



**HAL**  
open science

# SpiraList: A Compact Visualization Technique for One-Handed Interaction with Large Lists on Mobile Devices

Stéphane Huot, Eric Lecolinet

► **To cite this version:**

Stéphane Huot, Eric Lecolinet. SpiraList: A Compact Visualization Technique for One-Handed Interaction with Large Lists on Mobile Devices. 4th Nordic Conference on Human-Computer Interaction, NordiCHI 2006, 2006, Oslo, Norway. pp.445-448, 10.1145/1182475.1182533 . inria-00550600

**HAL Id: inria-00550600**

**<https://inria.hal.science/inria-00550600>**

Submitted on 29 Dec 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# SpiraList: A Compact Visualization Technique for One-Handed Interaction with Large Lists on Mobile Devices

Stéphane Huot, Eric Lecolinet

École Nationale Supérieure des Télécommunications (GET) – CNRS LTCI UMR 5141  
46 rue Barrault - 75013, Paris, France  
huot | elc @enst.fr

## ABSTRACT

This paper presents *SpiraList*, a focus+context visualization technique for interacting with large lists on handheld devices. *SpiraList* has been specifically designed to fit the constraints of small tactile screens. It provides a global view of large lists by way of a compact layout based on a spiral representation. Its design also allows for direct interaction with fingers. The combination of this compact layout and one-handed interaction makes *SpiraList* particularly suitable for mobility.

**Author Keywords:** Mobile interfaces, spiral layout, focus+context visualization, finger interaction.

**ACM Classification:** H5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces.

## INTRODUCTION

Most applications for handheld devices still rely on traditional GUI paradigms that have been developed for desktop computers. This model does not fit well on small devices because of their limited screen size and input capabilities. In particular, the efficiency of standard interfaces and interactions remains limited for applications that must display a large amount of data on such small screens.

Figure 1a shows the “Contacts” application available in Windows Mobile 5.0. As shown in the picture, only 8 of 100 items can be displayed simultaneously. Other items are accessed by scrolling the list or by entering textual queries. In addition, alphabetical indexes (on the toolbar) make it possible to shift the list directly to the first item starting with the chosen letter. This design suffers from several limitations. First, scrolling the list may become tedious if it contains many items (i.e. 100 or more). Textual queries require a virtual keyboard that uses a large amount of space, or character recognition that requires training. Besides, it is not possible to have a global view of the address book. This prevents the display of dynamic information about items in a non intrusive manner. For instance, an

instant messaging application could use the list for showing the persons who are currently connected, who sent a new message, etc. A partial view of the address book would thus require opening intrusive pop up windows to provide such information to the user. Finally, most of the interaction with traditional GUI requires using a stylus. An important goal of our new technique is to enable direct interaction with fingers. As a specific case, “thumb interaction” is especially promising because the application can be controlled by using only one hand (a feature that is adapted to mobility).

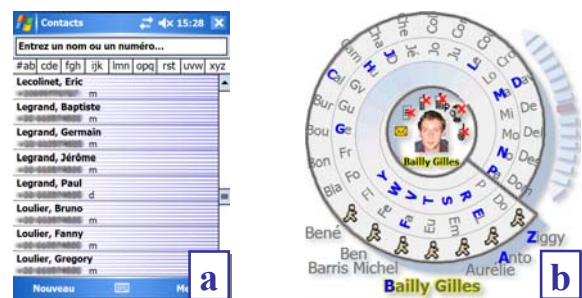


Figure 1. (a) The WM 5.0 standard address book display. (b) A *SpiraList* displaying 100 contacts on a VGA PDA.

Instead of adapting standard GUI techniques to mobile device, our approach consists in designing new techniques that take into account the strong constraints of mobile devices. The work presented in this paper introduces *SpiraList* (Figure 1b) a new interaction and visualization technique for displaying large lists (100 items or more) in an efficient way on small tactile screens. Inspired from previous works on visualization of serial periodic data [4,5], its spiral layout makes the most of concentric spiral revolutions to display a large number of items in a limited amount of space. We have also designed appropriate interaction techniques that can be performed with fingers.

## THE SPIRALIST VISUALIZATION TECHNIQUE

In terms of visualization, our main concern is to provide a *global view* of the list. This “visual access” approach has two main advantages. First, it provides fast and direct access to interesting subparts of the list without any interaction. Second, it makes it possible to display dynamic information in a non intrusive way. To achieve this goal, we have designed a compact spiral representation combined with a *focus+context* mechanism. The spiral layout is divided into equally sized sectors. Each item is displayed

inside a sector in such a way that the first displayed item and the last one are contiguous in the list.

**Focus+Context**

To avoid text overlapping and improve readability, SpiraList rotates and adapts the labels of the items according to their position in the spiral. Labels within the *Focus zone* are fully visible. This zone correspond to a subpart of the outer revolution (it is located on the right side of the spiral in Figure 2). Other labels are truncated in order to fit the spiral outline (3 letters are shown on the outer revolution, then 2 letters, etc.). The zone with truncated labels is called the *Context Zone*.

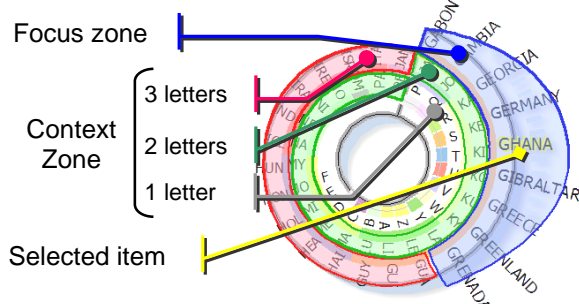


Figure 2. The display zones of SpiraList.

The location and number of items of the focus zone are automatically adapted to the orientation of the screen. In landscape mode, the focus zone is located on the right of the spiral and contains 9 items (Figure 2) whereas in portrait mode it stands at the bottom with 7 items (Figure 1).

**Automatic item grouping**

A spiral layout makes it possible to show more items simultaneously than a linear list. But because of the very limited amount of screen estate, this is not sufficient to display long lists with full labels. Hence, items of the context zone are automatically grouped according to their labels. This means that when several successive items have the same truncated label, only one is shown (for instance, in Figure 3, 13 items are collapsed in the same sector).

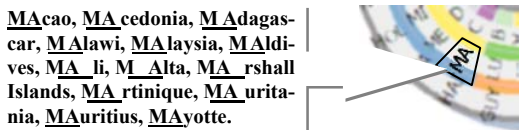


Figure 3. Detail of the alphabetical grouping of labels.

This mechanism can be seen as an automatic collapse/expand technique. In that sense, it is also a fo-

cus+context technique because this progressive alphabetical grouping strategy reduces the size occupied by the list in the context zone, while expanding focused items.

**Visual Clues and Dynamic Information**

Visual access to items is intensified by highlighting each first letter that differs from the previous one in the spiral (see Figure 1b). This alphabetical index reveals the structure of the list and allows the user to visually localize interesting portions of the list. Dynamic information is notified by changing sector background color. For example, this can be used in an address book application to notify events such as an entering call or a new email message. According to our global view goal, those “colored items” are not grouped in the context zone of the spiral so that they are always fully visible. Consequently, there is no need to show a pop up alert nor to automatically change the focus when an event occurs. Several visual artifacts are also used to augment information displayed about items: text color, icons in the focus zone or at the center of the spiral, etc.

**SELECTING ITEMS IN SPIRALIST**

The selection is always at the same location in the spiral. This item is highlighted and stands in the middle of the focus zone (see Figure 2). Hence, changing the focus of SpiraList also changes the selection. The current selection is also shown in the center of the spiral which provides complementary information to the user (for instance, in Figure 4a, the selected item is ‘Didier’). To change the selection, we provide two complementary techniques that are usable in one-handed mobility conditions.

**Continuous focusing**

The first technique is used to reach successive items. It provides a smooth way to browse contiguous items and to expand those that are grouped. It is initiated by touching the screen on the arrows located at the right of the spiral (Figure 4a). Then, the list slowly starts to turn inside the spiral. The speed and direction of the rotation are controlled by the direction and extent of the thumb movement: the higher the user goes, the faster the list turns counterclockwise and vice versa (see Figure 4b-c). The interaction stops when the thumb is lifted off the screen. This technique acts like using an infinite knob to roll out the list through the focus zone. However, controlling the direction and the speed of the rotation instead of the angle bypasses difficulties to perform radial movements, especially when interactions are performed with the hand that holds the device.

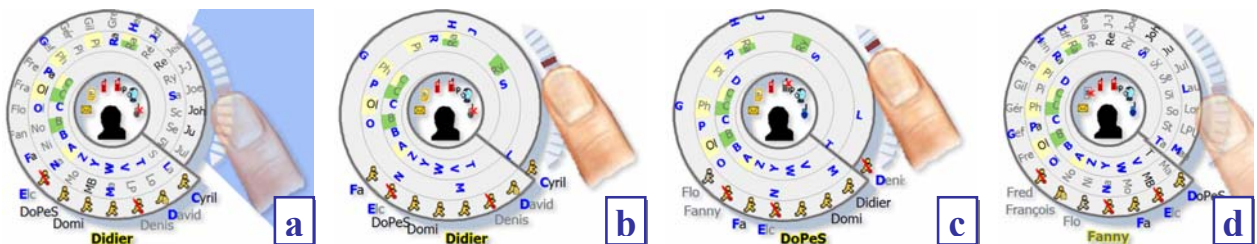


Figure 4. Continuous focus changing by rolling the list inside the spiral.

### Direct Selection

Tapping the screen with a finger on the item of interest is a natural way to directly select an object with a touch screen interface. But this is not possible with sufficient accuracy because of the small size of the screen and the high information density of our representation. Therefore, the SpiraList direct selection uses a “Touch-Drag-Lift” strategy like the “Take-off” technique in [8].

First, *touching* the screen initiates the direct selection by displaying a crosshair cursor on the screen (see Figure 5a). Precise selection is performed during the *drag* operation. Finally, the *lift* on the top of the expected item validates the selection. When the thumb is lifted over a group of labels, the first item of the group is selected. Hence, this technique also permits to expand grouped items on the outermost revolution and to select one of them afterwards. The transition from the original position of the item to the focus zone is notified by an animation where the item goes around the spiral (Figures 5b & 5c). This helps the user to understand that the list position has changed.

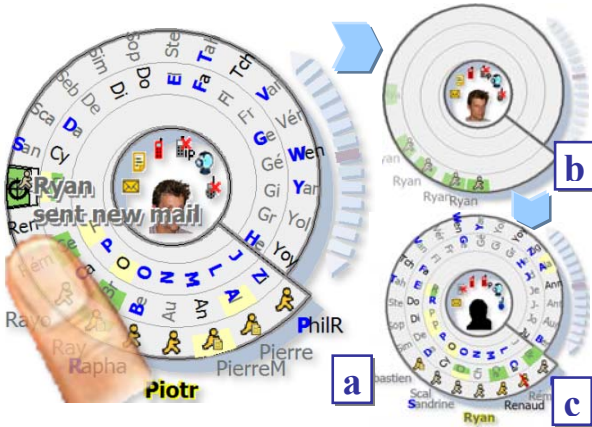


Figure 5. Direct selection: point (a) and release (b) to select the expected item (c).

Although SpiraList gives visual access to all items or groups of items, the compactness of the layout does not make it possible to display all the details. Hence, the corresponding spiral sector is highlighted and a tooltip displays additional information when the cursor flies over an item (see Figure 5a). If the sector contains a group, the tooltip displays information about the first item in the group.

This technique requires precise pointing tasks during the drag. For that reason, we will present the mechanisms we used to improve pointing with fingers on a PDA screen.

### Fingertip Offset and Cursor Stabilization

Finger-based interaction on touch-screens poses two main problems: visual occlusion and pointing/moving precision. To solve these problems, we provide a cursor with a fixed vertical offset, as POTTER et al. in [8]. Using a cursor is more accurate than directly pointing targets with the finger and applying a vertical offset is a simple but efficient way to avoid fingertip occlusions. However, using the thumb to interact on a PDA also makes it difficult to reach the left

and right borders of the screen. To improve access to borders, the cursor is shifted with an adaptive horizontal offset that is computed according to the following function:

$$\delta x = \frac{(x - W/2)^3}{W * \lambda}$$

Figure 6. Adaptive horizontal offset function.

In this equation,  $x$  is the non-corrected position along the horizontal axis,  $W$  the width of the screen and  $\lambda$  the strength of the offset. The behavior of this offset is to grow as the cursor goes away from the center of the screen (see the plot at the right of the Figure 6.). The  $\lambda$  parameter controls the amount of the offset around the central position. An experimental value of 80 give promising results: the offset is unnoticeable around the center of the screen and augments slightly when approaching the edges.

Finally, cursor moves are stabilized to withdraw the lack of precision of PDA touch-screens. Practically, the input hardware returns the mean of real contact points. In case of continuous interactions with a finger (several moving contact points), it can lead to erratic and vibrating cursor jumps. So, the cursor trajectory is smoothed by means of a real-time adaptive low-pass filter [3] that ensures a minimal offset between the filtered and original positions (so that there is no lag, nor slow down of the cursor).

To conclude, the smooth focusing technique is designed to browse the list continuously in the neighborhood of the focus zone. Conversely, the direct selection is the most useful when the expected item is far from the focus zone.

### RELATED WORKS & DISCUSSION

The spiral layout was extensively studied by CARLIS and KONSTAN in [4], for the visualization and interpretation of serial and periodic data. In *SpiraClock* [5], a spiral is augmented by an analog clock to provide a non intrusive display for upcoming events. *RankSpiral* [9] also uses spiral layout to display large sorted results of multiple search engines, but it is not designed for small devices. Finally, CARLIS and KONSTAN already pointed out the possibility to extend the spiral layout with a focus+context scroll mechanism [4]. Even if it deserves different purposes than serial/periodic data analysis, our work achieves this goal for the display of large lists.

Comparatively to traditional scrollable lists, as in the contact list application that was presented in the introduction, SpiraList improves the overall visibility of items by providing a global view of the list. Items or item groups can be found without interacting and dynamic information about items is displayed in a non-intrusive manner. Combined with direct selection of items, SpiraList also avoids the use of scrollbars and does not require using a stylus. Furthermore, alphabetical shortcuts (like in the “Contacts” application) will require many scrolling operations when the list is

not uniformly distributed. In contrast, the automatic grouping of labels in SpiraList ensures the consistency of the layout and always provides direct access.

Other approaches like *Fisheye Menus* [2] improve the overall visibility of large lists with focus-dependant geometrical zoom. Such methods could also be used to display lists on PDAs [7]. But even if all items are visible, only a few of them are readable, above all on a small screen. The compact display of SpiraList achieves the same level of visibility of the list. But its non-geometrical grouping strategy overcomes the legibility problem that arises with focus+context techniques based on geometrical zooming.

Research about touch screens rarely focus on the use of small touch screens in mobility situation. Some recent works on touch screens introduced interesting advanced techniques [1], but they do not suit the constraints of handhelds. Our work extends the “Take-off” technique [8] in the case of mobile devices. In particular, the adaptive horizontal cursor offset of SpiraList, that could be annoying on large touch screens [1], gives promising results on small screens. In *AppLens* and *LaunchTile* [6], the authors introduced one-handed interaction techniques for controlling mobile devices applications. These interactions rely on discrete zooming and gestures performed using the thumb. Although this approach is quite different and does not deal with list display, it defines one-handed interactions that could be compared to SpiraList interactions.

#### IMPLEMENTATION

SpiraList is implemented in C# with the .NET Compact Framework 2.0. It consists in a reusable .NET control that can handle different kinds of items. To achieve interactive framerates on standard handheld devices (Intel ~300 MHz and 128MB RAM), we used several optimization techniques. Even so, graphic details need to be reduced for interactions like continuous focusing. During this interaction, only the text of the focus zone and the alphabet index are displayed (see Figure 4b & c): it maintains interactivity while not decreasing the perception of movement.

#### CONCLUSION & FUTURE WORKS

We have presented *SpiraList*, a new visualization and interaction technique that improves the display of long lists on handheld devices. It combines the compactness of a *spiral layout* with a *focus+context* strategy that allows visual access to all list items. We have designed specific interaction techniques that are consistent with this spiral layout and for one-finger interaction. SpiraList could be used in a wide variety of mobile applications that require lists such as address books, multimedia databases, TV remote, etc. We think that it is also potentially a good way for displaying hierarchical lists and trees. In that sense, we have started to explore some improvements such as displaying categories around the spiral outline or using several SpiraLists hierarchically. However, the main challenge is to keep those enhancements consistent with the SpiraList design goals: *visual access* and *thumb interaction*.

Finally, the preliminary user feedbacks we have collected are encouraging and show good user acceptance. Future works will include more extensive usability studies comparatively to other approaches.

#### ACKNOWLEDGMENTS

This work has been done in collaboration with Guillaume DORBES and Bruno LEGAT from Alcatel Research. We want to thank them for their useful advices and their help in evaluating SpiraList. Many thanks to Renaud BLANCH and Gilles BAILLY for the valuable discussions about this work.

#### REFERENCES

1. Benko, H., Wilson, D. A. and Baudisch, P. Precise Selection Techniques for Multi-Touch Screens. In Proceedings of *CHI'2006 Human Factors in Computing Systems* (April 22-27, Montréal, QC, CA), ACM/SIGCHI, NY, 2006, pp. 1263–1272.
2. Bederson, B. B. Fisheye Menus. In Proceedings of *13<sup>th</sup> ACM Conference on User Interface Software and Technology (UIST 2000)* (November 5-8, San Diego, CA, USA), ACM/SIGCHI, NY, 2000, pp. 217–226.
3. Blanch, R., Ferley, E., Cani, M.-P. and Gascuel, J.-D. Non-Realistic Haptic Feedback for Virtual Sculpture. Research Report 5090, INRIA, France, 2004.
4. Carlis, J.V. and Konstan J.A. Interactive Visualization of Serial Periodic Data. In Proceedings of *11<sup>th</sup> ACM Conference on User Interface Software and Technology (UIST 98)* (November 1-4, San Francisco, CA, USA), ACM/SIGCHI, NY, 1998, pp. 29–38.
5. Dragicevic, P. and Huot, S. SpiraClock: A Continuous and Non-Intrusive Display for Upcoming Events. In *Extended Abstracts of CHI'02 Human Factors in Computing Systems* (April 20-25, Minneapolis, MN, USA), ACM/SIGCHI, NY, 2002, pp. 604–605.
6. Karlson, A. K., Bederson B. B. and SanGiovanni, J. AppLens and LaunchTile: Two Designs for One-Handed Thumb Use on Small Devices. In Proceedings of *CHI'2005 Human Factors in Computing Systems* (April 2-7, Portland, OR, USA), ACM/SIGCHI, NY, 2005, pp. 201–210.
7. Lecolinet, E. and Nguyen, D. Focus+context visualization of zoomable hierarchical lists. In Proceedings of *French-Speaking Conference on Human-Computer Interaction (IHM 2006)* (April 18-21, Montréal, QC), ACM, NY, 2006, pp. 195–198.
8. Potter, R.L., Weldon, L.J. and Shneiderman, B. Improving the Accuracy of Touchscreens: An Experimental Evaluation of Three Strategies. In Proceedings of *CHI'88 Human Factors in Computing Systems* (June 15-19, Washington, DC, USA), ACM/SIGCHI, NY, 1988, pp. 27–32.
9. Spoerri, A. RankSpiral: Toward Enhancing Search Results Visualizations. In Proceedings of *IEEE Symposium on Information Visualization (InfoVis 2004)* (October 10-12, Austin, TX, USA), IEEE, 2004, pp. 18–19.