Computing and Informatics, Vol. 29, 2010, 585–600

MODELING SKETCHING PRIMITIVES TO SUPPORT FREEHAND DRAWING BASED ON CONTEXT AWARENESS

Cuixia Ma, Shengfeng Qin, Hongan Wang, Guozhong Dai

Institute of Software
Chinese Academy of Sciences
South Fourth Street 4, Beijing, China
e-mail: cuixia@ios.cn

Revised manuscript received 24 March 2010

Abstract. Freehand drawing is an easy and intuitive method for thinking input and output. In sketch based interface, there lack support for natural sketching with drawing cues, like overlapping, overlooping, hatching, etc. which happen frequently in physical pen and paper. In this paper, we analyze some characters of drawing cues in sketch based interface and describe the different types of sketching primitives. An improved sketch information model is given and the idea is to present and record design thinking during freehand drawing process with individuality and diversification. The interaction model based on context is developed which can guide and help new sketch-based interface development. New applications with different context contents can be easily derived from it and developed further. Our approach can support the tasks that are common across applications, requiring the designer to only provide support for the application-specific tasks. It is capable of and applicable for modeling various sketching interfaces and applications. Finally, we illustrate the general operations of the system by examples in different applications.

Keywords: Sketch, interaction model, context awareness

1 INTRODUCTION

Sketches are useful for conceptual design and thinking record. Sketch-based interface provides the new method for users to design naturally and efficiently. Users can focus on design tasks following their operation habits which are similar with those in pen and paper; and users find freehand sketch less cognition load to draw what they want [1]. Sketch-based interface is one of new generation natural user interfaces, which can break the bottleneck of current WIMP interface and adopt the nature and efficiency of pen and paper based interface.

Intelligence in a sketch based interface enhances its interaction ability in terms of context awareness. Systems can adjust behaviors based on different context, context change and context history to increase the conversational bandwidth between human and computer [5, 6] and support the individual interaction. For the communication between human and human, they are able to use different situational information, or context, to increase the conversational bandwidth. Unfortunately, this ability to convey ideas does not transfer well to humans interacting with computers. In most systems, computers are not enabled to take full advantages of context awareness in interaction application such as a sketch-based interface for conveying ideas and communication. There is a lack of general sketching and interaction model to address sketch description, interaction mechanism, intelligence and context-sensitive interaction with humans.

There have been many applications developed based on sketch based interface. A survey was performed about the sketching functions shared in the applications [2], which can benefit the description of functions for sketch based interface. Mainly, there are two aspects about the research of sketch based interface, including: 1) recognition and model of sketch, like preprocessing, segmentation, classification, parsing and so on; 2) specific applications, using the domain knowledge, many applications have been developed from different domain, like conceptual design, conceiving thinking, music pad, math recognition, animation, hair style, garment design, floral modeling, etc. [3, 4]. For sketch based applications, they share some common functions, including:

- sketch as input
- pen gesture as command
- time and space as indexing of strokes
- ambiguous input
- interpreters that act on strokes due to the context
- understanding of sketch to other semantic objects
- continuous interaction and feedback
- typical characteristics of individuality
- informal sketch output.

The main idea behind sketch-based interfaces is to mimic pen and paper that represents a natural way of thinking about ideas and communicating them. With real pencil-and-paper sketching, in the initial stages of the design process a user will often faintly sketch primitive shapes to define the overall form of an object, and then use many small strokes to complete the sketch. Users also prefer to emphasize the content or take any annotation using sketching marks for improving later organization; but support for natural sketching with drawing cues, like sketching primitives of overlapping, overlooping, hatching, etc. are lacking in current sketch based interfaces [1].

In this paper, we describe certain sketching primitives for enhancing users' cognition and improving users' experience using sketching and gestures. The sketch information model and interaction model are presented to record the sketching primitives and gestures. We will not only improve the understanding of general features of sketch-based interface and modeling, but also provide a common ground for research and development by addressing the architectural issues needed to capture context and deliver it to interested consumers in ubiquitous computing environment.

2 RELATED WORK

Sketching is a fundamental tool to express design intent of users. The research on it can be dated back to the early 1960s when Ivan Sutherland published his seminal work on the Sketchpad [7]. With the development of pen device and interactive technologies, sketch based interface can be used in many application domains. Sketch representation and sketch understanding are two of the key technologies in order to improve the human-computer interaction in sketch based interface.

ASSIST(A Shrewd Sketch interpretation and Simulation Tool) interprets and understands a user's sketch as it is being drawn, and provides a natural-feeling environment for mechanical engineering sketches, in which to represent and user context to resolve ambiguities [8, 9]. Users input additional sketching on the former sketching object for some certain recognition. Architectural drawings and building depictions have been the subject of much work. A comprehensive sketch-based architectural modeling package is provided to develop 3D building models from conceptual sketches [10, 11]. Sketch-based interfaces have also been used for virtual garment design [12, 18], whiteboard [13, 17], note-taking [14, 16], annotation [19] and so on. Besides sketching, widget of supporting for operation also provides the natural interaction [15], which is another research challenge in future.

There are also many definitions of context [20, 21]. If a system is context-aware, it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task. A classic example of a context-aware application is a tour guide that provides relevant information and/or services to a user based on its location [22].

Although much work on context-awareness has been conducted in the pervasive computing area, there is less work existing in sketch-based interface and modeling research field. One of the main reasons why context is not used more often in sketch-based applications is that there is no common way to acquire and handle context. The ways of using context information in user interface can be divided into two broad categories: context-dependent and context-sensitive user interfaces.

A context-dependent user interface is specific for a context, but does not adapt to context. Context sensitive user interfaces adapt to the context and react to changes in context. Most of existing sketch interfaces are context-dependent. For example, in sketch-based modeling systems like SKETCH and TEDDY, gestural sketching interprets freehand sketches in a specific way so that some sketching gestures actually mean 3D reconstruction commands, besides being the objects profile lines [23].

Geometric correlations approach [24] introduced an optimization based 3D reconstruction approach based on a set of geometric correlations between a 3D object and its projection on a 2D plane. Perceptual approach [25] reconstructs 3D objects from line drawings along the identified principal axis in the axonometric view. All of these systems integrate a set of pre-defined static context to produce 3D models from 2D sketching and interaction. Static context integration means that the context is consulted before a user interface that uses this information is presented. Sketch-based interface has been used for many different applications. All these systems have application-specific context.

3 MODELING SKETCH AND CONTEXT

We analyze the drawing and interaction features in sketch based interface and describe sketching primitives. We develop sketch and interaction models based on context. Based on the model, the system can identify context types and examine how applications can effectively use context information through context-aware functions, such as sensing, interpreting and fusing context, presenting information and service to the use or using context to propose appropriate selections of actions to the user, automatically executing a service that triggers a command or reconfigure the system on behalf of the user according to context changes.

3.1 Sketch Primitives

During the process, thinking prefers to be divergent for versatile possibilities, then getting into convergence at the end in which status changes continuously. However, in most systems based on WIMP metaphor instead of pen and paper, users' train of thought is easily interrupted by excessive selections of menus, operations on buttons and keyboard input. The kind of sketch interaction is continuous rather than purely discrete. Meanwhile, context integrated in the process can benefit for intelligent interaction.

Designers often work with freehand sketches to quickly communicate since free form sketching is more intuitive than the formal shapes to lead to more design alternatives. In order to facilitate the design process, modeling the sketching process can improve the human–computer interaction. There are some characteristics for sketching which are different with those of WIMP interface. First, sketching using a pen is still more natural than using keyboard and mouse with frequent interruption. Second, the sketching can reflect individual and diversiform drawing style of users compared with those in WIMP interface (Figure 1).

Continuous sketching provides continuous experience of input and feedback for better communication and working through the problems. The hierarchy structure

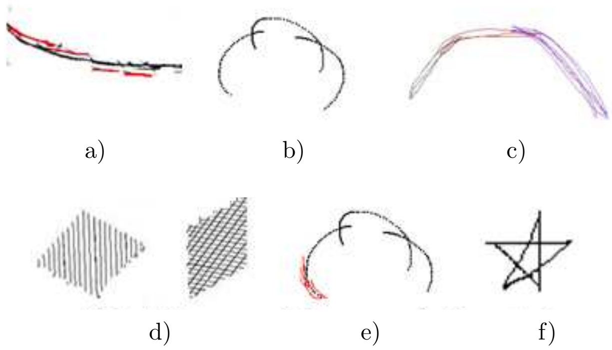


Fig. 1. a) one skeleton and some small or additional strokes (in red) to complete one strokegroup object-overlapping; b) more than one strokes for the same arc; c) drawing the 3 sub-arcs with overlapping, for each arc, you may draw a stroke with forward and backward motions, as is overlooping one. d) the hatching is the use of fine, parallel lines drawn closely together; e) an example to show the composition of two kinds of strokegroups; the overlapping and the joint are combined together. f) a pentacle mark shown here to highlight the content.

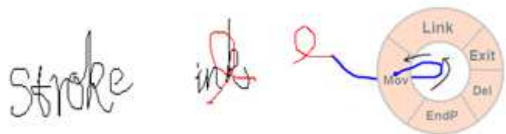


Fig. 2. a) sketching over the ink object leads to delete operation; b) the same gesture on different location and context provides the different function; it triggers one pie menu for further operation

of sketch is given based on the incremental sketching input process, including sketch, strokegroup and stroke.

Stroke can be structured following the operation of pen down and up with related attributions, involving time-stamping, space relationship, direction, inflexion, velocity, etc. Strokegroup consists of different drawing style types. Some typical features are provided in this paper. Different types of strokegroup can be assembled together for composed objects. We define a basic unit of a sketch as a drawing stroke with different context involving constraints, operation history, user context or interaction context A drawing stroke (line) is a set of sequential points captured (sensed) from an input device during a time period between a pair of pen-down and pen-up events. This structure can also be expanded to support more sketching types.

Pen gestures can be seen as simple sketch or strokegroup, whose structures are similar with those of sketch. Gestures are implemented in terms of contexts during the freehand drawing process. For example, a set of strokes covering some certain sketches can be recognized as delete gesture, while, if the strokes are drawn far from sketches, maybe it will trigger a pie menu (Figure 2).

We present the BNF paradigm to describe the sketching primitives which include the structure of sketch, strokegroup and stroke shown in this section.

$$
\begin{array}{rcl}
\langle Sketch\rangle & ::= & \{\langle sketch\rangle\}\{\langle StrokeGroup\rangle\}[Constraints]\langle Operations\rangle \\
\langle StrokeGroup\rangle & ::= & \{\langle StrokeGroup\rangle\}\{\langle Stroke\rangle\}\langle RecognizedResult\rangle \\
 & & \langle BoundingBox\rangle\langle Operations\rangle[Constraints]\langle GroupType\rangle \\
\langle GroupType\rangle & ::= & \langle Overlapping\rangle\backslash\langle StrokeGroup\rangle\}[Constraints]\langle Operations\rangle \\
\langle GroupType\rangle & ::= & \langle Overlapping\rangle\backslash\langle Overlooping\rangle\backslash\langle Hatching\rangle\backslash\langle joint\rangle\backslash \\
 & & \langle mark\rangle\backslash\langle Custom\rangle \\
\langle Stroke\rangle & ::= & \langle Points\rangle\langle DownTime\rangle\langle UpTime\rangle\langle Velocity\rangle\langle Inflexion\rangle \\
 & & \langle Direction\rangle\langle Operations\rangle \\
\langle Points\rangle & ::= & \{\langle Point\rangle\}\langle Time\rangle \\
\langle StrokeType\rangle & ::= & \langle Main\rangle\backslash\langle Additional\rangle \\
\langle Operations\rangle & ::= & \langle Move\rangle\backslash\langle Rotate\rangle\backslash\langle Zoom\rangle\backslash\langle Delete\rangle\backslash\langle Copy\rangle\backslash\langle Mirror\rangle \\
\langle Constraints\rangle & ::= & [GeometryConstraints]\backslash[DomainConstraints]\backslash \\
 & & [ContextConstraints]
\end{array}
$$

The corresponding hirarchy sturcture is given below to show the sketching primitives (Figure 3).

When designers perform their conceptual design with sketching, usually first a rough sketch stroke is drawn, which shows the overall characteristics of the lines to be drawn; this stroke is then refined to work out the local characteristics of the lines in terms of their width and brightness.

Therefore, some strokes need to be grouped together to have a collective meaning. Figure 4 gives an example; several strokes together represent an edge of an object [26]. Figure 4 a) illustrates initial input sketches which were grouped into 10 parts. The paths of each part are a straight line for an edge and their edge styles are captured. Furthermore, 2D constraints such as incidence and parallelism were obtained. The 2D tide-up gave a vertex-edge graph and the 3D understanding produced a 3D object (Figure 4 b)). After knowing 3D positional information for each group path, the brightness of rendered edge was updated after a rotation (Figure 4 c)). The closer to the view point the point is, the darker the rendering strokes are.

By incorporating the interaction model into the sketch model, a unified operational model of sketching has been developed to support object oriented operations. This data can be exchanged easily for different applications. The figure classes can be shown (Figure 5).

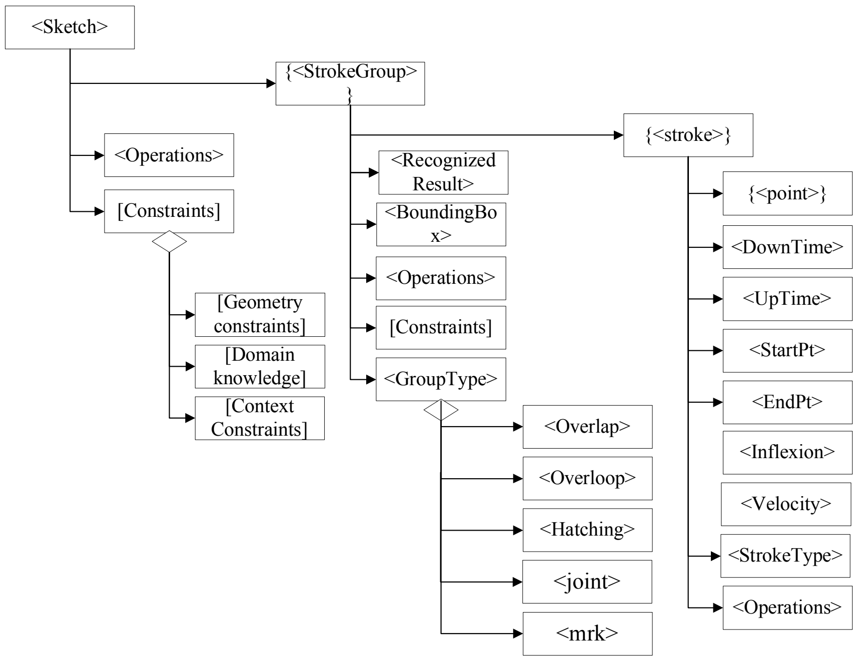
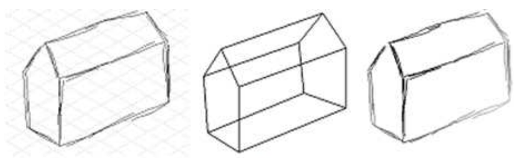


Fig. 3. Sketch information model



a) b) c)

Fig. 4. Sketch example

3.2 Context Aware Interaction Model

The proposed sketch model is focused on its data structure and hierarchy. That model has to be connected with its interaction model with different context so that a sketch-oriented object can be developed to support various operations and applications. Context is obvious and important in many environments. It is not easy to provide the exact definition for context in computing environments. Here, we focus on a set of state and variables which present information related with current application in sketch based interaction, as can be classified from the point of view of users and systems.

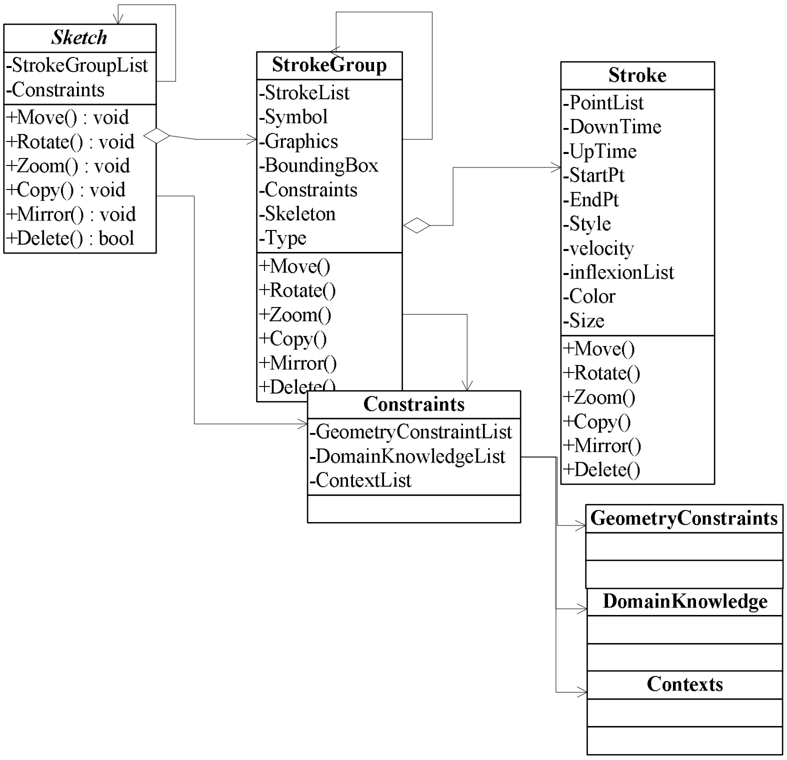


Fig. 5. Class structure relationship

The input environment context will include types and characteristics of input devices, technical parameters of the input devices such as resolutions and sensitivity to pressure, rendering stroke style, trade-off of brightness or width of line, paper or window sizes. The user context will affect the user-machine interaction including the user's design experience, drawing skills, preference, sketching and thinking modes, hand motions, and ROI (Region of Interest of eyes). The user context also links to the user's visual perception on input sketches, the user's design intent (design context) and typical ways of interacting with the interface.

From the user's input, the interface can obtain the user's choices and selections on menus or buttons; it can also get machine-sensed raw input context such as colors, positions of a stroke and sequences of points and strokes and time. Based on this, the interface can get machine interpreted context such as drawing speed (and pressure) and acceleration at each point, types of strokes such as overshooting, over-tracing and overlapping, drawing paces or rhythm. The speed information can be used to determine rendering style and the acceleration can be used to segment drawing strokes.

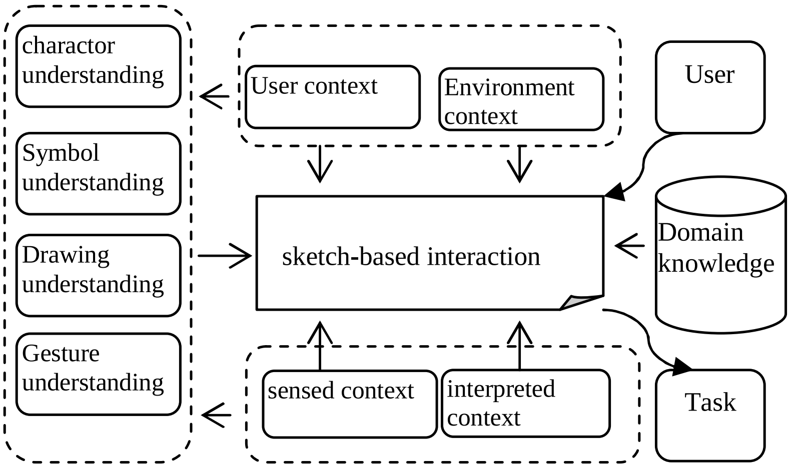


Fig. 6. Interaction model with context

The appropriate view of context contained in a design process depends on its application. In this system, context includes:

- *user context*: experienced designers or novice;
- *environment context*: device, pen locations, working mode and certain containers;
- *machine-sensed context*: time, sequence, color and positioning information.
- *machine-interpreted context* (first level derived context from sense context): classification of texts, symbols, drawings, gestures, etc.; (second level derived context): geometry, topological relations, and constraints.

The first constructor is used to draw a single-pixel-wide solid line. The second one specifies a constant thickness. The last three constructors require the settings for cap and segment joining. When a path is unclosed, then a cap is used on both ends of the path. If the path is explicitly closed, then segment joining is used between the last segment and the first one. Actually, the width of a drawing stroke is varied at each point depending on its drawing pressure or speed. In general, the higher the drawing pressure (slower speed), the wider the line is.

Based on machine sensed context, the interface can segment and classify the input contents as texts, symbols, edge drawings and gestures. This is the machine-perceived context. The interface can then call different application agents for performing different tasks with machine-sensed and interpreted context to understand input sketches and gestures, and to produce and present results to the user for interaction. The intelligent applications (or tasks) may include capturing 2D and 3D constraints and concurrent editing, understanding of symbols or gestures. The applications can be regarded as high-level interpreted context resulting from multiple context-fusions.

4 ILLUSTRATION AND EVALUATION

Based on the sketch model and interaction model, we can provide some application demonstrations about the sketch based interface and context understanding. During a design process, the status can change continuously with users' intention following the different context.

There are two kinds of application based on sketching and gesturing operations in this part. One is about the process planning design of knitting garment. It provides the sketching garment parts design, which supports the process planning formula input and calculation. Another is about the video editing, which supports conceptual design of video clips re-organization and annotation.

Symbol understanding and drawing understanding with domain knowledge. We have developed a sketch-based interactive process planning system, which supports sketching input, editing, and provides new methods to input and organize formulas and variables (underlying knowledge and experience) and to support requirements for flexible interaction [27] (Figure 7).

The system allows users to input their own knowledge into the database, which includes process planning formulas and experience variables in domain. Formulas can be input by drawing different symbols with ease. Experience variables can be input in the same way. The formulas they produce can in turn be added to the knowledge base for further reuse and reediting. If a skilled worker leaves, the loss will be prevented because his/her design knowledge and experience will have already been saved in the system. At the same time, novices who have not been well trained for the job can obtain much more experience and knowledge through the software.

To evaluate the usability performance of the sketch based interactive process planning system, we conducted one simple experiment. We selected 4 people to participate in our user experience study at the same workplace, including 2 novices and 2 skilled designers, to implement the same task by the traditional method and by the sketch based interactive process planning system. By using the system, the task time is shortened. In comparison to the traditional method, users spend much time on ideation of the process planning with this sketch based interface, like sketching and editing formulae in the system directly based on garment styles, instead of on the calculation and results checking. Users can obtain and modify formulae easily through sketching and gestures work well to reduce their cognitive load during the process. Feedback was positive as users listed the benefits of the system with sketch based interface. They found it to be natural and fluent to use during the planning process. And based on some feedback and requirements from the technologists, they have the good view about the garment for better decision

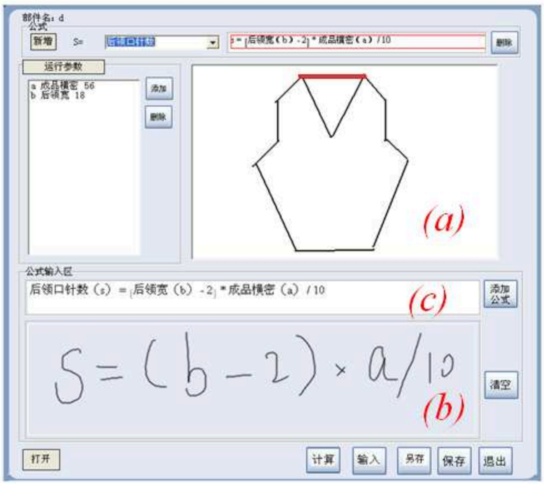


Fig. 7. garment sketch and formulas input interface with the knitting domain knowledge. a) In this area, what users draw on it will be recognized as geometry graphics to represent the garment sketch. b) What users draw on it will be recognized as letters or symbols to represent the formulas for calculating the process planning of knitting garment. Meanwhile, the formulas in b) reflect the calculation on the corresponding part of the drawing in a). Both are synchronous. c) The semantic representation is also given to facilitate users' understanding.

of next step, such as reduce the cost and improve the efficiency. A sketch based interface for video operation is under development. The flow chart using sketching is used here to present the organization of video clips. Users can design and re-organize different video clips to implement one whole video based on the sketching flow chart. It is easy and convenient to indicate the order of video clips. Annotations about the video clips can be taken using sketching and are integrated into the flow chart (Figure 8).

Users can take the annotations on the frames of video using sketching. Free form visual forms such as symbols and hand-drawing illustrations are used in the system. There are descriptive ion annotation and behavioral annotation (Figure 9).

Gesture understanding and continuous interaction. Gestures issued with freehand sketch strokes have one-to-one semantic mappings with the application. The semantics in the system include, for instance, creation, selection, deletion, modification, and constraint confirmation and cancellation operation. The user can sketch out gestures to implement the corresponding functions [28].

Gestures are typical commands in sketch-based interface, interpreted from the users' drawn marks. Commands and operations can be specified in gestures, which make the system faster than using menus and icons.

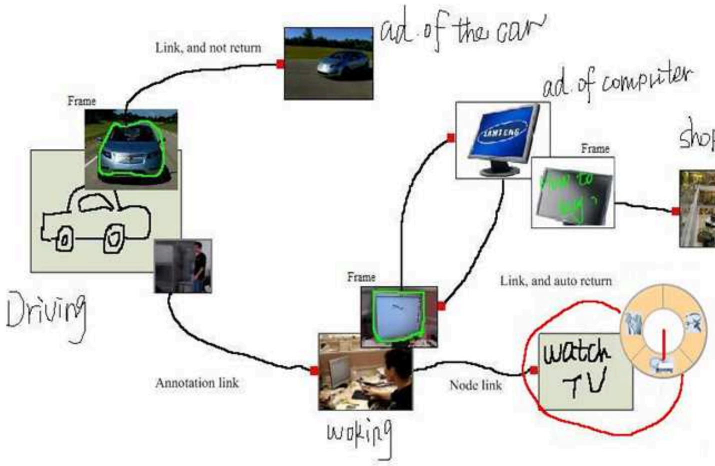


Fig. 8. Sketch based interface for video editing

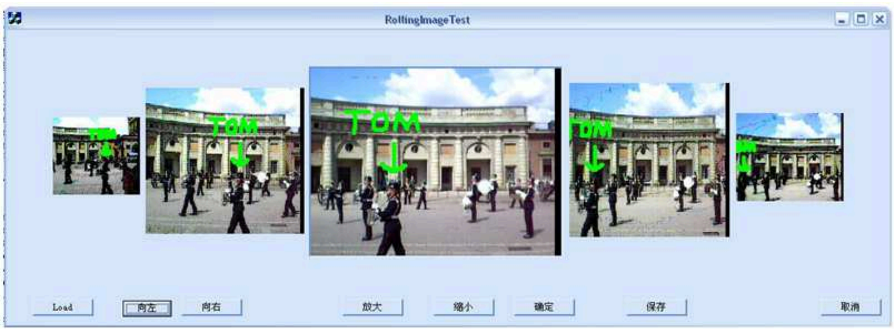


Fig. 9. Description annotation example

Continuous interaction differs from discrete interaction which are events driven that it takes place over a relatively longer period of time in which there is an ongoing relevant exchange of information between the user and the system. The user's continuous input goes with continuous visual feedback, which provides full interaction information through a user-friendly interface with reduced cognitive load.

Sketching not only enables continuous input in process, but also provides continuous user experience of input and feedback for better communication. Users will behave naturally as human beings without adopting a simplified behavior that characterizes a state of technological awareness. Consequently, it is important for the system to adapt to users, to be aware of their operating context, and to be able to take autonomous decisions to some extent.

5 CONCLUSION AND FUTURE WORK

Sketch based interface provides the operations of support for natural sketching with its characters which are different with those of WIMP based interface. How to specify the characters and make benefit for certain applications are popular research points. This paper presents the description of sketching primitives in order to improve user experience in sketch based interface. A framework of modeling sketching interface and interaction based on context are given. These pieces of context have been embedded into the proposed sketch and interaction models. The models can guide and help new sketch-based interface development and support object-oriented operations. From a software engineering point of view, the interaction model can be implemented as a basic class with a set of standard operational functions. New applications with different context contents can be easily derived from it and developed further.

However, the range of sketching primitives is limited for natural sketching requirements. Further work will be done to enhance the operations. We would like to improve the sketch information model based on the deep analysis of sketching primitives which support different applications. The framework will automatically support all the tasks that are common across applications, requiring the interface designer to only provide support for the application-specific tasks. It also potentially provides a sketch-based application data exchange platform for design collaboration.

Acknowledgement

This work was supported by the National Key Basic Research and Development Program under Grant No. 2006CB303105 and the National Natural Science Foundation of China under Grant No. 60703079.

REFERENCES

[1] Olsen, L.—Samavati, F. F.—Sousa, M. C.—Jorge, J. A.: Sketch-Based Modeling: A Survey. Computers and Graphics, Vol. 33, 2009, pp. 85–103.

[2] Hong, J. I.—Landay, J. A.: SATIN: A Toolkit for Informal Ink-based Applications. Proceedings of the 13^{th} Annual ACM Symposium on User Interface Software and Technology (UIST 2000), pp. 63–72.

[3] An Introduction to Sketch-Based Interfaces. SIGGRAPH 2006 Course.

[4] Sketch-based interfaces: Techniques and Applications. SIGGRAPH 2007 Course.

[5] Abowd, G. D.—Mynatt, E. D.: Charting Past, Present, and Future Research in Ubiquitous Computing. ACM Transactions on Computer-Human Interaction, Vol. 7, 2000, No. 1, pp. 29–58.

[6] Henricksen, K.—Indulska, J.: Developing Context-Aware Pervasive Computing Applications: Models and Approach. Pervasive and Mobile Computing, 2006, Vol. 2, No. 1, pp. 37–64.

[7] Sutherland, I. E.: Sketchpad – A Man-Machine Graphical Communication System. Proceedings of Spring Joint Computer Conference, 1963, pp. 329–346.

[8] Alvarado, C.—Davis, R.: Resolving Ambiguities to Create a Natural Sketch Based Interface. Proceedings ofIJCAI-2001.

[9] Davis, R.: Sketch Understanding in Design: Overview of Work at the MIT AI Lab. AAAI Technical Report SS-02-08. Compilation copyright 2002.

[10] Leclercq, P.: Invisible Sketch Interface in Architectural Engineering. Lecture notes in Computer Science, Vol. 3088/2004, pp. 353–363, Springer, Berlin 2004.

[11] Juchmes, R.—Leclercq, P.—Azar, S.: A Freehand-Sketch Environment for Architectural Design Supported by a Multi-Agent System. Computers and Graphics, Vol. 29, 2005, No. 6, pp. 905–915.

[12] Turquin, E.—Wither, J.—Boissieux, L.—Hughes, J. F.: A Sketch-Based Interface for Clothing Virtual Characters. IEEE Computer Graphics and Applications, January/February 2007, pp. 72–81.

[13] Moran, T. P.—Chiu, P.—Melle, W. V.: Pen-Based Interaction Techniques for Organizing Material on an Electronic Whiteboard. In Proceedings of UIST '97. Banff, Alberta (Canada), October 14–17, 1997, pp. 45–54.

[14] Davis, R. C. et al.: Lightweight Note Sharing by the Group, for the Group. Proceedings of CHI '99, Pittsburgh, PA, May 15–20, 1999, pp. 338–345.

[15] Schmidt, R.—Singh, K.—Balakrishnam, R.: Sketching and Composing Widgets for 3D Manipulation. Computer Graphics Forum, Vol. 27, 2008, No. 2, pp. 301–310.

[16] Davis, R. C.: NotePals: Lightweight Note Sharing by the Group, for the Group. In Proceedings of CHI '99, Pittsburgh, PA, May 15–20, 1999, pp. 338–345.

[17] Mynatt, E. D.—Igarashi, T.—Edwards, W. K.—Lamarca, A.: Flatland: New Dimensions in Office Whiteboards. Proceedings of CHI '99, Pittsburgh, PA.

[18] Ma, C. X.—Dai, G. Z.—Wang, H. A.: Perceiving the Invisible: Natural Interactive Collaboration Based on Sketch in Conceptual Design. The 7th Asia-Pacific Conference on Computer-Human Interaction (APCHI), October 2006, Taipei, Taiwan, pp. 11–14.

[19] Schilit, B. N.—Golovchinsky, G.—Price, M. N.: Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations. In Proceedings of CHI '98, Los Angeles, CA, April 18–23, 1998.

[20] Deya, K.—Salber, D.—Abowd, G. D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Hum.-Comput. Interact. J., Vol. 16, 2001, No. 24, pp. 97–166.

[21] Schlitt, B.—Theimer, M.: Disseminating Active Map Information to Mobile Hosts. IEEE Network, Vol. 8, 1994, No. 5, pp. 22–32.

[22] Brown, P. J.—Bovey, J. D.—Chen, X.: Context-Aware Applications: From the Laboratoryto the Marketplace. IEEE Pers. Comm., Vol. 4, 1997, No. 5, pp. 58–64.

[23] Qin, S. F.—Wright, D. K.—Jordanov, I. N.: From On-Line Sketching to 2d and 3d Geometry: A System Based on Fuzzy Knowledge. CAD (Computer Aided Design), Vol. 32, 2000, No. 14, pp. 851–866.

[24] Company, P.—Contero, M.—Conesa, J.—Piquer, A.: An Optimisation-Based Reconstruction Engine for 3d Modelling by Sketching. Computers and Graphics (Pergamon), Vol. 28, 2004, No. 6, pp. 955–979.

[25] Lamb, D.—Bandopadhay, A.: Interpreting a 3d Object from a Rough 2d Line Drawing. Proceedings of the 1st conference on Visualization '90,IEEE Computer Society Press, pp. 59–66.

[26] Kud, C.—Qin, S. F.—Wright, D. K.: A Sketch Interface for 3D Modeling of Polyhedrons. SBM '06. pp. 83–90.

[27] Ma, C. X.—Teng, D. X.—Wang, H. A.—DAI, G. Z.: A Sketch-Based Interface for Interactive Process Planning Design in Pervasive Computing Environment. Proceedings of the 2nd International Conference on Pervasive Computing and Applications, Birmngham 2007.

[28] Ma, C. X.—Dai, G. Z.—Teng, D. X.—Chen, Y. D.: Gesture-Based Interaction Computing in Conceptual Design. Journal of Software, Vol. 16, 2005, No. 2, pp. 303–308.

Cuixia Ma is Associate Professor at Institute of Software, Chinese Academy of Sciences. Her major research interests are sketch based interface, sketch based video processing and conceptual design. She graduated from Institute of Software, CAS and was a research associate of the Department of Computer Science in Naval Postgraduate School, USA (2005–2006). She has been principal investigator of projects sponsored by Natural Science Foundation of China and National Science and Technology Supporting Program of China. She has published more than thirty research papers in international academic journals and conferences.

Shengfeng Qin is Senior Lecturer at Brunel University, UK. His research interests include design method and computer graphics.

Hongan Wang is Professor at Institute of Software, CAS. His research interests include real time intelligence and human computer interaction.

Guozhong Dai is Professor at Institute of Software, CAS. His research interests include human-computer interaction and computer graphics.