Using a Painting Metaphor to Rate Large Numbers of Objects

Patrick Baudisch
Integrated Publication and Information Systems Institute (IPSI)
German National Research Center for Information Technology (GMD)
64293 Darmstadt, Germany
+49-6151-869-854
baudisch@gmd.de

1 The problem

When retrieving information from databases or search engines, or when configuring user profiles of information filtering systems, users have to describe what objects to retrieve. While some systems require users to describe their needs using textual input, other systems simplify the users' task by proposing a set of items, so that the user's task is reduced to picking or rating these items (e.g. relevance feedback (Haines 93)). This task is generally much simpler than writing queries from scratch. In interfaces of this type, users have to provide information of the type "does this item represent my information interest", "do I like this item" or "how much do I like this item". Similar problems are encountered in utility theory, when assessing the user's value functions (Keeney and Raiffa 76). As an example, Figure 1 shows such a selection user interface. It allows users the selection of TV channels, e.g. to configure their user profile for a TV recommender system. The interface contains about sixty toggle switches.

Assessing a large number of items can be time consuming. How can interfaces handle hundreds of such selection or assessment tasks in an efficient way? Sometimes it is possible to aggregate objects or to provide good defaults, so that only a few objects have to be manipulated in the first place. But what to do if there are no good defaults or too many of them? The set of TV channels that users can receive depends not only on the carrier, such as cable or satellite, but also varies widely depending on the local provider and subscriptions to pay-TV. If the interface provided extra buttons for every useful default configuration, the number of these buttons could easily exceed the number of actual toggles in the interface.

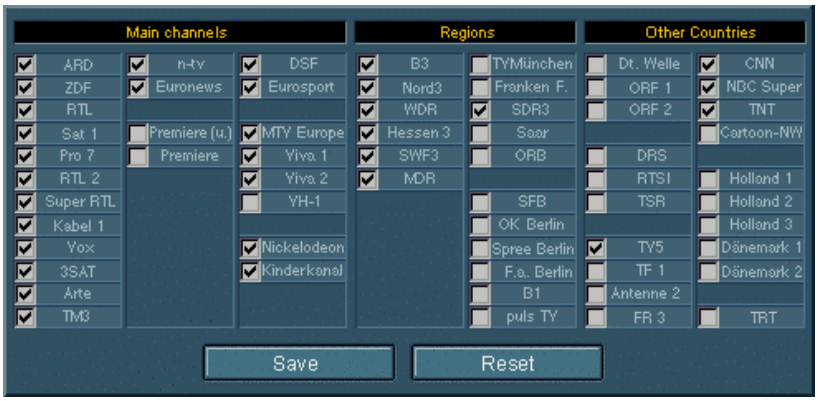

Figure 1: Dialog allowing users to input their personal TV channel profile. Each TV channel name is associated with a toggle switch.

2 Multiple select and painting

How are large numbers of interface objects handled in other application areas? One common technique is multiple select. First, users select a subset of items, e.g. cells in spreadsheet programs, icons in desktop GUIs, or pixels in paint programs. Usually this can be done with a mouse drag operation or using multiple mouse clicks while keeping a qualifier key depressed. Then, users select a method to be applied, e.g. clear the selected cells, move the selected icons, or set the selected pixels to a specific color. This order (select items first, then select method) is called *noun-verb* order (Smith 82). The noun-verb order allows restricting the list of available methods to those methods that are applicable to the items within the current selection (e.g. “empty trashcan” only to non-empty trashcans).

Paint programs offer more possibilities. Since painting programs deal with a single object type only, i.e. pixels, the noun-verb application order is not imperative. Therefore, paint programs provide both the noun-verb order as described above, and the verb-noun order, called painting. In the latter case a function is chosen, e.g. a painting tool such as pen or an airbrush, and then applied continually to all subsequently selected pixels.

The noun-verb order is preferable if several methods are to be applied to the same, possibly complex, selection. Otherwise painting has two advantages over selection. First, if the same painting tool is used several times in a row, then the tool has to be chosen only the first time, which saves interactions in all subsequent paint actions. Second, since the manipulation of painted items takes place immediately, painting gives better visual feedback.

We propose to apply these interaction techniques to the problem of object assessment stated at the beginning of this article. An interface object providing a Boolean value range (e.g. a toggle switch) is functionally equivalent to a black and white pixel; an object with a wider, but finite value range can be represented by a gray scale pixel. Consequently, larger layouts of such interface objects can be handled as if they were black-and-white or grayscale image respectively. Consequently, all advantages of pixel painting can be made available to the efficient assessment of objects. The resulting sets of interface objects combined with one or more painting methods are called “toggle maps” (Baudisch 98).

3 Requirements and layout

To maximize the benefit from toggle maps, two requirements have to be met. 1) Individual items should bear no or only small labels and should not require much time for decision making. Otherwise, users might prefer to release the mouse button and click switches individually. 2) It must be possible to manipulate several toggle switches per mouse drag. Otherwise there is no speed-up. This requires two things. First, there must be a need to manipulate a significant number of switches during individual sessions. Second, a significant frequency of co-occurrence between toggles has to exist and to be reflected by the layout (see below).

The goal of toggle map layout is to maximize usage speed by optimizing the factors stated above. To minimize recognition time (Requirement 1), layout should group buttons according to subjective similarity, so that recognizing one button already provides information about the neighboring buttons. To maximize the scope of paint operations (Requirement 2), layout should group buttons that are frequently manipulated together. Common techniques to accomplish this are non-metric multidimensional scaling and disjoint cluster analysis, e.g. using bottom-up clustering (Lindsey 77). Due to space limitations, we cannot discuss toggle map layout in detail here. However, the requirements toggle map layout tries to meet are very similar to the layout requirements of spatial menus (e.g. Norman 91, p. 261-280), so that many concepts can be transferred. In general, layouts based on frequency of co-occurrence usually offer good results (McDonald 86). Nonetheless, toggle map layout is more demanding than menu layout. Toggle map layout not only has to assert relative proximity of similar items but also direct adjacency, optimized for the respective paining tools.

4 Assessing objects using toggle maps

Using the techniques described above, the TV channel interface shown in Figure 1 can be handled as a toggle map. In this example, a tool which paints filled rectangles proved to be particularly efficient. Experiments proved that

"paintable" interfaces such as the one shown in Figure 1 provide significant speedups over click-only interfaces (Baudisch 98).

Alternatively to creating layouts using multidimensional scaling, it is possible to apply the toggle map concept to domains whose a regular internal structure practically imply a layout. In (Baudisch 98) an efficient timer interface was presented that used a tabular layout with days forming rows and hours forming columns. Layouts based on such strong internal structure have the advantage of meeting the requirements especially well, because both subjective similarity and frequency of co-occurrence between neighboring objects are unusually high.

Figure 2 gives another example of such a domain with a strong internal structure. The shown toggle map allows users to express their preferences concerning computer monitors, e.g. to retrieve the corresponding classified ads from a database. In the shown state, the user is looking for a rather large screen with a high resolution, e.g. for CAD applications. The interface contains buttons only for those feature combinations that correspond to available objects. Buttons can take a discrete range of grayscales to represent different grades of like and dislike. They can be painted very efficiently, e.g. using an airbrush.

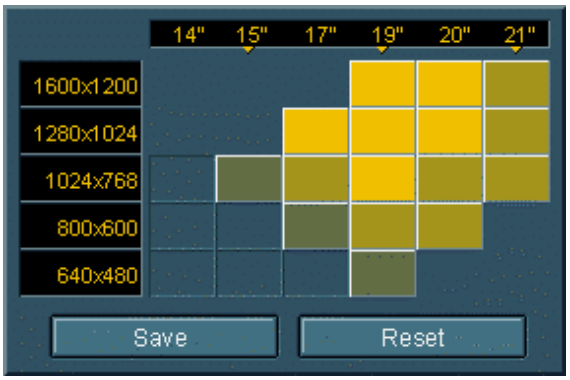

Figure 2: Toggle map allowing users to enter their preferences about computer monitors.

If the internal structure of the data objects is more than two-dimensional, it is possible to adapt the implied toggle map layout within bounds. Figure 3 shows two examples that are derived from the example shown in Figure 2 by successively introducing two more dimensions. To keep the original 2D painting interaction applicable, we used so-called *explosion displays* that solve the problem of occlusion. Layers of toggles are spread, the resulting gaps provide access to buttons that would otherwise be occluded. Using this technique, toggles at any depth can still be accessed directly with a 2D input device, such as a mouse. To provide maximum efficiency, additional n-dimension tools can be introduced. Figure 3a sketches the usage of a tool that allows painting filled (hyper) cubes by dragging the mouse from one vertex to the diametrically opposed one.

(a) (b)

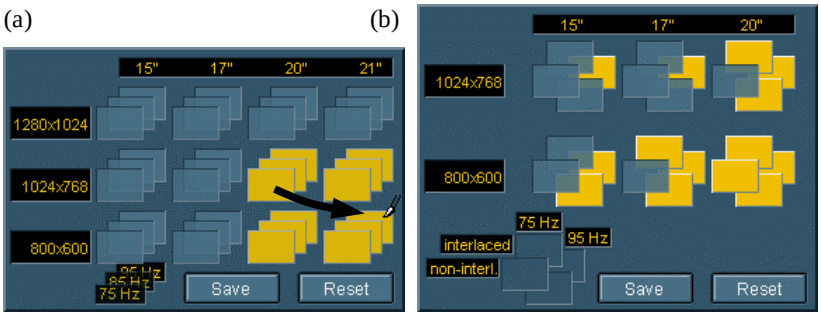

Figure 3: A three (a) and a four-dimensional (b) toggle map.

5 Conclusions and Future work

In this article, we discussed how painting interactions can be applied to the selection and assessment of objects, especially objects from domains having a regular internal structure facilitating layout. Future work will include experiments with computer generated layouts and n-dimensional painting tools.

6 References

Baudisch, P. (1998) Don't click, paint! Using Toggle Maps to Manipulate Sets of Toggle Switches. In *Proceedings of UIST '98*, San Francisco, pp. 65-66.

Haines, D. and Croft, W. (1993). Relevance Feedback and Inference Networks. In *Proceedings of SIGIR '93*. Pittsburgh, PA: ACM Press. pp. 2-11.

Keeney R.L., and Raiffa, H.R. (1976). *Decision with Multiple Objectives: Preferences and Value Tradeoffs.* John Wiley and Sons.

Lindsey, P.H., and Norman, D.A. (1977) *Human information processing*, New York: Academic Press.

McDonald, J.E., Dayton, T., McDonald, D.R. (1986). *Adapting menu layout to tasks* (MCCS-86-78). Las Cruces, NM: Memoranda in Computer and Cognitive Science, Computer Research Laboratory, New Mexico, State University.

Norman, K. L. (1991). *The Psychology of Menu Selection: Designing cognitive control at the human/computer interface,* Norwood, New Jersey, Ablex.

Shneiderman, B. (1998). *Designing the User Interface: Strategies for effective human-computer interaction*, 3rd edition, Reading MA: Addison-Wesley.

Smith, D.C. Irby, C. Kimball, R. Verplank, W., and Harslem, E. (1982). Designing the Star User Interface, *Byte* 7(4):242-282. McGraw-Hill, Inc.