[6] G. M. Clayton, S. Tien, A. J. Fleming, S. O. R. Moheimani, and S. Devasia, "Inverse-feedforward of charge-controlled piezopositioners," *Mechatronics*, vol. 18, no. 5–6, pp. 273–281, Jun. 2008.

[7] D. Hughes and J. T. Wen, "Preisach modeling of piezoceramic and shape memory alloy hysteresis," *Smart Materials and Structures*, vol. 4, pp. 287–399, 1997.

[8] D. Croft, G. Shed, and S. Devasia, "Creep, hysteresis and vibration compensation for piezoactuators: Atomic force microscopy application," *ASME J. Dynamic Syst., Measure. Control*, 2001.

[9] A. Dubra, J. Massa, and C. L. Paterson, "Preisach classical and nonlinear modeling of hysteresis in piezoceramic deformable mirrors," *Optics Express*, vol. 13, no. 22, pp. 9062–9070, 2005.

[10] K. Kuhnen and H. Janocha, "Inverse feedforwrad controller for complex hysteretic nonlinearities in smart-materials systems," *Control of Intelligent System*, vol. 29, no. 3, pp. 74–83, 2001.

[11] W. T. Ang, P. K. Kholsa, and C. N. Riviere, "Feedforward controller with inverse rate-dependent model for piezoelectric actuators in trajectory-tracking applications," *IEEE/ASME Trans. Mechatronics (Tmech)*, vol. 12, no. 2, pp. 134–142, Apr. 2007.

[12] B. Mokaberi and A. A. G. Requicha, "Compensation of scanner creep and hysteresis for AFM nanomanipulation," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 2, pp. 197–208, Apr. 2008.

[13] R. Bouc, "Forced vibration of mechanical systems with hysteresis," in *Proc. Conf. Nonlinear Oscillation*, Prague, 1967, pp. 32–39.

[14] Y. K. Wen, "Method for random vibration of hysteresis systems," *J. Eng. Mech. Div.*, vol. 102, no. 2, pp. 249–263, Mar./Apr. 1976.

[15] M. Jouaneh and H. Tian, "Accuracy enhancement of a piezoelectric actuators with hysteresis," in *ASME Jpn./USA Symp. Flexible Autom.*, 1992, pp. 631–637.

[16] T. S. Low and W. Guo, "Modeling of a three-layer piezoelectric bimorph beam with hysteresis," *J. Microelectromech. Syst.*, vol. 4, no. 4, pp. 230–237, Dec. 1995.

[17] C. W. Wong, Y. Q. Ni, and J. M. Ko, "Steady state oscillation of hysteretic differential model. I—Response analysis, II—Performance analysis," *J. Eng. Mech.*, vol. 120, pp. 2271–2325, 1994.

[18] S. K. Naha and Z. W. Zhong, "A microgripper using piezoelectric actuation for micro-object manipulation," *Sens. Actuators A: Phys.*, vol. 133, no. 1, pp. 218–224, Jan. 2007.

KnitSketch: A Sketch Pad for Conceptual Design of 2D Garment Patterns

Cui-Xia Ma, Yong-Jin Liu, Hai-Yan Yang, Dong-Xing Teng, Hong-An Wang, and Guo-Zhong Dai

***Abstract*—In this paper, we present a new sketch-based system — KnitSketch, to improve the efficiency of process planning for knitting garments at an early design stage. The KnitSketch system utilizes sketching interface with the pen-paper metaphor and users only need to draw outlines of different parts of the garment. Based on sketching understanding, the system automatically makes reasonable geometric inferences about the process-planning data of the garment. The system is designed for nonprofessional users and can design diverse garment styles by freehand drawings. The contributions of this work include contextual extraction of reusable data from sketches, a MDG structure for sketch beautification, and an integrated system with natural expression and effective communication that reduces the cognitive load of human beings. User experience shows that the proposed system helps designers focus on the task instead of the designing tools, and thus improves the efficiency and productivity of human beings.**

***Note to Practitioners*—The KnitSketch system is developed by a joint-project with a Chinese garment company located at Beijing. The target of KnitSketch is to provide a single and intuitive design platform for computer novices such as housewives. The integrated system presented in this paper uses a sketch-based input and a priori knowledge model, and successfully achieves this goal during a one-year probation in a garment factory of the company. As a practical application of sketch-based modeling, KnitSketch presents an integration solution with several distinct features that is useful in garment manufacturing industry.**

***Index Terms*—Conceptual design, domain knowledge, garment manufacturing, sketch-based interface.**

I. INTRODUCTION

KNITTING garments, including sweaters, T-shirts, polo shirts, tank tops, children and baby suits, is popular around the world. To facilitate design and manufacturing of knitting garment, 2D/3D CAD systems have been widely used in garment factory, among them some popular commercial systems [11] are SolidWorks and Gerber in the U.S., Toray-Acs and LookStailor in Japan, Investronica in Spain, and Lectra in France. These state-of-the-art CAD systems in the garment industry mainly support detail design with precise geometry information. However, they lack the capacity to support conceptual design due to the imprecise and incomplete information used at this early stage. Conceptual design is important since it enables designers to identify various design solutions before the product goes through the lifecycle process. A rapid garment modeler with reusable knowledge that supports conceptual design is in great demand in industry.

Manuscript received January 07, 2010; revised April 21, 2010; accepted October 01, 2010. Date of publication October 28, 2010; date of current version April 06, 2011. This paper was recommended for publication by Associate Editor Y. Li and Editor K. Goldberg upon evaluation of the reviewers' comments. This work was supported by the National Science Foundation of China (Project No. 60703079, 60736019, 60970099) and the National Basic Research Program of China 2011CB302205.

C.-X. Ma, H.-Y. Yang, D.-X. Teng, H.-A. Wang, and G.-Z. Dai are with the Intelligence Engineering Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China (e-mail: cuixia@iso.cn).

Y.-J. Liu is with the Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: liuyongjin@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2010.2086444

1545-5955/$26.00 © 2010 IEEE

For conceptual design, users should be able to easily and rapidly explore, compare, and communicate different design ideas with others. Most existing garment CAD systems utilize Windows, Icons Menus, Pointing (WIMP devices) style that does not match the traditional design habits of human beings well. Conceptual designers still prefer using paper and pencil, for its natural expression, effective communication and record of designers' ideas. Pen-based interaction and computing is a promising computer technique for conceptual design, since it can use various sketches with vague and imprecise information to rapidly express the creative ideas [6], [13]. Many sketch-recognition systems have been proposed to support pen-input interaction concerning general 2D geometry construction [9], [14]. In this paper, we make a step further by taking domain-specific knowledge such as garment sketching structures and styles, into consideration.

In this paper, we propose a KnitSketch system which is developed for supporting conceptual design for knitting garment. The core of KnitSketch is a sketch based system built on a sketching recognition platform, supported by certain domain knowledge. The distinct features inherent in KnitSketch include the following.

- *Effective communication*. A good conceptual design needs consider all the requirements in the lifecycle of product development. Usually, a multidisciplinary design team containing fashion designers, engineers, and customers is required [12]. How to efficiently communicate among the team and capture the design intent is a difficult problem [4]. As a practical solution to this difficult problem, the KnitSketch system adopts a sketch-based interface. By mimicking the traditional paper-and-pencil style, it provides a natural and effective communication environment.
- *Reuse of existing data and knowledge*. Conceptual design usually involves incomplete, inaccurate and inconsistent design specifications. Reuse of existing data and knowledge would benefit this fuzzy situation. In KnitSketch, formal knowledge is provided to facilitate the reuse process. By decomposing a sketching into a structurized form consisting of geometry, spatial and temporal structures, KnitSketch adopts two means to reuse existing data and knowledge. The first means is to import and navigate empirical formulas and parameters by sketching. The second means is to infer a detailed sketch based on user's rough sketches by best matching in a design database. The best matching is measured by similarity ranking based on a structurized form of sketches.
- *Efficiency and productivity improvement by pervasive computing*. The rapid progress of mobile devices offers new opportunities for improving the performance of knitting garment design. KnitSketch uses a paper-and-pencil-like sketching interface, which encourages personality and diversification of design operations with mobile devices in a pervasive computing environment. Users can design by sketching, searching, and modifying their ideas interactively, with immediate and continuous visual feedback. Users' knowledge and valuable experience can be captured and stored flexibly with mobile work.

II. Related Work

Most existing garment CAD systems focus on detail design with precise geometry, e.g., 2D pattern drafting and modification [11], 3D sewing and fitting from 2D pattern input [16], 3D garment design by sketching [7], [18], and 3D to 2D pattern transformation [10], [19]. In these works, akin to ours, Igrashi *et al.* [7] and Turquin *et al.* [18] also used sketching as design primitives. However, computer graphics effect is the main concern in [7], [18], and sketching in [19] is used to modify existing garment detail design with precise geometry. Contrastively, our work is based on a paper-and-pencil interface which allows ambiguous sketching and is suitable for communicating design ideas at an early stage.

To handle the fuzzy situation resulted from the imprecise and incomplete information obtained during conceptual design, reusing similar existing designs with similarity measures is a useful technique. Efficient searches by similarity comparison in existing design database have been extensively studied. For traditional 3D mechanical parts, similarity measures based on geometry, topology, design, and manufacturing features have been proposed [8]. For retrieval of 2D sketches or images, a wealth of methods [2] including signature extraction from features, similarity metrics using signatures, cluster and classification, and search enhancement by learning, were proposed. However, general 2D sketching recognition without the aid of domain-specific knowledge receives limited success in knitting garment development. In this paper, by decomposing sketching information into three terms consisting of geometry, spatial and temporal structures, we propose a structurized similarity comparison to reuse existing 2D sketching information for knitting garments.

Effective communication is important in conceptual design. By simulating the traditional paper-and-pencil working style, sketch-based interfaces turn out to be a natural tool that facilitates a rich variety of communication capabilities [9]. Sketch-based interactive design has already been used at the originality of CAD field when Ivan Sutherland published his seminal work on the SketchPad in the early 1960s [17]. State-of-the-art sketching interfaces support many common forms of human expression and leverage more implicit actions of human beings. In the application of knitting garment design, Eckert *et al.* [5] had revealed that interactive generative systems (including those equipped with sketching interfaces) can be powerful tools for human designers. Eckert [4] further analyzed the communication bottleneck in knitwear design between style designers and technicians, based on ethnography and knowledge level modeling. In our work, instead of improving the accuracy and reliability of ambiguous specifications between different users as achieved in [4], KnitSketch is developed for nonprofessional users who can design diverse garment styles by freehand drawings.

III. Overview of KnitSketch

The proposed sketch pad system, called KnitSketch, is an intuitive, pen-based sketching tool. Its working environment consists of a digital pen, a tablet, a display device, a sketching recognition, and management platform (Fig. 6 right). The conceptual design process with KnitSketch involves the following aspects.

- Sketch garment draft using a digital pen and a tablet. KnitSketch provides sketch-based design supported by gesture commands, such as deletion, zoom, copy, etc. (Fig. 2). The system processes the sketching operations in the following loop: (a) System regards the current sketching as a partial one and tries to complete it by partial matching in a sketch database. The partial matching function is optional to users. (b) Users continue to sketch by drawing more strokes.
- Sketch formulas using digital pen. KnitSketch allows users to input and modify formulas also using sketches. Based on the understanding of sketched formulas, KnitSketch can visualize the domain knowledge to facilitate the formula choice and modification (Fig. 4).
- Sketch-context classification. The sketches are classified into geometry part and semantics part such as dimensional constraints. Using dimensional constraints and sketch vectorization, the geometry part is further converted into detail design of precise geometry (Fig. 5).

The architecture of KnitSketch system includes three modules. The sketch-based interface module (Section IV) is used to support and improve the interaction intelligence. Sketch recognition and understanding module (Section V) reuses the domain knowledge with recommendation based on similarity and supporting sketch-based

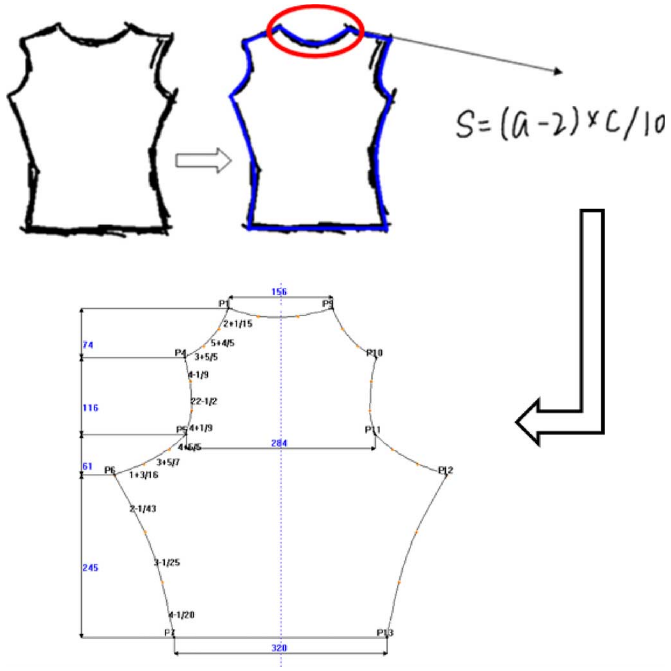


Fig. 1. A typical sketch process in KnitSketch.

widget to improve user experience. Sketch beautification and vectorization module (Section VI) converts sketches into precise geometry with a seamless connection with the downstream detail design process.

IV. SKETCH-BASED INTERFACE

The sketch-based interface of KnitSketch explores the tradeoff between effectiveness and naturalness. For knitting garment, the sketching information contains geometry, formulas and annotations. A typical sketching process is shown in Fig. 1. The top-left sketch in Fig. 1 represents the back part of a knitting garment: in this sketch, the collar shape was drawn following the user's intention. Meanwhile, the system recognized each stroke to different garment parts as shown in the top-right of Fig. 1. At this stage, formulas can be sketched naturally and referred to the intended part such as the circled collar in Fig. 1: the formula with variables s, a, c offers a constraint in this sketch and presents some semantic knowledge. Finally, the system converts the original sketch into a regularized vector graphics with precise sizes as shown at the bottom of Fig. 1. To achieve the above functionality, the sketch representation, gesture operations and sketch structurization method used in KnitSketch system are presented in the following sections.

A. *Sketch Representation*

KnitSketch uses context-aware sketching information to describe a knitting garment. The sketch information consists of two parts: graphics information and process information. The graphics information refers to the geometric shape of the knitting garment possibly with some underlying domain knowledge. The process information records the drawing orders, pen pressures and other customized sketching features.

KnitSketch describes the graphics information in three levels, sketch, stroke group and stroke, with the following definitions in terms of BNF description:

$$\begin{aligned}
&\langle Sketch\rangle ::= \{\langle Sketch\rangle\}\{\langle SketchGroup\rangle\}[Constraints]\\
&\langle StrokeGroup\rangle ::= \{\langle StrokeGroup\rangle\}\{\langle Stroke\rangle\}\\
&\qquad\qquad \langle Recognized\_Result\rangle\langle BoundingBox\rangle\\
&\qquad\qquad [Constraints]\langle GroupType\rangle\\
&\langle Stroke\rangle ::= \langle Points\rangle\langle DownTime\rangle\langle UpTime\rangle\\
&\qquad\qquad \langle Velocity\rangle\langle Inflexion\rangle\langle Direction\rangle\langle Style\rangle\\
&\langle GroupType\rangle ::= \langle Overlapping\rangle|\langle Overlooping\rangle|\\
&\qquad\qquad \langle Hatching\rangle|\langle Custom\rangle\\
&\langle Point\rangle ::= \{\langle Point\rangle\}\langle Time\rangle\\
&\langle StrokeType\rangle ::= \langle Core\rangle|\langle Complementary\rangle\\
&\langle Constraints\rangle ::= [GeometryConstraints]|\\
&\qquad\qquad [DomainConstraints]|[ContextConstraints]\\
&\langle GeometryConstraints\rangle ::=\\
&\qquad\qquad \{Constraint\_Set(Object1, Object2, Constraint)\}\\
&\langle DomainConstraints\rangle ::= TODO\\
&\langle ContextConstraints\rangle ::=, \{Sketch\_Context, User\_Model\}.
\end{aligned}$$

1) *Sketch:* is made up of several *StrokeGroup* or *Sketch* itself and types of constraints including *GeometryConstraints*, *DomainConstraints* and *ContextConstraints*. *GeometryConstraints* indicates the basic geometry constraints in graphics, such as parallel, perpendicular and intersecting. *DomainConstraints* reflects the domain of knowledge to which the system is applied. For example, the formulation method of collars and sleeves in KnitSketch is a type of domain knowledge. *ContextConstraints* records sketching process information together with users' drawing preference. All of the three types of constraints provide adequate information and efficient means to facilitate flexible and intelligent interaction between system and users.

2) *Strokegroup:* is composed of a group of strokes, a recognized result, a bounding-box and constraints. Four types of *StrokeGroup* are predefined in KnitSketch: *Overlapping*, *Overlooping*, *Hatching* and *Custom* are reserved. *StrokeGroup* can be moved, deleted or rotated as a whole.

3) *Stroke:* is the most basic operating element in KnitSketch. Each stroke is structured by following the sequential operations of pen down and up with related attributions, such as time stamp, spacial relations, directions, inflexions, and velocity.

B. *Gesture Operations*

Gesture understanding and manipulations are typical operations in sketch-based interface. The general gesture description in KnitSketch is defined as follows.

$$\begin{aligned}
Point\_set &= set(Point)\\
Stroke &= set(Point\_set)\\
Symbol &= \begin{cases} Low\_strokes(representation + context)\\ High\_strokes(representation + context)\end{cases}\\
Symbol(G) &= \begin{cases} \cup_{i=1}^{n} Symbol(G_i)\\ G = \{G_1, G_2, \ldots, G_n\}, G_i \text{ is a gesture}\end{cases}
\end{aligned}$$

$$\begin{aligned}
&\langle Gesture\rangle ::= \langle Symbol\rangle|\langle CONTEXT(Symbol)\rangle|\\
&\qquad\qquad \langle CONTEXT(Symbol)\rangle\\
&\qquad\qquad * \langle CONTEXT(Symbol)\rangle\\
&\langle Symbol\rangle ::= \langle Low\_stroke\rangle|\langle High\_stroke\rangle|\\
&\qquad\qquad \langle Customized\_stroke\rangle\\
&\qquad\qquad |\langle CONTEXT(Symbol)\rangle\\
&\langle CONTEXT\rangle ::= \langle User\_context\rangle\langle Environment\_context\rangle|\\
&\qquad\qquad \langle Machine\_perceived\_context\rangle\\
&\qquad\qquad |\langle Machine\_derived\_context\rangle\\
&\langle Low\_stroke\rangle := strokes\\
&\langle High\_stroke\rangle ::= \langle Grouping\_stroke\rangle|\\
&\qquad\qquad \langle CONTEXT(Low\_stroke)\rangle|\\
&\qquad\qquad \langle CONTEXT(Grouping\_stroke)\rangle.
\end{aligned}$$

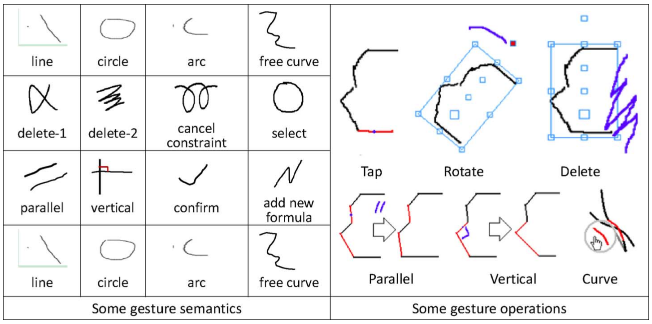


Fig. 2. The gesture description in KnitSketch.

The scenario of KnitSketch is classified into four parts: background, layout, graphics templates (e.g., front part, back part, sleeve part, and accessories such as pockets) and forms. The semantic linkage corresponding to different parts is established by direct manipulation of gestures and feature recognition. Accordingly, gestures identified by freehand strokes have one-to-one semantic mappings in KnitSketch. For example, the layout semantics include geometries (line, circle, arc, and free curve, etc.), operations (deletion, cancellation, and selection, etc.) and relations (parallelism and perpendicularity, etc.). See Fig. 2 (left) for some examples.

The operations with gesture semantics are simple since they mimic the traditional paper-and-pencil styles. Fig. 2 (right) illustrates some gesture operations in KnitSketch. Since each person has his/her own individual preference, KnitSketch also allows users to define their own customized gestures. A gesture management submodule is provided in KnitSketch. Users can add or delete gestures by interactively visualizing the gesture database. If a new customized gesture is added, KnitSketch searches the database by similarity measures (see Section V-B for details): if a similar gesture exists, the existing one is replaced by the new one. Therefore, by managing the gesture database, the operations in KnitSketch can be tailored to meet individual preference.

C. *Sketch Structurization*

Sketches consist of temporal and spatial information, involving different kinds of context, such as drawing order, interaction history, drawing cues and geometry constraints. In KnitSketch, a multiple decision graph model (MDG) is proposed to structurize the sketch context, in order to provide efficient sketch retrieval and recommendation. The MDG uses spatial relationship of strokes and strokes are added into MDG according to drawing orders (temporal information). The sketch node in MDG is defined as follows.

```
MDG node {
    SketchElement;
    SketchElement *PointerList; //pointing to child-nodes
};
SketchElement {
    Sketch;
    Graphic Attribute; //recognized graphics corresponding to sketch
    Constraint Relations; //constraint list with parent nodes
};
```

The MDG is created during sketching process following the users' interaction. The MDG construction algorithm is as follows and one example is presented in Fig. 3.

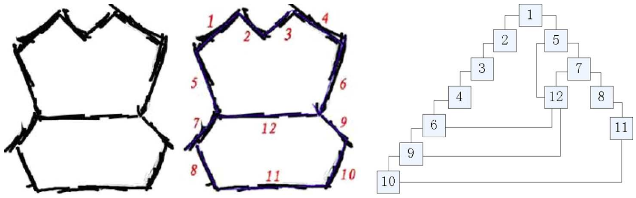


Fig. 3. The multiple decision graph (MDG) encodes sketches using both temporal and spatial information: the stroke node is inserted according to drawing orders and the graph is arranged with spatial relationship.

Algorithm 1: MDG construction

```
1. Create a new node $Nd$ for the stroke to be added;
2. Traverse the MDG by breadth-first search, for each node $Td$ in MDG do the follows:
    2.1. Compare the bounding box between $Nd$ and $Td$
    2.2. If the two bounding boxes are overlapped
        2.2.1. Calculate the distance $D$ between $Nd$ and $Td$
        2.2.2. If $D$ is less than a prescribed threshold of sketchgroup
            2.2.2.1. Add $Nd$ into the sketchlist of $Td$;
3. Add $Nd$ into MDG as a leaf.
```

Given the MDG, the constraints among strokes and sketches can be captured efficiently using the following algorithm.

Algorithm 2: Constraint construction

```
1. Traverse the MDG and test the constraint type between SketchElements;
    1.1. If a constraint exists between two SketchElements $pOb1$ and $pOb2$
        1.1.1. Create a constraint node $pCons$;
        1.1.2. Add $pCons$ into the constraint list of $pOb1$ and $pOb2$;
```

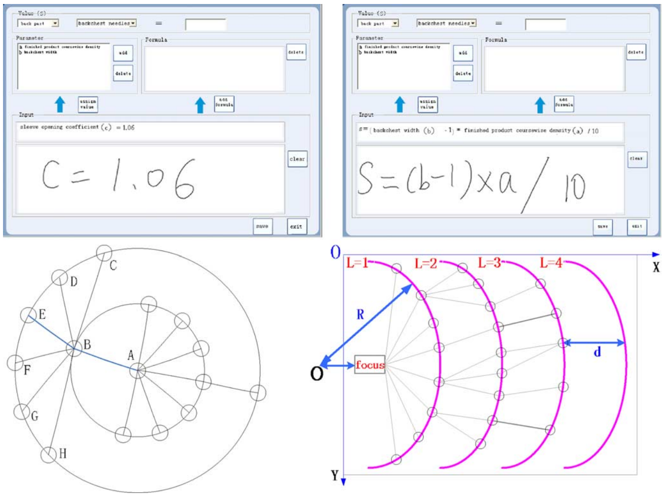
V. KNOWLEDGE MODELING AND REUSE

The domain knowledge used in KnitSketch includes process planning formulas and complete sketches in existing designs. KnitSketch allows users to input their own knowledge into the database interactively, which includes process planning formulas and empirical variables in domain. To reuse the sketches of existing designs in database, KnitSketch provides a similarity-based recommendation mechanism to assist current sketching.

A. *Formula Input by Sketching and Navigation*

Formulas and variables are important representations for knowledge and experience. Users usually prefer to provide empirical formulas and variables by writing them down on the paper or sketching on the interface. Sketched formulas and variables consist of mathematical notations and freehand diagrams. Two examples are shown in Fig. 4 (top) in which the sketched formula is as follows:

$$Backchest\ length(S) = (Backchest\ width(b) - 1) \times \frac{Finished\_product\_coursewise\_density(a)}{10}.$$



sleeve opening coefficient (c) = 1.06

C = 1.06

S = (backchest width (b) - 1) * finished product coursewise density (a) / 10

S=(b-1)xa/10

Fig. 4. Formula input by sketching and graphics navigation. Top left: parameter input and value specification. Top right: formula sketching. Bottom left: a classic radial layout [3]. Bottom right: the advanced radial layout in KnitSketch.

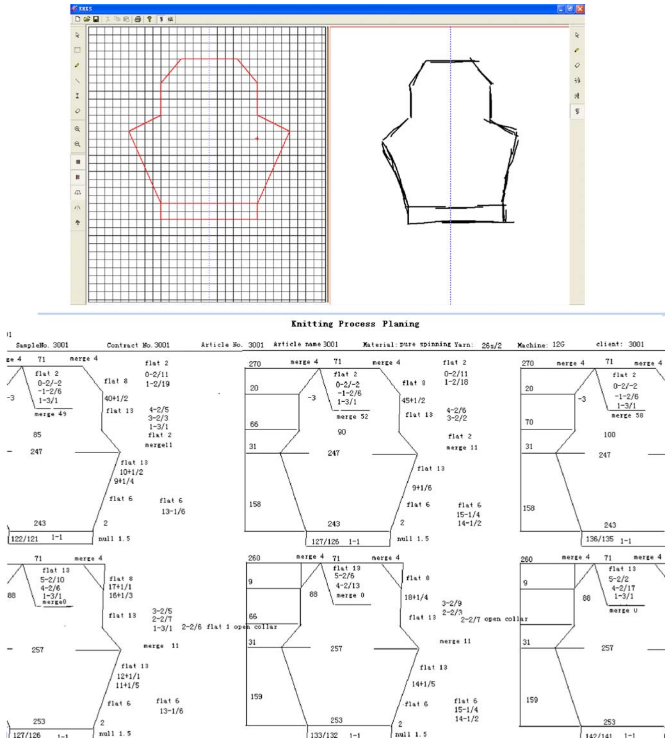
There are lots of empirical variables and formulas in process plans for diverse garment styles. How to organize them and provide a clear visual representation about the formulas and relationships among them is important for design and reuse. KnitSketch adapts a focus-context technique to visualize and navigate the inherent relations among the data elements. First all the variables have corresponding design terms in glossary relating to different garment styles. Then each variable is mapped into a node in a connected graph for visualization and navigation. A classic radial layout algorithm [3] is adapted in KnitSketch to provide multiple views of nodes by dynamically reconfiguring the connected graph (Fig. 4 bottom).

B. Sketch Recommendation Based on Similarity and Process Context

To reuse the complete sketches in existing designs, for current sketches (possibly incomplete), KnitSketch searches the sketch database to find the most similar sketches. Since sketches only present a rough idea with inaccurate geometry, the shape-context matching method [1] using elastic deformation is adopted in KnitSketch. Since the sketches to be matched into KnitSketch database are usually incomplete, Matching them to portions of existing sketches in database is a property desired by users. To achieve this, the sketches are segmented into several primitives based on MDG structure. For each sketch in database, several partial representations obtained by combining different primitives are stored and matched with the incomplete sketches. To obtain fast response time, point patterns of sketches in database are clustered in a hierarchical manner and an advanced pruning technique is used [15].

Different from static images, sketching is a continuous process with continuous feedback in the operations on multidimensional information, including pen pressure, pen tilting, pen speed, etc. Besides shape-context similarity, KnitSketch reuses and recommends sketches in database also based on process context. Since the users' preference can be reflected by drawing pressure and speed during sketching, KnitSketch represents the process context of sketching using pressure and speed attributes. The feature vector to be incorporated into sketch signature is as follows:

$$\begin{aligned} \max\_pres &= \max(pressure(i)) \\ \min\_pres &= \min(pressure(i)) \\ \max\_speed &= \max(speed(i)) \\ \min\_speed &= \min(speed(i)) \end{aligned}$$



SampleNo. 3001 Contract No. 3001 Article No. 3001 Article name 3001 Material: pure spinning Yarn: 26s/2 Machine: 12G client: 3001

Fig. 5. Sketch beautification and vectorization. (Top) The sketches by users. (Bottom) The specified detail design with precise geometry.

$$\begin{aligned} normalized\_pres(i) &= \frac{(pressure(i) - \min\_pres)}{(\max\_pres - \min\_pres)} \\ normalized\_speed(i) &= \frac{(speed(i) - \min\_speed)}{(\max\_speed - \min\_speed)}. \end{aligned}$$

The pressure and speed values are not very stable during sketching process. KnitSketch uses a smoothed version of these values by using a linear weighted moving average method:

$$x(i) = \frac{\sum_{t=i-k}^{i+k} (x(t)|t-i|)}{\sum_{t=i-k}^{i+k} |t-i|}$$

where k represents the interval of a moving average and x represents $pressure$ or $speed$.

VI. SKETCH BEAUTIFICATION AND VECTORIZATION

When the sketching process is completed, KnitSketch converts point-based sketches into a vector graphics by segmenting the sketches based on MDG and fitting each segment by a NURBS curve. The resulting vector graphics can be further modified by sketching and more constraints can be input to specify a detail design with precise geometry. An example is shown in Fig. 5.

The sketch structurization by MDG (Section IV.C) segments the sketches into groups of strokes and provides geometric constraints between stroke groups. The strokes capture the design intent of the user by a set of ordered points and KnitSketch system approximates them by a NURBS curves of degree 3.

Users frequently need to further modify the recognized parametric curves with precise geometry. We observed drawing behavior of several knit garment designers in a Chinese factory and found that most of them would rather edit curves by sketching than by control points. So

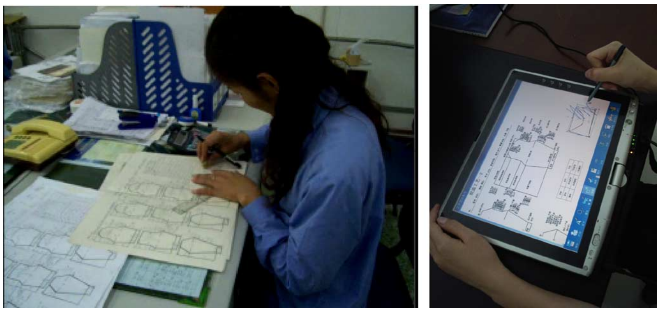


Fig. 6. System evaluation. Left: traditional design environment using physical paper, pen, eraser and ruler. Right: the experimental environment of the KnitSketch system with a tablet PC.

the KnitSketch system uses a sketch-based curve editing method. The sketch-based curve edit algorithm based on MDG is as follows.

Algorithm 3: sketch-based curve editing

```
Input. A sketch with MDG structure
Output. A sketch with updated MDG structure
1. When beginning to draw a stroke, set the pen-down point as $P_0$;
2. Initialize a point list $NPSet$ with $P_0$;
3. Find the nearest stroke $S$ to $P_0$ and add it to list $ReList$;
4. When pen is moving
        4.1. Record the pen position $P'$;
        4.2. Find the nearest point $P''$ to $P'$ on $S$;
        4.3. If $P'$ reaches the end of stroke $S$
            4.3.1. Find an adjacent stroke $S'$ that is nearest to $P'$;
            4.3.2. Add $S'$ to $ReList$
            4.3.3. Find the nearest point $P''$ to $P'$ on $S'$;
            4.3.3. set $S = S'$;
        4.4. Compute the distance $d$ between $P'$ and $P''$;
        4.5. If $d$ is less than a threshold $e$
            4.5.1. Set $P' = (P' + P'')/2$;
        4.6. Add $P'$ to $NPSet$
5. Traverse the list $ReList$ and for each stroke $S$ in $ReList$
        5.1. Refresh the point list of $S$ according to the point list $NPSet$;
        5.2. Re-fit stroke $S$ by a B-spline curve;
```

VII. SYSTEM EVALUATION

In order to aid a garment designer with his/her inventive work, the KnitSketch system must be compatible with current industrial practice in the workplace and allow the designer to follow his/her traditional habit. Fig. 6 shows a working instance. A user can sketch a drawing of a garment part with mobile devices or desktop PCs and empirical formulas can be input or reused to calculate the process planning data. The recognized process planning drawings are then generated with precise geometry. Finally, the detail designs are passed to workers for knitting the garment on machines.

To evaluate the usability performance of KnitSketch system, the following experiment is conducted.

A. Experiment Setup

The goal of the experiment is to examine how KnitSketch system benefits different operations in the garment process planning and how it affects the user's usability. Here, we use three methods to implement the process planning of a garment style, i.e., a general lady's pullover shirt with narrow waist. The three methods are: 1) using traditional paper; 2) using a commercial software SolidWorks which support conceptual design to some degree; and 3) using KnitSketch system. The description of the task is that the users carry out the design based on the given requirements for the same style garment and the output is a detailed engineering drawing. The users can refer to books or ask help from the skilled designers for empirical formulas or related parameter values in traditional mode.

B. Participants and Apparatus

Six people, aged from 22 to 40, were selected to participate in the user study at the same workplace. The six users include three novices and three skilled designers (who have engaged in garment process planning for more than one year). All users are required to implement the same task using the three different methods. To minimize the experimental bias due to computer skills, we ensure that there is no obvious difference on computer operation level between all participants according to users' self-reports and our observations. Fig. 6 (right) shows the working environment of KnitSketch system with a Toshiba Tablet PC running Windows XP system.

C. Procedure

The requirement form and some original data for the ladies pullover shirt with narrow waist were provided for the participants in advance. The original data included some formulas in existing designs (for other types of knitting garment). Ten minute introduction about the task and three different methods were offered. Participants were instructed to complete the experiment with the following general process. First, participants sketched the parts based on the garment style. Then, they formularized the resulting process planning data in terms of empirical formulas and sometimes they had to adjust the parametric values by trial and error. It was observed that some participants generated new formulas by editing the existing ones. Meantime, narrowing and widening data for curved shape were calculated to refine the design. Finally, data checking was performed. There were also some iterations among the operations during the process. The final output engineering drawings ware checked by another skilled designer for validation.

D. Results

The designing time using different methods and in different design phases were recorded for each participant. Fig. 7 (left) shows the time needed to complete a design using three different methods. From this result, we conclude that: 1) KnitSketch system achieve minimum designing time and 2) there is no significant difference in time between using physical paper and commercial SolidWorks system. Since SolidWork system did not show any benefits, we exclude it from the candidate list and next we only compare the performance between using physical paper and using KnitSketch.

To compare the two methods, we focus on two subtasks of the process planning with four different operations: 1) subtask of design with sketching and formula editing operations and 2) subtask of calculation with data calculation and verification operations. The statistic data obtained using traditional paper-and-pencil method and the KnitSketch system are shown in Fig. 7 (middle and right), respectively. From the data presented in Fig. 7, it is readily seen that: 1) using traditional method, users spend more time on the calculation subtask and 2) using KnitSketch system, users can focus more on design and ideation sub-task.

These results demonstrate that using KnitSketch system, the users can draft their ideas, obtain and modify empirical formulas easily through sketches and gestures, and their cognitive load is reduced during the conceptual design process. After the experiment, all the

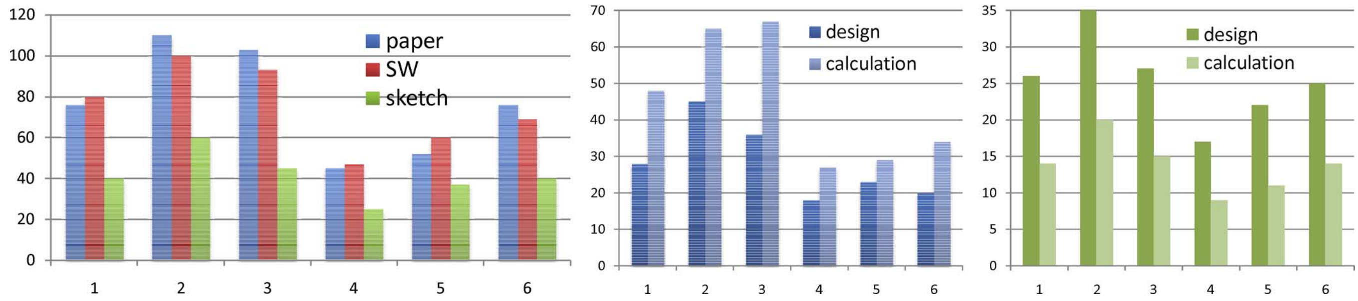


Fig. 7. Evaluation results. Left: the time needed to complete a design using three different methods: physical paper, commercial SolidWorks and KnitSketch. X axis represents the six participants (1–3 are novices and 4–6 are skilled designers). Y axis represents the time measured in minutes. Middle: the spent time of two different sutasks (design and calculation) using the traditional paper-and-pencil method. Right: the spent time of two different sub-tasks (design and calculation) using KnitSketch system.

participants were asked about their experience and preference. The feedback is positive: 1) all the participants thought that KnitSketch system is superior to both commercial SolidWorks system and traditional paper-and-pen method and 2) using KnitSketch, all the participants felt that they could concentrate more on the design task and draft the ideas easily, efficiently, and effectively.

VIII. CONCLUSION

Sketching with a natural user interface is a promising technique to alter the way that designers conduct the process planning for knitting garment. The key idea behind the KnitSketch system is to mimic a paper-and-pencil interface which reserves a natural way of thinking and communication of ideas. Furthermore, by integrating advanced computing techniques of design and calculation in pervasive environment, users can make conceptual designs more easily and keep a natural style of interaction. As a practical application of sketch-based modeling integrated with domain knowledge, KnitSketch would be useful in garment manufacturing industry.

ACKNOWLEDGMENT

The authors thank the reviewers for their comments that help improve this paper. The authors also highly appreciate Prof. C. C. L. Wang at CUHK for his detailed comments and English corrections of this paper.

REFERENCES

[1] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 24, pp. 509–522, 2002.

[2] R. Datta, D. Joshi, J. Li, and J. Wang, “Image retrieval: Ideas, influences, and trends of the new age,” *ACM Computing Surveys*, vol. 40, no. 2, pp. 5–60, 2008.

[3] P. Eades, “Drawing free trees,” *Bulletin of the Institute of Combinatorics and its Applications*, vol. 5, pp. 10–36, 1992.

[4] C. Eckert, “The communication bottleneck in knitwear design: Analysis and computing solutions,” *Computer Supported Cooperative Work*, vol. 10, no. 1, pp. 29–74, 2001.

[5] C. Eckert, I. Kelly, and M. Stacey, “Interactive generative systems for conceptual design: An empirical perspective,” *Artif. Intell. Engineering Design, Anal. Manuf.*, vol. 13, no. 4, pp. 303–320, 1999.

[6] M. Gross and E. Do, “Ambiguous intentions: A paper-like interface for creative design,” in *Proc. ACM Symp. User Interface Softw. Technol.*, 1996, pp. 183–192.

[7] T. Igrashi and J. Hughes, “Clothing manipulation,” in *Proc. ACM Symp. User Interface Softw. Technol.*, 2002, pp. 91–100.

[8] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani, “Three-dimensional shape searching: State-of-the-art review and future trends,” *Computer-Aided Design*, vol. 37, no. 5, pp. 509–530, 2005.

[9] J. LaViola, “Sketch-based interfaces: Techniques and applications,” ACM SIGGRAPH’07 Course Notes 3, 2007.

[10] Y. J. Liu, Y. K. Lai, and S. M. Hu, “Stripification of free-form surfaces with global error bounds for developable approximation,” *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 4, pp. 700–709, Oct. 2009.

[11] Y. J. Liu, D. L. Zhang, and M. M. F. Yuen, “A survey on CAD methods in 3D garment design,” *Comput. Industry*, vol. 61, no. 6, pp. 576–593, 2010.

[12] Y. J. Liu, K. L. Lai, G. Dai, and M. M. F. Yuen, “A semantic feature model in concurrent engineering,” *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 659–665, Jul. 2010.

[13] Y. J. Liu, C. X. Ma, and D. L. Zhang, “Easytoy: A plush toy design system using editable sketch curves,” *IEEE Computer Graphics and Applications*, no. DOI: 10.1109/MCG.2009.147.

[14] Y. J. Liu, K. Tang, and A. Joneja, “Sketch-based free-form shape modelling with a fast and stable numerical engine,” *Comput. Graphics*, vol. 29, no. 5, pp. 778–793, 2005.

[15] G. Mori, S. Belongie, and J. Malik, “Shape contexts enable efficient retrieval of similar shapes,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2001, pp. 723–730.

[16] H. Okabe, H. Imaoka, T. Tomiha, and H. Niwaya, “Three dimensional apparel CAD system,” *Comput. Graphics*, vol. 26, no. 2, pp. 105–110, 1992.

[17] I. Sutherland, “Sketchpad: A man-machine graphical communication system,” Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, 1963.

[18] E. Turquin, J. Wither, L. Boissieux, M. P. Cani, and J. F. Hughes, “A sketch-based interface for clothing virtual characters,” *IEEE Comput. Graphics Appl.*, vol. 27, no. 1, pp. 72–81, 2007.

[19] C. Wang, Y. Wang, and M. M. F. Yuen, “Design automation for customized apparel products,” *Computer-Aided Design*, vol. 37, no. 7, pp. 675–691, 2005.