概念设计中基于笔式手势的交互计算研究*

马翠霞+, 戴国忠, 滕东兴, 陈由迪

(中国科学院 软件研究所,北京 100080)

Research of Interaction Computing Based on Pen Gesture in Conceptual Design

MA Cui-Xia+, DAI Guo-Zhong, TENG Dong-Xing , CHEN You-Di

(Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-10-62540434, E-mail: macxia@hotmail.com, http://iel.iscas.ac.cn

Received 2003-03-23; Accepted 2003-07-02

Ma CX, Dai GZ, Teng DX, Chen YD. Research of interaction computing based on pen gesture in conceptual design. *Journal of Software*, 2005,16(2):303−308. http://www.jos.org.cn/1000-9825/16/303.htm

Abstract: Natural interaction based on pen gesture is efficient for innovation in conceptual design. This paper presents a hierarchy concept model for pen gesture design based on the interactive informal constraint sketch and context-awareness. Furthermore, it discusses the constraint foundation and solution. Feature gesture as an instance of gesture applications in conceptual design is given. Compared with the traditional modeling mode, the feature gesture is convenient. Gesture-based interaction gives a natural way to record the brain streaming quickly. This paper provides an efficient method for innovative design to improve human-computer interaction.

Key words: pen gesture; context-aware; constraint; feature gesture

摘 要: 基于笔式手势的自然交互是支持概念设计创新的有效方式.提出了一种笔式手势的层次概念模型,结合基于约束的自由勾画和上下文感知技术描述了手势设计方式,进一步讨论了手势内部的约束建立和求解算法;基于手势应用的范式,给出了概念设计中特征手势建模的应用实例,通过与传统建模和交互方式的对比,验证了特征手势建模的方便性.给出一个自然、高效的方法,以基于手势的方式来完成概念设计构思过程,采用笔式交互快速自然地记录下设计思路,所实现的系统方便了用户的创新设计,改善了人机交互模式.

关键词: 笔式手势;上下文感知;约束;特征手势

中图法分类号: TP391 **文献标识码**: A

*Supported by the National Grand Fundamental Research 973 Program of China under Grant No.2002CB312103 (国家重点基础研究发展规划(973)); the National Natural Science Foundation of China under Grant Nos.60033020, 60373056 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2003AA411330, 2002AA414010 (国家高技术研究发展计划(863))

MA Cui-Xia was born in 1975. She is an assistant researcher at Institute of Software, the Chinese Academy of Sciences. She received her Ph.D. degree from Institute of Software in 2003. Her research interests are human computer interaction and conceptual design. **DAI Guo-Zhong** was born in 1944. He is a professor and doctoral supervisor at Institute of Software, the Chinese Academy of Sciences. His research interests are human computer interaction and computer graphics. **TENG Dong-Xing** was born in 1973. He is a post- doctor at Institute of Software, the Chinese Academy of Sciences. His research interests are computer aided design and computer graphics. **CHEN You-Di** was born in 1933. He is a professor at Institute of Software, the Chinese Academy of Sciences. His research interests are computer aided design and computer graphics.

The interaction mode based on pen-paper metaphor is natural for users to express themselves between human and computer, which can provide an imprecise and quick sketching to implement the conceptual design[1,2]. To provide good tools and environment for the concept design will boost the designing efficiency, and thus enhance creativity. In most current design systems, designers are bounded in the WIMP interactive mode. They have to provide precise information and to operate according to the ordered steps of menu and icon.

Pen gestures are the important characteristic of pen-based user interfaces. In the following part, we will use abbreviation gesture for pen gesture. Command and operand can be specified in gestures, which makes them faster than menu and icon. The shape of gesture suggests its meaning, as it is easier to remember than textual commands. UC Berkeley made a survey that gestures are powerful and users would like to define more gestures. Specifically, users often find some gestures difficult to remember, and they become frustrated when computers misrecognize gestures[3]. In the paper, we provide a gesture interaction model. The method of gesture design is based on geometry level which provides a better recognition in terms of context-awareness. A netlike structure[4] is given to implement the spread and solution of the bidirectional constraints to improve the interaction. Finally it describes feature gestures by hyper-graph, which takes advantage of the feature and gesture to improve the efficiency in conceptual design.

1 Related Work and Hypothesis

Pen-based user interface is becoming more popular, versatile, and powerful. Some related works are as follows.

UC Berkeley develops two systems, Gdt and Quill, to improve the gesture set to be remembered and recognized easily. Gdt[5] provides visualizations to help designers discover and fix recognition problems. Quill[6] provides some better characteristics, such as the active feedback and similarity analysis. Because of the limitation of Rubin's recognition algorithm[7], they only support single-stroke gestures. They didn't give the gesture description and neglected the special pen-based interaction. The Fraunhofer Institute for Computer Graphics in Germany provides a 3D-sketch language for primitive creation and freeform objects[10]. It provides a simple BNF-grammar for the sketching language. But it is inconvenient to extend and design dynamically. The sketch system of Brown University processes 2D strokes while sketching on the plane to create predefined 3D primitives[8]. It is ambiguous while interpreting geometric properties of the sketched objects, such as type, position, alignment, size, etc. Teddy prefers to polygonal freeform surfaces from sketched 2D silhouettes[9]. Chateau introduces a new type of interface for 3D drawings——suggestive interface, as hints about a desired operation to the system by highlighting the related geometric components in the scene, infers possible operations, and presents results of the operations as small thumbnails.

Research hypothesis is given that gestures can be used efficiently as an independent resource like the menu and icon in WIMP interface. On the one hand, gestures are natural and fast. It is flexible to use gestures instead of menus and icons. On the other hand, gestures are created on the precise geometry information. So a better recognition would be provided through the effective algorithm and active feedback to achieve the process.

2 User Interactive Gesture Resource

2.1 Gesture design model based on constraints

We give a gesture design model to improve the designing of gestures. There are two aspects on which we have to focus[10]. One is that the gestures should be recognized reliably. The other is that the gesture should be easy for people to learn and remember. First, some definitions are given as follows.

Definition 1. An element is a stroke which constitutes a gesture by sketching, such as point, line, circle,

ellipse, arc and freeform figure.

Definition 2. A customized element is the strokes to constitute a gesture, including some close or special figures, such as polygon and so on.

Definition 3. Constraints and informal geometry constraints between strokes can be captured automatically.

Definition 4. A symbol is a stroke set composed of elements, customized elements and constraints, to constitute a gesture or a sub-set of a gesture.

Definition 5. Combination of constraints among symbols, such as contain, intersect, tangent, coherent and so on.

Definition 6. Object constraint model, based on the non-directed graph is given as $G=(V,E)$, $V=\{v_1,v_2,v_3,\ldots,v_n\}$; V is the vertex set and represents the symbols. $E=\{Con_1,Con_2,Con_3,\ldots,Con_n\}$; E is a directed edge set and represents the constraints among vertex, $Con_t=\langle V_p,V_t\rangle$; $v_t\in V$, is the target point of the directed edge; $v_p\in V$, the ordered n–elements, the pre-nodes on which the target point depends.

Definition 7. Gesture, command issued with a pen, has the semantic mapping with operations.

$$Symbol(G)=\begin{cases}\bigcup_{i=1}^{n} Symbol(G_i) & (G_i \text{ is a gesture, } G=\{G_1,G_2,\ldots,G_N\}\\ G\end{cases}$$

Gesture = symbol | *CONS* (*symbol*) | *CONS (symbol)* CONS* (*symbol*)

Symbol = element | *customized stroke* | *CONS* (*element*) | *CONS* (*element*)* *CONS* (*element*)

CONS = Constraints | *Combination*

Element= stroke | *point*| *line*| *arc*| *circle* | *ellipse*

Stroke=set (*point-set*)

Point-set=set (*point*)

Designers should be able to maintain their thinking without being broken by excessive switches among the selections of menus, operations on buttons, and keyboard input of words. What's more, constraints in gestures are captured automatically during the process, which can be kept in later operations.

Definition 8. A constraint structure graph is composed of two parts. One part is the geometry element chain, which is generated according to the creation order; the other is the constraint chain, which is the binary-netlike graph, and each node has two sub-nodes, pointing to the next constraint of the two related elements.

Property 1. A geometry element chain is sequential and described by 5-tuple ⟨type, ID, Cons-list, attributes, behavior⟩; "Cons-list" points to the related constraint chain, non-ordered, described as ⟨type, ElefirstID, ElesecondID, ConsNextforfirst, ConsNextforsecond, attributes, behavior⟩; "ConsNextforfirst" points to the next constraint of the first element got by the "ElefirstID"; "ConsNextforsecond" points to the next constraint of the second element got by the "ElesecondID";

According to the structure, we give an algorithm for the bi-directed constraint solution. One element is selected,

Step 1. Initialization. The editing-element stack is set null, $P=\Phi$ and the current element is Ψ. Its *m_Edit*=1 (represents being edited), add Ψ to the stack, push(Ψ);

Step 2. If $P=\Phi$, then go to Step 5; otherwise, $P=\{\Psi,\ldots,\Psi_i\}$, pop (Ψ), get the current element Ψ;

Step 3. Get the constraint list, $L(\Psi)=\{Con_1,Con2,\ldots,Con_i\}$, if $L(\Psi)$=null, then go to Step 2;

Step 4. Get the first constraint node of the list, get the other element Ψ', which has the constraint with Ψ,

If *m_Edit*=0, then {push(Ψ'); *m_Edit*=1;}

About the current constraint node, if *Cons_Flag*=0, then identify its constraint type, deal with the constraint, modify the related elements, get the new result, and set *Cons_Flag* as 1;

Finish computing its left and right sub-lists with recursion, and then go to Step 2;

Step 5. End.

2.2 Gesture interaction

Similar to people's everyday skills, ergonomically natural and agile, pen interaction is popular in common life[11]. We provide three modules about the gesture interaction: gesture agent, object service, and application agent. Gesture agent describes the gesture design model and the mapping with the rules of structure and language description. Object service gives the services for gesture applications such as service recognition, simple indicative service, quick feedback, and context-aware service. Application agent provides functions for domain applications.

About object service, the context-aware computing is provided as follows.

Definition 9. A context-aware condition is captured according to the existing objects and relative behaviors such as object type, operating position and so on. Different context-aware conditions lead to different operations.

Context-Awareness is an integral part of the interaction computing[12]. Time, identity and constraint can be the values of the context. We give the description of context in Fig.1. For example, the functions of move and scale can be implemented in terms of the capture and analysis of the context-awareness, including the different identities and pen positions.

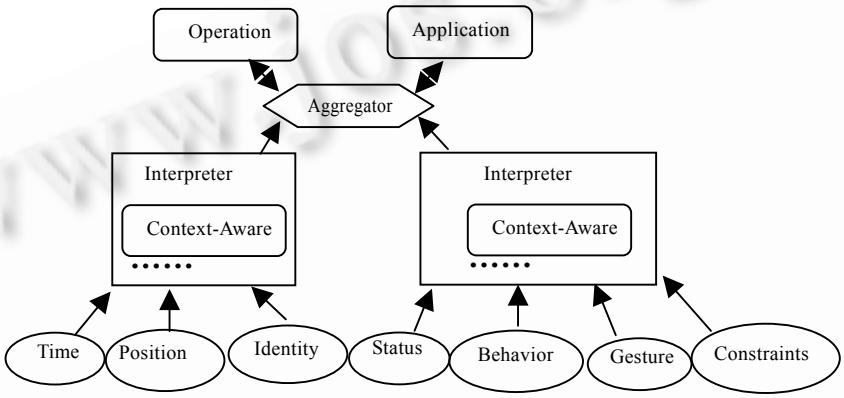


Fig.1 The context-aware computing

Application agent provides a gesture interaction service for different applications, such as the feature gesture. As mentioned above, we present the structure of the gesture interaction (Fig.2). It is to reduce the gap between applications and pen gesture. The lowest is the input level, which gets the original ink. The data are transferred from the input to the application and changed from low to high level information based on the gesture description and interaction.

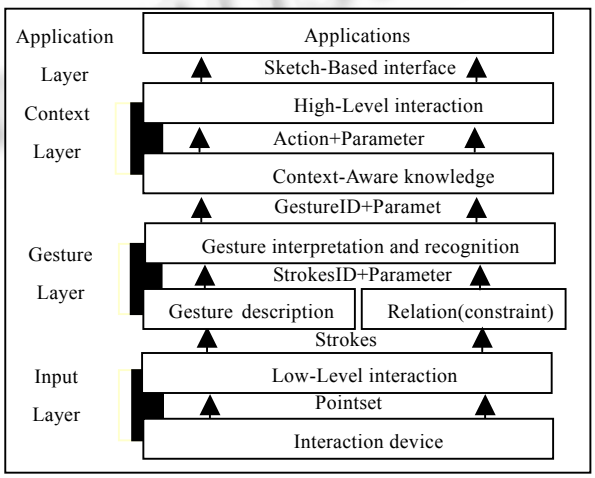


Fig.2 The hierarchy structure

2.3 Feature gesture

For many years, researchers have realized that the current CAD technology is too restrictive to use for thinking with, thus they have been trying to find ways to allow a more intuitive interface. The feature gesture is one kind of high–level modeling way matching the designers' intent instead of the traditional feature modeling, which takes advantage of the gestures and features for the innovative design.

Definition 10. Feature gesture, the integration of gesture and feature based on feature, consists of gestures and has the characters of geometry and topology constraints with its own attribute and function.

Features have a hierarchical structure as well as the strokes[12], so do feature gestures. Feature gestures at a high level are composed of those at a low level. It is efficient to represent the feature gesture in terms of hyper-graph, in which the vertex may act as a target point at most once but act as a source point in several edges. From the above discussion, we provide a feature gesture modeling in Figure 3. Users sketch feature gestures directly to achieve the designing. Users can manipulate objects without paying much attention to menus and icons for command items. During the process, constraints are captured and solved to provide a better method for modeling.

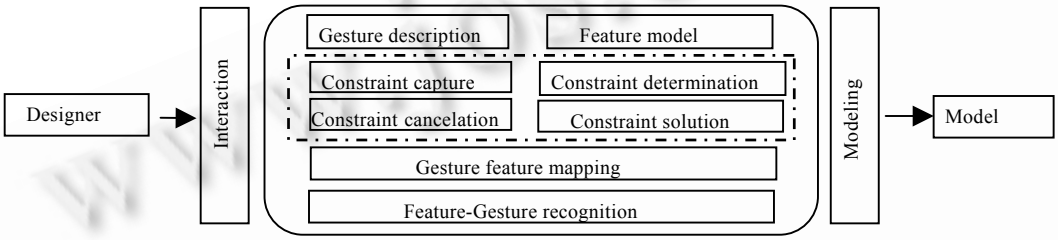
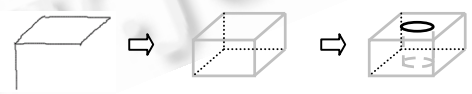


Fig.3 Feature-Gesture modeling

Most current CAD systems provide somewhat inconvenient process based on the traditional modeling method. It requires implementing the modeling under the ordered steps. First, it is needed to select the operation plane, give the drawing command, and then draw. It needs to provide the constraints and give the scanning direction before scanning command to get the last result. During the process it needs to operate the menu and icon complicatedly. The feature gesture modeling can implement the process by sketching without too much command switches. Figure 4 shows the obvious difference between the two ways.

(a) Traditional modeling



(b) Feature gesture modeling

Fig.4 Different modelings

3 Discussion

The independence among modules could benefit their communication as well as expansibility. In the gesture resource foundation module, the visualization mode of designing gesture provides a convenient way to implement tasks. In the gesture applications module, it provides a natural and flexible designing mode in terms of the interaction technology based on different applications.

There remain many difficulties. In some conditions, pen can take the place of mouse, but what fits in with a

mouse is not similar with a pen, and vice versa. For any given task, the ideal pen-based UI should not be limited to techniques developed for GUIs, but incorporate pen-specific techniques that take advantage of the special characteristics of pen. Gestures invoke operations for a direct manipulation to objects. However, it is difficult to recognize gestures according to users' intent if the operation information is fuzzy. Whereas misrecognition of a character is easily detected by users, misrecognition of an operator may not be. Furthermore, an unintended operation is likely to be more difficult to be corrected than an incorrectly recognized character. The gesture design model presented above based on the constraint and context-awareness can provide a better performance.

4 Conclusions

In this paper, we have provided an approach to gesture interaction. We first describe a gesture design model based on sketch and informal constraints. Furthermore, the contextawareness is discussed. Feature gestures, one instance of the gesture application, are given to improve the designing process in conceptual design. The advantage of our method is that we provide users with a fast and direct gesture based interface for rapid operations.

With the development of pen and new interaction technology, gestures are the valuable aspect of pen-based UIs. We believe that gesture operation, unlike the majority of WIMP techniques based on mouse motions, has the greatest potential for maximizing the interactivity of pen-based environments since operation modes are offloaded from focusing on the technology to users and tasks themselves.According to the definition and recognition of gestures, it is significant to allow context-aware gestures, which can provide an efficient assistance to users. How to represent and make the best of the context is still a challenge for us to confront. Future research could also involve the investigation of gestures in ubiquitous computing.

References:

[1] Gregory DA, Elizabeth DM. Charting past, present and future research in ubiquitous computing. ACM Trans. on Computer-Human Interaction, 2000,7(1):29−58.

[2] Long AC, Landay JA, Rowe LA. PDA and gesture use in practice: Insights for designers of pen-based user interfaces. Technical Report, UCB//CSD-97-976, U.C. Berkeley, 1997.

[3] Long AC, Landay JA, Rowe LA, Michiels J. Visual similarity of pen Gestures. In: Proc. of the Human Factors in Computing Systems SIGCHI 2000. 2000,2(1):360−367.

[4] Ma CX, Meng XX. Research on object constraint model and inverted constraints in parametric design. Chinese Journal of Computers, 2000,23(9):991−995 (in Chinese with English abstract).

[5] Long AC, Landay JA, Rowe LA. Implications for a gesture design tool. In: Human Factors in Computing Systems (SIGCHI Proc.). ACM Press, 1999. 40−47.

[6] Long AC. Quill: A gesture design tool for pen-based user interfaces [Ph.D. Thesis]. Berkeley: University of California, 2001.

[7] Rubine D. Specifying gestures by example. In: Computer Graphics. ACM SIGGRAPH, Addison Wesley, 1991. 329−337.

[8] Zeleznik RC, Herndon KP, Hughes JF. Sketch: An interface for sketching 3D scenes. In: Computer Graphics (Proc. of the SIGGRAPH'96, Annual Conference Series). 1996. 163−170.

[9] Igarashi T, Hughes JF. A suggestive interface for 3D drawing. In: ACM SIGGRAPH 2001, Technical Sketch, Conf. Abstracts and Applications. 2001. 265−276.

[10] Bimber O, Encarnacao LM, Stork A. A multi-layered architecture for sketch-based interaction within virtual environments. Computers & Graphics, 2000,24:851−867.

[11] Hauptmann AG. Speech and gestures for graphic image manipulation. In: Proc. of the CHI'89. Addison-Wesley, 1989. 241−245.

[12] Xue HY, Zhou J. Binary tree algorithm of two dimension drawing expressed by constraint network. Journal of Computer-Aided Design & Computer Graphics, 1996,8(6):470−474 (in Chinese with English abstract).

附中文参考文献:

[4] 马翠霞,孟祥旭. 参数化设计中的对象约束模型及反向约束的研究.计算机学报,2000,23(9):991−995.

[12] 薛鸿源,周济.用约束网络表示二维图形的二叉树算法.计算机辅助设计与图形学学报,1996,8(6):470−474.