# Parallel world - gestures: nice to use, but difficult to learn

Rebecca Linke

**Abstract**— Gestures on an interactive surface are mostly hard to discover and therefore difficult to learn. That is the major drawback which should be resolved for allowing gesture-based input to develop into an easy and widely used input-technique. A well known gesture is in average faster, than moving to a small button. Gestures do not consume place of an interface and you can apply them on the whole surface. Nevertheless this is exactly the point, where the problem starts. As it is pointed out in the following, the main problem in using gesture-based input is this kind of parallel world. As they are not part of the graphical-interface and most of the time, especially in expert-mode invisible, they are very hard to discover. Overall in most cases gestures are not self-explanatory. That is why users are deterred from using an apparently not intuitive interface. To represent the opposite point of view many interaction-techniques have been explored to give feedback, to support learning and training of gestures, and to make the parallel world of gestures - the level beneath the visible interface - evident.

**Index Terms**—guidance, self revealing, parallel world, intuitive

## 1 Introduction

We are living in a world where the usage of computers and other technical input devices like tablets or smartphones is indispensable. Nevertheless there are people of the older generation who are not familiar with making use of all these technical devices. Furthermore trends indicate a change to a society with even more old people and in the paper 'Aging, Motor Control, and the Performance of Computer Mouse Tasks' from John W. Smith [12] it is proven, that they have difficulties to access fields and buttons with a mouse using the double-click function. It is even worse when older generations are not used to mouse-based input at all.

Why not changing the way of adressing interfaces to a more familiar and intuitive way? We have to specify an alternative form to communicate with computers in an easy way. We do not even have to touch a device anymore. Furthermore it is a very natural way to communicate - sign language demonstrates this effectively. Why do we force users to use an artificial pointing device, although we have perfect pointers always around: our arms and hands. On top of this gesture-based input is very effective. For example if you want to execute a command like 'save' without shortcut, you have to navigate through the main-menu. And this action costs time: Imagine for example a 'S'-like gesture applied on the whole screen. With regard to Fitt's law, this simple movement should be faster than navigating through a small linear menu. Using gesture-based input offers an immense improve of speed and effectiveness for expert users.

And thats's where the main problem of gesture-based input is hidden: it's not self-revealing. If you are not an expert and have never used gestures for operation of an application, it costs you a lot of training to rise from novice to expert mode. Your hands may be a natural pointing device but gestures do not provide a visible path, where you can follow your interaction. On the contrary the set of gestures is like a parallel world beneath the main graphical interface. In addition only an expert user can navigate there fast and benefitts the most from the intuitive way of communication. In this paper the different approaches and improvements of interaction techniques for single- and multi-touch-gestures are presented. They range from marking menues to the new idea of crosslearning and cover aspects from revelation of gestures, motivation, training areas and dynamic guides. All together, their overall aim is to make the parallel world of gestures visible.

• *Rebecca Linke is studying Media Informatics at the University of Munich, Germany, E-mail: R.Linke@campus.lmu.de*

• *This research paper was written for the Media Informatics Proseminar on "Interactive Surfaces", 2015*

## 2 Learning by watch and imitate

How could we break through this parallel world making gestures easily to use for the majority? Gestures are completely different from WIMP-interfaces, as they do not include buttons, menues or other on-screen-fields. That's why they are not immediately discoverable. When applying gestures in the wild they have to be learned - and this can be a very difficult process, like you can see it with the famous 'zoom-to-pinch' gesture [7]. At the beginning of its publication on smartphones almost nobody knew, how to use it or even worse, that the possibility of zooming with a gesture exists. Of course there's is no button anymore, which tells the user: "'press here to zoom". But the usage spread by a "'watch and imitate"' propaganda. The main Problem here is the revelation of gestures. As the paper "'StrikeAPose"' [11] states, people do not recognize by themselves, the possibility of using gestures for input. They need a clear starting-point and besides a strategy like spatial devision, which shows the gesture, to start.

Nevertheless, if we want to introduce a wider set of gestures of at least eight different motions, we have to accelerate the process of teaching the user - 'how to communicate with gestures'. That is the point where we should start to make the invisible world of gestures apparent and bring them closer to the user.

## 3 Rehearsal of gestures

### 3.1 Looking up for gestures in a crip sheet

A first approach of Kurtenbach [8] was to provide cheat sheets in an overlay. If the novice user wants to know how to carry out a gesture, he is able to look it up there and this enables a user to practice gesture based input. Yet it has several drawbacks: As it is an overlay you can not see the guide and perform the gesture at the same time. Moreover mixing up some gestures in an important project is definitely not worth striving for. Furthermore it's not too effective to look up every single gesture and therefore it belittles the advantage of gesture-based input. Besides long-term-recognition is hardly supported. All together lots of training, crib sheets, videos or tutorials, which are not directly included to the programm, will cause the same drawbacks. They create a barrier of entry, because users expect a programm to be intutive enough.

### 3.2 Practise area and demos in application

That is the reason why it is necessary to look for alternative aproches to make the level of gestures visible and understandable. In the application 'GestureBar' [2] a training free approach to disclosing and teaching getures, a first attempt is made: There are practice areas for gestures with animated demonstrations directly included in the interface of a diagramming application. This limits errors caused by looking up the gesture in an extra crib sheet. Additionally the user can start without precedent training. It outperforms a 'state of the art crib sheet'. But nevertheless a novice-user still has to spend a lot of time in the training section, before he can apply the gesture securly on the project. This may distract possible users. Obviously, the gestures are experiencable in the training area, but not while using. In addition a novice user may not be willing to spend lot of time learning new gestures, if he has the choice to skip the training.

### 3.3 Motivating Gesture Learning: Gesture Play

It is a similiar approach like the one defined in [2]. But the concept of 'Gesture Play', which is described in [1] wants the user not to be afraid of learning new gestures. The aim is, that users always will choose gesture-based input, even if they have the possibility to use a WIMP-interface, too. To make the process of learning gestures funny and motivating, it is the next step to bring the remote level of invisible gestures and a novice user as far together, that he can easily operate a device with gestures. The idea of 'Gesture Play' is to make gesture learning fun and enjoyable. Therefore they use little puzzles with animated spring widgets, physical props, and simple button widgets. The system works without upstream explanations. However the aim is to let the user discover the gestures on his own and if he succeeds he is rewarded with trophies. To make the puzzles more intuitive the components are reacting physically correct and follow the users movement. That's why the learning curve is relatively high. Main aim of gesture play is to be intuitive and motivating. Without stress or annoying training the gestures are learned and now understandable for the user. Nevertheless we still need an extra training-interface and time to learn.

## 4 Leave the training-area with dynamic guides

### 4.1 Smooth transition from novice to expert mode: Marking Menus

One step further it is time to get rid of the training interface and to the idea of improving menus by being faster and more efficient to use. This culminates in hierachical marking menus, which allow users to select a menu-item by carrying out a single-stroke-gesture. For example [6] points out, how you can improve the standard linear-menu and replace it through a so called pie-menu. When using linear menus, users have to scroll through a long list of items. If you are not exact enough, the menu closes, even if that was not the intention. Furthermore you have to search through the full list, if the item to select is not the first one. It's shown that pie menus, are significantly faster to use.

Kurtenbachs [8] marking-menus extend pie-menus. Basically you still start in the middle of the pie. Only when the user is in novice-mode and hesitates too long, the options are shown circular around the pointed spot. A simple marking-menu stops here, but hierachical marking menus describe nested lists as well. That's why after moving to the desired point you can navigate with the press-and-hold gesture for novice-mode into the next step. The extraordinary archievement is, that unconscious and step-by-step users are taught the single-stroke gestures for the menu-items. And if in the end the expert-user-mode is reached, the gesture can be executed very quickly. In [9] Marking Menus are said to be three times faster, than ordinary pull-down menus. And a simple press-and-hold gesture makes the logic laying beneath visible.

### 4.2 Continously feedback: dynamic menus

Nevertheless with the setting of a pie-menu, marking-menus are very limited in their number of gestures, because there can not be presented to many menu-items at one level. Furthermore the exact path of the gesture is difficult to detect. Besides, there is no feedback provided for the path between the different levels of hierachy. A novice user is still relatively slow and leaks information at some points. Thats is why dynamic guides like OctoPocus [9] or 'Apèrge' [4] provide continious feedback and the same amount of feedforward information. The advantage here is, that for example 'Apèrge' uses chords with multiple fingers, which is similar to playing the piano. And therefore it is more comprehensive for users. In comparison 'OctoPocus' can be adopted to almost every single-stroke gesture-set. Like in marking-menus a press-and-hold gesture evokes novice mode and all possible gestures are shown. If the user chooses one direction, the opacity of the other paths become weaker until they disapear. Finally the user is on the right way. Over all, the user is provided with lasting feedback at any stage of executing the gesture. As the risk to make errors is very low, even novice-users can proceed very fast.

Another dynamic guide called "ShadowGuide" [3] exceeds this concept, as it is designed for multi-touch gestures. It is an on-demand assistance, which helps users to get to know the varity of initial contact postures and contacts throughout the performance. Therefore "ShadowGuides" consist of two parts: a "registration pose guide", which is a pop-up panel with informations about possible registration poses and a "user shadow annotations" to guide the user from his current hand pose to the complete gesture. Therefore the shadow of the hand is shown and help is provided by arrows, shape deformation keyframes and dynamic markers.

## 5 Getting more intuitive

### 5.1 Create mental links

With dynamic guides we are at a point, where we can facilitate the revelation and physical training. But still the gesture itself is like a vocabulary to learn without any aide-memoire. That means a very weak performance for long-time recognition. The gestures itself are easier to learn but still remain in the "‘parallel-world" beneath the user-interface, because the logical link is not easy to detect and therefore gestures are forgotten very fast, without regular training. With MIME a mid-air gesture-set [5] developers try to merge the two levels of user-interaction togehter. In this context it's shown, that gestures which have an association to a certain icon in the userinterface are the easiest to learn and to remember. In comparison textual links and arbitrary connections between gesture and command have a significantly weaker performance.

### 5.2 Crosslearning - create own mental links

In the end the last step of making gestures easily appliable, is to combine the advantages of a dynamic guide and a gesture which provides an contextual-link for users. In [10], a paper about user- and system cross-learning, it is described, what could be the next steps into a gesture-based interface. Users are wanted to understand gestures from the beginning on, because they have defined the set themselves. Therefore it is necessary, that the system and the recognizer learn the gestures at the same time. After a few repetitions, which is training for both the user and the system, the new input-set can be applied on the interface. In this case, it was tested on a simple picture editor. Firstly it was proven, that self-defined gestures are easier to remember then using a cheat sheet. However secondly even customizable gesture menus brought a convincing result. In this case the advantages of continious feedback and a contextual link are combined to a concept, that breaks the gap between graphical interface and "‘world of gestures"'.

A common interface, which reacts to click-events or simple tap-events is normally very easy to use, because the user is told what to do on every single button. For example an arrow which points to the left will redirect to the previous page. If users don't have the buttons or a help text, which tell them the command they will trigger, then self-defined gestures are a good alternative.

## 6 Conclusion

Gesture-based input is a very effective way of adressing an interface-item or to execute a certain command. Once one has reached the expert mode the gesture-based input system will be one of the best methods competing at the moment. Nevertheless Users expect interfaces to be usable with little or any training. Even older people should be enabled to learn the gestures quickly - so that they won't have problems to perform the task because of too slow motor control. There are lots of interaction techniques and in the end an usable interface needs a little bit of all of them to make the layer of gestures visible and understandable. In conclusion designers of an interface should be encouraged to try crosslearning approach: where users have a cheat sheet, dynamic guides and self-defined gestures, which can be rembered for a longer time.

## REFERENCES


[1] A. Bragdon, A. Uguray, D. Wigdor, S. Anagnostopoulos, R. Zeleznik, and R. Feman. Gesture play: Motivating online gesture learning with fun, positive reinforcement and physical metaphors. In *Proceedings of the ACM International Symposium on Interactive Tabletops and Surfaces 2010 (ITS 2010)*, pages xx–xx, New York, NY, USA, 2010. ACM.

[2] A. Bragdon, R. Zeleznik, B. Williamson, T. Miller, and J. LaViola. Gesturebar: A training-free approach to disclosing and teaching gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*, pages 2269–2278. ACM, 2009.

[3] M. R. M. Dustin Freeman, Hrvoje Benko and D. Wigdor. Shadowguides: visiualizations for in-situ learning of multi-touch and whole-hand gestures. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)*, pages 165–172, 2009.

[4] O. B. Emilien Ghomi, Stéphane Huot, M. Beaudouin-Lafon, and W. E.Mackay. Arpège: Learning multitouch chord gestures vocabularies. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces(ITS '13)*, pages 209–218. ACM, 2013.

[5] S. Ismair, J. Wagner, T. Selker, and A. Butz. Mime: Teaching mid-air pose-command mappings. In *Proceedings of the 17th international conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '15, pages xx–xx, New York, NY, USA, 2015. ACM.

[6] M. W. Jack Callahan, Don Hopkins and B. Shneiderman. An empirical comparison of pie vs. linear menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems(CHI '88)*, pages 95–100. ACM, 1988.

[7] D. V. Jeff Avery, Mark Chor and E. Lank. Pinch-to-zoom-plus: An enhanced pinch-to-zoom that reduces clutching and panning. In *Proceedings of the 27th annual ACM symposium on User interface software and technology(UIST '14)*, pages 595–604. ACM, 2014.

[8] G. Kurtenbach. The design and evaluation of marking menus. In *PhD thesis*, pages 1099–1108, 1993.

[9] W. E. Olivier Bau. Octopocus: A dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST '08)*, pages 37–46. ACM, 2008.

[10] E. A. PeiYu Li, Manuel Bouillon and G. Richard. User and system cross-learning of gesture commands on pen-based devices. In *Human-Computer Interaction – INTERACT 2013, 14th IFIP TC 13 International Conference, Cape Town, South Africa, September 2-6, 2013, Proceedings, Part II*, pages 337–355. Springer Berlin Heidelberg, 2013.

[11] G. B. Robert Walter and J. Müller. Strikeapose: Revealing mid-air gestures on public displays. In *Proceedings of the SIGCHI Conference of Human Factors in Computer Systems (CHI '13)*, pages 841–850. ACM, 2013.

[12] J. W. Smith, J. Sharit, and S. J. Czaja. Aging, motor control, and the performance of computer mouse tasks. In *Human Factors: The Journal of the Human Factors and Ergonomics*, pages 389–396. 2013 Journal Citation Reports® (Thomson Reuters, 2014), 1999.