Microsoft® BROWN DTO

2007 Invited Workshop On Pen Computing

Abstracts

March 26 – 28, 2007
Providence, RI

2007 Workshop on Pen Computing

Math Recognition

Designing Pen-Centric Applications
Bob Zeleznik, Tim Miller

MathBrush: Investigating Pen-Math Environments
George Labahn, Edward Lank, Mirette Marzouk, David Tausky

Sketch Understanding II

Pen-based Tutoring Systems
Thomas Stahovich

Sketch Recognition at Texas A&M University
Tracy Hammond

Theoretical Implications

Activity-based Pen Computing
James A. Landay, Yang Li, and Richard Davis

From Laws of Action to Shape Writing: Pen-Centric UI Research at IBM Almaden
Shumin Zhai

Ink Analysis and Education

Overview of Ink Parsing Research and Productization at Microsoft with emphasis on Digital Note-Taking
Sashi Raghupathy

Classroom Learning Partner: Improving Teaching and Learning via Pen-Based In-Class Excersises
Kimberle Koile

Table of Contents

Diagramming

INTERPRETING ANNOTATED ENGINEERING DRAWINGS
2007 Workshop on Pen-Centric Computing Research

Nuria Aleixos, Dept. of Mechanical Engineering and Construction. Universitat Jaume I,
E-12071, Castellon, Spain. <naleixos@emc.uji.es>

Ferran Naya, ETSII-DEGI, Polytechnic University of Valencia,
E-46022, Valencia, Spain. <fernansan@degi.upv.es>

Manuel Contero, ETSII-DEGI, Polytechnic University of Valencia,
E-46022, Valencia, Spain. <mcontero@degi.upv.es>

Joaquim Jorge, Eng[a]. Informática, IST, Av.Rovisco Pais
1049-001 Lisboa, Portugal. <jorgej@acm.org>

Peter Varley, Dept. of Mechanical Engineering and Construction, Universitat Jaume I,
E-12071, Castellon, Spain. <varley@emc.uji.es>

Pedro Company, Dept. of Mechanical Engineering and Construction, Universitat Jaume I,
E-12071, Castellon, Spain. <pcompany@emc.uji.es>

Keywords: Sketching, Solid Models, Computer-Aided Design, Annotations, Gestures

1. ABSTRACT:

In previous work, we have developed two pure sketching applications aimed at creating 3D models of medium complexity from *thinking sketches*. But what about typical WIMP CAD applications? Can we add a sketching interface to them without losing functionality? Can we use *prescriptive sketches* to give instructions to a CAD application to create a 3D model? In order to answer to these questions, we are developing an application, *ParSketch*, which combines a calligraphic interface with a commercial 2D parametric engine.

Our general aim is to benefit from simpler paradigms based on sketching and drawing to reduce unneeded complexity in both the conceptual and the detailed design phase. Prescriptive sketches are used during the product development process to convey geometric characteristics of a new product for 3D modelling or generation of 3D drawings. Engineering designers increasingly use computer-based tools to produce and process such sketches. For example, many draughtsmen now have automated drawing boards which record their strokes as they are made and keep an electronic record.

As part of the design process, engineering designers annotate their designs with standard annotations with conventional meanings, such as “make concentric”, “make parallel”, “make perpendicular”. These annotations will also be recorded by the computer and reproduced when the computer prints out the drawing. Humans viewing these drawings understand the conventional meanings attached to the annotations.

Our objective is that the computer not merely records the annotations but interprets them and applies them to the design in the same way that a human would.

METHOD: In principle, there are four stages to interpreting annotations:

1. Capture and record the data
2. Separate annotation data from drawing data
3. **Interpret annotations**
4. Apply the annotations to the drawing

The first stage, capturing and recording the data, is not interesting theoretically: it should be straightforward, and it will vary depending on the particular equipment used (it is a job for equipment vendors, not academic researchers!). The fourth stage, applying the interpreted annotations to the drawing, is the *geometric constraint satisfaction* problem, which is currently being studied by several other groups.

In our preliminary demonstration, we assume that annotations are added to an existing drawing, i.e. all new strokes are part of the annotation, not part of the drawing.

Our work therefore concentrates on interpreting annotations. Our gesture recogniser provides an alphabet of geometric and dimensional constraints and gesture commands to manage the sketch and to create and refine 3D geometry.

Our geometry recogniser supports multi-stroke gestures. Strokes with a path length under a length threshold are deleted. Geometric entities are identified and their control parameters obtained as follows:

1. We use pre-processing image analysis techniques to smooth and remove noise from the sketched gesture.
2. The size of the sketched gesture is normalized to provide the same concentration of digitized points throughout the gesture. From this, two signatures of the shape of the gesture are calculated: a) using the distances of each point from the gesture centroid and b) using a mean shift procedure (Yu, 2003) to identify vertices and find changes in the direction of consecutive points.
3. We apply a fast Fourier transform (FFT) to move to the spectrum domain of the two signatures.
4. A standard non-linear discriminant analysis is used to classify the gestures by analysis of the Fourier descriptors.

Our recognizer is interactive, running continuously during a sketching session and parsing strokes.

CURRENT RESULTS:

Currently, *ParSketch* interprets strokes which can be recognized as geometry (line, arc, circle, ellipse, or composed entities that are automatically segmented into those basic entities), or annotations representing constraints. At present the system *ParSketch* uses pen pressure to decide whether input corresponds to geometry or to gesture strokes. The criterion is that drawing applying high pressure means a geometry stroke; otherwise stroke is analyzed as a gesture. Both geometry and gesture analyzers make use of two geometric signatures: the direction and curvature graphs of each stroke.

Currently, *ParSketch* interprets annotations to engineering drawings with greater than 90% accuracy. This work has been submitted to an international conference (Aleixos et al, 2005).

The twelve annotations which we recognise, identified in (Naya, Contero, Aleixos, Jorge, 2004) and listed below, are “borrowed” from sets of meaningful engineering symbols currently defined in international standards (ISO, ASME…): Dimension; Diametric dimension; Concentric; Parallel; Perpendicular; Horizontal; Vertical; Extrude; Rotate left; Rotate right and Erase. These are illustrated here:

http://pacvarley.webspace4free.biz/Figures/Annotations/Annotations.html

A description of the work to date and a summary of results will also appear (in both English and Spanish) on the official REGEO web site: http://www.regeo.uji.es/

HARDWARE:

In aiming to interpret annotations added to engineering drawings, ideally, we should wish our process to be independent of the way the drawing is acquired. It might, for example, be entered via a sketch tablet or an electronic drawing board, or transmitted to the computer as an electronic image captured by a mobile phone, as suggested by (Farrugia et al, 2004). The applicability of other approaches like Logitech® io™2 Digital Writing System can be considered too.

Since the preliminary implementation which we shall demonstrate relies on stroke order, it is limited to those hardware configurations which can provide this information. We shall use a laptop computer with built-in graphics tablet.

2. DESCRIPTION OF DEMONSTRATION

We shall demonstrate recognition of each of the twelve gestures listed above, by entering these into a laptop computer with built-in graphics tablet. We hope (but cannot guarantee) to match the recognition rate of 92% which we have achieved in our own experiments.

If time permits and there is sufficient interest, we should also wish to demonstrate *REFER*, our existing application for interpreting wireframe drawings as 3D solid objects (Company et al., 2004). It is our intention to integrate recognition of annotation gestures with *REFER*. *REFER* can be downloaded from:

http://www.regeo.uji.es/software_eng.htm.

We could also demonstrate a more recent application, *CIGRO* (Naya et al., 2003), which provides a reduced instruction set calligraphic interface to create polyhedral objects using an incremental wireframe drawing paradigm evocative of paper and pencil drawings. Users draw lines on an axonometric projection, which are first automatically beautified and connected to existing elements of the drawing, then converted into a three-dimensional model in real time through a reconstruction process based on axonometric inflation.

3. REMAINING RESEARCH ISSUES

Although our existing work is successful in meeting its immediate objectives, it relies on annotations being drawn in a particular manner. In particular, we have made several assumptions concerning orientation and stroke order. For this reason, it is not independent of the way the data is acquired. As our demonstration will show, it can be used successfully for interpreting annotations entered via a tablet PC, but would not, for example, be useful for interpreting annotations included in a digital image (these do not preserve stroke order). Future work will try to overcome these limitations.

The choice of annotations which our existing implementation recognises, and the gestures which are used to draw them, are based on the requirements of a small sample of people. This is unsatisfactory, and in we shall carry out a survey to determine which other common annotations should be recognised.

In order to accommodate a wider range of annotations, we plan to adopt ideas from the field of handwritten kanji recognition such as Kawatani's cascade arrangements of complementary classifiers (Kawatani, 1999).

We also wish to integrate recognition of annotations with *REFER*, as mentioned above.

ACKNOWLEDGEMENTS

This work has been supported financially by UJI-Fundació Caixa Castelló project Code P1-1B2004-02, and by the Spanish Ministry of Science and Education and the European Union, through the project Ref. DPI2004-01373. All of this support is acknowledged with gratitude.

REFERENCES

Aleixos, N., et al, 2007. *Geometry and Gesture Recognizers for Prescriptive Sketch Interpretation,* submitted to IbPria 2007.

Company, P., Contero, M., Conesa, J., Piquer, A., 2004. An optimisation-based reconstruction engine for 3D modelling by sketching. *Computers & Graphics, 28(6),* 955–979.

Farrugia, P.J., Borg, J.C., Camilleri, K.P., Spiteri, C. and Bartolo, A., 2004. A Cameraphone-Based Approach for the Generation of 3D Models from Paper Sketches, in *ed. J. F. Hughes and J. A. Jorge, Sketch-Based Interfaces and Modelling, Eurographics Symposium Proceedings,*33–42.

Kawatani, T., 1999. Handwritten Kanji Recognition Using Combined Complementary Classifiers in a Cascade Arrangement, in ***Proceedings of the 51h ICDAR,*** Bangalore, 503–506.

Naya F., Conesa J., Contero M., Company P. and Jorge J., 2003. Smart Sketch System for 3D Reconstruction Based Modeling. *Smart Graphics, Proceedings. Lecture Notes in Computer Science*. Volume 2733, 58–68.

Naya F., Contero M., Aleixos N. and Jorge J., 2004. Parametric Freehand Sketches. *Lecture Notes in Computer Science*. Volume 3044, 613–621.

Yu, B., 2003. Recognition of Freehand Sketches Using Mean Shift. *Proc. of the 8th International Conference on Intelligent User interfaces (IUI '03), ACM Press,* 204–210.