Test Automation for Gesture-Based Interfaces

Marc Hesenius[1], Matthias Book[2], and Volker Gruhn[1]

[1] University of Duisburg-Essen, Essen, Germany
{firstname.lastname}@uni-due.de
https://se.wiwi.uni-due.de
[2] University of Iceland, Reykjavík, Iceland
book@hi.is

Abstract. With the increasing ubiquity of touch-enabled devices, gestures are quickly becoming a mainstream interaction modality. The implementation of gesture-based user interfaces however requires specialized engineering methods. Testing gestures is particularly crucial to ensure smooth and intuitive interaction, but is often manual work, i.e. cumbersome, time-consuming, and error-prone. Unfortunately, hardware and framework providers failed to incorporate adequate gesture testing abilities into their products. We therefore argue that test automation techniques for gesture-based interfaces are urgently needed. However, they require further investigation into suitable methods and tools. We therefore propose a research agenda that focuses on evaluating gestures' quality (e.g. in terms of ergonomics, ambiguity, cultural connotations etc.), and automatically testing gestures' functionality through generation and injection of realistic gesture trajectories into a UI under test.

Keywords: Gestures · Test Automation · UI Tests · Interaction Tests

1 Introduction

Gestures are rapidly becoming a mainstream – and nowadays often primary – User Interface (UI) modality on a broad variety of devices, ranging from smart watches and phones, tablets and computer screens to large wall-mounted or tabletop displays. In addition to the multi-touch surface gestures performed on these devices, spatial gestures and poses play a key role in the interaction with augmented and virtual reality applications.

Designing and implementing effective gesture-based user interfaces however poses a multitude of engineering challenges – from the challenges of precisely yet efficiently and readably specifying transient motions of various limbs in relation to UI widgets, and of recognizing gestures given the unavoidable fuzziness and variance in the input signals, to the challenges of accommodating very different device types, dimensions and sensor capabilities, and the challenges of gesture quality (in terms of ergonomics, mnemonics, motoric and semantic ambiguity, etc.).

These challenges make extensive – and thus automated – testing of applications' gesture-based interface with regard to the above-mentioned functional and quality properties essential. Suitable test automation technology for gesture-based interfaces is, however, almost non-existent. While powerful tools for app development exist, manufacturers failed to introduce sufficient gesture testing facilities into their respective Software Development Kits (SDKs) and – even more importantly – their devices. Several interesting testing approaches, especially for surface gestures, have been developed in the past (e.g. [4,10]). However, these approaches should only be considered the beginning of what should in our opinion be a much broader field of research towards a much more powerful set of methods and tools to be put at software engineers' disposal. In this position paper, we describe a research agenda for developing this field in more detail.

2 Open Issues and Research Questions

The fundamental difference between gesture-based and traditional keyboard- and mouse-based interaction is that gestures are much more user-specific (to the point of being biometric markers [1]): Gesture trajectories vary by users' motor skills and experience, physical state (e.g. stress/exhaustion), environment and context parameters, etc., and are never precisely reproducible.[3] This makes testing gesture-based interfaces more complex than traditional WIMP (windows, icons, menus, pointers) UI tests from a quality and functional perspective.

In the following subsections, we describe these challenges and sets of concrete research questions that can be derived from them in more detail.

2.1 Evaluating Gestures' Quality

The whole point of gesture-based user interfaces is to make the interaction with an app (e.g. the manipulation of data) feel as natural as possible, opposed to more artificial interaction modalities such as typing or navigating menus. The gestures designed for interaction with an app therefore need to be evaluated early in the development process regarding their quality, especially in terms of usability.

This includes an evaluation of whether the intended gestures can be ergonomically executed or require user movements that can lead to immediate strain or long-term fatigue [7,8]; whether their shape is robust against the unavoidable minor variances introduced in each and every manual execution, or larger variations due to user responses to an application's physical context or digital state; and whether they are distinct enough to be reliably distinguished from each other [11]. These issues are exacerbated when considering spatial gestures, not only due to the additional degrees of freedom in the involved limbs and trajectories, but also due to the issues in segmenting spatial gestures and distinguishing intentional command input from natural but irrelevant motions.

Notably, these evaluations are affected by technical as well as cultural context: For one thing, devices' form factors have significant impact on which limbs are

[3] While the same is true for mouse trajectories and typing intervals, such variations have no impact on input recognition in those modalities.

involved in executing a gesture (e.g. phones and tablets are held differently and thus (finger-)operated subtly differently, while large wall-mounted and tabletop screens allow more involvement of both arms than hand-held devices do, but differ in how they cause fatigue, etc.). For another thing, given the various ways in which some gestures may be interpreted by different users, an evaluation of real-life and in-app gesture semantics, and their alignment or divergence between users and cultures (e.g. meanings of the *thumbs up* sign), can additionally be helpful to uncover usability issues not caught by traditional testing methods. Last but not least, different user groups with different motor capabilities (e.g. elderly people, disabled people, users wearing work gloves) need to be considered depending on the app's target audiences.

Due to the iterative-incremental nature of modern application development, gesture quality evaluation cannot be an initial up-front activity, but must – much like functional regression testing – be performed throughout the project. This is to ensure that gestures introduced for new features in later iterations do not introduce ambiguities with earlier-defined gestures, that they maintain the same level of usability, and maybe even that they follow a consistent interaction style. Gesture quality regression testing thus requires automatic evaluation of gesture quality as part of a larger test suite, and could possibly be integrated as a step in existing continuous integration processes.

Along these lines, we propose to pursue the following research questions:

- In which ways and to which degrees can physiological, psychological, cultural and contextual influences affect or restrict the execution of gestures?
- How can such influences and/or the resulting gesture variances be expressed in ways that are compatible with the idealized trajectories that are described by gesture specification languages?
- How can such influences and/or variations be mapped to quality metrics, e.g. for particularly robust/distinct gestures, particularly straining gestures, particularly memorable gestures, etc.?
- Which guidelines, constraints and best practices should be recommended to gesture designers, in order to achieve high gesture quality in terms of ergonomics, distinctiveness, etc.?
- How can gesture specifications or implementations be evaluated or tested with regard to the above criteria?

To work towards answering these questions, we suggest the following actions:

- Empirical studies identifying the impacts of physiological, psychological, cultural, and contextual influences on gesture execution
- Extension and adaption of gesture specification languages for expressing gesture variations
- Definition of guidelines and best practices for gesture ergonomics and usability
- Creation and evaluation of tools for (semi-)automated gesture quality analysis

2.2 Testing Gestures' Functionality

Testing UIs is a central element in Behavior-Driven Development (BDD) and often referred to as *User Acceptance Tests*. Users check the UI according to a scenario that defines input, interaction, and output. Several frameworks seeking automation of this process and suitable notations to define test scenarios have been developed in recent years – in simple terms, they take a scenario, mimic user behavior (e.g. which buttons are clicked and what data is entered into text boxes), and investigate the outcome.

To test gestures' functionality, however, adopting classic WIMP UI testing techniques of injecting predefined – and thus perfectly spec-matching – inputs into an app is insufficient: In productive use, the app will be exposed to a broad variance in how the gestures are executed, so the injected trajectories must exhibit realistic variance, both in terms of unavoidable muscle jitter and deliberate shape transformations to accommodate the data or widgets being manipulated. This means that in addition to the already challenging task of specifying gesture inputs, we need specification languages or language extensions that are capable of specifying gesture test cases in all their variability, and testing frameworks that can generate and inject according trajectories and examine the expected application reactions. Several gesture notations have been proposed in recent years (e.g. [3,6,9]) that can be reused and extended for testing purposes, e.g. by adding information on the degree of trajectory variance to be tested for. To create realistic test cases, the variance should not just be a random variation of the specified ideal gesture, but reflect different user groups, device types, interaction specifics (e.g. involved limbs), etc.

Injecting gesture information into the application then requires a gesture simulator: The simulated trajectories must be transformed into input events (e.g. motion events in Android), which in turn trigger the corresponding reactions in the application under test. Existing UI automation techniques and algorithms from related areas (e.g. for generating manual signatures [2] or synthesizing sign language [12]) could be used as a starting point for generating synthetic gestures. A suitable simulator should incorporate means to create and adapt gestures depending on additional information, e.g. how fast and where the gesture should be performed, to mirror the wide range of possibilities gesture-based interaction makes possible. Furthermore, some special cases must be considered. For example, the various options that cooperative gestures [13] allow in multi-user applications require the simulator to generate several gestures and synchronize their execution.

To develop the necessary testing tools, we propose to pursue the following research questions in this area:

- Which of the previously identified physiological, psychological, cultural, and contextual factors that influence gesture variance are relevant to which kinds of gestures and how critical are they for different types of inputs (e.g. selection, manipulation, navigation, etc.)?
- How can tolerance thresholds for gesture variance be defined, both in terms of concrete metrics for different influencing factors and in terms of technical compatibility with gesture and test case specification languages?

- How can gesture variance be introduced into automated UI testing? Which types and degrees of variance should be specifiable? What strategies should be used to automatically generate and test such variances?
- How can individual, deliberate transformations/adaptions of gestures (e.g. in response to dynamic UI elements) be distinguished from inadvertent jitter in specification and recognition? How can test cases for complex gesture interactions with dynamic UI elements be specified?

To answer these questions, we suggest the following actions:

- Transfer of insights from research on gesture variance *analysis* for the purpose of specification evaluation (Sect. 2.1) to research on gesture variance *synthesis* for the purpose of test case generation
- Definition or extension of gesture specification languages to incorporate test-specific information, especially with regard to factors influencing gesture variance
- Definition or extension of gesture specification or test definition languages to incorporate gesture transformations in response to dynamic UI elements
- Development of gesture simulators synthesizing realistic gesture variance
- Development of test frameworks injecting realistic gesture trajectories into applications and evaluating responses

3 Conclusion

We described open research questions and necessary actions to develop automated UI tests for gesture-based applications. We envision two different types of tools: a gesture checker testing defined gestures against a set of quality rules to validate if they are ergonomical, unambiguous, etc., and a system for automating UI tests that automatically injects realistic gesture trajectories into an application.

As an outlook to even further research opportunities, considerations of multimodal interaction also suggest that other modalities (e.g. voice recognition) are subject to similar challenges and require suitable test automation techniques [5]. In addition, multi-device and multi-user scenarios introduce additional technical and semantic challenges in the recognition and interpretation of gestures (e.g. regarding synchronization, conflicts, etc.). A solid engineering foundation for single-user/single-device gesture interaction is however necessary to ensure that each component of a multi-modal, multi-device, or multi-user interface will contribute reliable inputs to the system, to ensure a smooth user experience.

References

1. Bevan, C., Fraser, D.S.: Different Strokes for Different Folks? Revealing the Physical Characteristics of Smartphone Users from their Swipe Gestures. International Journal of Human-Computer Studies **88**, 51–61 (2016)

2. Galbally, J., Plamondon, R., Fierrez, J., Ortega-Garcia, J.: Synthetic On-Line Signature Generation. Part I: Methodology and Algorithms. Pattern Recognition **45**(7), 2610 – 2621 (2012)
3. Ganhör, R., Spreicer, W.: Monox: Extensible Gesture Notation for Mobile Devices. In: Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services. pp. 203–212. MobileHCI '14, ACM, New York, NY, USA (2014)
4. Hesenius, M., Griebe, T., Gries, S., Gruhn, V.: Automating ui tests for mobile applications with formal gesture descriptions. In: Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services. pp. 213–222. MobileHCI '14, ACM, New York, NY, USA (2014)
5. Hesenius, M., Griebe, T., Gruhn, V.: Towards a Behavior-Oriented Specification and Testing Language for Multimodal Applications. In: Proceedings of the 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems. pp. 117–122. EICS '14, ACM, New York, NY, USA (2014)
6. Hesenius, M., Gruhn, V.: Introducing gesturecards: A prototyping gesture notation. In: Proceedings of the 9th Nordic Conference on Human-Computer Interaction. pp. 111:1–111:6. NordiCHI '16, ACM, New York, NY, USA (2016)
7. Hoggan, E., Nacenta, M., Kristensson, P.O., Williamson, J., Oulasvirta, A., Lehtiö, A.: Multi-touch pinch gestures: Performance and ergonomics. In: Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces. pp. 219–222. ITS '13, ACM, New York, NY, USA (2013)
8. Hoggan, E., Williamson, J., Oulasvirta, A., Nacenta, M., Kristensson, P.O., Lehtiö, A.: Multi-touch Rotation Gestures: Performance and Ergonomics. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 3047–3050. CHI '13, ACM, New York, NY, USA (2013)
9. Kammer, D., Henkens, D., Henzen, C., Groh, R.: Gesture Formalization for Multitouch. Software: Practice and Experience **45**(4), 527–548 (2015)
10. Khandkar, S.H., Maurer, F.: A domain specific language to define gestures for multi-touch applications. In: Proceedings of the 10th Workshop on Domain-Specific Modeling. pp. 2:1–2:6. DSM '10, ACM, New York, NY, USA (2010)
11. Long, Jr., A.C., Landay, J.A., Rowe, L.A., Michiels, J.: Visual similarity of pen gestures. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 360–367. CHI '00, ACM, New York, NY, USA (2000)
12. Losson, O., Vannobel, J.M.: Sign Specification and Synthesis. In: Braffort, A., Gherbi, R., Gibet, S., Teil, D., Richardson, J. (eds.) Gesture-Based Communication in Human-Computer Interaction, Lecture Notes in Computer Science, vol. 1739, pp. 239–251. Springer Berlin Heidelberg (1999)
13. Morris, M.R., Huang, A., Paepcke, A., Winograd, T.: Cooperative gestures: multiuser gestural interactions for co-located groupware. In: Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06. p. 1201. ACM Press, New York, New York, USA (apr 2006)