G3: Bootstrapping Stroke Gestures Design with Synthetic Samples and Built-in Recognizers

Daniel Martín-Albo
Universitat Politècnica de València
Camí de Vera, s/n, 46022 Spain
`damarsi1@upv.es`

Luis A. Leiva
Sciling
CPI UPV, V.9B.0.063, Spain
`name@sciling.com`

MobileHCI '16 Adjunct, September 06–09, 2016, Florence, Italy
ACM 978-1-4503-4413-5/16/09.
http://dx.doi.org/10.1145/2957265.2961833

Abstract
Stroke gestures are becoming increasingly important with the ongoing success of touchscreen-capable devices. However, training a high-quality gesture recognizer requires providing a large number of examples to enable good performance on unseen, future data. Furthermore, recruiting participants, data collection and labeling, etc. necessary for achieving this goal are usually time-consuming and expensive. In response to this need, we introduce G3, a mobile-first web application for bootstrapping unistroke, multistroke, or multitouch gestures. The user only has to provide a gesture example once, and G3 will create a kinematic model of that gesture. Then, by introducing local and global perturbations to the model parameters, G3 will generate any number of synthetic human-like samples. In addition, the user can get a gesture recognizer together with the synthesized data. As such, the outcome of G3 can be directly incorporated into production-ready applications.

Author Keywords
Gesture Synthesis; Bootstrapping; Gesture Recognition; Strokes; Marks; Symbols; Unistrokes; Multistrokes; Multitouch; Kinematics; User Interfaces; Rapid Prototyping

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation]: User Interfaces; **I.5.4** [Pattern Recognition]: Applications

Introduction

Any application that is driven by gestures must rely on some recognition-based techniques. These techniques in turn require expert knowledge in machine learning, something that is often beyond the reach of many developers. Furthermore, data collection and labeling are usually time-consuming and expensive. Thus, it is important to investigate how to assist developers to quickly design a gesture set and deploy it together with a gesture recognizer.

In this paper we describe G3, a web application to design custom gesture sets by bootstrapping synthetic samples. G3 has been built with Bootstrap,[1] a mobile-first web development framework, and is optimized for touchscreen input. The fundamental advantage of this application is that the user only has to provide one example per gesture, and G3 will generate any number of synthetic human-like samples. Together with the synthesized data, a number of gesture recognizers are available in different programming languages, thus allowing developers to quickly create a competitive, ready-to-use recognizer. In sum, G3 aims for easily designing gesture sets and creating gesture recognizers, eliminating the overhead of recruiting and data collection, and reducing the need for expert knowledge in machine learning.

Related Work

We can find in the literature a number of gesture authoring tools to support explorative prototyping; e.g., A CAPpella [2], EventHurdle [3], or MAGIC 2.0 [4]. These tools allow users to "program" their desired behavior without writing any code, by demonstrating it to the system and by annotating the relevant portions of the demonstration. G3 preserves these core interactions, though its goal goes further, by allowing the user to quickly create a ready-to-use gesture recognizer that is suitable for use on UI prototypes in different programming languages. Currently, G3 integrates the popular "$ family" of gesture recognizers: $1 [10], $N [1], and $P [9]. These recognizers are a very viable and a relatively simple solution, and can be adapted to personalized user gestures.

For gesture synthesis, G3 taps into the foundations of the Kinematic Theory [6], which provides the most solid framework to date for the study of the production of human movements [7]. The Kinematic Theory approaches the neuromuscular network involved in the production of a human movement as a lognormal function of velocity profiles. This knowledge has been recently exploited to synthesize human-like gestures, showing that synthetic samples improve an existing recognizer's accuracy [5]. Currently, G3 has a revamped web service with a simpler API, and more recognizers are available in more programming languages (see next section).

In short, G3 provides the user with an unprecedented capability in terms of time savings, since (1) only one gesture example has to be specified in order to (2) generate any number of additional samples, and (3) a built-in recognizer can be quickly deployed together with the synthesized gestures. Furthermore, since G3 is web-based, the user does not have to install any dedicated software to start using it.

System Description

G3 supports unistroke, multistroke, and multitouch gestures drawn on a web canvas (Figure 1). It also supports uploading a custom gesture dataset (1g), so that it can be augmented with additional samples or deployed with a gesture recognizer. Currently there are 14 combinations of recognizers and programming languages (1f). Below we introduce G3 by following a developer, Mary, as she implements a recognizer for a set of customized gestures.

[1] http://getbootstrap.com

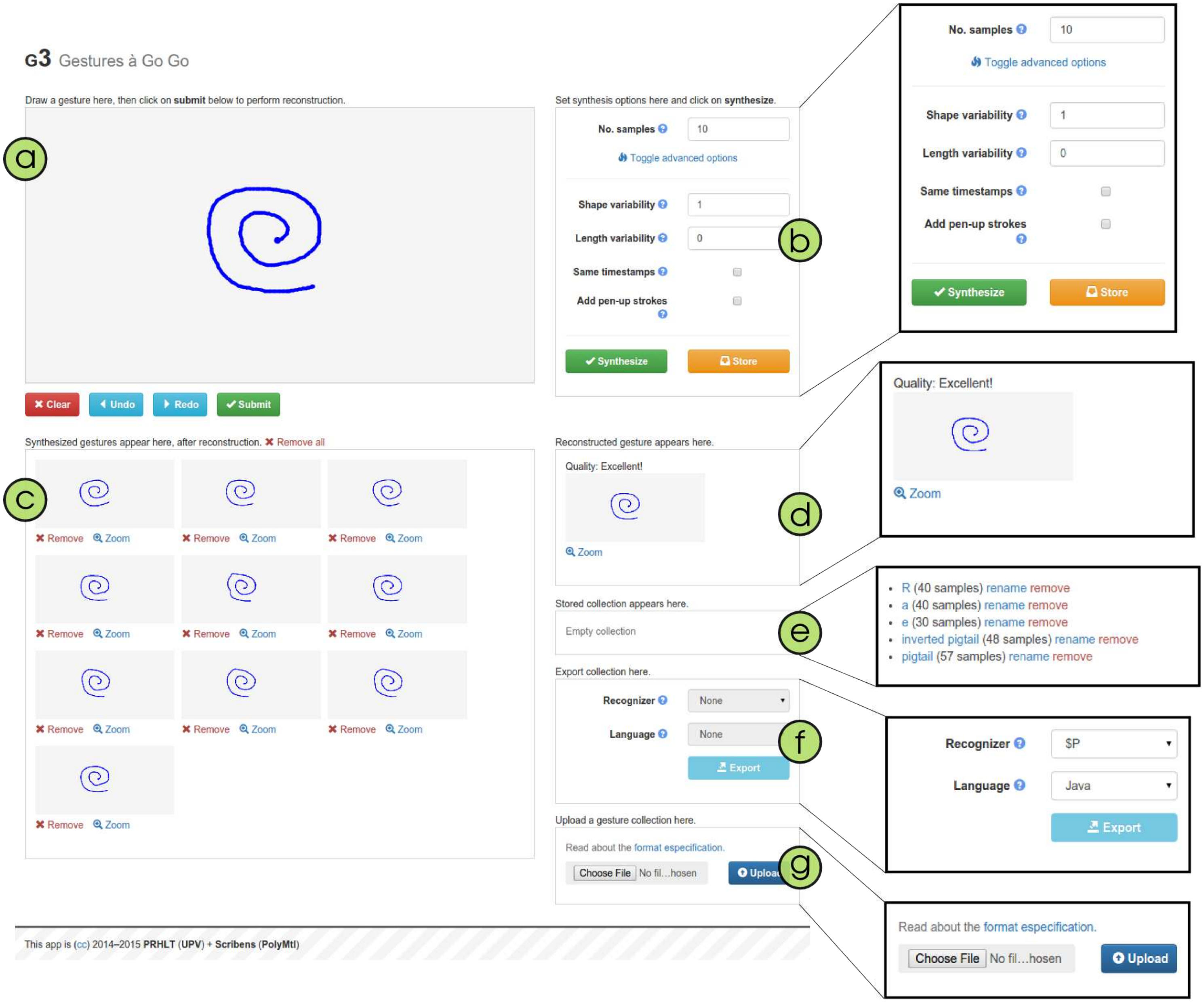
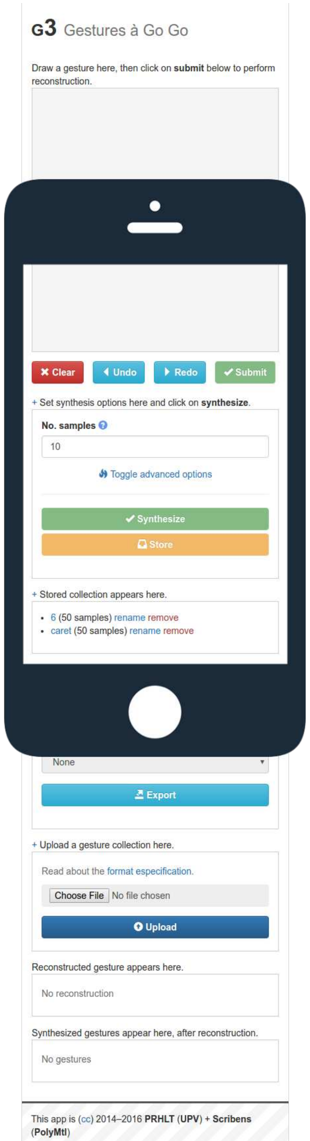


Figure 1: G3 user interface. [a] Drawing panel: the dot indicates the starting point of each gesture stroke. [b] Options panel: besides the number of gestures to generate, advanced (new) options are available. [c] Synthetic samples panel: each sample can be inspected and/or removed from the current gesture set. [d] Reconstruction panel: comprising the current gesture model. [e] Gesture collection panel: each gesture set can be renamed or removed. [f] Export panel: the user can export a gesture recognizer available in different programming languages. [g] Import panel: a JSON file comprising a custom gesture collection can be uploaded.



Mobile user interface.

Mary wants to build a recognizer that handles 5 different gestures. For each gesture, she draws just one example on the web canvas and clicks on 'Submit' (1a). Next, G3 models the submitted sample and verifies that the reconstruction is of enough quality (1d), by inspecting its signal to noise ratio with respect to the original gesture, otherwise Mary will be informed so that she can draw the gesture again.

Once a gesture is reconstructed, Mary chooses the desired number of samples that will be synthesized, as well as other advanced options such as the degree of variability for such synthetic samples (1b). Next, she clicks on 'Synthesize' and G3 generates the desired number of samples (1c). Mary can remove any of the synthesized samples if it does not meet her expectations. When she is happy with the results, she clicks on 'Store' and assigns a label to the synthesized gestures in order to save it (1e).

By the time the 5 gestures examples are drawn, Mary has a gesture set of an arbitrary size (e.g., 20 samples for each of the 5 gestures provided, 100 samples overall). By clicking on 'Export', the dataset is exported as a JSON file (1f). However, since Mary was interested in implementing a recognizer, she now selects one of the available recognizers in the programming language of her choice and clicks on 'Export'. As a result, Mary gets the recognizer together with the synthesized gesture set.

Conclusion and Future Work

G3 is an innovative application that offers active design exploration by prototyping, bootstrapping, and deploying gesture datasets together with ready-to-use recognizers. At the moment, G3 can only process 2D gestures. Thus, in future work we would like to extend it for processing 3D gestures, derived from e.g. Wii controllers and similar mobile devices. G3 is available at https://g3.prhlt.upv.es.

Acknowledgments

This work has been partially supported by the EC's H2020 program through grant 674943 (READ project).

References

1. L. Anthony and J. O. Wobbrock. 2010. A lightweight multistroke recognizer for user interface prototypes. In *GI '10*.
2. A. K. Dey, R. Hamid, C. Beckmann, I. Li, and D. Hsu. 2004. A CAPpella: Programming by demonstration of context-aware applications. In *CHI '04*.
3. J.-W. Kim and T.-J. Nam. 2013. EventHurdle: Supporting designers' exploratory interaction prototyping with gesture-based sensors. In *CHI '13*.
4. D. Kohlsdorf, T. E. Starner, and D. Ashbrook. 2011. MAGIC 2.0: A web tool for false positive prediction and prevention for gesture recognition systems. In *FG '11*.
5. L. A. Leiva, D. Martín-Albo, and R. Plamondon. 2015. Gestures à go go: Authoring synthetic human-like stroke gestures using the kinematic theory of rapid movements. *ACM Trans. Intell. Syst. Technol.* 7, 2 (2015).
6. R. Plamondon. 1995. A kinematic theory of rapid human movements. Part I: Movement representation and control. *Biol. Cybern.* 72, 4 (1995).
7. R. Plamondon, A. M. Alimi, P. Yergeau, and F. Leelerc. 1993. Modelling velocity profiles of rapid movements: a comparative study. *Biol. Cybern.* 69, 1 (1993).
8. B. Poppinga, A. S. Shirazi, N. Henze, W. Heuten, and S. Boll. 2014. Understanding shortcut gestures on mobile touch devices. In *MobileHCI '14*.
9. R.-D. Vatavu, L. Anthony, and J. O. Wobbrock. 2012. Gestures as point clouds: a $P recognizer for user interface prototypes. In *ICMI '12*.
10. J. O. Wobbrock, A. D. Wilson, and Y. Li. 2007. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In *UIST '07*.

Appendix

Figure 2 illustrates how G3 can be used to recreate an existing dataset. We picked one example at random from the Shortcut Gestures Dataset [8], and synthesized 100 samples of each gesture (2000 samples overall). Due to space limitations, only 10 of these samples are shown in the figure.

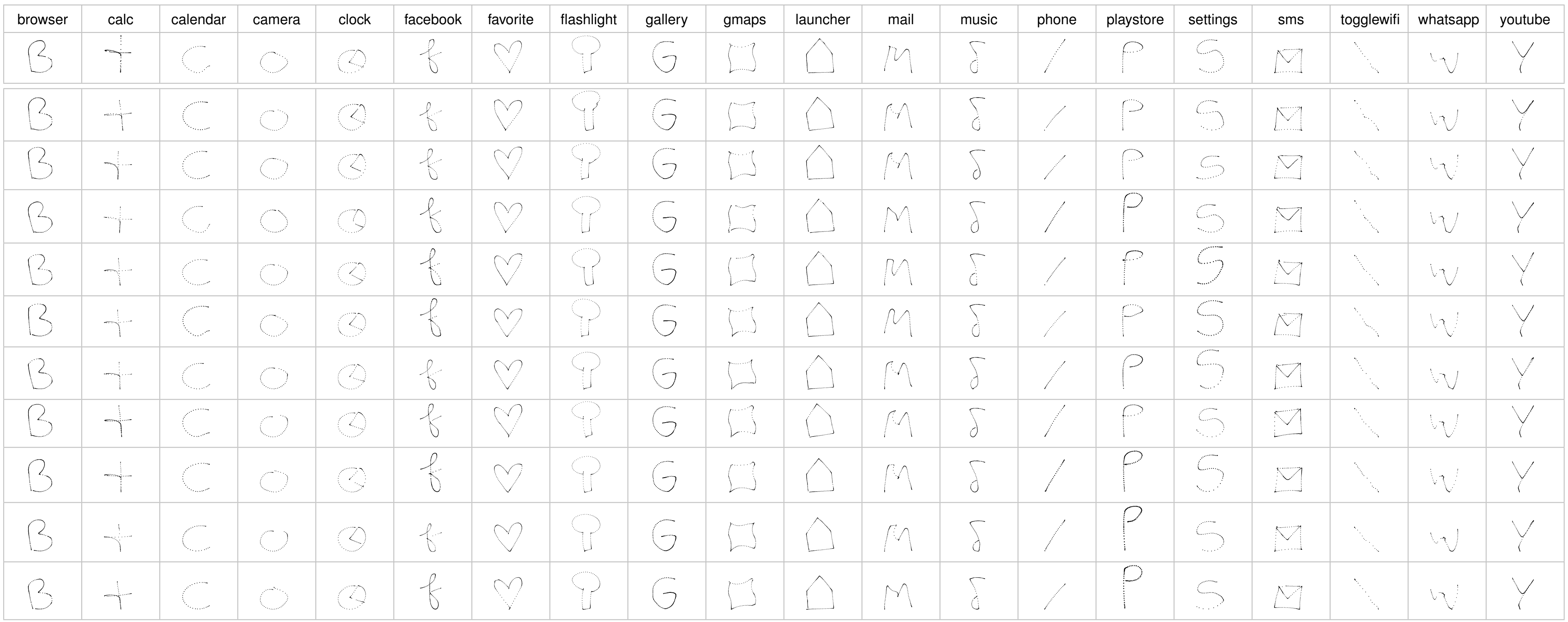


Figure 2: Recreating the MobileHCI Shortcut Gestures Dataset. One human gesture was picked as random seed (top row) to generate 100 samples per gesture. Zoom in for details.