

End-User Authoring of Gestural Interactions
Koç University

Mehmet Aydın Baytaş

2014-09-01

Contents

1	Introduction	5
1.1	Motivation	5
1.2	Scope	6
1.3	Aim	8
1.3.1	Research Questions	8
1.3.2	Hypothesis and Expected Result	8
2	Background and Related Work	11
3	Method	13
4	Hotspotizer: Authoring Mid-Air Gestural Interactions	15
5	Hotspotizer: Design and Evaluation	17
6	Conclusion	19
A	Attributions	25

Chapter 1

Introduction

1.1 Motivation

Historically, using the alignment and motion of human limbs as an input modality for human-computer interfaces has been accomplished through intrusive methods - by placing markers or sensors on the body. Up until recently, non-intrusive sensing of human limb positions has been limited to research efforts [12, 27, 28]. In recent years, vision-based skeletal tracking sensors have become commercially available from a variety of established vendors such as Microsoft¹ (Figure 1.1) and Asus². Non-intrusive - or *perceptual* [5, 39] - sensing of body movements has thus become widely accessible for both commercial and hobbyist applications [9].



Figure 1.1 – The Microsoft Kinect sensor is equipped with a depth camera that can “see” the positions and motion of human limbs.

There are a variety of computing applications where the non-intrusive detection of human limb positions can be desirable as an input modality. Gaming is an obvious one (see Figure 1.2), where using movements with “prior mappings” to real-world happenings increases immersion [4]. Another one is user interfaces on public interactive systems: An input modality that does not require physical contact is often cheaper to deploy and maintain, and more hygienic to use. Of course, there are numerous other contexts where hygiene considerations make a touchless interface may be desirable: Cooking, gardening, working on a dirty mechanism, and performing surgery [42] come to mind. Other applications for perceptual interfaces include convenient control of smart homes [38], interactive art and musical instruments³, interfaces for manipulating 3D images [11], and spatial medicine [16, 24, 37].

The design and development of perceptual interfaces requires that *gestures* - limb positions and movements that constitute inputs to the interface - be *authored* in a machine-readable manner and mapped to events within the interactive system. This can be done in a textual programming environment using tools supplied by vendors of gesture-sensing hardware^{4,5} or third parties⁶. Using textual programming to author gestures, however, has drawbacks - both for adept software developers and for comparatively non-technical users such as designers, artists, hobbyists or researchers in fields other than computing. (I will henceforth refer to these users who produce software not as an end, but as a means towards goals in their own domain, as *end-users* [19]). Specifically; for end-users, textual programming embodies a significant *gulf of execution* – a chasm between the user’s goals and the actions taken within a system to achieve

¹microsoft.com/en-us/kinectforwindows

²www.asus.com/Multimedia/Motion_Sensor_Products

³vimeo.com/45417241

⁴microsoft.com/en-us/kinectforwindowsdev

⁵softkinetic.com

⁶kinecttoolbox.codeplex.com



Figure 1.2 – Gaming with the Microsoft Kinect.

those goals – since it introduces additional tasks like setting up the programming environment and getting used to the development ecosystem. For both end-users and software developers, textual programming embodies a significant *gulf of evaluation* - a gap between a system's output and the users's expectations and intentions - since they do not fully support the embodied [6], reflective [33], and experiential [22] practices inherent in the design, construction and evaluation [15] of these highly interactive artifacts [21, 29]. (See Figure 1.3 for a visualization of the *gulf of execution* and the *gulf of evaluation*. See [31, 32] for a detailed discussion on the gulfs.)

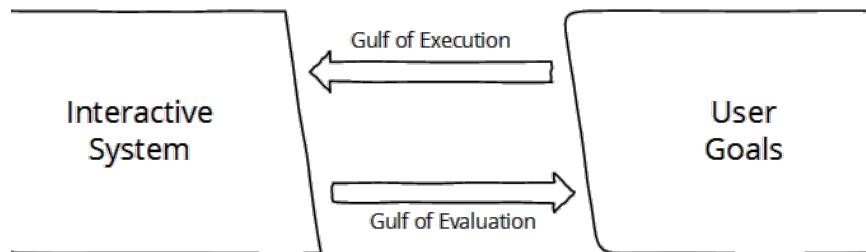


Figure 1.3 – The *gulf of execution* and the *gulf of evaluation*. The gulfs pertain to unidirectional aspects of interaction: The *gulf of execution* lies between the user's goals and the interactive system; the *gulf of evaluation* divorces the interactive system from the users's expectations and intentions.

Using appropriate tools for a job helps bridge the gulfs of evaluation and execution.

This thesis presents my attempt at producing an appropriate tool to support the design and development of perceptual gesture-based interfaces by end-users.

1.2 Scope

In human-computer interaction (HCI) and interaction design (IxD) literature, the usage of the word *gesture* is ambiguous: Depending on the context, it may denote finger strokes on a touchscreen [25], deformations inflicted on a tangible input device [41], full-body poses [40], even finger movements on a keyboard [43]. For the purposes of this manuscript; the following definition, adapted from [20], will be used:

A gesture is a movement or position of a human body part that conveys information.

By design, in order to accomodate existing works, this definition is broad. It allows for the use of any body part in gesturing as well as the use of sensing devices such as mouse, styli

and gloves. It does not require an explicit intention to justify gesturing, thus accommodating non-command user interfaces [30] such as those that respond to affective [17] and habitual [23] gestures.

More specifically, I use the term *mid-air gesture* to denote gestures that carry the following two characteristics:

1. The gesture should be performed in a volume where limbs can move freely in 3 dimensions; e.g. free space. This excludes gestures that are constrained to affect a tangible medium; e.g. a keyboard, a touch-sensitive surface, or a shape display [8].
2. The gesture-sensing hardware should detect the movements and/or location of human body parts perceptually, without requiring physical contact; e.g. with a camera. This excludes systems that sense gestures using devices that must be worn, wielded, or touched; e.g. a mouse, a ring^{7,8,9}, or an accelerometer [1, 18].

The gesture sensing input device used during the course of this work was a Microsoft *Kinect for Xbox 360*; chosen due to its availability. The device employs an infrared projector-camera pair to capture 3D *depth images*. If what resembles a typical human body is present in the depth image, the positions (relative to the sensor) of its large limbs are detected using machine learning [13, 34–36]. A *skeletal model* of the user is produced in this fashion (Figure 1.4). The alignment and motion of the skeletal model is then used to control interactive applications.

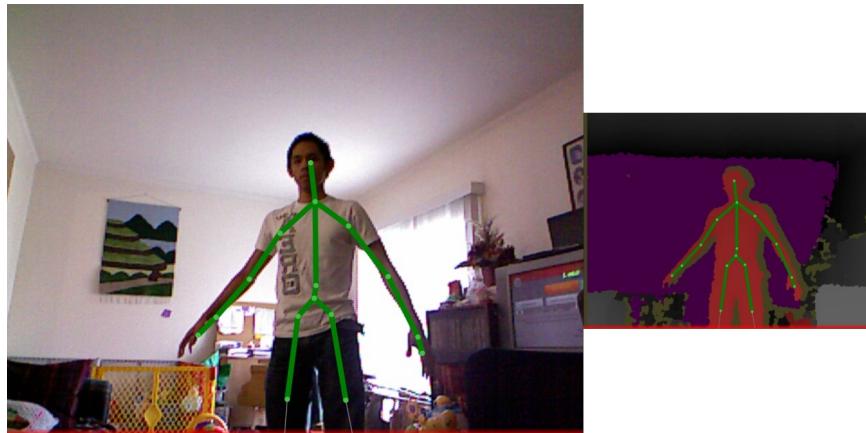


Figure 1.4 – Skeletal tracking from a depth image.

The Microsoft *Kinect for Windows Software Development Kit (SDK)* version 1.8 was used to implement gesture sensing. The capabilities of the Kinect sensor and the SDK are not limited to skeletal tracking; they also include the detection of hand gestures, speech recognition, background removal from videos, facilitating proxemic interaction [2], fusing color and 3D images, and fusing data from multiple sensors¹⁰. These topics, however, lie outside the scope of this work.

In kinesiology, human movements are classified according to movement precision [14]: *Gross motor skills* denote large and comparatively imprecise movements produced by large muscles; e.g. jumping, or lifting weights. *Fine motor skills* involve smaller movements with higher accuracy and precision; e.g. typing, or writing. Note that gestures do not always belong strictly to one of two discrete classes. Rather, the distinction between fine and gross gestures forms a continuum characterized by the size of the engaged musculature and the trade-off between force and precision [7] (Figure 1.5). One limitation of the skeletal tracking technology used for this work is that it can only detect gross gestures¹¹. Thus, this work deals only with issues related to the use of *gross movements* of the human limbs as an interaction modality in computing.

⁷wearfin.com

⁸hellonod.com

⁹logbar.jp/ring

¹⁰blogs.msdn.com/b/kinectforwindows/archive/2013/09/17/updated-sdk-with-html5-kinect-fusion-improvements-and-more.aspx

¹¹Currently, the Kinect SDK does have support for hand gestures. However, this feature was not available while the design work described in this thesis was done.

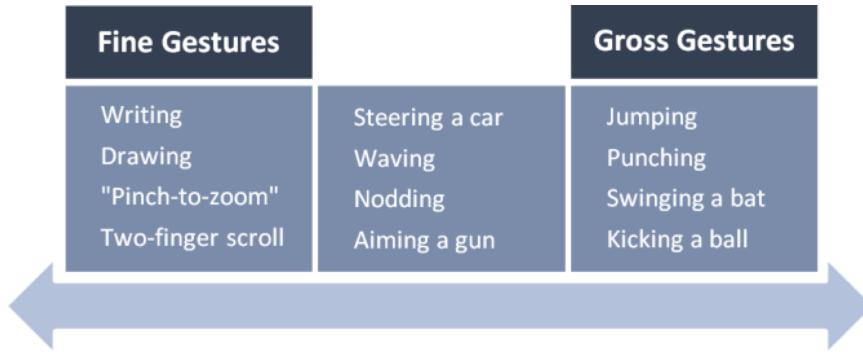


Figure 1.5 – The continuum of *fine* vs. *gross* movements.

In sum, the scope of this work covers the design and development of a *software tool for authoring gross mid-air gestures* for interactive computing systems that employ *skeleton tracking perceptual input devices*.

1.3 Aim

The aim of this thesis is to document the design, development, deployment and evaluation of a *software application for authoring gross mid-air gestures for skeletal tracking perceptual input devices*.

The features and the user interface of this application are geared towards supporting the activities of *end-users* rather than adept computer programmers. The methods employed in the design and evaluation of the application are selected to be appropriate for this purpose. The practices employed for the construction and deployment of the application also reflect its end-user focus: The application must perform well and reliably on users' computers, be easy to obtain and set up, and be maintainable to facilitate rapid adaptation to evolving technologies and user needs [3, 26].

1.3.1 Research Questions

The main research question pursued in this thesis is as follows:

How can end-users' authoring of gross mid-air gestures for skeletal tracking interfaces be supported with a software tool?

The main question breaks down into the following sub-questions that align with the research activities:

- What are the desiderata and design considerations that would pertain to mid-air gesture authoring software for end-users?
- What methods should be used to elicit the desiderata and design considerations from the target users?
- What are the technical considerations that would pertain to the application?
- What methods are appropriate to evaluate the application?

1.3.2 Hypothesis and Expected Result

I hypothesize that a suitably designed gesture authoring tool will accomplish the following:

- It will enable *end-users* with no experience in textual programming and/or gestural interfaces to introduce gesture control to computing applications that serve their own goals.
- It will provide *developers* and *designers* of gestural interfaces with a rapid prototyping tool that can be used to experientially evaluate designs.

- By providing an appropriate representation of the design space, it will expose the capabilities and limitations of the technology and be fit to serve *educational* purposes.

This application constitutes an artifact of *research through design* [10]. Thus it is expected to fulfill the following criteria proposed in [44]:

- *Process*. The methods employed must be selected rationally and documented rigorously.
- *Invention*. Various topics must be integrated in a novel fashion to create the artifact.
- *Relevance*. The artifact must be situated within a real, current context; while supporting a shift towards a justifiably preferable state.
- *Extensibility*. The work must enable the future exploitation of the knowledge derived from it.

The expected result from this work is a software application that will accomplish the goals above and constitute an authentic contribution as an artifact of research through design.

Chapter 2

Background and Related Work

Chapter 3

Method

Chapter 4

Hotspotizer: Authoring Mid-Air Gestural Interactions

Chapter 5

Hotspotizer: Design and Evaluation

Chapter 6

Conclusion

Bibliography

1. Ashbrook, D., and Starner, T. MAGIC: A Motion Gesture Design Tool. *Proc. CHI '10*, 2159–2168.
2. Ballendat, T., Marquardt, N., and Greenberg, S. Proxemic Interaction: Designing for a Proximity and Orientation-aware Environment. *Proc. ITS '10*, 121–130.
3. Brooks, F. P. *The Mythical Man-Month: Essays On Software Engineering*. Pearson Education, 1995.
4. Cairns, P., Li, J., Wang, W., and Nordin, A. I. The Influence of Controllers on Immersion in Mobile Games. *Proc. CHI '14*, 371–380.
5. Crowley, J. L., Coutaz, J., and Bérard, F. Perceptual User Interfaces: Things That See. *Communications of the ACM*, 43(3), 2000: 54–ff.
6. Dourish, P. *Where the Action Is: The Foundations of Embodied Interaction*. MIT press, 2004.
7. Edwards, W. *Motor Learning and Control: From Theory to Practice*. Cengage Learning, 2010.
8. Follmer, S., Leithinger, D., Olwal, A., Hogge, A., and Ishii, H. inFORM: Dynamic Physical Affordances and Constraints Through Shape and Object Actuation. *Proc. UIST '13*, 417–426.
9. Francese, R., Passero, I., and Tortora, G. Wiimote and Kinect: Gestural User Interfaces Add a Natural Third Dimension to HCI. *Proc. AVI '12*, 116–123.
10. Frayling, C. Research in Art and Design. *Royal College of Art Research Papers*, 1(1), 1993: 1–5.
11. Gallo, L. A Study on the Degrees of Freedom in Touchless Interaction. *Proc. SA '13*, 28:1–28:4.
12. Gavrila, D. M. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, 73(1), 1999: 82–98.
13. Girshick, R., Shotton, J., Kohli, P., Criminisi, A., and Fitzgibbon, A. Efficient Regression of General-activity Human Poses from Depth Images. *Proc. ICCV '11*, 415–422.
14. Haibach, P. S., Reid, G., and Collier, D. H. *Motor Learning and Development*. Human Kinetics, 2011.
15. Hartmann, B., Klemmer, S. R., Bernstein, M., Abdulla, L., Burr, B., Robinson-Mosher, A., and Gee, J. Reflective Physical Prototyping Through Integrated Design, Test, and Analysis. *Proc. UIST '06*, 299–308.
16. Huang, J.-D. Kinerehab: A Kinect-based System for Physical Rehabilitation: A Pilot Study for Young Adults with Motor Disabilities. *Proc. ASSETS '11*, 319–320.
17. Kapur, A., Kapur, A., Virji-Babul, N., Tzanetakis, G., and Driessens, P. F. Gesture-Based Affective Computing on Motion Capture Data. *Proc. ACII'05*, 1–7.
18. Kela, J., Korpijää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L., and Marca, D. Accelerometer-based Gesture Control for a Design Environment. *Personal and Ubiquitous Computing*, 10(5), 2006: 285–299.
19. Ko, A. J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., Scaffidi, C., Lawrence, J., Lieberman, H., Myers, B., Rosson, M. B., Rothermel, G., Shaw, M., and Wiedenbeck, S. The State of the Art in End-user Software Engineering. *ACM Computing Surveys*, 43(3), 2011: 21:1–21:44.

20. Kurtenbach, G., and Hulteen, E. A. Gestures in Human-Computer Communication. In: *The Art of Human-Computer Interface Design*. Ed. by B. Laurel. Addison Wesley, 1990, 309–317.
21. Lim, Y.-K., Stolterman, E., and Tenenberg, J. The Anatomy of Prototypes: Prototypes As Filters, Prototypes As Manifestations of Design Ideas. *ACM Transactions on Computer-Human Interaction*, 15(2), 2008: 7:1–7:27.
22. Lindell, R. Crafting Interaction: The Epistemology of Modern Programming. *Personal and Ubiquitous Computing*, 18(3), 2014: 613–624.
23. Liu, J., Zhong, L., Wickramasuriya, J., and Vasudevan, V. uWave: Accelerometer-based Personalized Gesture Recognition and Its Applications. *Pervasive and Mobile Computing*, 5(6), 2009: 657–675.
24. Lozano-Quilis, J. A., Gil-Gómez, H., Gil-Gómez, J. A., Albiol-Pérez, S., Palacios, G., Fardoum, H. M., and Mashat, A. S. Virtual Reality System for Multiple Sclerosis Rehabilitation Using KINECT. *Proc. PervasiveHealth '13*, 366–369.
25. Lü, H., and Li, Y. Gesture Studio: Authoring Multi-touch Interactions Through Demonstration and Declaration. *Proc. CHI '13*, 257–266.
26. McConnell, S. *Code Complete*. Microsoft Press, 2009.
27. Moeslund, T. B., and Granum, E. A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding*, 81(3), 2001: 231–268.
28. Moeslund, T. B., Hilton, A., and Krüger, V. A Survey of Advances in Vision-based Human Motion Capture and Analysis. *Computer Vision and Image Understanding*, 104(2), 2006: 90–126.
29. Myers, B., Hudson, S. E., and Pausch, R. Past, Present, and Future of User Interface Software Tools. *ACM Transactions on Computer-Human Interaction*, 7(1), 2000: 3–28.
30. Nielsen, J. Noncommand User Interfaces. *Communications of the ACM*, 36(4), 1993: 83–99.
31. Norman, D. A. Cognitive Engineering. In: *User Centered System Design*. Ed. by D. A. Norman and S. W. Draper. CRC Press, 1996. Chap. 3, 31–61.
32. Norman, D. A. *The Design of Everyday Things*. Basic Books, 2002.
33. Schön, D. A. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, 1984.
34. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. Real-time Human Pose Recognition in Parts from Single Depth Images. *Proc. CVPR '11*, 1297–1304.
35. Shotton, J. Conditional Regression Forests for Human Pose Estimation. *Proc. CVPR '12*, 3394–3401.
36. Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., and Blake, A. Efficient Human Pose Estimation from Single Depth Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12), 2013: 2821–2840.
37. Simmons, S., McCrindle, R., Sperrin, M., and Smith, A. Prescription Software for Recovery and Rehabilitation Using Microsoft Kinect. *Proc. PervasiveHealth '13*, 323–326.
38. Tang, J. K. T., and Igarashi, T. CUBOD: A Customized Body Gesture Design Tool for End Users. *Proc. BCS-HCI '13*, 5:1–5:10.
39. Turk, M., and Robertson, G. Perceptual User Interfaces. *Communications of the ACM*, 43(3), 2000: 32–34.
40. Walter, R., Bailly, G., and Müller, J. StrikeAPose: Revealing Mid-air Gestures on Public Displays. *Proc. CHI '13*, 841–850.
41. Warren, K., Lo, J., Vadgama, V., and Girouard, A. Bending the Rules: Bend Gesture Classification for Flexible Displays. *Proc. CHI '13*, 607–610.
42. Wen, R., Nguyen, B. P., Chng, C.-B., and Chui, C.-K. In Situ Spatial AR Surgical Planning Using Projector-Kinect System. *Proc. SoICT '13*, 164–171.

43. Zhang, H., and Li, Y. GestKeyboard: Enabling Gesture-based Interaction on Ordinary Physical Keyboard. *Proc. CHI '14*, 1675–1684.
44. Zimmerman, J., Forlizzi, J., and Evenson, S. Research Through Design As a Method for Interaction Design Research in HCI. *Proc. CHI '07*, 493–502.

Appendix A

Attributions

- Figure 1.1 by James Pfaff / [CC BY-SA 3.0](#)
- Figure 1.2 by Sergey Galyonkin / [CC BY-SA 2.0](#)