Beyond Recognition: Using Gesture Variation for Continuous Interaction

Baptiste Caramiaux
Department of Computing
Goldsmiths, University of London
bc@goldsmithsdigital.com

Frédéric Bevilacqua
Ircam STMS CNRS UPMC
Paris, France
bevilacq@ircam.fr

Atau Tanaka
Department of Computing
Goldsmiths, University of London
atau@goldsmithsdigital.com

CHI 2013 Extended Abstracts, April 27 – May 2, 2013, Paris, France.

Abstract
Gesture-based interaction is widespread in touch screen interfaces. The goal of this paper is to tap the richness of expressive variation in gesture to facilitate continuous interaction. We achieve this through novel techniques of adaptation and estimation of gesture characteristics. We describe two experiments. The first aims at understanding whether users can control certain gestural characteristics and if that control depends on gesture vocabulary. The second study uses a machine learning technique based on particle filtering to simultaneously recognize and measure variation in a gesture. With this technology, we create a gestural interface for a playful photo processing application. From these two studies, we show that 1) multiple characteristics can be varied independently in slower gestures (Study 1), and 2) users find gesture-only interaction less pragmatic but more stimulating than traditional menu-based systems (Study 2).

Author Keywords
Gestural Interaction; Continuous Interaction; Gesture Recognition; Motor Control

ACM Classification Keywords
H.5.2. User Interfaces: Input devices and strategies; I.5.5. Pattern Recognition: Implementation;

General Terms

Algorithms, Design, Human Factors

Introduction

Gestural interaction has become commonplace in consumer electronics. Finger gestures captured on touch screens provide intuitive ways to interface with complex tasks, and some gestures such as pinch-zoom have become iconic. Most techniques for coding gesture are based on forms of activity detection that involve recognition of gesture in unitary form. Meanwhile users' casual perception of the potential of gestural input goes beyond simple recognition. People imagine gestural interaction to be intuitive and continuous, where aspects of a gesture organically map onto the response of an interactive system. What are the ways in which we might capture gesture quality to enable forms of continuous interaction?

These challenges point out the need to better understand the elements at play when a user swipes a finger across a touchscreen. What qualities make up that gesture? Are they reproducible? Are they similar across users? If a user does the same gesture in a different way, what does it reflect? Rather than suppress inter-gesture and inter-user variation, can we use these differences to extract expressive vectors from a gesture and its component qualities?

This paper looks at ways to capture gesture variations as expressive vectors for continuous human-computer interaction. The goal is to exploit the simultaneous identification and estimation of gesture and its variation for use in user interface operations that link UI actions of selection and modification. Gesture-based selection requires low latency, early recognition that identifies an incoming gesture as quickly as possible. Modification of an application UI parameter is then controlled by continuous variation in the way the gesture is articulated.

Using gestures in such a compound fashion raises several challenges. First, there is a question of which gesture characteristics can be controlled. What characteristics of gesture can users consciously control? What are the conditions for simultaneous control of multiple characteristics? And does it depend on the gesture in play?

Second, there is a question of how to use manipulating gesture variation in an expressive way in continuous interaction. Can we combine unitary selection and continuous parameter control? How does the user perceive such control?

To explore these challenges, we propose a two-stage study. The first experiment is a quantitative study on gesture performance with combinations of variations. Then we implement an advanced machine learning system that allows recognizing gesture and assessing its variation. The second experiment is a qualitative study in which we examine subjective impressions of the attractiveness of gesture selection/manipulation in a ludic real world application. The results provide important complementary clues to guide the design of gesture-based continuous interaction.

Related Work

Gesture Design

Designer-centered approaches to gesture consider ergonomics and technical constraints [14]. Ergonomics form the underlying metrics for characterizing ballistic movements like pointing [7] or reaching [15]. Specific features of recognition systems can steer the design of gesture vocabularies to guarantee high recognition success rates [14].

In user-centered approaches to gesture design, Long et al. [11] asked users to rate similarity between shape-based gestures to define a vocabulary avoiding ambiguity. Wobbrock et al. [21] asked participants to perform gestures corresponding to a given command in order to arrive at a tabletop gesture vocabulary. Kane et al. [9] seek to better understand the difference between gesture vocabularies created by sighted and blind people. Bragdon et al. [4] extend this approach by looking at environmental demands on attention.

Recognition and Classification of Gestures
There exist a number of techniques for gesture recognition. Rubine using a Single-Path recognizer [17], and Wobbrock using the $1 Recognizer [20] show that shape-based methods are efficient in HCI applications. Some techniques such as Dynamic Time Warping take into account temporal profiles of gestures by defining one template per class.

Methods based on machine learning make use of multiple examples to derive gesture classes. Established methods include Hidden Markov Models (HMM) [12] for time-dependent signals and Support Vector Machines (SVM) for static patterns. A training procedure is needed to estimate model parameters that best fit the data - the resulting models are able to take into account gesture variations present in the database. Building up comprehensive databases, however, are time-consuming and are not well suited for user-centric approaches.

Hybrid methods implement template-based approaches while using statistical recognition algorithm [2, 3, 5, 16]. The method presented in this paper belongs to this category.

Variations and Invariance
Inter-user variation is a key challenge for recognition and generalization. Recognition accuracy is generally diminished when a particular user's data is not in the original training set. Large multi-user databases are a brute-force solution that remains inelegant and impractical. Adaptation procedures that modify the class description during recognition have been described for both template-based methods [5, 10] and those using statistical learning [18, 19]. Some work makes direct use of gesture variation in the interaction process [6, 19]. These approaches remain largely unexplored and confined to the study of gesture in subjective art performance contexts but provide potentially powerful tools for continuous interaction scenarios.

Continuous interaction and early recognition
Most recognition systems output results in discrete time, typically upon gesture completion. There are some systems that continuously report estimation on gesture classes or characteristics [2, 5, 13], allowing for prediction and “early recognition” that could enhance the fluidity of continuous interaction [1]. A tradeoff exists between low latency recognition and recognition accuracy.

Study 1: Control of Gesture Variations
We first conducted a task-oriented study to look at dependencies between gestures and their variations. We recruited 13 participants (3 female, 10 male) for

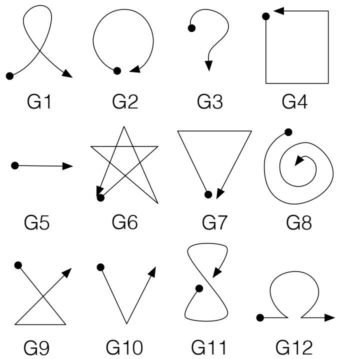

Figure 1 Gesture vocabulary used in the first experiment

Variations	
Id.	Description
V1	Slower
V2	Faster
V3	Change size
V4	Change orientation
V5	Slower and change size
V6	Faster and change size
V7	Slower and change orientation
V8	Faster and change orientation
V9	Change size and orientation
V10	Slower and change size and orientation
V11	Faster and change size and orientation

Table 1 Set of 11 variations combinations used in Step 2 of Study 1.

this experiment, ranging in age between 25 and 49 with a mean age of 33. Each subject took an individual 45-minute session and was recompensed GBP10 for their participation. Participants' gestures were captured using a custom application based on TuioPad[1] running on an Apple iPad. The application traced finger stroke gestures onscreen and sent the drawing as a series of time-stamped x, y coordinates over Wifi to a data logging computer (running the real time graphical programming environment Max/MSP 5 on a MacBook Pro OS X 10.7.4).

Procedure

We used Wobbrock's [21] taxonomy of surface gesture, of the form, *static pose and path with one finger*. We based our gesture vocabulary on perceptually different gestures [11] typically used for gesture recognition systems [20] (Figure 1).

We looked at how participants perform the gestures while modifying characteristics such as: *speed*, *size*, and *orientation*. From these characteristics we created four unitary transformations: *slower*, *faster*, *change size*, *change orientation*. In order to study the dependency of the variations on the gesture itself and the inter-dependency between variations, we created a set of 11 combinations of these four unitary transformations. This final set of variations is reported in the Table 1. The goal of the study was to understand if users could successfully control multiple characteristics simultaneously and whether this ability depended on the gesture in question.

[1] See http://code.google.com/p/tuiopad

The experiment followed a within-subject design with two factors: GESTURE (G1 to G12, Figure 3) and VARIATION (V1 to V11, Table 1). The study was comprised of three steps: 1) Gesture execution with no variation, 2) Gesture execution with all combinations of variations, 3) Outgoing questionnaire. From Step 1, we collected 12 *gestures* x *3 repetitions* = 36 *trials* for each of the 13 participants. Step 2 generated 12 *gestures* x 3 *repetitions* x 11 *variations* =396 *trials* for each participant. We performed an *a posteriori* analysis, computing the mean relative size defined as:

$$\text{relative size} = \sqrt{\left(\frac{1 - \text{modified size}}{\text{original size}}\right)^2}$$

We also computed the mean speed, and rotation angle for each gesture trial.

Results

SIZE VARIATION

We analyzed how both factors, GESTURE and VARIATION, affect the mean value of relative size (where relative size is based on original gesture performance with no variation) on two distinct subsets of variation tasks: I) a subset of VARIATION containing "change size" (called VARIATION-SUBSET I with V3, V5, V6, V9, V10, v11), and II) a subset containing the remaining combinations (called VARIATION-SUBSET II with V1, V2, V4, V7, V8).

Looking first at the two factors, GESTURE and VARIATION-SUBSET I, a repeated-measure ANOVA reveals a significant effect of GESTURE on the relative mean size ($F(11,792)=16.0$, $p<0.01$) while VARIATION-SUBSET I does not affect mean size. A post-hoc analysis using

Tukey's HSD (Honestly Significant Difference) shows that the GESTURE factor affects mean size since Gesture 5 (straight line) has a significantly higher relative size than the other gestures. All the other gestures do not exhibit significant difference in relative size.

Next, looking at the influence of factors GESTURE and VARIATION-SUBSET II, a repeated-measure ANOVA reveals a significant effect of GESTURE on the relative mean size ($F(11,660)=5.9$, $p<0.01$) while VARIATION-SUBSET II does not affect the mean size. As with Subset I, post-hoc analysis shows that the influence of GESTURE is due exclusively to Gesture 5, with the others giving a statistically insignificant difference in relative size.

SPEED VARIATION

Slower and *faster* were considered separately as distinct unitary transformations and formed the basis of two subsets of variations involving one or the other. VARIATION-SUBSET I contains: V1, V5, V7, V10. VARIATION-SUBSET II contains: V2, V6, V8, V11. We analyzed, for each (I, slower; II, faster), whether mean speed changes when varying one of the other gesture characteristics.

Looking first at both factors GESTURE and VARIATION-SUBSET I, a repeated-measure ANOVA reveals a significant effect of VARIATION-SUBSET I on the mean speed ($F(3,528)=4.4$, $p<0.01$) while GESTURE does not affect the speed. A post-hoc analysis shows that speed is significantly different for the following pairs of variations: V1–V10 and V5–V10 (see Table 1).

We next consider the factors, GESTURE and VARIATION-SUBSET II. A repeated-measure ANOVA with these two factors reveals a significant effect of VARIATION-SUBSET II on the mean speed ($F(11,528)=4.5$, $p<0.01$) as well as GESTURE ($F(3,528)=11.9$, $p<0.01$). A post-hoc analysis shows that only Gesture 5 induces a difference in speed: the mean speed is significantly higher than for the other gestures. With VARIATION-SUBSET II, a significant variation in speed exists between the following pairs of variations: V2–V6, V2–V11, V8–V6 and V8–V11 (see Table 1).

ORIENTATION VARIATION

We looked at the participants' choices when asked to modify a gesture's orientation, and whether or not they mapped onto a set of obvious angular rotations. The analysis showed that the dominant angle of rotation chosen by the participants was -90º (28.6% of all trials including a change of orientation) followed by -180º (15.4%).

PERCEIVED TASK DIFFICULTY

In Step 3, we asked the participants to indicate which gesture characteristic they found the easiest to modify as well as the one they found most difficult. 69.2% of the participants rated size as the easiest characteristic to control, followed by speed (30.8%). Nobody (0%) rated the orientation as the easiest variation while 76.9% of participants rated it the most difficult characteristic to control. This was followed by the speed (according to 23.1%). Size was never perceived as the most difficult characteristic. These results allow us to sort each gesture characteristic according to the *perceived difficulty* indicated by the participants. We obtained the following order (from easiest to most difficult): *Size, Speed, Orientation*.

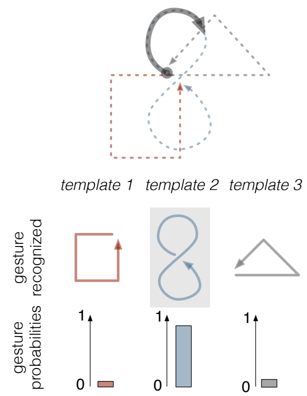

Figure 2 A gesture is recognized across a set of templates in real time. A running probability is assigned to each, the maximum being the recognized gesture

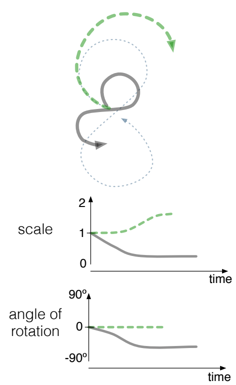

Figure 3 Adaptation of two iterations with variations (dashed, solid) in scale and/or rotation. Continuous estimation of the characteristics in time.

Observations from Study 1

At slow gesture execution, a change of size does not affect the speed. Speed then increases if varying the orientation and even more when varying the size and the orientation. Since participants rated speed more difficult than size, and orientation more difficult than speed, this seems to indicate that the inter-characteristic dependency increased as difficulty increased. At faster gesture execution, the dependencies were different. Speed increases with changing size. A complementary analysis shows that users tend to perform larger gestures when asked to change size. This dependency does not exist between speed and orientation. However, orientation has been shown to be rather obvious.

Moreover, these results are independent of the gesture considered as long as we do not consider Gesture 5 (which, as a straight line, is an exception due to its simple nature).

Technique for Gesture Variation Estimation: Recognition with Adaptation

We sought to use the knowledge gained from the previous results in a concrete application implementation that recognizes gesture and adapts to user controlled variation. In this section, we present a template-based technique for recognition coupled with an adaptation procedure. The method is based on particle filtering (PF), and allows for continuous recognition and early estimation of spatial and temporal variations while performing recognition in real time. A detailed description of the algorithm can be found in our prior work [5] (Chapter 8).

Recognition

Figure 2 shows the process to recognize one of 3 stroke gestures: *square*, *figure-8*, *triangle*. In an initial learning phase, the user (or gesture designer) performs each gesture once by way of example. This is stored as a template comprised of a time series of multi-dimensional coordinate points. In the recognition phase, gesture input is captured sample by sample. The algorithm estimates the probability of the input against each gesture template. This forms a set of weights, each between 0 and 1 (illustrated underneath the template in Figure 2). At the first input sample, the probability distribution is uniform across the gesture templates. For every new incoming sample, the algorithm continuously updates the weights. The gesture is assigned to the template having the maximum weight.

Adaptation

In addition to recognition, the algorithm is able to adapt to a set of spatial and temporal variations: phase (position along the gesture template); speed (execution relative to timestamps in the template); scale (instantaneous size relative to template); orientation (relative angle of rotation relative to template). The algorithm updates the values of each characteristic at each new input sample, continuously estimating the characteristics' degree of variation with respect to the original templates. Figure 3 illustrates two gesture inputs on the left side of the figure: a gray curve is the template performed with a smaller size and twisted; the dashed curve is the template performed more largely. On the right, we report the estimation of the two characteristics, scale and orientation. The estimation is continuous with the initial point at time 0 corresponding exactly to the original template (scale is

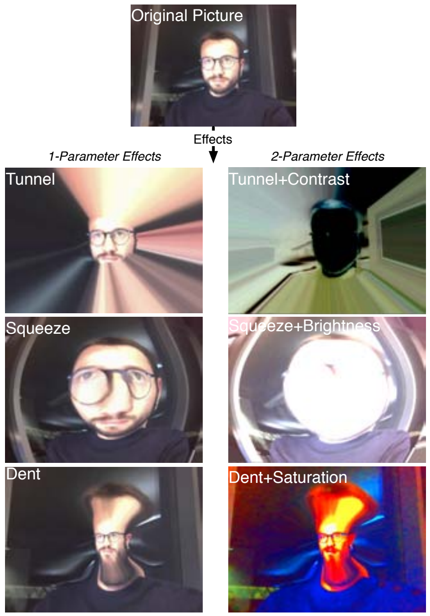

Figure 4 Effects considered in Study 2

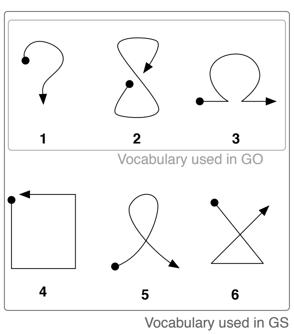

Figure 5 Gesture vocabulary used in Study 2. Items 1-3 are used in the Gesture Only (GO) mode. Items 1-6 are used in the Gesture/Slider mode (GS)

1 and orientation is 0). The estimated characteristics can be thought of as invariants in the recognition process.

Study 2: Application scenario in Continuous interaction

We implemented the recognition/adaptation technique in an end user application for use in a follow up study. The interface took advantage of users ability to modify gesture characteristics across a number of gestures. We tested this technique in a qualitative study to assess the attractiveness of using continuous variation of gesture to control parameters in a graphical application.

We recruited 16 participants, (7 female, 9 male), ranging in age between 23 and 43 with a mean age of 30. Each subject took an individual 30-minute session and were recompensed GBP10 for their participation. We developed a custom image processing program using the Max/MSP Jitter environment to emulate Apple's Photo Booth (playful image processing on self-portraits) and added the possibility for effects parameters to be controlled by different interface elements. Participants' gestures were captured on the touchscreen of an iPad and sent using the OpenSoundControl protocol to the host computer.

Use case and procedure

The application allows the user to select from one of six (6) different effects. The first three effects were geometric deformations (*Tunnel*, *Squeeze*, *Dent*) in which the deformation intensity could be controlled by the user. The other 3 effects paired geometric deformations with chromatic distortion (*Tunnel* + *Contrast*, *Squeeze* + *Brightness*, *Dent* + *Saturation*) giving two continuous parameters (see Figure 4). The program was set up to be controlled by one of 3 modes:

Menu/Slider (MS) drop-down menu for selecting effect and vertical slider(s) to control continuous parameter(s)

Gesture/Slider (GS) gesture recognition for selecting the effect and an onscreen vertical slider(s) to control the continuous parameter(s)

Gesture Only (GO) gesture recognition for selecting the effect and gesture variation(s) to control the continuous parameter(s)

Both GS and GO require a gesture vocabulary. We used a subset of the vocabulary from Study 1. We eliminated the simplistic (*line*, Figure 1, Gesture 5) and complex (*star*, Figure 1, Gesture 6) extremes. The independence of gesture characteristic from gesture demonstrated in Study 1 enabled us to assemble a vocabulary for Study 2 in a relatively straightforward manner (see Figure 5).

In GS interaction, the machine learning classifier recognized the stroke input as one of 6 different gestures (Items 1-6 in Figure 5), and used the recognized gesture index to select the visual effect. The user then dialed in the effects parameter using an on-screen slider. In this way, GS interaction replicated MS interaction, where the menu for effects selection was replaced by gesture recognition.

In GO interaction, classification was complemented by adaptation to gesture variation. The adaptation feature of the machine learning algorithm reported the amount

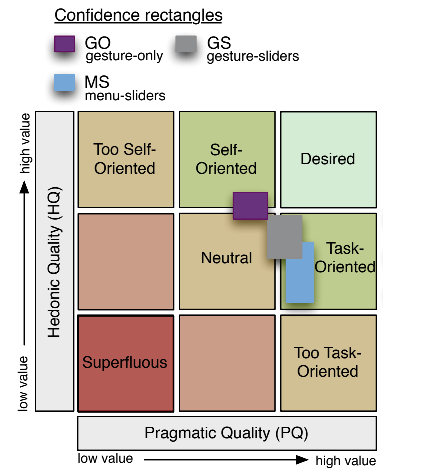

Figure 6 AttrakDiff Portfolio confidence rectangles, PQ vs HQ of the three tested modes of interaction “GO”, “GS” and “MS”.

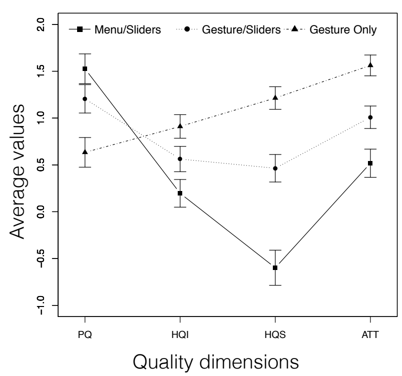

Figure 7 Average values of AttrakDiff quality dimensions: pragmatic quality, hedonic quality (identity/ stimulation), attractiveness. With standard errors.

of variation for each of 2 variations (phase and size) as a continuous value (0.0 - 1.0). Gesture variation was used to control one or two effects parameters separately, and consequently this mode used 3 gestures instead of 6 (Items 1-3 in Figure 5). This was used alongside gesture recognition to combine the two step select/parametrize process into a single gestural act. The algorithm we used allows the visual effect to be selected while still drawing, and the variation on the gesture to parametrize, in a continuous fashion, the intensity of the effect.

The experiment followed a within-subject design. The participants were asked to explore each of the 3 control modes (MS, GS, GO) by using the program taking a photo self-portrait and applying an effect. After trying each control mode, participants were asked to complete a questionnaire to indicate the attractiveness of the interaction. We used AttrakDiff[2], an evaluation method in which the user rates a product according to a series of word pairs, along a 7-point scale in order to assess the pragmatic quality (PQ), hedonic qualities (stimulation HQ-S and identity HQ-I), as well as the overall attractiveness of a product (ATT). Participants evaluated 28 pairs of words, 7 pairs in each category (PQ, HQ-S, HQ-I, ATT), rating each pair on a scale from -3 (negative) to 3 (positive). Scores are then averaged to have one mean per category for each of the 3 modes of interaction (MS, GS, GO) leading to 16 x 28 x 3=1,344 scores.

Results

The questionnaire results are summarized in an AttrakDiff *Portfolio* (Figure 6). Menu/Slider interaction was rated by participants as being highly pragmatic, but of medium hedonic quality. Gesture/Slider interaction was seen nearly as pragmatic, and slightly more hedonic. The confidence rectangle overlaps with the MS results, crosses over to a less pragmatic zone, but stays within the same hedonic zone overall. Gesture Only interaction does not overlap with either MS or GS interaction, and is in a less pragmatic zone and more hedonic zone than MS interaction. On the pragmatic scale, the users felt that GO was distinctly less task-oriented than MS or GS interaction. On the hedonic scale, GO straddles the Neutral and Self-Oriented zones, resulting in a rating of "fairly self-oriented".

Let us now compare the three modes of interaction along each of the four dimensions, PQ, HQ-S, HQ-I and ATT. Figure 7 reports averaged results for the 3 interaction modes for each dimension.

MS is seen as most pragmatic. We collected scores for all word pairs in the questionnaire in the pragmatic category. A statistical test (ANOVA) shows that the modes of interaction are not equally rated ($F(2,333) = 8.4$, $p<0.01$), with the MS mode being evaluated as having a more pragmatic quality than GO mode ($p=0.0002$).

GO ranks highest in Hedonic-Identity, Hedonic-Stimulation, and Attractiveness, with the largest difference in Stimulation. In the HQ-I category, there is also a significant difference in how the participants rated each mode of interaction ($F(2,333)=6.8$, $p<0.01$). The GO mode returns a higher global score on how the users identify themselves with this mode than the MS mode ($p=0.0007$).

---

[2] See http://www.attrakdiff.de/

In Hedonic-Stimulation, we found that GO gives significantly higher scores than the other two interaction modes ($F(2,333)=34.9$, $p<0.01$). GS also gives higher HQ-S scores than MS.

For Attractiveness, users' scores were also significantly different according to the mode of interaction ($F(2,333)=16.6$, $p<0.01$). Gesture-only interaction was perceived to be more attractive compared to Menu-slider or Gesture-slider interaction.

Discussion

Control of Gesture Characteristics

Study 1 shows that, at slow execution, users are able to reliably control variations of multiple gestural characteristics simultaneously. At fast execution, certain characteristics become dependent, notably the speed and the size (coherent with motor control constraints). Globally, user ability to vary gesture execution does not depend on the gesture in question.

Our machine learning algorithm was successfully deployed in an application use scenario. Users were able to use gesture and its variation as an alternative to classical menu-slider interaction to control an image processing application. Users responded positively to Gesture Only interaction. While they felt that traditional Menu-Slider interaction was more pragmatic, Gesture Only interaction scored high on the Hedonic scale, indicating that interfacing with the application in this manner was a pleasant experience.

Study 2 demonstrates the potential in an open-ended creative task of interaction that exploits subtle variations in gesture input. This is grounded in Study 1 which established the ability of users to reliably and deliberately exercise gestural control of temporal and spatial variations. This provides interaction designers insight on ways in which variation in gesture could be used expressively in the user interface, and design guidelines for leveraging forms of continuous control.

Beyond Recognition

Machine learning is typically optimized for recognition and classification tasks, suppressing inter-user or inter-gesture variation by identifying invariants. Our technique, based on particle filtering, provides early recognition then follows user gesture variation, making these nuances available for continuous adaptive control. The application in gestural UIs of the statistical and information analysis processes embodied in machine learning, is a manifestation of the convergence of intelligent systems and interaction design described by Grudin [8].

The question remains why GO interaction was deemed less pragmatic but more attractive. This could be due to a number of reasons. Although the machine learning technique is robust, the technology is still young and it is not error free. As these techniques improve in the future, they could become perceived as more reliable for task oriented commands. The use of gesture in this way was new for subjects in our studies. This has dual consequences – it is possible that a novelty effect made GO interaction attractive. Would the attractiveness decrease as the user gets accustomed to this kind of interaction? At the same time, as the user becomes used to GO interaction, it is possible that they would become more adept at using it in pragmatic applications.

This kind of interaction may take on more practical applicability with improvements in machine learning, and wider take up by users of this form of interaction. For the moment, GO interaction seems well suited for open-ended tasks such as image processing and exploration. The studies point out the usefulness of a gesture tracking technique that takes into account and uses gesture variation instead of suppressing them. Indeed, we can control them.

References

[1] Appert, C. and Bau, O. 2010. Scale detection for a priori gesture recognition. *Proceedings of CHI* (2010), 879–882.

[2] Bevilacqua, F. et al. 2010. Continuous realtime gesture following and recognition. *In Embodied Communication and HCI, volume 5934 of LNCS*. Springer Verlag. 73–84.

[3] Black, M.J. and Jepson, A.D. 1998. Recognizing temporal trajectories using the condensation algorithm. *Proceedings of the IEEE AFGR* (1998), 16–21.

[4] Bragdon, A. et al. 2011. Experimental analysis of touch-screen gesture designs in mobile environments. *Proceedings of CHI* (2011), 403–412.

[5] Caramiaux, B. 2012. *Studies on the Relationship between Gesture and Sound in Musical Performance*. University of Paris VI.

[6] Fdili Alaoui, S. et al. 2012. Dance Movement Qualities as Interaction Modality. *Designing Interactive Systems (DIS 2012)* (Newcastle, UK, 2012), 761–769.

[7] Grossman, T. and Balakrishnan, R. 2005. A probabilistic approach to modeling two-dimensional pointing. *ACM ToCHI*. 12, 3 (Sep. 2005), 435–459.

[8] Grudin, J. 2009. AI and HCI: Two fields divided by a common focus. *AI Magazine*. 30, 4 (2009), 48–56.

[9] Kane, S.K. et al. 2011. Usable gestures for blind people: understanding preference and performance. *Proceedings of CHI* (2011).

[10] Kratz, L. et al. 2012. Making gestural input from arm-worn inertial sensors more practical. *Proceedings of CHI* (2012), 1747–1750.

[11] Long, a. C. et al. 2000. Visual similarity of pen gestures. *Proceedings of CHI* (2000), 360–367.

[12] Lucchese, G. et al. 2012. GestureCommander: continuous touch-based gesture prediction. *Proceedings of CHI EA* (2012), 1925–1930.

[13] Mori, a. et al. 2006. Early Recognition and Prediction of Gestures. Porceedings of *ICPR* (2006), 560–563.

[14] Nielsen, M. et al. 2003. A procedure for developing intuitive and ergonomic gesture interfaces for HCI. *Int'l Gesture Workshop 2003, LNCS vol. 2915*. Heidelberg: Springer- Verlag. 409–420.

[15] Nieuwenhuizen, K. et al. 2010. Insight into goal-directed movement strategies. *Proceedings of CHI* (2010), 883–886.

[16] Rajko, S. et al. 2007. Real-time gesture recognition with minimal training requirements and on-line learning. Proceedings of *CVPR* (2007), 1–8.

[17] Rubine, D. 1991. Specifying gestures by example. *ACM SIGGRAPH Computer Graphics* (Jul. 1991), 329–337.

[18] Wilson, A.D. and Bobick, A.. 2000. Realtime online adaptive gesture recognition. *Pattern Recognition, 2000. Proceedings. 15th International Conference on* (2000), 270–275.

[19] Wilson, A.D. and Bobick, A.F. 1999. Parametric hidden markov models for gesture recognition. *IEEE TPAMI*. 21, 9 (1999), 884–900.

[20] Wobbrock, J.O. et al. 2007. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. *Proceedings of UIST* (2007), 159–168.

[21] Wobbrock, J.O. et al. 2009. User-defined gestures for surface computing. *Proceedings of CHI (*2009), 1083–1092.