PlayCLC Manual

written by Xiang Cao

(May 28, 2007)

PlayCLC is a research demonstration of the CLC (Curves, Lines & Corners) model for pen stroke gestures developed by Xiang Cao and Shumin Zhai (ref. [1]). The main purpose of the tool is for researchers to experiment with the CLC model and to support further research in this aspect.

The CLC model is a quantitative model for predicting the production time of single-stroke pen gestures drawn by users. It may serve as a foundation for the design and evaluation of existing and future gesture-based user interfaces at the basic motor control efficiency level. The CLC model works by breaking a given gesture into elements of *Curves*, straight *Lines* and *Corners*. The production time of each type of element is modeled as follows:

$$T(line) = mL^n$$

$$T(curve) = \frac{1}{K}\int_0^S R(s)^{-\beta} ds$$

$$T(corner) = 0 \quad (approx.)$$

where m, n, K, β are parameters of the model, whose values may vary across individuals. And the production time of the complete gesture is the summation of all elements (omitting the *T(corner)*):

$$T = \sum T(line) + \sum T(curve)$$

For details of the CLC model and the related experimental studies, please refer to [1].

Using the PlayCLC tool, you can create and save samples of pen gestures, organized as gesture sets. The system can analyze the gesture and predict the production time according to the CLC model parameters provided by you. The predictions can be both displayed in real-time, and exported as a text format report file. Currently the system cannot automatically segment a gesture into elements, instead the segmentation is indicated interactively as part of the gesture definition process.

The program runs under Windows operating system, and **requires Microsoft .NET Framework 2.0**. It can be used either with a mouse or a stylus (recommended) input.

File List

Under the unzipped folder, you will find the following files:

PlayCLC.exe – main program

graffiti.set – Gesture set file for Graffiti gestures

unistrokes.set – Gesture set file for Unistrokes gestures

shapewriter.set – Gesture set file for 36 examples of ShapeWriter gestures (ref. [2], also known as SHARK gestures)

Readme.pdf – User Manual

WelcomeMsg.rtf – welcome message used by the main program

User Interface

File Menu (top)

New Set – Create an empty gesture set

Open Set – Open an existing gesture set file (*.set)

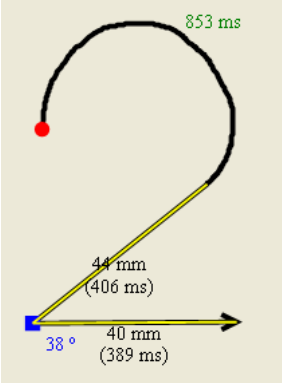
Save Set – Save the current gesture set as a *.set file

Save Set As – Save the current gesture set with a different file name.

Generate Analysis Report – Generate an analysis report (text file) of the current gesture set. Each line includes one gesture's estimated production time, numbers of straight lines, corners, and curve segments. The CLC model parameters used for the calculation are also included in the top of the file. You may then import this report into statistical software such as SPSS for further analysis. Depending on the number and complexity of the gestures, the report generation may take a few seconds to a few minutes.

Authoring Panel (middle)

You can view existing gestures (in the *Reviewing* mode) or define a new gesture (in the *Creating* mode) in the authoring window. The legends for displaying gesture are as follows:



Red dot – Starting point

Arrow – Ending point

Blue squares – Corners (marked with the degree of the corner angle)

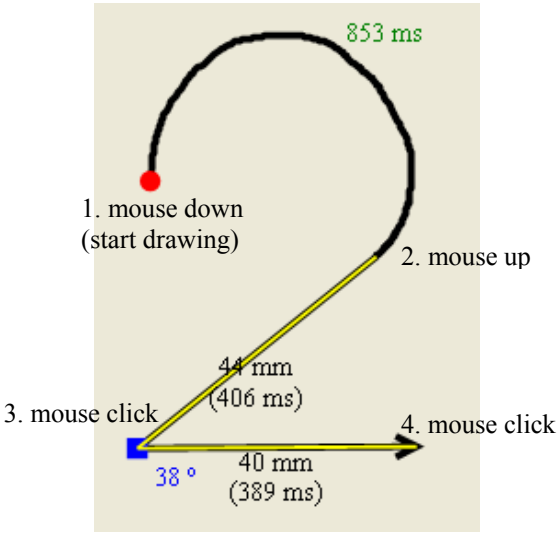
Yellow lines – Straight line elements (marked with the length and estimated production time)

Black lines – Curve elements (marked with estimated production time)

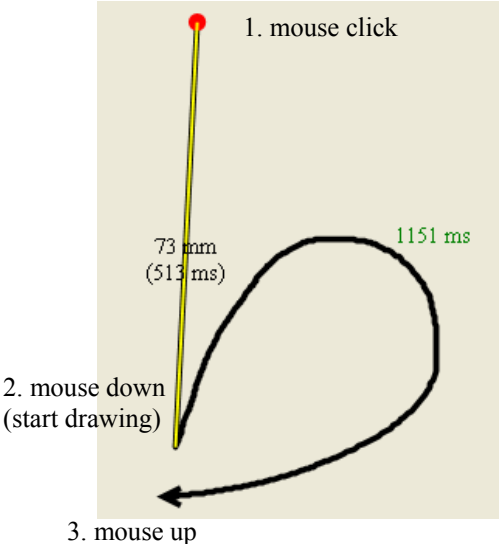
In the *Creating* mode, you can create a new gesture. The elemental segments (curves or straight lines) of the gesture are added in sequence. To add a straight segment, click the starting and the ending position; to add a curved segment, draw the curve using the stylus (or using the mouse cursor while holding the left mouse button). Disjoint segments are automatically connected by straight lines, and corners between straight lines are automatically detected. Therefore, to add a curve following a straight line, you can directly start drawing the curve without clicking the end point of the straight line first (see point 2 in example 2).

Below are 2 examples that shows the action sequences for defining a gesture (time ordered by the number) in the case of using a mouse. The actions using a stylus are similar, in which mouse button up/down actions are replaced by pen up/down actions.

Example 1



Example 2



Control Panel (bottom)

Gesture Name – Name of the currently displayed gesture. You can edit it by typing in a different name.

Gesture # – Index of the gesture. To jump to another gesture, enter a different index and press <Enter>.

Mark Elements – Whether to mark the curve, straight line and corner elements in the gesture.

Show Grids – Display grid lines in the authoring panel.

Time Estimation – Estimated production time of the gesture, in milliseconds.

Browse Gestures:

Previous – Go to the previous gesture in the set.

Next – Go to next gesture in the set. If this is the last gesture in the set, create a new blank gesture.

Clear – Make the current gesture blank. You can then redefine this gesture.

Delete – Remove the current gesture from the set.

Transform Gesture:

Rotate (90 degree, 45 degree) – Rotate the gesture clockwise.

Mirror (Vertical, Horizontal) – Mirror the gesture.

Model Parameters: (K, Beta, m, n)

Parameters of the CLC model used to estimate the gesture production time. By default, these parameters are set to the values acquired from the experiment reported in [1]. However, these values are not universal, and may vary for different people. You can adjust the parameter values in the text boxes. Press <Enter> or the *ReCaculate* button to update the estimation.

Known issues

A single point is not considered a valid gesture. The program ignores gestures consisting of only one point.

The program currently does not detect corners formed between two curves, or one curve and one straight line. This does not influence the estimation since *T(corner)* is omitted in the model. However if you want to explicitly indicate these, you may draw short straight lines around the corner, which smoothly connect with the surrounding curve segments.

Because of numerical error of the calculation, estimated time may be slightly different when the gesture is rotated or mirrored.

Contact

If you have any questions, or want to suggest corrections or improvements, please feel free to contact Xiang Cao (caox@cs.toronto.edu) or Shumin Zhai (zhai@almaden.ibm.com).

References

[1] Xiang Cao, Shumin Zhai. (2007). Modeling human performance of pen stroke gestures. Proceedings of CHI 2007, ACM Conference on Human Factors in Computing Systems.

[2] Per-Ola Kristensson, Shumin Zhai. (2004). SHARK2: a large vocabulary shorthand writing system for pen-based computers. Proceedings of UIST 2004, the 17th annual ACM Symposium on User Interface Software and Technology, p. 43-52.