SIGArt Multi-touch Gesture Recognition Interface for the Tacchi Table

SPECIAL INTEREST GROUP ON ARTIFICIAL INTELLIGENCE

Project Summary

We present a real-time multi-touch gesture recognition implementation primarily for use by the Tacchi Table built by SIGMusic for music-related programs. The algorithms and interface used are general enough to be applicable to a variety of other applications. To accomplish our gesture recognition, we build upon the TUIO interface to acquire raw gesture data and perform gesture segmentation and classification on our orientation-free gesture model through MATLAB.

Tacchi

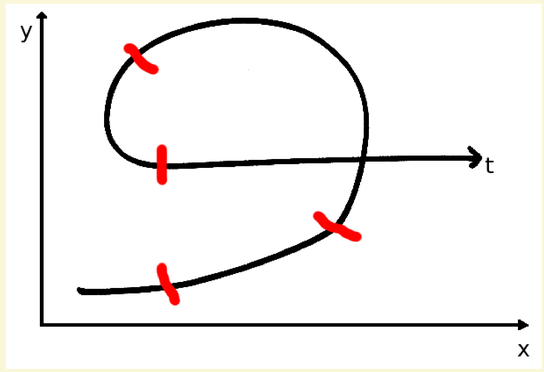
Tacchi is SIGMusic's 46" fully multi-touch table. Complete with two cameras and four lasers, Tacchi is capable of unlimited users. SIGMusic first unveiled Tacchi at Engineering Open House in 2010, winning "Best Undergraduate Student Research" as well as the Lockheed Martin Computer Science Award. Tacchi was constructed over a period of four months, starting from commodity hardware components. SIGMusic is a special interest group dedicated to the study and production of electronic and computer music.

Environment

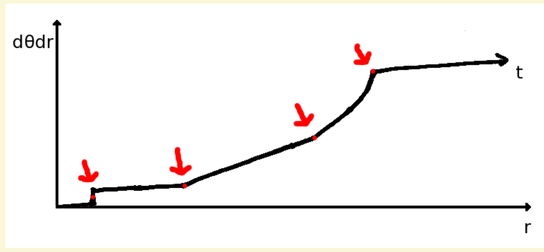
There are several factors that must be considered when building a gesture interface for a multi-touch table such as Tacchi. Unlike most multi-touch devices where there is only a single user, Tacchi enables multiple users providing input simultaneously. Moreover, as the orientation of the users may be varied, we may not rely on a fixed coordinate system of input. As the Tacchi table is geared toward music applications, our recognition system must also operate in real-time. Therefore we require our algorithms to be both robust and efficient.

Gesture Model

To simplify the analysis and detection of incoming data, and to facilitate the detection of different aspects of gestures and transitions, we developed our own representation of each potential gesture based on certain data. The basis of our model involves visualizing each gesture via a plot of $d\theta dr$ (the change in the angle between data points multiplied by the change in distance between points) versus r (the length of the gesture). This representation makes detecting points, straight lines, circular arcs, and other curves very easy, while additionally allowing computation in an orientation-free manner.



A hypothetical gesture, as it would appear in a standard Cartesian plane. Transition points are marked with red slashes.

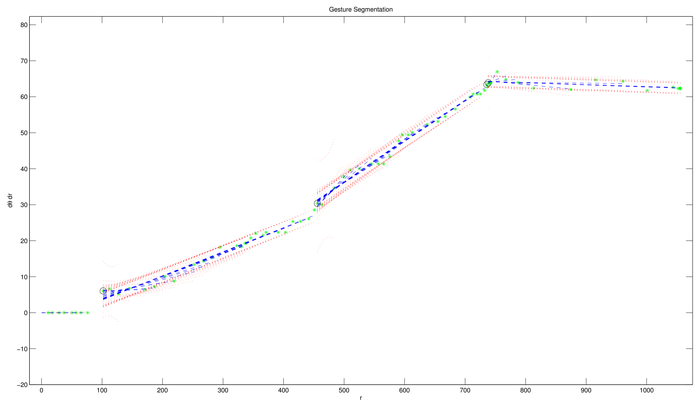


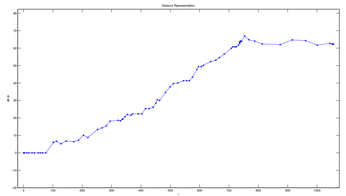
The corresponding representation of this gesture within our model. Transitions points are marked with red arrows.

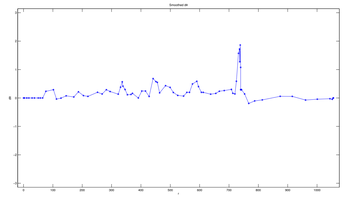
About Us

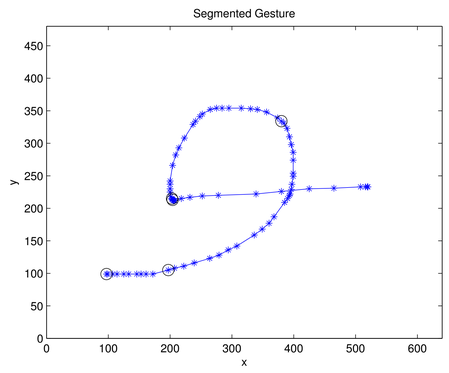
SIGArt is a student group of ACM (Association for Computing Machinery) that is dedicated to working on projects that pose interesting problems in and related to the field of artificial intelligence. Past projects have included a Wikipedia Knowledge Extractor for a common sense database as well as an instant messenger chatbot.

Gesture Segmentation and the Golden Ratio



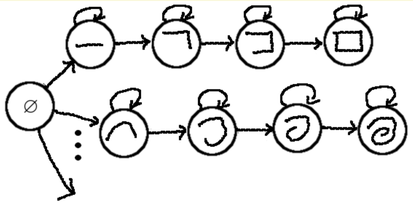






Segmentation is performed by analyzing the transformed gesture representation alongside the change in θ over a small region in space, seen in the above right figures. We find the transitions between the current segment and the start of the next segment by comparing a new gesture data point with the 95% confidence interval of the linear fit over the set of current data points factored by the reciprocal of the golden ratio (0.618). A transition is marked if the new point lies outside this interval, otherwise the point is added to the current segment and an updated linear fit and confidence interval is calculated. The segmentation can be seen in the above left figure: the blue dashed lines represent the linear fit at each iteration, the red dotted lines represent $0.618 \times 95\%$ confidence interval, and the green dots are the gesture data points. The resulting transition points can be seen on the original gesture on the figure to the left.

Gesture Classification



Provided a sequence of gesture segments (point, line, arc...), we classify the gesture whole by feeding it to our hidden Markov model (HMM) for gestures. An illustration of the structure of our HMM states is shown on the left: we begin in an initial state that transitions to various state trajectories for each trained gesture. At each transition point, the probabilities for each state given the current sequence of gesture segments is calculated. We then estimate the gesture the user is currently performing simply by evaluating the trajectory containing the state with the highest calculated probability.

Contributers

Xiao Cheng: Department of Computer Science (UIUC) - cheng88@illinois.edu
Nikoli Dryden: Department of Computer Science (UIUC) - dryden2@illinois.edu
Darin Pantley: Department of Computer Science (UIUC) - pantley2@illinois.edu
Abhishek Pradhan: Department of Computer Science (UIUC) - apradha2@illinois.edu
Felix Wang (*SIGArt Chair*): Department of Electrical and Computer Engineering (UIUC) - fywang2@illinois.edu