protocol, all that is required by any framework to use this will be a simple wrapper written in the language of the framework. Using this augmented communication protocol, applications will register elements as GestureListeners, and also register the context of interaction - the set of gestures that are currently available for the user to perform.

The project I am proposing will draw from machine learning literature to use bayesian networks to model learnt gestures. Such techniques allow multitouch application developers to use complex gestures by simply teaching them by a few examples. I have previously demonstrated the feasability of such techniques using Hidden Markov Models to successfully learn and subsequently recognize gestures that have multiple points of contact (http://tinyurl.com/mtGestureLearning).

I will be using AMELiA (http://ame4.hc.asu.edu/amelia/) as the gesture recognition toolkit for this project. AMELiA is a generic gesture recognition toolkit that allows developers to build domain specific models for learning and recognizing gestures. I will be using this toolkit to build models specifically for multitouch input. The probabilistic bayesian models availabe within the AMELiA toolkit are appropriate for creating Interaction Contexts with a set of gestures that the application supports. As the library includes the training and recognition algorithms, my project will only require implementing the domain specific code for the modeling multitouch data. This makes the project tractable within the timeline of summer of code.

This project takes it's inspiration from the gestures developed by Wayne Westerman during for his PhD thesis that evolved into the company FingerWorks and are now in Apple's new multitouch trackpads. One of the key difference to note between my proposed project and Westerman's work is that a multitouch input-only device was used to develop gestures for his devices, whereas today we are working with interactive screens, that merge display and input into the same space. To use gestures affectively, we will need to experiment with multiple mappings for operations with different gestures. This project develops the framework for interaction designers to easily create and use multitouch gestures within their application.

Orientation: The orientation of a multitouch gesture for a recognition algorithm is tied to the problem of dimensionality. Since touchlib/tBeta assigns id's in increasing order to contacts, multiple fingers may have varying order across samples. For example, consider a three finger gesture with the thumb, index and middle finger. During the first sample, the fingers may have id's (0,1,2) whereas

the second time the gesture is performed their id's might be (1,0,2) respectively. This is a problem for the recognizer. A heuristic that has worked to solve this problem uses the orientation of each finger to consistently re-assign id's to the contacts across samples. I will apply a shape matching heuristic that can quickly re-assign id's by matching the shape of the contacts. This will make all samples of a gesture have consistent id ordering, and allow the gestures to be recognized independent of the orientation.

Development Methodologies:

My proposed project will be developed in a C++ makefile environment that can compile cross-platform. This project will be developed through a series of iterations, with a concrete deliverable at every stage.

- Month 1: A simple learning and recognition system for non-parameterized, fixed dimensionality gestures. This would include a gui to allow gestures to be inputed as training samples. The communication protocol for recognized gestures will also be established. I will use pyMT as my initial implementation framework to receive recognized gestures. The OSC parser would be expanded to include messages such as /gesture/recognized gestureId x y frameNum. The parser would then generate a gesture event and relay it to GestureListeners to perform the corresponding action.

- Month 2: Adding varying dimensional gestures to the model. Gestures that have taps, multiple touch-up and touch-down events will be learnt and recognized. This phase would involve allowing the models to connect to each other, allowing a chain of gestures to function as a single gesture. More complex gestures will be efficiently learnt and recognized.

- Month 3: Parameterized Gestures. Operations such as scale canvas, pick color, skew object require additional parameters to provide the user with fine-control during their performance. This will be added during the last month, and the communication protocol will be augmented with the required parameter structure.

During the development of these deliverables, the accuracy of the models will be continuously optimized. Also, samples from multiple users will be tested against gesture models that have been taught by a single user. I will ensure that the model is

not user dependent, so a gesture taught once by anyone can be used without further retraining. Mentor and community feedback at every stage will be vital to help me understand how best this toolkit can be integrated with the rest of nuigroup's frameworks.

Reason for Picking Specific Project:

Gestural interaction is still not fully understood. One of the reasons for this is the lack of a framework to easily define and use gestures within a multitouch application. With this project, apps can be more engaging and allow better expression. I am tempted to increase to scope of the project to various other tasks that immedeatly follow building a gesture recognition system. The visualization of the gesture during recognition, interfaces for displaying possible gestures to the user are unsolved issues with interesting solutions. However, I am limiting myself to a scope that I can confidently complete a robust toolkit with high recognition accuracy independent of user.

For the past year, I have been dedicated in understanding and proving the benefits of multitouch surfaces in real-world applications. As I spend more time, I realize that we are still scratching the surface with what can be done. As part of my master's thesis, I have created a set of tools that allows me to iteratively test recognition models using saved samples. This tool will be invaluable in helping me analyze the stream of input during gestures, and fix various problems that will be extremely difficult if I had to perform each gesture to test the system. I will be working on this toolkit for long after summer of code is gone, whether or not I get accepted. However, it is incredibly exciting to have an oppurtunity to work with such a community in completing such a project. The enthusiasm and devotion shown by mentors and students during this summer will no doubt motivate all of us to perform at our best.

Activity level within the NUI Group Community:

Long time forum reader, IRC lurker. (Joined circa spring 2007)

Time working in related fields:

Multitouch specifically - 1.5 years. Mostly on recognition aspects. Currently funded to work on a creativity support tool, which requires connecting the domains of semantic web, web crawling, metadata management to information visualization, HCI and interaction design.

Personal Profile

Name: Sashikanth Damaraju

Email: d.sashikanth@gmail.com

Location/Timezone: College Station, TX - CST

Website: http://www.ecologylab.net/people/sashikanth.html

Age: 23

Education/Qualifications: Bachelors in Computer Science and Engineering, Jawaharlal Nehru Technological University, India. Currently a Masters Student in the Department of Computer Science, Texas A&M University. (expected graduation : summer 09)

Academic and Industry Background: HCI/InfoViz/Information Storage and Retreival/UI/Interaction Design

Open source development experience: Summer of code 2006 (OpenMRS): I worked on overloading sparkline visualizations to help doctors diagnose clinical patient history. OpenMRS is a medical record system based on a jsp/sql architecture, and my project involved integrating flash based visualizations through javascript (ajax).