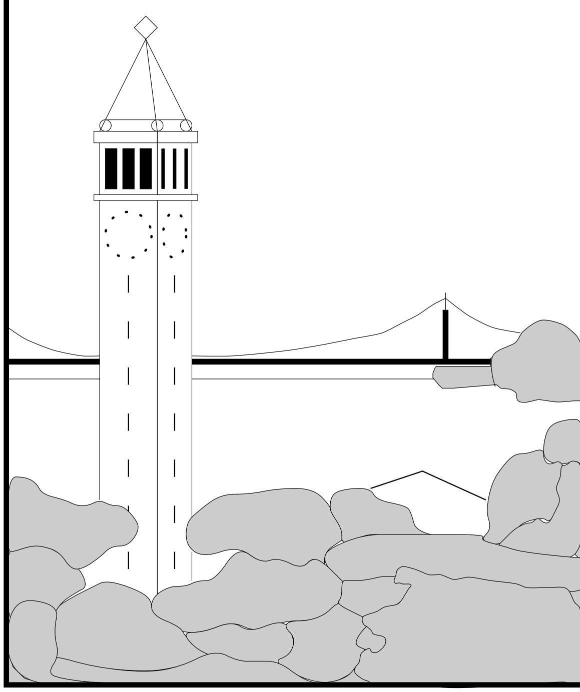
Incremental Sketch Understanding for Intention Extraction in Sketch-based User Interfaces

Li, Yang



Report No. UCB/CSD-3-1284

October 2003

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Incremental Sketch Understanding for Intention Extraction in Sketch-based User Interfaces[1]

Yang Li
Group for User Interface Research, Computer Science Division
University of California, Berkeley
Berkeley, CA 94720-1776 USA
yangli@cs.berkeley.edu

ABSTRACT

The inherent ambiguity of sketch-based user interfaces makes the intention extraction process quite different from traditional user interfaces. It is a critical problem of how precise computers can efficiently understand and naturally tolerate ambiguous sketch-based interactions. This paper proposes incremental sketch understanding. Based on the cognitive attributes of humans, a software framework is designed for incremental sketch understanding that is demonstrated by a note structuralizing application.

Keywords

Sketch-based user interface, cognition, incremental sketch understanding, note structuralizing

INTRODUCTION

Sketch-based user interfaces allow users to naturally express their ideas with freeform writing and drawing. With the support of computers, more functions can be enabled than with pen and paper. The more semantics of sketches that computers can discern, the more functionality computers can provide.

It is difficult for users of sketch-based user interfaces to let computers know what they are sketching. The poor accuracy of recognition engines is frustrating. The computer-centered interaction control is designed to force the user to accomplish the designated tasks in a certain region or at a certain time, e.g. box-based segmentation for continuous handwriting and maximum-interval-based multi-stroke gestures.

[1] This work was supported by the National Science Foundation under Grant No. IIS-0205644.

In most situations, users know exactly what they are sketching. However, it is quite difficult and impossible to ask computers to completely understand various sketches. Psychologically, as a mirror of thinking, sketching on paper is an efficient way to enhance our working memory and ease abstract thinking. It is not necessary for computers to completely understand various sketches [2]. Most sketch-based user interfaces are more prone to present multiple interpretations to the user. We propose that computers partially and incrementally share their understanding to match the cognitive habits of humans and to facilitate the processing of computers. Incremental sketch understanding intends to fulfill the computability of freeform sketches without requiring extra cognitive load on users.

Note taking is a universal activity used for idea or event capture. To enable semantics-based note manipulations rather than interactions with single stroke, note structuralizing is the precondition. We demonstrate incremental sketch understanding in the context of note structuralizing.

INCREMENTAL SKETCH UNDERSTANDING

In contrast with traditional user interfaces, sketch-based interactions are continuous, ambiguous and implicit. The information collected and analyzed is generally reversible and nonlinear. The process of incremental sketch understanding can be considered as a negotiation between the user and the computer. However, this negotiation is supervised by the user, i.e. it is a human-centered interaction control, and it is conducted implicitly.

Human cognition is performed incrementally and iteratively. It is based on current observations and past experiences. Likewise, in incremental sketch understanding, a parsing engine incrementally and continuously collects and analyzes users' sketches and automatically adjusts the parsing context based on history. User mediation is highlighted in incremental sketch understanding and it not only enhances the feeling of participation but also relieves the parsing engine. Implicit or explicit feedback by the user interactively refines the parsing result.

The parsing engine and the user are two independent threads that communicate with each other in an asynchronized manner. The user does not need to wait for the parsing engine to show all parsing results and the parsing engine does not need to wait for the user to finish all inputs before parsing. Users can either accept or ignore the feedback of the parsing engine. Users can also ask the parsing engine to synchronize its understanding with the user's. The computation of the parsing engine frequently lags behind or is ahead of the user's actions, e.g. when the engine needs to collect more input to give parsing results or when it presents anticipated user actions.

SOFTWARE FRAMEWORK

Based on human cognitive mechanisms [1], we designed a software framework, as shown in figure 1, to support incremental sketch understanding.

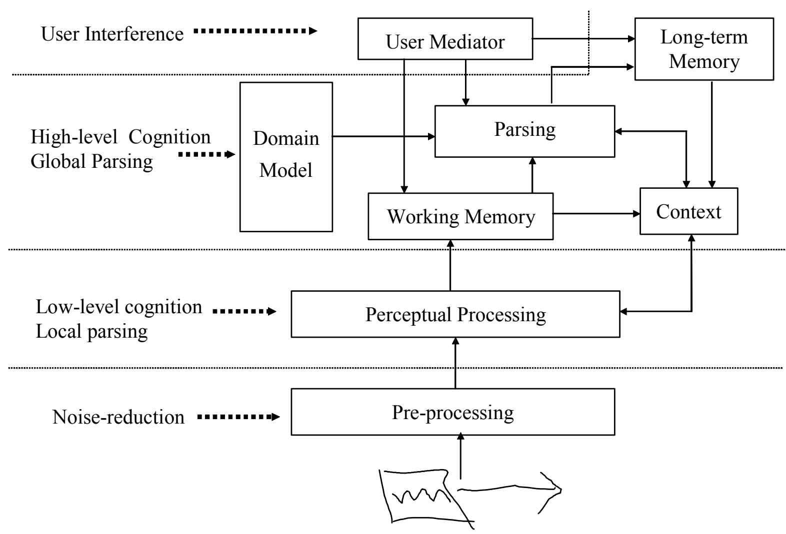


Figure 1: software framework for incremental sketch understanding. Arrows indicate internal data flows for parsing purpose.

Users' sketches are delivered to a perceptual processing component after noise reduction. The perceptual processing is low-level cognition, which collects the basic features of strokes, e.g. the bounding box or local spatial relationships under certain interaction contexts, e.g. identifying clusters of strokes.

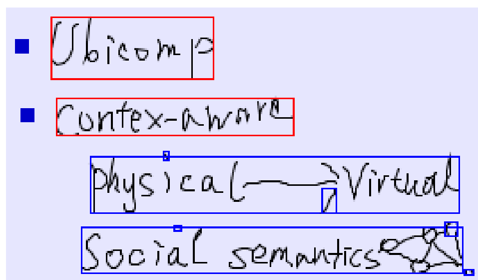
The results of perceptual processing, which is equivalent to what humans directly perceive from the visual system without thinking, are kept in the working memory. With the help of domain knowledge and the parsing history in long-term memory, the parsing engine can make global and semantics-rich interpretations of sketches. Parsing results are internally maintained by long-term memory, which will contribute to the context of later parsing.

Low-level parsing and high-level parsing indirectly affect each other through the context component. Parsing globally under the context of a parsing history will enhance the tolerance of locally ambiguous interactions. The user mediator is the representative of the user in parsing and it enables the user to give feedback on the parsing process.

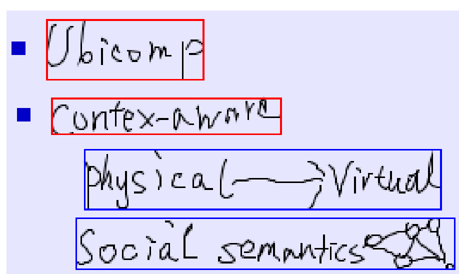
NOTE STRUCTURALIZING

We demonstrated the framework in a note structuralizing interface for an electronic notebook [3]. Structuralized notes are easy to maintain and manipulate.

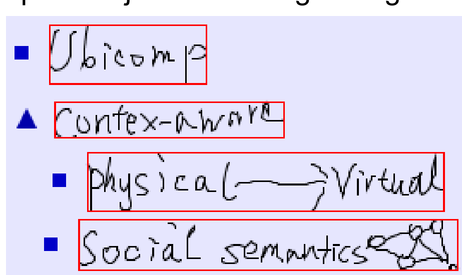
The example shown in figure 2 shows the entire process of note structuralizing in three steps. The basic units that the engine can discern are strokes. Based on the features of strokes (e.g. curviness) and the spatial relationships between them (e.g. adjacency or containment), these strokes are grouped into clusters as seen in figure 2a. Influenced by high-level cognition, the perceptual processing can adjust its results. As shown in figure 2b, the small fragmented clusters of figure 2a are merged into adjacent clusters. These two steps are carried out by the perceptual processing and all of the produced clusters are stored in working memory for further parsing.



(a) perceptual processing groups strokes into clusters



(b) perceptual adjustment merges fragment clusters



(c) high-level cognition parses the structure semantics of sketches

Figure 2: example of note structuralizing. Clusters with red boxes and semantic decorations are internally stored in long-term memory. Other clusters are stored in working memory.

The structure semantics of “list” and “title-list” are kept in a domain model. The engine conducts its parsing based on current observations, i.e. clusters in working memory, the knowledge of structuralizing, i.e. the structures in domain model and its memory, i.e. parsed results (history) in long-term memory. Those parsed sketches are decorated with lightweight semantic feedbacks, e.g. solid blocks and triangles in figure 2c, which are used to manipulate notes, e.g. for collapsing or expanding lists.

The high-level parsing is invoked when it is necessary. Users also can intentionally trigger it by tapping the notes, which hints to the engine that “My sketches are complete. What can you provide?” For incorrect results, users can explicitly correct them or just ignore the results and continue to sketch. The parsing engine will refine its parsing incrementally and implicitly.

CONCLUSION

Efficient understanding of users’ sketches is the foundation to fulfill the computability of informal sketches. We proposed the idea of incremental sketch understanding and devised a software framework that was demonstrated in the context of note structuralizing.

REFERENCES

1. Kieras, D.E. and Meyer, D.E. An overview of the EPIC architecture for cognition and performance with application to human-computer interaction, Human Computer Interaction, vol.12, pp.391-438, 1997.
2. Landay, J.A., Hong, J.I., Klemmer, S.R., Lin, J. and Newman, M.W. Informal PUIs: No Recognition Required. In Proceedings of AAAI 2002 Spring Symposium (Sketch Understanding Workshop). Stanford, CA 2002, pp. 86-90.
3. Li, Y., Guan, Z., Wang, H., Dai, G. and Ren, X. Structuralizing Freeform Notes by Implicit Sketch Understanding, In Proceedings of AAAI 2002 Spring Symposium (Sketch Understanding Workshop). Stanford, CA 2002, pp.91-98.