# MojiBoard: Generating Parametric Emojis with Gesture Keyboards

**Jessalyn Alvina***
Dept of Computer Science
University of British Columbia
Vancouver, BC, Canada
jalvina@cs.ubc.ca

**Joanna McGrenere**
Dept of Computer Science
University of British Columbia
Vancouver, BC, Canada
joanna@cs.ubc.ca

**Chengcheng Qu**†
Dept of Computing and Communications
Lancaster University
Lancaster, United Kingdom
c.qu3@lancaster.ac.uk

**Wendy E. Mackay**
LRI, Univ. Paris-Sud, CNRS,
Inria, Université Paris-Saclay
Orsay, France
mackay@lri.fr

## ABSTRACT

Inserting emojis[1] can be cumbersome when users must swap through panels. From our survey, we learned that users often use a series of consecutive emojis to convey rich, nuanced non-verbal expressions such as emphasis, change of expressions, or micro stories. We introduce MojiBoard, an emoji entry technique that enables users to generate dynamic parametric emojis from a gesture keyboard. With MojiBoard, users can seamlessly switch between typing and parameterizing emojis.

## CCS CONCEPTS

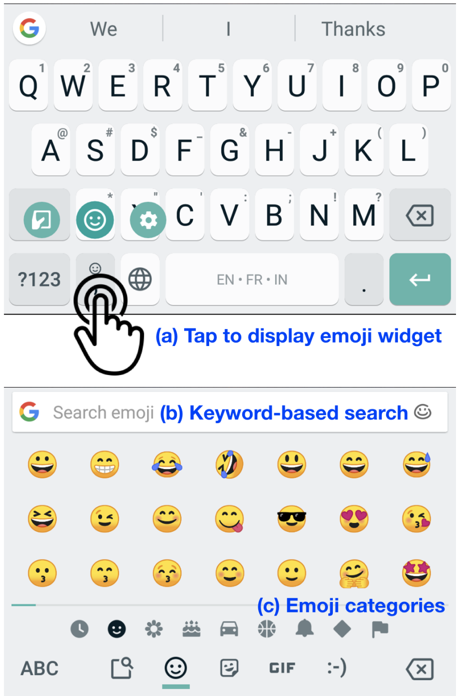
• **Information Systems** → **User Interfaces**; **Interaction Styles**.

*CHI 2019, May 4–9, 2019, Glasgow, Scotland UK*
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-5970-2/19/05...$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

*Also with .

†Also with .

[1] A set of two-byte pictographic unicode characters. Not to be confused with emoticons: a representation of a facial expression e.g. :) formed by various combinations of characters. See www.unicode.org/emoji for further details.



(a) Tap to display emoji widget

(b) Keyword-based search

(c) Emoji categories

**Figure 1: (a) To insert an emoji, users must first tap a button to display the emoji widget, then search for the desired emoji while swapping panels. Alternatively, users can filter emojis by (b) typing a keyword on the search bar, or (c) tap an emoji-category tab to immediately jump to the panel under that category. This process is cumbersome and can disturb the flow of writing.**

## KEYWORDS

emoji; continuous interaction; expressive communication; gesture input; gesture keyboard; mobile

**ACM Reference Format:**
Jessalyn Alvina, Chengcheng Qu, Joanna McGrenere, and Wendy E. Mackay. 2019. MojiBoard: Generating Parametric Emojis with Gesture Keyboards. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

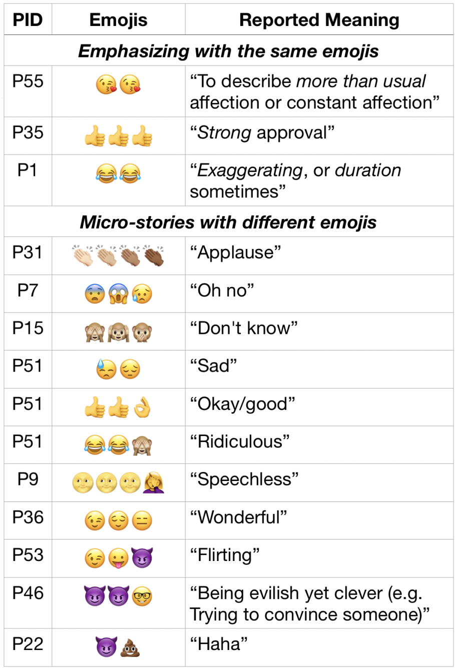
## INTRODUCTION

Around 40% of mobile activities is accounted for text-based communications [3]. Prior work has focused on improving text input efficiency, for example improving typing speed, word prediction, or spelling and grammar. However, text messaging is not only about producing text but also other non-verbal expression. Users often use emojis (e.g. 😊) to substitute or to supplement the textual message. Lee at al. described emoji common usage patterns in three categories: 1) as an expression of emotion (e.g. emotional context, expressing intensity, and emphasis); 2) for strategic purposes (e.g. reaction, self-representation, forming impression and social presence); and 3) for functional purposes (e.g. substitution or supplement for text) [6].

Most text-messaging applications on mobile devices provide a long list of emojis, even animated ones (called stickers), from which users can choose and insert into their conversation. Due to screen real estate limitations, the list is displayed on many panels that are sorted under different categories such as ‘face’, ‘animal’, ‘flag’, etc. (as in GBoard [2], see Fig. 1). This entry technique is inefficient and cumbersome: users must perform a linear search task while swapping between different panels [7, 8]. The entry rate declines the more users swap panels. EmojiZoom [8] has tried to address this issue by displaying all emojis in a smaller scale in one panel, enabling a focus+context exploration. However, the context panel can only include a certain number of emojis before the scale becomes too small to support an effective exploration. Furthermore, users often use a series of consecutive emojis, both the same or different [4]. As such, the emoji entry rate probably reaches its lowest peak when users are inserting a rarely-used emoji [7] or a series of consecutive emojis [4].

We are interested in adding more expressivity to emojis, in a simple, fun and easy-to-learn way. We began our exploration by conducting a survey to better understand how and in which context users form a series of consecutive emojis in their daily communication. Based on the result, we designed MojiBoard, an emoji input technique that enables users to parameterize dynamic emojis based on a gesture-typed emoji keyword, while minimizing the cost of panel swapping.

[2] https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin&hl=en



| PID | Emojis | Reported Meaning |
|---|---|---|
| ***Emphasizing with the same emojis*** | | |
| P55 | 😘😘 | "To describe *more than usual* affection or constant affection" |
| P35 | 👍👍👍 | "*Strong* approval" |
| P1 | 😂😂 | "*Exaggerating*, or *duration* sometimes" |
| ***Micro-stories with different emojis*** | | |
| P31 | 👏🏻👏🏼👏🏽👏🏾 | "Applause" |
| P7 | 😨😱😥 | "Oh no" |
| P15 | 🙈🙉🙊 | "Don't know" |
| P51 | 😓😔 | "Sad" |
| P51 | 👍👍👌 | "Okay/good" |
| P51 | 😂😂🙈 | "Ridiculous" |
| P9 | 🌝🌝🌝🤦‍♀️ | "Speechless" |
| P36 | 😉😌😑 | "Wonderful" |
| P53 | 😉😛😈 | "Flirting" |
| P46 | 😈😈🤓 | "Being evilish yet clever (e.g. Trying to convince someone)" |
| P22 | 😈💩 | "Haha" |

**Figure 2: Example of using consecutive emojis in text messages. Consecutive same emojis are often used to express emphasis, unusual occurrence, or duration. Consecutive different emojis are used for expressing complex emotions or telling micro stories. (PID=Participant ID)**

## SURVEY ON USAGE OF CONSECUTIVE EMOJIS

We focus on the issue of inserting a series of consecutive emojis since it is particularly inefficient, especially if the emojis are under different panels. We conducted a survey with 62 participants to understand how and when users form the series of consecutive emojis in text-messaging apps.

Participants were mainly young adults using text-messaging apps to communicate with their closest friends and family: 41% mostly send messages to their partner, 31% to best friends or siblings, 12% to other family members. Only 16% of the participants mentioned that they used text messaging most often in a professional context (e.g. colleagues or employers). The participants were heavy users of text-messaging apps: 96.8% messaged their texting-partner at least once per day, with 38.7% of the participants messaged their texting-partner more than 6 times a day.

We asked their experience using a series of consecutive emojis: the frequency of use and how they use it, either the same (e.g. 😂😂😂) or different emojis (e.g. 😉👏🏻🎉). The majority of the participants (75%) use a series of emojis in their everyday life, with 61% providing a rating of at least 3 in a 1-to-5 scale of frequency of use (1='Never', 5='Constantly'). Of those 75%, there is an equal tendency of using the same (37.2%) or different emojis (32.6%). Moreover, 30.2% of the participants use both. This suggests that young users are motivated to use emojis, even a series of emojis that takes more time to form, since they mostly text message their closest friends and family.

We traced the emoji similarity based on the classification done in [7]. Some participants (e.g. P7, P15, P51) combined similar emojis to express a rich, complex emotion, for example "🙈🙉🙊" to express a feeling of having no clue or "don't know". Others used completely different emojis consecutively to express subtle change of emotion (e.g. P53 with "😉😛😈") or to describe a story or an action (e.g. P46 with "😈😈🤓", Fig.2). Altogether, this suggests that young users form a series of emojis to convey richer, nuanced expressions that is not easily done with only a single emoji. This highlights a design opportunity: how to let users form emojis with a higher granularity of expressivity in a more fun way.

## EMOJI INPUT TECHNIQUE DESIGN

### Background Rationale

Past research has explored the possibility of increasing the granularity of expressive typed text. Lee et al. added several command buttons into a desktop text-messaging app to create animated text that changes in color, size, or position over time [5]. Expressive Keyboards [2] uses a gesture keyboard [9] to generate rich, expressive text output (e.g. dynamic typography) based on gesture properties such as speed. As such, users can generate expressive text output simply by typing.

Some emoji entry systems (e.g. GBoard) allow users to type a keyword to filter emojis on the emoji search bar (Fig. 1). Also, Pohl et al. observed that users often immediately jump to another category if

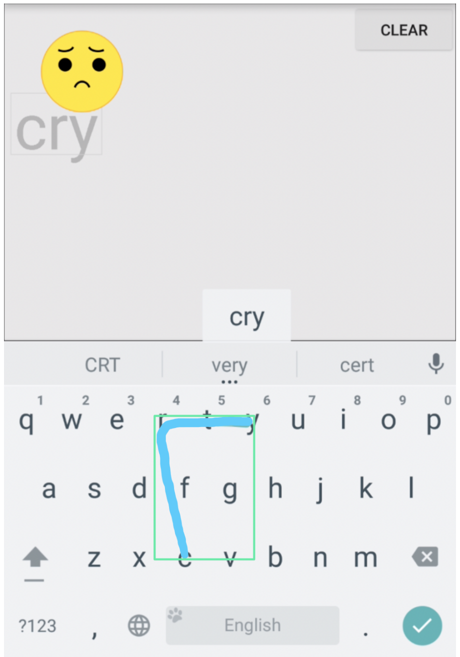


**Figure 3: MojiBoard augments a gesture keyboard to be both text and emoji input system. Users can gesture type as usual to insert text. If what they are currently gesture typing happens to be an emoji keyword, e.g. "cry", an emoji preview appears. Lifting their finger on the keyboard will insert the word "cry" and remove the preview. To insert the emoji as previewed, users must continue the gesture to the upper space, perform the 'execute' /\ gesture. As such, users can switch from inserting text and emoji without swapping panels.**

the first panel displayed does not contain the desired emoji [7]. This indicates that users may *recall* the category name or the keyword with which the desired emoji is associated.

The picking v.s. querying in emoji input can be similarly observed in the *recognition* v.s. *recall* mechanism in command selection. CommandBoard [1] leveraged a gesture keyboard to enable users to select a command by gesture-typing the command keyword. A gesture that ends on the keyboard area is interpreted as a word, while a gesture that ends on the area above the keyboard is interpreted as a command selection.

Similarly, we are interested in the possibility of leveraging users' ability to recall a keyword to insert an emoji in a more fun and efficient way, so that users can "*just type the emoji*" from their keyboard without switching to the emoji widget. We aim to reduce the cost of searching while increasing the granularity of expressivity in emoji entry system.

## MojiBoard

We introduce **MojiBoard**, which augments a gesture keyboard to generate animated parametric emojis. This enables users to be able to convey a nuanced expression, such as emphasis, a change of emotion, or micro story. We drew the inspiration from Expressive Keyboards [2] and CommandBoard [1]: we leverage the otherwise-unused gesture input variations, and with a tail gesture at the end, transform them into parametric emojis which expressions changes depending on how the users perform the gesture. We opt for a gesture keyboard since it is already widely used and a unistroke can potentially generate more properties than a tap gesture. Previous work has demonstrated that users can reliably control their gesture variations on mobile devices [2].

MojiBoard considers three discrete interaction spaces: keyboard, command bar, upper space (Fig.5). The keyboard serves both as a text input (i.e. typing) and emoji input space. The emoji input is initiated by gesture typing an emoji keyword, such as "cry" then continuing the gesture to the upper space. We expand the functionality of the keyboard in CommandBoard [1]: not only to type the keyword, but also to parameterize the emoji output.

*Generating Parametric Emoji.* Most emoji systems have a set of keywords associated to an emoji (e.g. "smile" to 😊, "sad, cry" to 😭). MojiBoard lets users gesture type the keyword and quickly insert it into their text message. Every time users gesture type, the gesture keyboard displays the four most-likely word candidates to users: the first one as the chosen word output and the rest on the keyboard's suggestion bar (Fig.3), that sometimes auto complete the word (as illustrated in Fig.5). MojiBoard checks progressively if any of the words is an emoji keyword. If the keyword matching is successful, MojiBoard previews the generated emoji. To transform the word into an emoji, users can continue gesturing to the area above the keyboard, and perform a /\ gesture (Fig. 4). MojiBoard calculates the *size* (i.e. bounding box area, the green box shown in Fig.3 and Fig.4) and the *curviness ratio* (i.e.

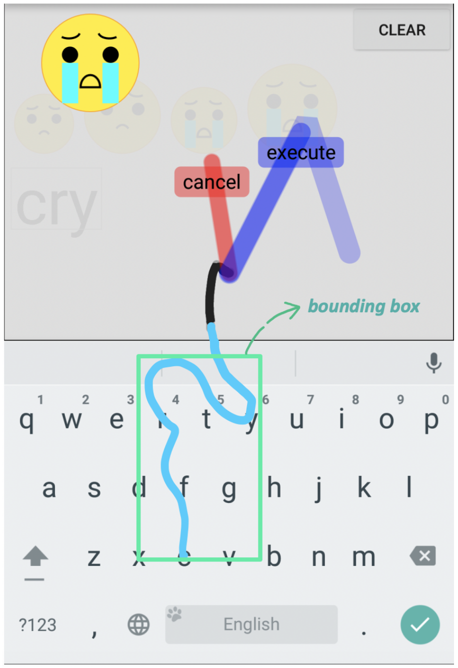


**Figure 4: Generating a parametric crying emoji with MojiBoard. A user gesture type "cry" by drawing a wobbly, inflated gesture (inside green bounding box). The preview of the final result appears (top left), and as the user continues drawing to the area above the keyboard, users can either draw an 'execute' gesture (dark-blue gesture) or a 'cancel' gesture. By following the 'execute' gesture and lifting the finger, an animated emoji is inserted into the text field.**

radius of curvature) in real time. The bigger the bounding box, the bigger the emoji is. Likewise, the more curvy the gesture, the more intense the emoji expression is. For example, the gesture in Fig.3, a relatively straight and smaller gesture, generates a sad face with a small frown. As users increase the curviness (i.e. by wiggling the gesture) and the bounding box size, the emoji grows bigger and the expression changes from a small frown into a crying face (Fig.4).

The matching keyword is held until users lift their finger or they type a completely different word, hence lowers the chance of the keyword changing as they wiggle or inflate the gesture. If there are more than one emojis associated with the keyword, the frequently-used options are displayed on the *command bar* (Fig. 5) above the keyboard, and users can cross the desired emoji as they go to the upper area. The generated emoji animates the change of expression in two seconds, e.g. from a small frown to a crying face, giving a more dramatic expression. Users can tap the emoji to replay the animation, repeated three times (hence, 6 seconds in total). To cancel the emoji generation, users simply make a straight gesture (instead of /\) on the area above the keyboard.

*Picking Random Emojis.* Aside from generating an animated parametric emoji, we explore the possibility of inserting a series of random emojis based on the categories. This can be a fun, simplistic alternative to form a micro story with rarely-used emojis. Users can gesture type "random" to start picking the emojis. A preview appears with all the categories displayed on the command bar (Fig.5). As users slide up, they can cross the categories from which they want to include. To pick two or more emojis from the same category, the gesture exits the command bar, and then enters it again through the desired category. The preview displays the picked emojis. Users perform the /\ after they finish picking to insert the emojis, or cross the 'cancel' option on the command bar to cancel immediately.

## CONCLUSION AND FUTURE WORK

We found that users often use emojis to convey a nuanced expression such as emphasis, change of emotion, or micro stories. With MojiBoard, users can manipulate aspects of their gestures to parameterize an animated emoji, such as the intensity of the expression. Users do not need to display the emoji widget, hence they can seamlessly switch between typing and inserting emojis with a higher degree of expressions from their keyboard. Since the parameters are measured from users' gestures, we argue that it is more personal than choosing from a pre-defined emoji. However, a more complete, sophisticated emoji engine is needed to achieve personalized emoji keyboards. While using a random function to pick emojis can be fun, future work should focus on enabling users picking specific emojis to form micro stories more efficiently.

Technically, the emoji generated with MojiBoard can be as cheap as a three-byte unicode character. Emojis are currently represented as a two-byte unicode character. The additional parametric features introduced in MojiBoard (e.g. curviness ratio) can be added as an additional byte to the emoji. Each

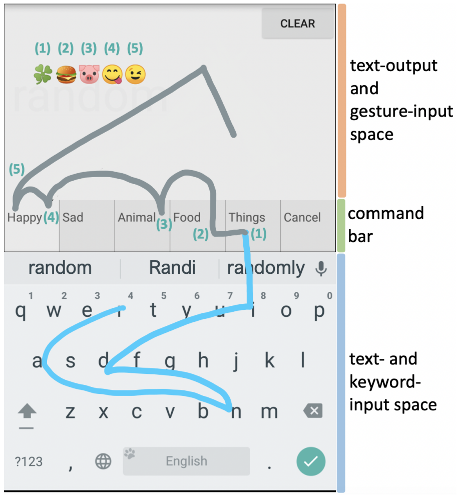


**Figure 5: Inserting a series of random emojis based on categories. Users gesture type the keyword "random" to start picking. MojiBoard auto-completes the gesture if the keyword already appears in the suggestion bar, hence typing "randi" is enough to initiate it. In this figure, the user picked 5 emojis from four different categories (as numbered) by crossing the different categories on the command bar. Users can pick as many as desired by going in and out the command bar area while crossing the desired categories.**

platform or text-messaging applications can decide on the rendering independently, that is not limited to animation but also other rich output types such as stickers, skin-tone modifiers, or GIF image parameters. As such, we can potentially increase the personalization and the expressivity aspects of an emoji, with fewer bytes than if users insert multiple consecutive emojis in order to convey essentially the same expression. Nevertheless, we believe the approach should be an additional alternative on top of the current system, instead of a replacement.

In the future, we want to expand MojiBoard so the parameterization can be done both from the keyboard and the emoji widget. We are particularly interested in creating a parametric emoji engine that interpolates between different expressions, for example from happy to shock to crying, to see how users appropriate such system in their daily conversation.

## ACKNOWLEDGMENTS

We thank Xiaojun Bi for providing the gesture keyboard prototype used in MojiBoard. This work was partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and European Research Council (ERC) grant no 321135 CREATIV: Creating Co-Adaptive Human-Computer Partnerships.

## REFERENCES

[1] Jessalyn Alvina, Carla F. Griggio, Xiaojun Bi, and Wendy E. Mackay. 2017. CommandBoard: Creating a General-Purpose Command Gesture Input Space for Soft Keyboard. In *Proc. ACM UIST'17*. ACM, NY, USA, 17–28. https://doi.org/10.1145/3126594.3126639

[2] Jessalyn Alvina, Joseph Malloch, and Wendy E. Mackay. 2016. Expressive Keyboards: Enriching Gesture-Typing on Mobile Devices. In *Proc. ACM UIST'16*. ACM, NY, USA, 583–593. https://doi.org/10.1145/2984511.2984560

[3] Barry Brown, Moira McGregor, and Donald McMillan. 2014. 100 Days of iPhone Use: Understanding the Details of Mobile Device Use. In *Proc. ACM MobileHCI'14*. ACM, NY, USA, 223–232. https://doi.org/10.1145/2628363.2628377

[4] Zhenpeng Chen, Xuan Lu, Wei Ai, Huoran Li, Qiaozhu Mei, and Xuanzhe Liu. 2018. Through a Gender Lens: Learning Usage Patterns of Emojis from Large-Scale Android Users. In *Proc. WWW'18*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 763–772. https://doi.org/10.1145/3178876.3186157

[5] Joonhwan Lee, Soojin Jun, Jodi Forlizzi, and Scott E. Hudson. 2006. Using Kinetic Typography to Convey Emotion in Text-based Interpersonal Communication. In *Proc. ACM DIS '06*. ACM, NY, USA, 41–49. https://doi.org/10.1145/1142405.1142414

[6] Joon Young Lee, Nahi Hong, Soomin Kim, Jonghwan Oh, and Joonhwan Lee. 2016. Smiley Face: Why We Use Emoticon Stickers in Mobile Messaging. In *Proc. ACM MobileHCI '16*. ACM, NY, USA, 760–766. https://doi.org/10.1145/2957265.2961858

[7] Henning Pohl, Christian Domin, and Michael Rohs. 2017. Beyond Just Text: Semantic Emoji Similarity Modeling to Support Expressive Communication;. *ACM Trans. Comput.-Hum. Interact.* 24, 1, Article 6 (March 2017), 42 pages. https://doi.org/10.1145/3039685

[8] Henning Pohl, Dennis Stanke, and Michael Rohs. 2016. EmojiZoom: Emoji Entry via Large Overview Maps &#128516;&#128269;. In *Proc. ACM MobileHCI '16*. ACM, NY, USA, 510–517. https://doi.org/10.1145/2935334.2935382

[9] Shumin Zhai and Per Ola Kristensson. 2012. The Word-gesture Keyboard: Reimagining Keyboard Interaction. *Commun. ACM* 55, 9 (Sept. 2012), 91–101. https://doi.org/10.1145/2330667.2330689