Supportive Interfaces for Creative Visual Thinking

Takeo Igarashi

The University of Tokyo
takeo@acm.org
http://www.mtl.t.u-tokyo.ac.jp/~takeo/

Abstract

Current graphical user interfaces (GUI) are too cumbersome to use in the early stage of creative activities. Nested menus, arrays of buttons, and dialog boxes interfere with the flexible exploration. As a result, especially in visual task domain, most people rely on traditional pen and paper for the initial exploration, and move to the computers later. Our goal is to provide fluent user interfaces that can facilitate these early, exploratory stages of visual thinking activity on computational systems. Basic idea is to make interfaces *transparent* so that the user can directly interact with the target visual representations without using menus and buttons. We will introduce three example systems. Pegasus interactively beautifies freeform drawings satisfying possible geometric constraints. Teddy allows the user to design freeform three-dimensional objects by drawing their silhouette shapes. Flatland provides various computational supports for simple note-taking activity on an office whiteboard. We report the lessons learned from these experimental systems, and clarify the open problems.

1. Introduction

Typical creative activities, such as paper writing and diagram drawing, consist of successive development stages, ranging from early exploration to final refinement. In the early exploration stage, people gather various possible ideas and evaluate each of them. Some ideas survive for further development, while most are discarded. In this stage, the ability to quickly experiment with alternative scenarios is essential. In the final refinement stage, people focus on the detailed features and drastic modifications do not occur.

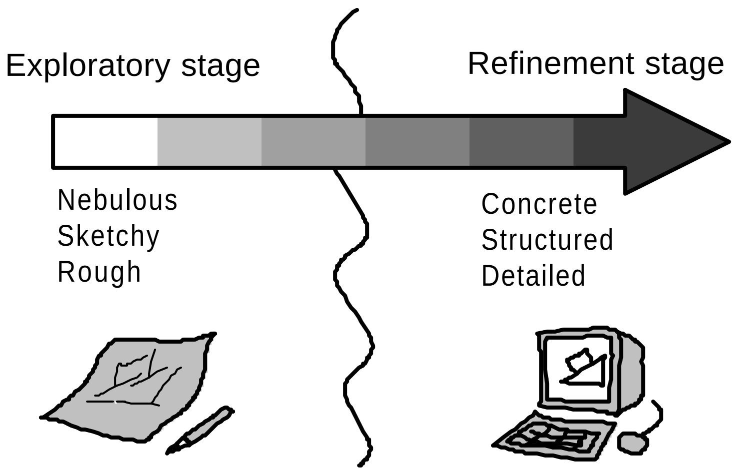


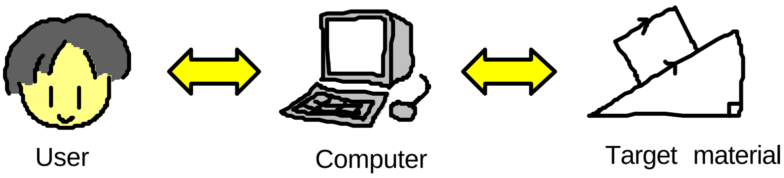
Figure 1: Development stages of creative activity.
Computers are mainly used in the refinement stage.

Various computational systems have been developed to support intellectual activities, such as word processor, spreadsheets, presentation tools, and diagram editors. However, these tools are generally designed for the final refinement stage of the entire creative process, and support for the early exploration stage has been limited, especially for visual materials such as diagrams. While more and more people use computational systems such as word processors or outlining tools in the early stage of text writing, most people still use traditional pen and paper of diagram drawing or shape design [6, 7,11]. The goal of our research is to develop computational systems that can facilitate these early exploration stages for visual materials(Figure 1).

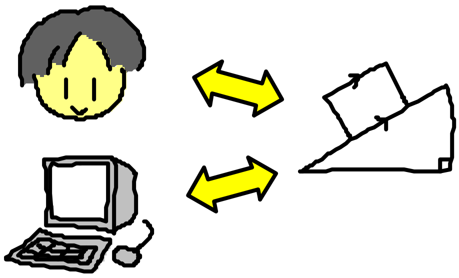
2. Supportive Interfaces

Fundamental limitation resides in the currently dominant graphical user interface (GUI) framework. While GUI significantly improved the previous command-line interfaces, it is still based on the notion of explicit *control* of computers via *command* operation [9,10]. From the user's point of view, the target materials (such as database, document, and diagram) are concealed by the GUI widgets, and the user modifies the material by giving appropriate commands using the widgets. The user can interact with the target material only indirectly through the computer. This interaction style is appropriate for well-structured information such as mathematical equations, databases, and spreadsheets, but is too tedious for more flexible materials such as freeform drawings.

As an alternative user interface framework, we propose *supportive* user interfaces. In this framework, the user interacts with the target material directly without using GUI widgets. A computer is a kind of *assistant* working with the user on the same material, rather than an agent that stands between the user and the target material (Figure 2). While we believe that the basic idea is applicable to wide variety of computational applications, our current focus is on stroke-based drawing programs[2]. The user draws arbitrary shapes on a canvas using freeform strokes, and the computer provides appropriate support when necessary. We will introduce three example applications based on the idea of supportive interfaces below.



a) Traditional GUI model.
The user interacts with the target material indirectly using GUI widgets and explicit command operation.



b) Supportive Interface model.
The user directly interacts with the target material, and the computer provide additional support when necessary.

Figure 2: Supportive User Interfaces

3. Example Interface Systems

Pegasus

Pegasus [3,4] is a prototype 2D drawing application for pen-based systems. Using Pegasus, a user can construct precise geometric diagrams easily without struggling with complicated editing commands. Two novel interaction techniques are introduced: interactive beautification and predictive drawing. Interactive beautification receives the user's free stroke input and beautifies it by considering possible geometric constraints among segments. The system generates multiple candidates to prevent recognition errors. Predictive drawing predicts the user's next drawing operation based on the spatial relationship among existing segments on the screen. The user can construct various configurations just by successive clicking as long predicted segments contain the desired one. A user study showed that the user can draw simple geometric diagram more rapidly and precisely using Pegasus than commercial drawing editors and CAD programs.

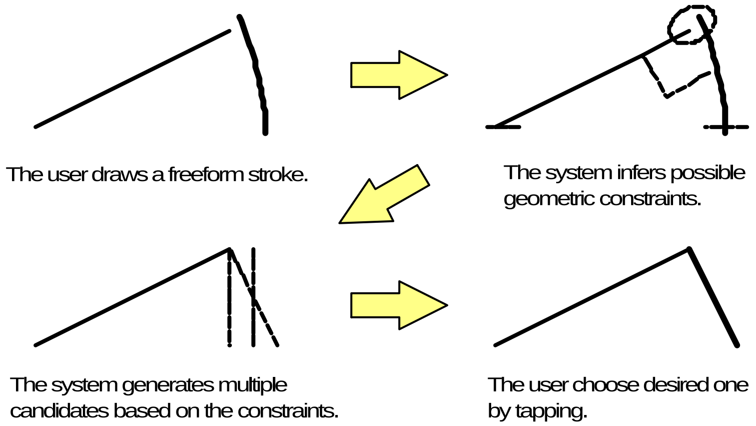


Figure 3: Interactive Beautification
The system automatically beautifies the user’s freeform stroke, and generates multiple candidates. The user can select desired one by tapping on it.

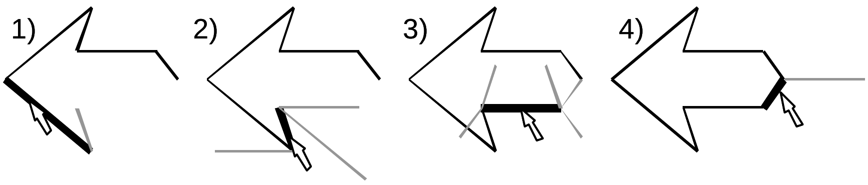


Figure 4: Predictive Drawing
The system predicts the next drawings and shows them as multiple candidates. The system generates next predictions as soon as the user taps the desired one.

Teddy

Teddy [5] is a sketching interface for quickly and easily designing freeform 3D models such as stuffed animals and other rotund objects. The user draws several 2D freeform strokes interactively on the screen and the system automatically constructs plausible 3D polygonal surfaces. Our system supports several modeling operations, including the operation to construct a 3D polygonal surface from a 2D silhouette drawn by the user: it inflates the region surrounded by the silhouette making wide areas fat, and narrow areas thin. Our informal user study showed that a first-time user typically masters the operations within 10 minutes, and can construct interesting 3D models within minutes.

Figure 5: Teddy in use (left) and 3D models created using Teddy (right).
The user can construct these 3D models simply by drawing silhouettes without using any menus or control points.

Flatland

Flatland is an augmented whiteboard system designed to assist informal activities in one's office environment [8]. The system is designed to provide additional computational support to physical whiteboards without damaging the original ease of use. As the user draws strokes on the board, the system automatically groups adjacent strokes together, and allows the user to organize the board flexibly based on the groups. When necessary, the user can attach application behavior to the surface. These applications observe the user's stroking operations, and provide appropriate feedback, without requiring explicit command operation. The system automatically records entire operation history, and allows the user to retrieve them based on temporal or contextual keys.

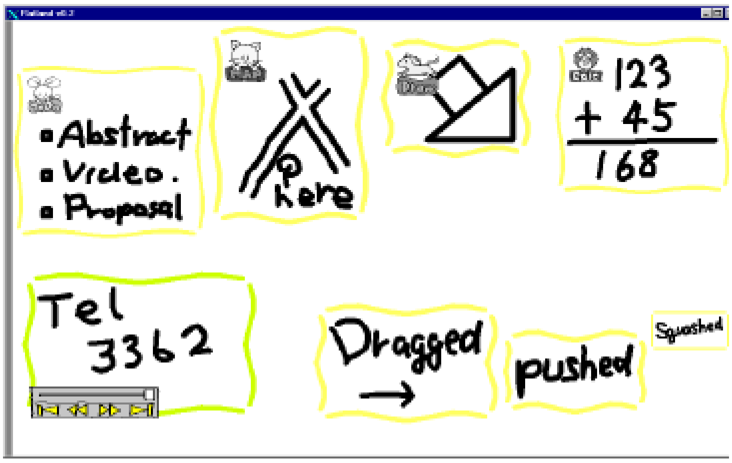


Figure 6: An example screen snapshot from Flatland.
The user can attach an application to each working area. This example shows to-do list, map, geometric drawing, and hand-writing calculator.

4. Discussions

The design of supportive user interface systems is much more challenging than traditional command-based systems. In command-based systems, the application programmer simply programs an action for each command. The number of commands is limited, and the relationships between a command and corresponding action is straightforward. However, in supportive user interface systems, the application programmer has to program actions for arbitrary user activity on the target material. In our stroke-based applications, the programmer has to process freeform strokes of infinite variety. In addition, the system's action should be context dependent. Same user input can have different meaning depending on the surrounding context [1]. Supportive user interfaces are also highly user dependent. Desirable action can vary among different users. Thus, explicit customization or automatic adaptation mechanism should be provided. Our prototype systems are currently designed for authors themselves, and we expect that it requires significant development efforts to make it usable for general users.

5. Conclusion

We introduced the idea of supportive user interfaces, and three example application systems based on the idea. In supportive user interface systems, the computer works as a personal assistant working together on the subject material, rather than the user manipulates the material by sending commands to the computer. We believe that supportive user interfaces allow the user to concentrate on the essential problems instead of command operations, and thus facilitate the flexible, exploratory activity in creative tasks.

References

1. Gross,M.D., Do,E.Y.L. Ambiguous intentions: A paper-like interface for creative design. Proceedings of UIST'96, pp. 183-192, 1996.
2. Igarashi,T., Freeform User Interfaces for Graphical Computing, Doctoral dissertation, the University of Tokyo, 2000
3. Igarashi,T., Matsuoka, S., Kawachiya, S., Tanaka,H. Interactive Beautification: A Technique for Rapid Geometric Design, Proceedings of UIST'97, pp.105-114, 1997.
4. Igarashi,T., Matsuoka, S., Kawachiya, S., Tanaka,H., Pegasus: A Drawing System for Rapid Geometric Design, CHI'98 summary, pp.24-25, 1998
5. Igarashi,T., Matsuoka,S., Tanaka,H., Teddy: A Sketching Interface for 3D Freeform Design, SIGGRAPH 99 Conference Proceedings, pp.409-416, 1999.
6. Lakin,F., Wambaugh,J., Leifer,S., Cannon,D., Steward,C., The electronic notebook: performing medium and processing medium, Visual Computer, Vol.5, pp.214-226, 1989.
7. Landay,J.A., Myers,B.A. Interactive sketching for the early stages of user interface design. Proceedings of CHI'95, pp. 43-50, 1995.
8. Mynatt,E.D., Igarashi,T., Edwards, W.K., LaMarca, A., Flatland: New Dimensions in Office Whiteboards, Proceedings of CHI'99
9. Nielsen,J. Noncommand User Interfaces. Communications of the ACM, Vol.36, No.4, pp.83-99, 1993
10. van Dam,A. Post-WIMP User Interfaces, Communications of the ACM, Vol.40, No.2, pp. 63-67, 1997.
11. Zeleznik,R.C., Herndon, K.P., Hughes,J.F. SKETCH: An interface for sketching 3D scenes. SIGGRAPH 96 Conference Proceedings, pp. 163-170, 1996.