Pre-print version
ARDITO C., COSTABILE M. F., H.-C. JETTER. (2014). Gestures that people can understand and use. *Journal of Visual Languages & Computing*, vol. 25(5), 572-576. doi: 10.1016/j.jvlc.2014.07.002

JVLC View point

Gestures that people can understand and use

Carmelo Ardito[1], Maria Francesca Costabile[1], Hans-Christian Jetter[2]
[1]Dipartimento di Informatica, Università di Bari Aldo Moro, via Orabona 4, 70125 Bari, Italy
[2]Intel ICRI Cities, University College London, Gower Street, London, UK
carmelo.ardito@uniba.it, maria.costabile@uniba.it, h.jetter@ucl.ac.uk

Introduction

In recent years, the form factors of computers and the ways in which we interact with them have undergone a dramatic change. Three decades after the widespread adoption of graphical user interfaces with mouse and keyboard, there are now more smartphones in the world than there are desktop PCs. Interaction by touch is becoming the norm, rather than an exception. The ongoing advances in sensor and display technologies, CPUs, and wireless networks are a continuous source of innovation with novel devices ranging from very large displays to small wearables such as smart watches or augmented reality glasses. All these new devices will keep pushing researchers to envision new interaction possibilities that extend or completely replace traditional mouse and keyboard input for non-desktop scenarios.

One particularly successful example are large interactive displays or whiteboards that rely on pen or touch input and can now be found in many meetings and classrooms or even in our public spaces. Already very early attempts like Liveboard recognized the need for interaction beyond mouse and keyboard [1], since interacting with these devices felt comparably primitive and cumbersome and only very simple applications could be implemented. Over time, size and resolution of these displays increased and they also became far more affordable and widespread, so that the focus of large display development is now on complex applications that meet users' real-world needs in various situations. Therefore, in an attempt to make interaction more "natural", new modalities and interaction languages without mouse and keyboard are studied to improve the interaction with these new systems.

A particular area of interest is the use of human gestures and body movements. For instance, the presence of a human body in the proximity of a display can be detected by using different sensors (e.g., cameras, microphones, pressure sensors, Bluetooth, RFID scanners) to let the system automatically react to the presence or movement of users nearby. Ballendat et al. propose such a system as an interactive home media player on a large vertical display in a living room [2]. The system adapts displayed content and interaction possibilities based on proxemic information, i.e. distance, orientation, movements, and identity of people in relation to an ecology of multiple devices and objects in their nearby environment.

Today, thanks to advances in computer vision that permit real-time body, hand, and finger tracking, it is also possible to recognize human motion from a distance. Users can communicate with a system by performing a *gesture*, which in human-computer communication is defined as "a motion of the body that contains information" [3]. Gestures are claimed to enable a more natural and intuitive communication between people and devices. Ideally users do not think in terms of handling an input device, but naturally use their body to execute tasks or make use of their skills for gestural communication with other humans. The big challenge here is: how to design gestures that people can effectively understand and use?

Hand gestures, in particular, have been studied for a long time. One of the first papers on the topic was published in this journal in 1994 by Bordegoni [4]. It described a system supporting hand gestures for interacting with 3D user interfaces, which also provided a visual programming environment for the design of gestural languages that consisted of a set of hand gestures with each one containing information (as in the definition in [3]). Since then, however, only a few papers dealing with gesture design and use for human-computer communication have been published in JVLC. This is surprising, since, as we point out in the next section, gestural languages are indeed visual languages. With this article, we therefore hope to stimulate the research community interested in languages for gestural interaction and gestural user interfaces (not only limited to large displays) to consider this journal as an appropriate venue for their research.

Learning from Visual Languages

Let us recall the definition of visual language (VL) provided by the JVLC Editors in the foreword of the first issue of this journal in 1990: “By *visual languages* we mean the systematic use of visual expressions to convey meaning” [5]. The focus was on formal visual languages, which were studied with the goal of easing computer programming as well as human-computer communication through the use of graphics, drawings or icons. But this definition is also appropriate for natural and less formal languages. Examples of such more natural visual languages are the many different sign languages that are used worldwide to enable communication among deaf people or people who cannot speak: they use hand and body gestures to convey meaning. Similarly, gestures used by a human to communicate with a computer (performed either by the whole human body or by a part of it, e.g. the hands) are “visual expressions to convey meaning”, i.e. visual languages.

We believe that for designing future visual-gestural languages for human-computer interaction we can learn from the past and capitalize on the great experience and lessons learned from three decades of VL work. 2014 marks thirty years from the first IEEE Workshop on VL that was held in Hiroshima, Japan, in 1984. That workshop stimulated the research on VL and started a series of workshops now held every year. In the mid-80s, the market availability of “high-resolution” graphical screens generated an enormous enthusiasm and the hope to greatly facilitate human-computer communication and programming by using VL. The use of graphics promised to enable visual interaction by manipulating visual representations of objects and a better support of our human skills for visual information processing. In other words, it promised to solve an important real-world problem of the time by using newly available interaction technology, much like gestural languages for novel devices are now expected to solve interaction problems of our time. More concretely, the question of how to design a universal “standard set” of gestures is now a recurring theme in books, blogs, workshops, or special interest groups (e.g., see [6-9]).

Back in the 80s, one of the challenges of the VL community was to create visual programming languages that could be general-purpose, like Fortran or C. Their aim was to make programming easier for non-technical people. Glinert et al. addressed this challenge in [10], discussing several open problems that need to be solved to make this possible, for example creating a sound “graphical vocabulary”, defining and validating metrics for assessing the relative merits of visual environments and programs, or developing scalable approaches. Other authors remarked the importance of finding new domains and various forms of visual languages where using graphics would be truly beneficial. Over the years, the idea of general-purpose visual programming languages demonstrated to be a failure. Visual representations have several advantages, but also many disadvantages: they are inherently ambiguous and often hard to understand or can only be interpreted within a certain context. Highly abstract concepts are too complex to be expressed visually. For example, many attempts have been performed to visualize *recursion*, but they resulted in very complicated images, difficult to understand. However, there are also success stories: the research on visual languages to facilitate database querying from the 90s (see [11]) resulted in visual interfaces which are much more usable than SQL for laypeople and are currently adopted by DBMS. Several domain-specific languages have been developed, which proved to be successful in practical applications [12].

General-Purpose vs Context-specific Gestures

Can this experience inform on how we should approach designing gestural languages today? We believe it can. Like the VL community three decades ago, some researchers in HCI are now working on defining general-purpose gesture sets that are intended to be universally accepted by most people. To this aim, a specific meaning has to be assigned to each gesture. This results in *symbolic gestures*, i.e., a gesture becomes a symbol for communicating the particular meaning that is assigned to it [13, 14]. Again, the problem is how to ensure that the designer assigns a meaning to the visual expression that is readily understood by the user and the receiver of the message (either human or computer). Thereby gestures present interpretation difficulties pretty similar to those of other visual expressions used in VLs. In particular, depending on context, users and tasks the pictorial component of a visual expression can be understood in different ways. Conversely, the same meaning can be expressed with different visual representations. The relationships between pictorial component and meaning were deeply analyzed and formalized by VL researchers, with the aim of handling ambiguous interpretations of visual expressions (for example, see [15] and [16]). In order to design gestures that are easier to understand, a possibility is to assign meanings that exploit users' familiarity with gestures from everyday human-human communication and culture. Figure 1 shows two examples of *symbolic gestures*, namely "✓", "✗", for accepting and rejecting an item, respectively. These gestures are the result of finger strokes drawn on a physical surface. A symbolic gesture is also the "thumbs up" gesture to express agreement, which is shown in Figure 2. In this case, it is a mid-air gesture, i.e. a gesture performed in the 3D space by a user standing in front of a display. We have to be aware of gesture dependence from users' cultural and social context. For example, the "thumbs up" gesture is considered an insult in some cultures.

Figure 1. Examples of symbolic gestures. The arrows in (a) and (b) show possible ways to draw the gestures through a finger movement on a physical surface.

Figure 2. Example of symbolic mid-air gesture, i.e. a gesture performed in air by a user standing in front of a display.

The failure of general-purpose visual languages in the past should warn researchers about working towards a future general-purpose gestural language. Instead, researchers should learn the lesson that VL researchers learned before them and consider using gestures only in specific contexts and domains and only for specific tasks, so that users easily understand the meaning that is assigned to a gesture. Even if this is the case, the users' effort for learning gestures still remains an important issue to consider. Gestural interfaces are not necessarily self-revealing, forcing the user to discover and learn the set of available gestures and how to perform them. For applications in a specialized context and when tasks are performed frequently, it might pay off for users to learn a particular set of gestures. The familiarity with the domain might also help in remembering and using a gesture once it has been learnt. However, in less specific contexts with everyday applications and diverse users, we have to be aware that users will not be happy about a device that requires them to learn specific gestures before they can use it.

The above discussion leads us to conclude that enabling people to use supposedly "natural" everyday gestures does not necessarily result in more efficient and effective interaction. This was also pointed out by Norman [17], who critiques the naturalness of gestural interfaces in terms of their claimed intuitiveness, usability, learnability and ergonomics. But when is a system easy to use? Difficult things become easy once users feel that they are in control, they know what to do and when to do it, what to expect from the system every time they perform an action. Users' understanding is facilitated when the system is capable of revealing a clear conceptual model of itself. Such a model should be based on elements that provide a large number of affordances and are homogenously connected to each other. Thus, gestures should be tailored to the specific tasks to be performed and a good system representation should be provided.

Gestures vs Manipulations

From our experience, a good system representation and a clear conceptual model are best provided to the users in the form of a simple, yet powerful, visual "model-world" interface. Thereby the rules that govern this model-world should be based on our commonly shared spatial and physical experiences. For example, almost every user has experienced how it feels to use their hands to move and arrange physical objects on a desk. Shouldn't we then try to exploit this fact when designing gestural user interfaces? Shouldn't we enable users to directly manipulate, arrange, and pile the objects on the screen rather than introducing artificial gestural languages that require users to first learn *symbolic gestures* and then to execute them to indirectly tell the system what to do?

The famous "pinch-to-zoom" or "two-finger-rotate" gestures (see Figure 3), probably the most common multi-touch gestures in user interfaces today, are great examples of *manipulations* that exploit users' familiarity with natural spatial and physical principles. Even if there is no real-world material that stretches and shrinks like the simulated rubber sheet during a "pinch-to-zoom", their spatial and physical characteristics are understood by users within a few seconds. If successfully applied, such an approach can shift the focus of gestural interaction from learning and executing *symbolic gestures* to an almost effortless use of direct natural *manipulations* of the objects on the screen. Therefore we believe that the key to better gestural languages are such *manipulations*. Our claim resonates with Mauney et al.'s study of user agreement on user-elicited gestures. They report there was a "clear trend towards higher agreement scores on actions that could be performed through direct manipulation and lower agreement scores on actions that were symbolic in nature" [18].

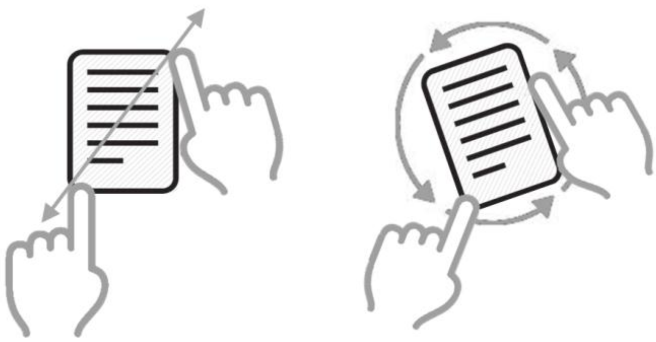

Figure 3. Examples of manipulations to zoom (a) or rotate (b) an object visualized on a display.

Manipulations work because they draw on the way the everyday physical and spatial world works or, perhaps more accurately, the ways we experience the everyday physical and spatial world. Instead of relying on users' familiarity with human-human gestural communication or elaborate real-world metaphors on the screen, they are based on our most fundamental sensorimotor and spatial experiences. As humans we all share and memorize such experiences, e.g., as image schemas [19] or naïve physics [20], since our early childhood because of the similarities of our bodies, our senses, and our early upbringing in an environment that was governed by the same natural physical and spatial laws [21]. As discussed in [13], introducing large numbers of *symbolic gestures* instead of focusing on *manipulations* is similar to expecting users to learn a great number of keyboard shortcuts or command line expressions or an entirely new language. Thus, if only used for *symbolic gestures* and not for *manipulations*, gestural user interfaces can actually become indirect and pseudo-natural, thus meaning a step backward into the era of pre-graphical user interfaces.

Conclusion

In the light of these considerations, we wish to summarize the two key points we addressed. First of all, the failure of general-purpose visual languages in the past should warn gesture researchers about working towards a future general-purpose gestural language. Researchers aiming at the design of a universal "standard set" of gestures should take into account the inherent ambiguity of visual expressions (like gestures) and the difficulty of expressing abstract concepts through symbolic gestures. Rather than trying to create a general-purpose gestural language, they should focus their design on specific domains and contexts. Secondly, interaction designers should always consider using *manipulations* and onscreen physical and spatial model-worlds before resorting to elaborate symbolic gestural languages. Unlike culturally-laden or context-specific symbolic gestures, manipulations are a part of a physical and spatial "mother tongue" that we all share. Manipulations do not require us to learn and remember new gestural symbols like a non-native second or third language and are thus the key to gestures that people can understand and use.

Acknowledgements

This work is partially supported by the Italian Ministry of University and Research (MIUR) under grant PON 02_00563_3470993 "VINCENTE" and by the Italian Ministry of Economic Development (MISE) under grant PON Industria 2015 MI01_00294 "LOGIN".

References

[1] S. Elrod, R. Bruce, R. Gold, D. Goldberg, F. Halasz, W. Janssen, D. Lee, K. McCall, E. Pedersen, K. Pier, J. Tang, B. Welch, Liveboard: A Large Interactive Display Supporting Group Meetings, Presentations, and Remote Collaboration, in proc. of: SIGCHI Conference on Human Factors in Computing Systems, ACM, Monterey, California, USA, 1992, pp. 599-607.

[2] T. Ballendat, N. Marquardt, S. Greenberg, Proxemic Interaction: Designing for a Proximity and Orientation-Aware Environment, in proc. of: ACM International Conference on Interactive Tabletops and Surfaces, ACM, Saarbrücken, Germany, 2010, pp. 121-130.

[3] G. Kurtenbach, E.A. Hulteen, Gestures in Human-Computer Communication, Addison-Wesley Publishing Co, 1990.

[4] M. Bordegoni, An Environment for the Specification and Recognition of Dynamic Gestures, Journal of Visual Languages & Computing, 5 (1994) 205-225.

[5] S.-K. Chang, S. Levialdi, Foreword, Journal of Visual Languages & Computing, 1 (1990) 311.

[6] D. Saffer, Designing Gestural Interfaces: Touchscreens and Interactive Devices, O'Reilly Media, Inc., 2008.

[7] D. Saffer, A Call to Arms for Interaction Designers, 2007, http://www.adaptivepath.com/ideas/a-call-to-arms-for-interaction-designers/, accessed on March 31st, 2014.

[8] M.L. Maher, T. Clausner, B. Tversky, D. Kirsh, J. Kay, A. Danielescu, K. Grace, Gesture-Based Interaction Design: Communication and Cognition, Workshop at Chi '14, 2014, http://hci.uncc.edu/~mmaher9/CHI-gesture-interaction/, accessed on March 31st, 2014.

[9] D. Wigdor, G. Morrison, Special Interest Group: Designing User Interfaces for Multi-Touch and Surface-Gesture Devices, CHI '10 Extended Abstracts on Human Factors in Computing Systems, ACM, Atlanta, Georgia, USA, 2010, pp. 3193-3196.

[10] E.P. Glinert, M.E. Kopache, D.W. McIntyre, Exploring the General-Purpose Visual Alternative, Journal of Visual Languages & Computing, 1 (1990) 3-39.

[11] T. Catarci, M.F. Costabile, S. Levialdi, C. Batini, Visual Query Systems for Databases: A Survey, Journal of Visual Languages & Computing, 8 (1997) 215-260.

[12] M. Fowler, Domain Specific Languages, Addison-Wesley Professional, 2010.

[13] H.-C. Jetter, J. Gerken, H. Reiterer, Natural User Interfaces: Why We Need Better Model-Worlds, Not Better Gestures, presented at CHI '10 Workshop on Natural User Interfaces, 2010.

[14] R. George, Ocgm (Pronounced Occam['S Razor]) Is the Replacement for Wimp. , 2009, http://blog.rongeorge.com/, accessed on March 31st, 2014.

[15] C. Shi-Kuo, Visual Languages: A Tutorial and Survey, IEEE Software, 4 (1987) 29-39.

[16] P. Bottoni, M.F. Costabile, S. Levialdi, P. Mussio, Formalising Visual Languages, in proc. of: IEEE International Symposium on Visual Languages, 1995, pp. 45-52.

[17] D.A. Norman, Natural User Interfaces Are Not Natural, interactions, 17 (2010) 6-10.

[18] D. Mauney, J. Howarth, A. Wirtanen, M. Capra, Cultural Similarities and Differences in User-Defined Gestures for Touchscreen User Interfaces, CHI '10 Extended Abstracts on Human Factors in Computing Systems, ACM, Atlanta, Georgia, USA, 2010, pp. 4015-4020.

[19] J. Hurtienne, J.H. Israel, Image Schemas and Their Metaphorical Extensions: Intuitive Patterns for Tangible Interaction, in proc. of: International Conference on Tangible and Embedded Interaction, ACM, Baton Rouge, Louisiana, 2007, pp. 127-134.

[20] R.J.K. Jacob, A. Girouard, L.M. Hirshfield, M.S. Horn, O. Shaer, E.T. Solovey, J. Zigelbaum, Reality-Based Interaction: A Framework for Post-Wimp Interfaces, in proc. of: SIGCHI Conference on Human Factors in Computing Systems, ACM, Florence, Italy, 2008, pp. 201-210.

[21] H.-C. Jetter, H. Reiterer, F. Geyer, Blended Interaction: Understanding Natural Human–Computer Interaction in Post-Wimp Interactive Spaces, Pers Ubiquit Comput, (2013) 1-20.

Figure captions

Figure 1. Examples of symbolic gestures. The arrows in (a) and (b) show possible ways to draw the gestures through a finger movement on a physical surface.

Figure 2. Example of symbolic mid-air gesture, i.e. a gesture performed in air by a user standing in front of a display.

Figure 3. Examples of manipulations to zoom (a) or rotate (b) an object visualized on a display.