

Tactile Interaction for Novice User

Uncolocated Gestures

Denis Chêne^{1(✉)}, Vincent Pillot², and Marc-Éric Bobillier Chaumon²

¹ Orange Labs, 28 Ch. du Vieux Chêne, 38243 Meylan, France
denis.chene@orange.com

² GRePS (EA 4163), 5 av. P. Mendès-France, 69676 Bron, France
vincent@pacmail.fr, marc-eric.bobillier-chaumon@univ-lyon2.fr

Abstract. This paper introduces the concept of tactile interaction for novice elderly users. Cognitive difficulties, motor constraints, visual overloads and lacks of feedback lead to hardly usable tactile smartphone among elderly users. An optimized tactile interface was produced, offering continuous and secure gestures, and introducing “uncolocated gestures”. Comparative tests to a classic tactile interface show that those gestures solve interaction problems but generate other difficulties. Uncollocation is a solution of interest but has to be learned and has to be progressively acquired through activity. A final enhanced profile for Elderly users was set and solves this situation, enabling uncollocation manipulation for Back and Up-Down commands and preventing it for Validation command, until it is totally acquired by the user.

Keywords: Handheld devices for the elderly · Use and design of smart phones for the elderly · Novice profile · MenuDFA · Tactile interaction

1 Introduction

Elderly users are often novices in smart phones use and they face touchscreen use as a hard experiment [1]. One may talk of illectronism, a kind of numerical world illiteracy. In fact, taps, multi-finger taps, repeated taps, slides, swipes, multi-touch gestures and others abstract interface objects are certainly many sources of confusion [2, 3]. Indeed, to interact tactilely on a touchscreen is not so easy and it requests good visual-motor abilities. Due to this, hence elderly user and other novices are left out. Firstly, this paper analyzes why tactile interaction generates troubles. Secondly an alternative tactile interaction mechanism is described. Then a user comparative user test between a classical tactile interface and an optimized tactile interface for novice user is described. Finally results and limitations of the innovative interaction method are discussed and solved.

2 Limitations of Classical Tactile Interface

Standard interaction method pushes novice user (and also expert user) to do numerous types of errors. Indeed many inadvertent actions are done by novice use. As the “tap”

gesture required by standard touchscreen devices is short delayed it leads to many missed validations and missed backward navigation. This is direct consequence of merging selection and validation phases in a single operation. Through a personal computer interaction mediated with a mouse, it is not really a problem as user can point out, move his mouse over an item (pointing phase), and then clicks it (selection phase) or double-clicks it (selection-validation phase). During web browsing through hyperlinks the double click is useless as in this case selection-validation phase is achieved through a single click. But the user can still move his mouse over the item before acting, in order to obtain some information about the pointed item.

Tactile Device. On the opposite on a tactile device, the user loses this opportunity to point over the item¹. In parallel he is losing as well the choice to have a single or a double click. Touching the screen is a trigger threshold. Or, on the contrary, novice user is in a situation of discovering and learning. He needs time to decide and clear information about available functions. How to be aware about new elements if they are directly triggered? He needs time to ask for information before execution, and even time during the command execution. Finally, on a classical tactile interface the user will have probably to get that information himself, mainly through errors and missed actions. He has to go there, in order to know where he goes. This lack of division between information, selection and validation phases leads to an interaction style that could be named “blind arbitrarily exploration”.

Moreover, the gesture itself of “doing a tap” on a screen is not so easy to operate for a novice user. It needs some cognitive, motor and visual accuracy. Novice users start from real buttons metaphor and apply it to tactile buttons. Real buttons are much easier to operate as they offer many feedbacks (pressure, abutment). But tactile buttons are refrained from giving any details and finally mislead user. Indeed, despite of their “real button” affordance, that should lead to push them, they are not push-able but push-off-able. Main action is triggered after the tap-off phase, not before. But second level action (as Edit mode or Contextual menu) is triggered after a long duration tap-on phase. Then novice users are frequently doing a long press as they will do for a “real button”, but are faced with a non-expected second level action. Tap nature (on or off) and tap duration are not the only trap. Lack of information about requisite pressure leads novice user to press too hard, generating side effect of a wider contact zone, and pointing errors.

Pointing errors have many other causes. Ergonomics real buttons are often curved and have a concave shape in order to help the finger to fit in and to feel the limits. That is not the case on a tactile interface where, in addition, items are smaller. To slip even slightly is not recommended. Moreover on a tactile phone the hand hide a part of the pointed item, which further accentuates pointing errors. Moreover, tactile virtual buttons at the bottom of the phone are often invisible and are erroneously activated. Consequently, standard tactile interface overstretch cognitive, motor and visual user capacities.

User Capacities. If that is not bad enough, user capacities themselves may be altered. It is often the case with elderly users [3]. Finger shaking, lack of finger sensibility, low

¹ Some recent new technologies will offer to detect that the finger is just over the screen, without touching it. It may be an outcome.

pointing capacities, or finger electrical conductivity, visual impairment, and audio disabilities are minor and major disabilities that often goes hand in hand with elderly. Largely speaking short and localized gestures should be avoided in case of accessibility needs [4].

Situation of Use. Situation of use has also to be taken into account. Indeed even if novice users are often trying to use their mobile phone with two hands, they are often glad to do it with a single hand [5] as mobility is related to multitask activities. By example user will make a call and will walk at the same time, sharing is attention between the street interactional path and the virtual tactile one [6]. Daily, users will also hold something in one hand and have to answer a call with the other one. When user is sited, his mobile phone use is totally different from a standing use [7].

Such contextual use leads generally to one-handed use [8]. In such situation, interaction is achieved through thumb activity and this has consequences on comfort of use [9] and on performance [10]. Indeed, comfort of use may be severely altered. As an example dramatic increase of thumb interaction may multiply the occurrence of related musculo-skeletal disorders [11]. Indeed a pointing interface where interaction items to be pointed out are spread all over the screen is not really adapted to thumb limitations.

Divided in 12 parts (Fig. 1) the screen shows structural deficiencies for tap activity through the thumb. Only 5 and 8 zones seem accurately tap-able [12]. But researchers explored other kinds of gestures: horizontal and vertical swipes. 3, 5, 6, 7, & 8 zones seem accurately swipe-able [13]. It has to be noted that this is true for novice users as well as for expert users.

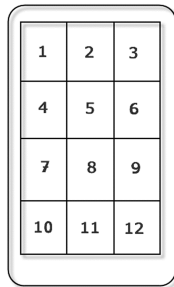


Fig. 1. Partition in 12 zones of the smartphone screen [10]

Classical tactile interface displays tiny items all over the screen, and use short time-lapse gestures, that are also strongly localized and that finally lead to many execution and pointing constraints. On the contrary vertical and horizontal swipe may be more easily executed, through the thumb with one hand, or through other fingers in case of a two-handed use. Metatla et al. even recommend using a “no direct pointing” interface [14]. It means to be able to act without being dependent from the item and the finger collocation (unco-location command). Moreover, such continuous gestures are not generating any trouble in case of disability user constraints [3], provided that they are not too long [15].

3 An Optimized Tactile Interface for Novice Users

What could be an optimized tactile interface for novice user? First, classical tactile interface need to be optimized in terms of cognition. Few interface objects are easier to understand than many. Vertical list component is a good candidate to better understanding as it is quite the only component that can be a substitute to every kind of interface objects.

It needs also to be optimized in terms of available basic commands. Information phase need to be available and usable before command execution. If Selection and Validation phases are dissociated one from each other, hence, after Selection phase achievement, user has the choice to choose between Information or Validation phase.

Such an optimized interface needs as well to be optimized in terms of gesture execution through continuous gestures (as drag and swipes gestures), through uncolocated items manipulations or at least few pointing constraints, and with sufficient time delays.

Taking into account all those arguments leads to define a tactile interface that is managed only through cascading lists components, with a user-controlled focus, and a secure tactile manipulation made of long presses and swipes. Such a tactile interface optimized for novice elderly users (N interface) was designed based on the MenuDfA component [16]. It applies to user interfaces whose functions (e.g. list items, menu options, settings) can be organized in a hierarchical way (e.g. top levels lists and sub lists). As there is no “average” user it offers several interaction profiles optimized to each kind of user. Beyond the “Design for All” principle, MenuDfA technique provides a mean for adjusting finely each interaction profile, and more specifically gestures that can be executed through continuous focus handling or in a discrete manner (shortcut gesture, as a rapid swipe). It organizes the various items of the intended application in the form of a set of hierarchical lists where the items of each list are displayed vertically. It provides a sequential access to the interface elements, as well as a direct access. In both cases selection and validation may be set to two separated operations. Interaction mode can be set to pointless or pointed at, and in this case, uncolocated or colocated actions can be chosen. Thus it covers a large diversity of user needs and situations.

A selection focus enables user to navigate the entire application just by moving the focus in the four directions. Vertical displacements are used for moving through the items of the current list, and horizontal displacements for moving through the hierarchy of lists. A slide towards Up moves the focus up, to the previous item; a slide towards Down moves the focus down to the next item; a slide towards Right validates the current item and goes down into list hierarchy; and a slide toward Left goes back and up in the hierarchy. In such a context, absolute positioning isn’t mandatory. Indeed, all the gestures make use of relative positioning. It means that they can begin anywhere on the screen, the finger and the focus has no colocation constraint.

A dedicated profile to novice users was set. It is called hereafter “N interface”. Tactile interaction in the N interface was designed to answer to novice constraints. They need time to ask for information and to be secured thus selection time was separated from validation time. This is achieved through a replacement of tap-shot by continuous focus manipulation. They need time then standard tap was replaced by a long tap delay (800 ms). They need support to pointing inaccuracy then large items were designed and

inaccuracy pointing square was added. They need access to any screen zone through the thumb, then uncolocation of the finger and the aimed item was allowed. Available navigation commands of the N interface are the following: the user has to first move the focus to the aimed item, with a continuous vertical gesture (Up or Down) or a Short Press in case of direct pointing (less than 800 ms). But no validation occurs at this time. Only a vocalization of the current focused item is displayed to the user. The Up or Down gestures may be colocated (Fig. 2) or uncolocated (Fig. 3).

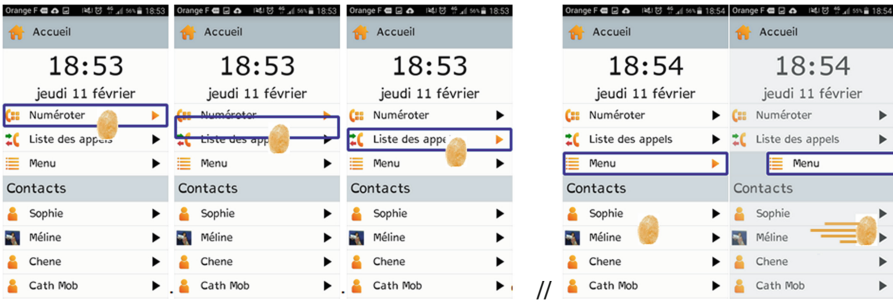


Fig. 2. Colocated continuous up-down focus manipulation (selection) and colocated right gesture (validation)

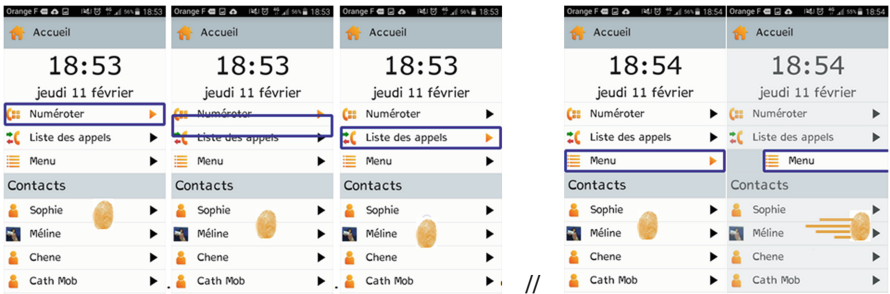


Fig. 3. Uncolocated continuous up-down focus manipulation (selection) and uncolocated Right gesture (validation)

When the short press is executed on a previously selected item, then it repeats vocalization of the focused item. It enables user to ask for information as many as he likes. Then he can validate the focused item with his choice of a continuous colocated (Fig. 2) or uncolocated (Fig. 3) gesture to the right. This validation can also be done with a colocated-only long press (Fig. 4).

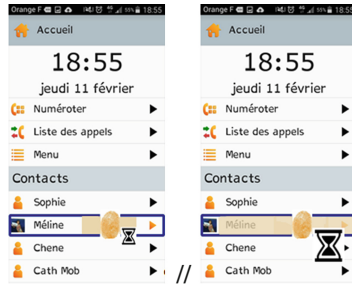


Fig. 4. Colocated Press (less than 800 ms) for selection and Hold (more than 800 ms) for validation

The back command is a continuous gesture to the left similar to the right one. It can be either colocated to the focused item or uncolocated. We will see that even if Left and Right gestures are similar, their inherent commands attribution (respectively Back and Validation commands) has a great impact on understanding and manipulation. Note that there is no need of a Back button, and consequently such Back button wasn't displayed at all on the N interface.

4 Comparative User Test

In order to perform a meaningful comparison of the tactile interaction modes, an equivalent classical tactile interface was also designed (C interface). On the C interface only the Short Press (Tap) is available to achieve validation, and back action (on the back button) in a unique selection/validation phase. It needs precise pointing and fine motor abilities and has a high potential for unintended mistakes. Back action is available on the screen with a back button on the top left part of the screen. C and N interfaces are designed with the same functionalities enabling making a phone call from number dialing, or from contact selection, or from past phone call list. Contact list is available and contacts are editable.

The only visual differences between those two interfaces (Fig. 5) is the existence in the N interface of a focus element (a rectangle), of some visual feedbacks associated to continuous manipulation, and the lack of any back button. This novice optimized tactile interface (N) was compared to classical tactile interface (C). Experimental plan was the following: $S_{10} < I_2 * T_5 >$. N and C interfaces (I factor) were both used one after another by 10 elderly touchscreen expert users and 10 elderly touchscreen novice users (S). Users were from 70 years old (expert average; $\sigma = 6,70$) to 74 years old (novice average; $\sigma = 7,63$). Novice users had never touched any touchscreen devices. Expert users were used to use a tactile mobile phone. 5 tasks (T) had to be made on each interface. First they had to find a recent call in the call history list. Second they had to find a radio station in the radio list hide in the sub-level of the Menu item. Third they had to dial a number and to call someone. Fourth they had to find a specific contact in the call history list and to add it to the contact list, but they had to do it in a one-handed situation. Fifth, they had to modify a contact in the contact list.

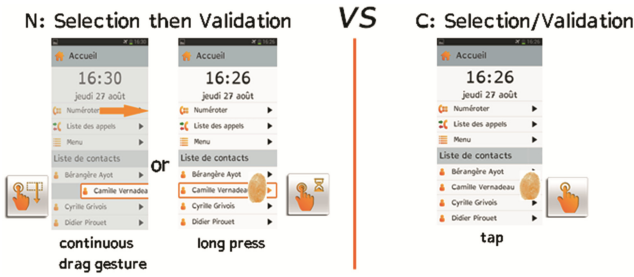


Fig. 5. N and C interfaces comparison

N and C interfaces order was counterbalanced. And before each of them some explanations about the tactile interaction mode was given to user. For the N interface the explanation was: «you have to move the focus in order to select the item you want. Then validate it doing a gesture to the right side». If the user had some difficulties to execute this pre-test, then it was added: «you can also do a long press to validate». For the C interface the explanation was: «you have to tap on the screen in order to validate the item you want». Time, task achievement, and errors were recorded on log files, and on videotapes. In the N condition 9 types of errors were coded. Type 1 is when the user wants to validate an item X, but forget to manipulate the focus that is still on item Y, and thus validates Y instead of X. Type 2 is when the user wants to validate an item X, points at it, but due to pointing inaccuracy during press, validates another close item. Type 3 is when the user tries to validate the selected item by a Right gesture, but switches to another one because its gesture starts too slowly and with inaccuracy. Actually its gesture has generated a Press & Hold on another item, leading to a missed validation. Type 4 is when the user tries a Tap in order to validate the current focused item. Type 5 is when the user operates a Tap or another gesture on the title zone of the screen that has no linked command. The user probably seeks for the Back button of the C interface. Type 7 is when the user wants to validate with a Right gesture and makes a Left one. Conversely for Type 8, the user wants to do a back with a Left gesture and makes a Right one. Type 9 is when the user wants to glide the focus over items but due to a too slow execution the item underneath finger is unfortunately selected.

In the C condition 10 types of errors were coded. Type 1 is about pointing inaccuracy: another item than the one aimed to is validated during a Tap. Type 2 is a missed vertical manipulation (up-in-the list with drag Down gesture) of the list that generates an unexpected validation. Type 3 is the same for down-in-the list (drag Up gesture). Type 4 is when the user does a wrong validation of an item during an attempt to go back with the back button softkey. Type 5 is the same in case of the use of the back button in the Title zone. Type 6 error is when the user tries to do a validation with a Right gesture that is totally inoperative in this C condition (persistence of the N interface procedure for part of the users that start with N condition). Type 7 is similar to the previous one, but for the Left gesture. Type 8 is when the user tries to navigate vertically in the list but is stopped by the list end or top abutment. Type 9 is when user tries to do a Tap on the title (inoperative). Type 10 is when the user tries to activate another non-actable item. Those errors where merged in more generic

categories in order to be compared. Pointing inaccuracies (N types 2, 3 and C types 1,2,3,4,5), focus adjustment difficulties (N types 9 and C types 2,3,8), interaction methods confusion (N types 4 and C types 6,7), non-actable items (N types 5,6 and C types 9, 10), specific N condition errors, types 7,8 about laterality confusion and type 1 missed validations due to uncolocation.

4.1 Results

A Wilcoxon signed ranks test shows that in N condition novice errors were as numbered as in C condition (novice N errors $M = 1.75$, $SD = 2.37$ and C errors $M = 1.78$, $SD = 2.34$) whatever was the task (Fig. 6). That was not expected as uncolocation and Selection-Validation dissociation were here to solve gesture inaccuracies and unintended mistakes in the N condition.

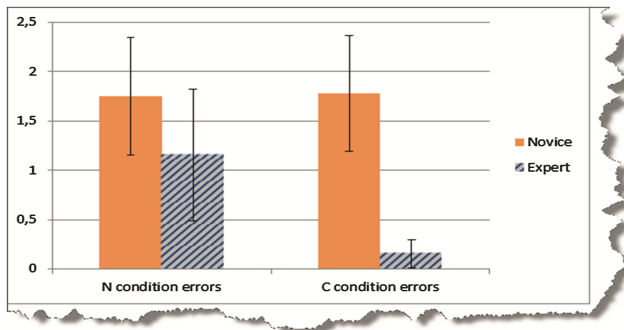


Fig. 6. Means (all) errors comparison between N and C condition for novice and expert users

And even expert users are doing more errors in N condition. That was expected as some gestures (up and down) are reversed compared to C condition but we can see in the details results that it's mainly due to other causes. Detailed results show that new kinds of error appear in N condition, and are mainly due to uncolocated actions, for novice and expert users. One main observation of this study is that uncolocated actions are very difficult to understand for both novice and expert user. Both are constantly stuck to direct pointing. This behavior was awaited for expert user of standard tactile pointing interface, as they are used to directly point at items, but it wasn't for novice users. But at the beginning of the task the "focus" element existence is not so obvious for them. They manipulate it clearly with vertical slides ($\uparrow\downarrow$), but in a located manner. They put precisely their finger on the focus and start to drag it to another item. They start to acquire that sliding vertically the focus may be done in an uncolocated way only when they are facing long lists and have to scroll them. At this point they understand slightly that they can move the focus without being stuck to it. But this possibility is not easily extended to validation phase. And many missed validations occur because the user was rapidly doing a right slide gesture directly on an item (that was not already focused). Doing so, they were validating another item without realizing that it was not the focused one. The Right gesture without colocation constraint (\rightarrow) is a trap for novice and expert users

until they have not acquired that the focus can be used in an uncolocated manner. On the contrary the Left uncolocated gesture \leftarrow doesn't generate error at all, as it is never dependent to location. Indeed it just goes back to the previous level. Starting from the focused item or not doesn't change anything in that case.

Details results show also that those errors due to Right gesture hide the effect of other interesting points of the N interface. If one centered on other expected errors than errors due to uncolocated Right gesture (*uRg*), then the N condition fulfills his goal. Indeed Mann-Whitney test shows, that expected errors (as gesture inaccuracies and located missed validation and pointing located errors) are significantly in a smaller amount ($z = 2.42, p < .01$) for novice in N condition (errors without *uRg* errors $M = 0.31, SD = 0.77$) than for novice in C condition (errors without *uRg* errors $M = 1.78, SD = 2.34$). In brief that is to say these usual errors due to short press gestures combining validation and selection (classical Tap) are mainly done on C prototype but not on N prototype, as expected (Fig. 7).

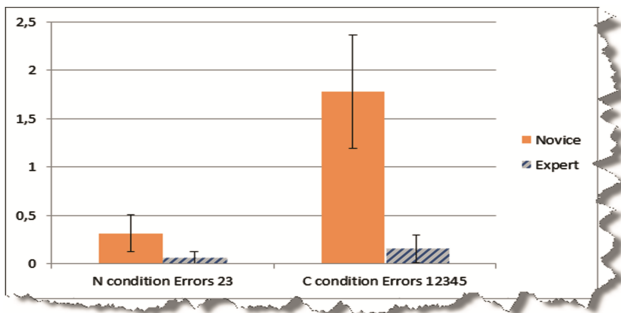


Fig. 7. Mean of expected errors (without *uRg* errors) in N condition vs C condition for novice and expert users

N condition with its “Selection & Validation” dissociation and its numerous facilitations has served its purpose to substantially reduce expected usual errors. Moreover, on a descriptive point of view, even expert users are doing fewer errors in N condition than in C condition (even if there is no significant difference in their case).

One of the task asked users to operate one-handed with the thumb. Similarly, centered on expected errors without uncolocation errors, results show that in N condition, novice users ($M = 0.5, SD = 0.93$) are doing quite as much errors as expert users (0), namely almost peanuts. In thumb interaction mode, N condition made the glass ceiling disappear between novice and expert users. It is not the case in the C condition where novice users ($M = 3.75, SD = 3.54$) are doing significantly more errors ($z = 2.383, p < .02$) than expert users ($M = 0.625, SD = 1.06$). Usual pointing interface (C) is still a trap for novice users, and also for expert users in on-handed situation. On the contrary N condition is well suited both for novice and expert users in a one-handed situation.

4.2 Novice Profile Adjustment

Accordingly to this study the novice profile was easily enhanced. Uncollocation errors on validation command (*uRg*) were quickly annihilated, switching collocation factor on “off” for the Right gesture. Square incertitude (4 mm) and zoom level were augmented a few in order to reduce even more imprecision errors. Medium tap delay was reduced from 800 ms to 500 ms (some users declare 800 ms was a little bit too long) in order to increase a little bit efficiency. Those user tests lead us also to improve displays specifications (size, contrast, and legible “Accessible DfA” font²), audios and tactile feedbacks type and timings.

5 Discussion

Uncollocation is very useful for novice and expert users, mostly in one-handed situation but this focus uncollocation concept needs to be understood in use. N condition experiment was only half an hour and this was a short delay to learn both for novice and expert users. Nevertheless Metatla proposal to use uncollocated navigation has to be carefully implemented. Uncollocation can be used in order to limit pointing errors and to facilitate back command. Indeed Left gesture enables to do a Back without asking for a lot of precision and generates very few manipulation errors. Up and Down command enabling to navigate inside the current list is also enhanced by uncollocation as user needs less precision. Of course, all this needs adequate continuous manipulations and clear feedbacks, feedforwards and affordances.

However, uncollocation for the validation command is a trap for novice and expert user that have not totally acquired this practice. Nevertheless both of them start to acquire it as they were easily doing uncollocated Left gesture and uncollocated Up-Down gestures. As a matter of fact, operative learning transfer was ongoing during this half an hour use of N interface. It starts clearly from the Up-Down navigation inside the current list, at the limits of the screen, when users were trying to reach the next item outside the screen. It was confirmed when users were more and more operating a less controlled Left gesture for going back, thereby freeing their own pointing constraints. The chances are better than not that the next operative learning transfer will be for the validation command. But it has to be brought progressively following a personalized learning curve, perhaps after some hours of use, or some days. Waiting that moment, a solution has to be set for validation command. And this solution must be compatible with a future availability of an uncollocated right gesture that is clearly well suited for thumb usage. This solution is to first reduce validation only to a long Press, and to keep other uncollocated gestures (Up-Down and Left). After some time of use, the adaptative interface will be able to offer to user to add this Right uncollocated gesture. Of course, some user ecological user tests have to be done in order to confirm this assertion.

² Available at <https://github.com/Orange-OpenSource/font-accessible-dfa>.

6 Conclusion

Through this experimental study, innovative touchscreen interaction optimized to novice users was defined, tested and adjusted. Right validation gesture, as source of many errors wasn't kept in the final profile. The separation of selection from validation phase was confirmed as appropriate. Long press of 800 ms delay was validated for secure navigation but will gain in efficiency to be a bit shorter. Left for back command was validated. Dragging focus Up and Down was also appropriate for novice users. Simple Tap for getting information about the current focused item was estimated as really useful.

Enhancement of the novice profile was easily done after the test thanks to the general great flexibility offered by the MenuDfA component. Other profiles are currently underway (low vision, blind, illiteracy, motor, cognitive) and should be useful for answering diversity usually found in elderly population.

References

1. Hwangbo, H., Yoon, S.H., Jin, B.B., Han, Y.S., Ji, Y.G.: A study of pointing performance of elderly users on smartphones. *Int. J. Hum. Comput. Interact.* **29**(9), 604–618 (2013)
2. Trewin, S., Swart, C., Pettick, D.: Physical accessibility of touchscreen smartphones. In: *Proceedings of the 15th International ACM SIGACCESS Conference on Computer and Accessibility*, vol. 19, pp. 1–8 (2013)
3. Kobayashi, M., Hiyama, A., Miura, T., Asakawa, C., Hirose, M., Ifukube, T.: Elderly user evaluation of mobile touchscreen interactions. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (eds.) *INTERACT 2011, Part I. LNCS*, vol. 6946, pp. 83–99. Springer, Heidelberg (2011)
4. MacGookin, D., Brewster, S., Jiang, W.: Investigating touchscreen accessibility for people with visual impairments. In: *Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges*, pp. 298–307 (2008)
5. Karlson, A.K., Bederson, B.B.: ThumbSpace: generalized one-handed input for touchscreen-based mobile devices. In: Baranauskas, C., Abascal, J., Barbosa, S.D.J. (eds.) *INTERACT 2007. LNCS*, vol. 4662, pp. 324–338. Springer, Heidelberg (2007)
6. Bergstrom-Lehtovirta, J., Oulasvirta, A., Brewster, S.: The effects of walking speed on target acquisition on a touchscreen interface. In: *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 143–146 (2011)
7. Chourasia, A.O., Wiegmann, D.A., Chen, K.B., Irwin, C.B., Sesto, M.E.: Effect of sitting or standing on touch screen performance and touch characteristics. *J. Hum. Factors Ergon. Soc.* **55**(4), 789–802 (2013)
8. Karlson, A.K., Bederson, B.B., Contreras-Vidal, J.L.: Understanding single-handed mobile device interaction. In Lumsden, J., (Ed) *Handbook of Research on User Interface Design and Evaluation for Mobile Technologie*, pp. 86–101 (2008)
9. Trudeau, M.B., Young, J.G., Jindrich, D.L., Dennerlein, J.T.: Thumb motor performance varies with thumb and wrist posture during single-handed mobile phone use. *J. Biomech.* **45**, 2349–2354 (2012)
10. Trudeau, M.B., Udtamadilok, T., Karlson, A.K., Dennerlein, J.T.: Thumb motor performance varies by movement orientation, direction, and device size during single-handed mobile phone use. *Hum. Factors J. Hum. Factors Ergon. Soc.* **54**(1), 52–59 (2012)

11. Berolo, S., Wells, R.P., Amick, B.C.: Musculoskeletal symptoms among mobile hand-held device users and their relationship to device use: a preliminary study in a Canadian university population. *Appl. Ergon.* **42**(2), 371–378 (2011)
12. Park, J., Han, S.H.: Defining user value: a case study of a smartphone. *Int. J. Ind. Ergon.* **43**(4), 274–282 (2013)
13. Wobbrock, J.O., Myers, B.A., Aung, H.H.: The performance of hand postures in front-and back-of-device interaction for mobile computing. *Int. J. Hum. Comput. Stud.* **66**(12), 857–875 (2008)
14. Metatla, O., Martin, F., Stockman, T., Bryan-Kinns, N.: Non-visual menu navigation: the effect of and audio-tactile display. In: *Proceedings of the 28th International BCS Human Computer Interaction Conference on HCI 2014-Sand*, pp. 213–217 (2014)
15. Kane, S.K., Wobbrock, J.O., Ladner, R.E.: Usable gestures for blind people: understanding preference and performance. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing System*, pp. 413–422 (2011)
16. Petit, É., Chêne, D.: MenuDfA: navigation gestuelle tactile sans contrainte dans un menu linéaire hiérarchique. Article de recherche, Orange Labs. 2014. Archive ouverte HAL <hal-01056978> (2014)