Continuous marking menus for learning cursive pen-based gestures

Adrien Delaye, Rafik Sekkal, Eric Anquetil

Continuous Marking Menus for Learning Cursive Pen-based Gestures

Adrien Delaye Rafik Sekkal Eric Anquetil
Université Européenne de Bretagne, France
INSA, IRISA, UMR 6074, F-35708 RENNES
{adrien.delaye, rafik.sekkal, eric.anquetil } @irisa.fr

ABSTRACT

In this paper, we present a new type of Marking menus. Continuous Marking Menus are specifically dedicated to pen-based interfaces, and designed to define a set of cursive, realistic handwritten gestures. In menu mode, they offer a continuous visual feedback and fluent exploration of menu hierarchy, inviting the user to execute cursive gestures for invoking the desired commands. In marking mode, a specific gesture recognition method is proposed and proved to be very efficient for recognizing cursive gestures.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: Interaction styles; H.5.2 Information Interfaces and Presentation: Input devices and strategies

General Terms

Design, Human Factors

INTRODUCTION

Pen-based interfaces offer a very efficient interaction mode, as gestures are realized directly on the target area of the screen, with a stylus manipulated in a natural way. Moreover, gestural commands allow eyes-free selection and are thus specifically suitable for interacting with mobile devices [5]. With the progress of techniques for gesture recognition, applications now offer a vocabulary of complex and cursive gestures exploiting all the variety permitted by handwriting, including singularities, loops, inflections points. . . [1, 7, 8]. Such richer sets of gestures allow higher symbolic significance, and can be easier to remember than piecewise linear gestures. Despite the aforementioned advantages, gesture-based interfaces suffer that gestures are not *self-revealing* (list of possible gestures is not easily accessible to the user). To cope with this, shapes of gestures are often chosen to be as much as possible linked by a *logical relation* with the command to be executed (semantic, visual or mnemonic relation). For instance, the *Copy* action can be associated to a “C”-shaped gesture, or *Erase* action to a “cross-out” gesture. The more an action-gesture association respects some relation understandable for the user, the easier it is to learn and remember it [7, 8]. However, this logical association is often insufficient, and learning or remembering gestures can remain a tedious task, especially if the number of gestures is important. The *logic* of action-gestures associations can appear as poorly intuitive for some users, and is often unable to deal with purely abstract or non-representable actions.
Alternative solutions exist for making gestures self-revealing. In [4], authors propose an interactive guiding of pen trajectory by dynamically displaying gestures suggestions. This assistance facilitates memorization and no effort is needed to remember gestures. Another solution is to let users define their own gestures and associate actions, but this requires recognition engines able to deal with new shapes, while demanding as few training samples as possible. Designing this type of recognition methods is still a challenging pattern recognition problem [1].

Marking menus offer an alternative for assisting the user in learning gestures [5]. These menus offer two modes : item selection from a menu, and gesture command without the menu being displayed. First mode is a traditional use of a pie menu: user selects an item with a pen (or a sub-menu he wants to explore in the case of hierarchical menus). Second mode implies gesture recognition, and works like a gesture-based interface, where the user is familiar with the gestures commands for triggering actions. Kurtenbach’s idea is to link the gesture shapes to the layout of the underlying menu: the physical movement for selecting a menu item is identical to the physical movement required to make the associated gesture command [5]. By repetitively using the menu mode, user can thus memorize gestures without explicitly making any effort. One drawback of marking menus and variants, though, is they are limited in expressiveness as they are restricted to polygonal gestures [5], or sequences of linear gestures [10, 2] (except for Flower Menus that allow degrees of curvature [3]).

In this paper, we introduce Continuous Marking Menus, an improved type of marking menus implying a rich set of cursive gestures, that can be executed in a single continuous gesture with a pen or finger. We first propose some novelties for topology and interaction behavior to help the user make selection of menu items in a more fluent and gesture-like fashion. Integration of specific *inertia areas* permits to absorb imprecision in the change of direction, coupled with a continuous visual feedback, supports the user in the making of fluent gestures. Finally, we propose a specific gesture recognition technique that can handle the variability of resulting curved gestures in the marking mode. A review of marking menus related work is given in the first section, before we introduce Continuous Marking Menus design. The recognition method used in the marking mode is then presented briefly, together with an experiment that show its ability to properly recognize cursive gestures.

RELATED WORK

Marking Menus were introduced for making gestures self-revealing with the help of a radial menu that is popped-out according to the user need. They permit to implicitly learn the gesture commands by repetitively using the menu mode, and the transition to marking mode is facilitated by making user *rehearse the physical movement involved in making the gesture every time a selection from the menu is made* [5].

The basic, non-hierarchical version of marking menus involves simple radial gestures from menu center toward the chosen direction. In the hierarchical version (Compound Marking Menus), several levels can be explored by making a piecewise linear gesture, each part corresponding to the selection of an item or sub-menu at one level. Such gestures can be easily executed with the help of a mouse, even if the use of the pen is significantly more efficient [5]. However, strong constraints about their shapes (polygonal shape with abrupt angles) are actually in contradiction with the drawing of fluent handwritten gestures with a pen, because users have to pause at each menu level, and gestures are more a sequence of linear movements.

Simple Marking Menus [10], or variants as Zone and Polygon menus [9] offer to decompose the selection of sub-menus and items into several disconnected marks, one for each level. They permit to solve two problems of Compound Marking Menus. First, they reduce the space required for menu display and drawing of gestures in both modes. Secondly, they allow an increased depth (more successive levels in the hierarchy), by simply adjusting the number of menu selections (or marks) in the sequence. Zone and Polygon menus further permit more items at each level (increase breadth). Simple Marking Menus and alike, however, are less efficient in menu mode because the navigation path through the hierarchy is not visible to the user, and no previsualization (browsing of sub-menus for a fast inspection of available commands) is possible. More importantly, by nature they require several distinct gestures to invoke commands, which is a step away from our objective of permitting fully continuous gestures. Wave Menus [2] improve the novice mode of Simple Marking Menus and help learning gestures by providing enhanced feedback, but associated marks are of the same nature as Simple Marking Menus.

Recently introduced Flower Menus [3] handles a set of curved gestures. More items are displayed at each level by offering several items in each direction that are distinguished by their degree or sense of curvature. Advantages are an increased breadth up to 56 items (7 items in each of the 8 directions), and the sub-grouping of items within the same menu level that facilitates the learning of gestures. Although admitting different degrees of curvature, the resulting gestures are however still constrained to a limited variety of shapes. Moreover, the Flower Menu being another variant of Simple Mark Menu, a sequence of disconnected gestures is needed to invoke a command.

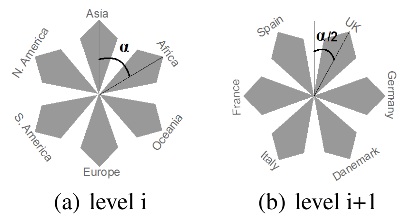


Figure 1. Two configurations of Continuous Menus alternately used in successive hierarchical levels: level i (a) and sub-menu level i+1 (b)

As a summary, the existing variants of Marking Menus only offer a rather limited vocabulary of gestures, mostly linear or piecewise linear gestures, not dedicated to direct interaction with pen or finger. Our work aims at improving the marking mode of these menus, by tolerating a much larger range of gestures expected to be faster and more efficient to execute. The challenge is twofold : how to design a new type of marking menu offering an enlarged gestural expressiveness while maintaining an efficient menu mode, and how this menu mode can be optimized for inviting to make fluent, gestural-like menu selections.

CONTINUOUS MARKING MENUS

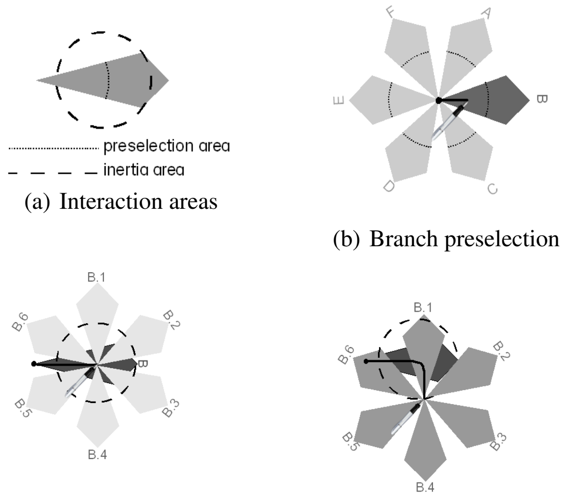
Continuous Marking Menus are a variant of Compound Marking Menus [5] with enhanced interaction and continuous user feedback in the menu mode that facilitates continuous and fluent movements for selecting commands.

Design

Continuous Marking Menus are based on hierarchical radial menus with a breadth of 4, 6 or 8 items at each level. All the examples shown in this article are based on a 6-breadth version, which we used for experiments. For each level, the menu consists of a set of kite-shaped branches, centered around the point where the menu was triggered by the cursor. The labels are displayed at the extremity of each branch, in a rectangle oriented perpendicularly to the direction of the branch. Contrarily to traditional pie menus, the active areas are not contiguous (there is an inactive blank zone between two neighboring branches), in order to guide the user make menu selections in a more trajectory-like way. While classical Marking Menus display only labels as targets to be selected, we propose to display directed branches that suggest a *path* to invoke a command. Figure 1 shows two configurations of this menu, with 6 branches each. The two configurations only differ by a rotation (axis-shift), and they will be alternately displayed when exploring the menu hierarchy levels, in order to avoid ambiguity problems by forcing a change of direction whenever a sub-menu is developed.

Interaction

A specific interactive behavior is required for assisting the user and invite him to make continuous gestures while exploring the menu. Specifically, Continuous Menus offer a



(a) Interaction areas

(b) Branch preselection

(c) Entering inertia area

(d) Leaving inertia area

Figure 2. Interaction areas

constant visual feedback by an instantaneous reaction to any of the user's movements. The branches offer two interaction areas, namely a *pre-selection area* and a *inertia area*, illustrated in figure 2(a).

Pre-selection area is useful for browsing the menu level and let the user look for desired item or sub-menu. The associated visual feedback highlights the branch more and more intensely as the user cursor moves toward the branch extremity. Simultaneously, the other branches of the menu level are faded out, as shown in figure 2(b). When the cursor crosses pre-selection limit, branch menu is considered as pre-selected, and the other alternative branches are completely invisible. As the cursor enters the *inertia area*, the sub-menu of the pre-selected branch fades in and its branches are more and more visible as the cursor moves away from the center of the inertia area (see figure 2(c)). In this area, the center of the sub-menu sticks to the movement of the cursor, until it crosses the outter boundaries of the area (see figure 2(d)). This area permits to absorb the inertia of the pen movement when a change of direction is executed in order to encourage the user to make fluent moves when exploring the menu hierarchy. Indeed, thanks to this mechanism, no stopping or dwelling in the user movement is needed when changing direction. Inertia area can be seen as the tolerance area where the change of direction can be done.

Detailed menu selection process

The dynamic interaction offered by the menu when user make a menu selection can be described in details by the following succession of steps, illustrated in figure 3[1].

(a) Initial menu display, centered on the cursor.

(b) Pre-selection area. The chosen branch ("Europe") is highlighted, the other ones fade out.

(c) Pre-selection of sub-menu ("Europe"), and entering in the inertia area. Only the chosen branch is visible.

(d) Inertia area. Sub-menu branches progressively appear, and the sub-menu follows the position of the cursor.

(e) Crossing of the inertia area limit. The sub-menu is now fixed. The parent branch ("Europe") is validated, and its shape changes to a smaller kite as a visual confirmation. Another cycle starts (back to step (a)).

At any time before Branch validation (step (e)), the user can move back to the parent menu and change the preselected branch by moving back the cursor along the branches toward the center of parent menu. The continuous feedback, including fading out of parent overlooked branches and fading in of subsequent child sub-menu not only stimulates the user for pursuing his gesture, but is also precious for maintaining an acceptable visual complexity. The chart in figure 3 shows that the number of items displayed during the selection process remains limited, and branches from two level of menus are never displayed simultaneously.

MARKING MODE WITH CURVED GESTURES

The marking mode of Continuous menus involves a gesture recognizer that do not provide any visual interaction during the execution of the gesture. Once the gesture is accomplished, several pre-processing are applied (normalization, resampling), and the Dynamic Type Warping (DTW) recognition technique is used : elastic matching distance is computed by dynamic programming between the input gesture and each gesture prototype (modeling the ideal gesture for each possible command). The input gesture is assigned the class corresponding to the closest prototype according to DTW distance, and the associated command is invoked. Since elastic matching distance does not involve a polygonal approximation of gestures, it is suited to handle properly cursive and distorted gestures. Moreover, the nearest-neighbor classification paradigm require no training phase, so it could be possible for the user to dynamically modify, add or remove some menu items according to his needs.
A database of gestures was constituted by asking subjects to execute 32 different gestures associated to a 3-level menu with a breadth of 6 items per level (see figure 4(b)). In order to assist participants in achieving expert behavior, we displayed a visual guide for the required mark, thus eliminating the need for familiarity with gestures. The visual guide corresponds to the display of starting and ending point and several fuzzy grey areas denoting the inertia areas, where the user should approximately make direction changes (figure 4(a) shows an example of visual guide). A Tablet PC with a pen was used for the input. 10 people participated in the experiment, with age between 18 and 45. Each of them executed 10 instances of each gesture, amounting to a total number of 3200 samples. Our experiment with DTW recognizer reached a correct recognition rate of 99,97%, confirming its great ability to discriminate between cursive gestures.

CONCLUSION

In this work we introduced Continuous Marking Menus for improving the expressiveness of marking menus and admitting a vocabulary of cursive, fluent handwritten gestures,

[1]See video at `http://www.irisa.fr/imadoc/CMM.html`

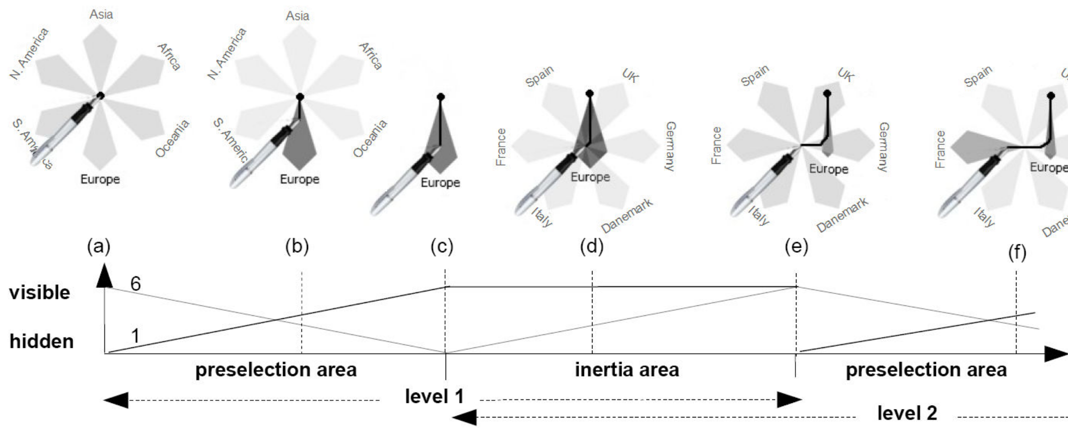


Figure 3. Steps for menu selection (a)-(f) and variation of the visual complexity.

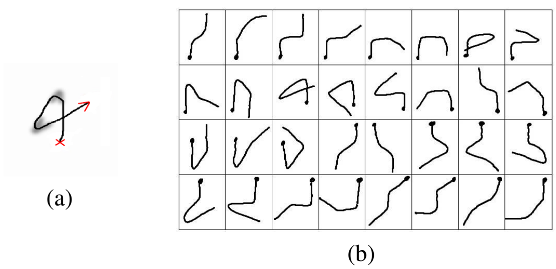


Figure 4. Example of guide for assisting expert behavior (a), and table of 32 gestures classes (b).

and thus offer a more natural and efficient interaction in the marking mode. We proposed an improved interaction in the menu mode and a specific mechanism for handling imprecise direction changes, suited to the specificities of pen based or finger based gestures. The method proposed for recognition of gestures in the marking mode is well adapted to cursive gestures, and its performance was experimentally validated. The very high recognition rate suggests that the breadth and depth choices for our menus are not a limitation for our classification method. Future work will focus on evaluating acceptability and efficiency of these menus in comparison with alternative techniques. The gain of efficiency and user satisfaction should be measured and compared in the marking mode when using cursive vs polygonal gestures. Finally, in order to push further the idea of efficient single-gesture commands, we will also consider combining both the selection and control of actions in a single gesture, in the light of the works of Pook[6].

ACKNOWLEDGMENTS
This work was partly supported by the Region Bretagne through the ScriptEverywhere project.

REFERENCES
1. A. Almaksour, E. Anquetil, S. Quiniou, and M. Cheriet. Evolving fuzzy classifiers: Application to incremental learning of handwritten gesture recognition systems. In *Proc. of the 20th ICPR*, 2010.
2. G. Bailly, E. Lecolinet, and L. Nigay. Wave menus: improving the novice mode of hierarchical marking menus. In *Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction*, pages 475–488, 2007.
3. G. Bailly, E. Lecolinet, and L. Nigay. Flower menus: a new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. In *Proc. of the working conference on Advanced visual interfaces*, pages 15–22, 2008.
4. O. Bau and W. Mackay. OctoPocus: a dynamic guide for learning gesture-based command sets. In *Proc. of the 21st annual ACM symposium on User interface software and technology*, pages 37–46, 2008.
5. G. Kurtenbach and W. Buxton. The limits of expert performance using hierarchic marking menus. In *Proc. of INTERCHI'93*, page 487, 1993.
6. S. Pook, E. Lecolinet, G. Vaysseix, and E. Barillot. Control menus: excecution and control in a single interactor. In *CHI'00 extended abstracts on Human factors in computing systems*, page 264, 2000.
7. D. Rubine. Specifying gestures by example. In *Proc.s of the 18th annual conference on Computer graphics and interactive techniques*, pages 329–337. ACM New York, NY, USA, 1991.
8. D. Willems, R. Niels, M. van Gerven, and L. Vuurpijl. Iconic and multi-stroke gesture recognition. *Pattern Recognition*, 42, 2009.
9. S. Zhao, M. Agrawala, and K. Hinckley. Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, volume 2, page 1077, 2006.
10. S. Zhao and R. Balakrishnan. Simple vs. compound mark hierarchical marking menus. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 33–42, 2004.