A Framework for Multi-Domain Sketch Recognition

Christine Alvarado, Tevfik Metin Sezgin, Dana Scott, Tracy Hammond, Zardosht Kasheff, Michael Oltmans and Randall Davis

The Problem: People use sketching to interact with one another and to record their ideas across many domains, including mechanical engineering design, software design, human resource management, and many other areas. Currently, almost all sketching is done on paper, and often theses sketches are transferred to the computer for recording or interaction purposes through menu-based systems.

Current hardware allows people to sketch directly on their computers; however, the ability to sketch directly onto the computer is not useful unless the computer can represent these sketches with the appropriate semantic meaning, rather than just as a bitmap image. And unfortunately, current recognition systems are brittle and difficult to construct.

This abstract outlines our approach towards a framework that would make recognition systems across various domains more robust and easier to construct.

Motivation: We have written sketch recognition systems in various domains, including mechanical engineering designs, UML diagrams, architectural floorplans and finite state machines. In each case, we have found that, as in speech recognition knowledge of the domain aids the recognition process and makes the recognition more robust. For example, if we know that the user is drawing finite state machies, it is likely that any closed shape the user draws is meant to be a circle.

Currently context must be built into each recognizer–the the routine for recognizing circles in one domain is different from the routine to recognize circles in another. We would like to avoid building context into each individual recognizer, and allow the designer of the recognition system to formally specify both the context rules and the shapes found in a given domain.

Previous Work: Our work pertains both to previous work on pattern description as well as to work on pattern recognition.

Formal grammars for describing shapes and patterns have been explored in architectural design for the last thirty years, starting with work by Stiny [10, 9] and more recently in [11] and [4]. Work in this area is mainly concerned with using grammars for the generation of shapes. In contrast, our work is concerned with using these grammars to guide recognition.

Recognition has been studied in various contexts. The most closely related to our work is other work in sketch recognition. Work by Gross and Do [3, 2], Landay and Meyers [6], and Stahovich [8] all explore various aspects of sketch interpretation within a limited domain. Forbus [1] takes a multimodal approach to building a sketch interpretation system for military diagrams.

Pattern recognition systems that use context and shape to constrain recognition have also been proposed in the field of computer vision, where the task is to recognize a pattern in a photographic image [7, 13, 5]. These works present a general framework of the type we are aiming for, but we start with hand-drawn stroke data rather than pixel data.

Approach: There are two pieces to our proposed solution. First we provide a grammar for programmers to specify higher level shapes in a particular domain. Next, the system compiles the specified shapes and builds a context sensitive recognition system that uses both bottom-up—driven by matching against recognizeres for simple strokes such as lines and circles—and top-down—driven by the high-level patterns specified by the user—processing to recognize incoming strokes.

A programmer wishing to build a recognition system in a new domain specifies the shapes in the domain in terms primitive shapes and geometric relationships. Figure 1 gives an example of the specification for an and gate.

Once the user has specified the patterns to be recognized and the context in which these pattens will appear, the system compiles the reconigzers into a system which can recognize strokes as they are put on the page. Each pattern will be represented with a fragment of a Bayesian network; low level interpretations (for example, a line and an arc in the and-gate) will provide support for higher level interpretation (the and-gate) which in turn provide support for the other low level interpretations in the

Sketch	Description
	1. define and-gate 2. NumberOfInputs 3. line l1 l2 l3; arc a1 4. parallel l1 l2 5. same-length l1 l2 6. same-horiz-position l1 l2 7. vertical l3 8. meets l1.p2 l3 9. meets l2.p2 l3 10. semi-circle a1 11. orientatin(a1, 180) 12. connected a1 l3 a1.p1 l3.p1 13. connected a1 l3 a1.p2 l3.p2

Figure 1: The formal description of an and-gate. Line 1 starts the definition and gives the name of the object. Line 2 lists attributes that are relevant to the recognized object. Line 3 lists the subcomponents that make up the and-gate. Lines 4–12 describe the relations that must be found among the subcomponents for the object to be recognized as an and-gate.

pattern (the other two lines).

Impact: Because sketching is useful across many domains, a system that can support multiple domains with less effort from the programmer is more useful than a system tailored to one domain. Furthermore, incorporating context into recognition systems will make these systems more robust and therefore more useful in practice.

Future Work: We have outlined a basic sketch interpretation system, but have not yet implemented this system. There are several challenges we must face in the implementation. For example, we have described a method for describing shapes within a domain, as well as a skeleton of a recognition system in terms of fragments of a Bayesian network. When we actually perform recognition of a user's strokes we will have to determine how to map those strokes to the recognition templates defined in our system.

Also, our current plan expects the programmer to write out the detailed description of each object in the domain, which is a tedious process itself. We would like the system to learn the relationships and subcomponents from drawings of the high-level pieces rather than forcing the designer to specify them explicitly.

Research Support: This work is supported in part by the Ford/MIT Collaboration, under the Virtual Engineering Project, by the National Science Foundation, through graduate research fellowships, and by MIT Project Oxygen.

References:

[1] Kenneth Forbus, R. Ferguson, and J. Usher. Towards a computational model of sketching. In *IUI '01*, 2001.

[2] Mark Gross and Ellen Yi-Luen Do. Ambiguous intentions: a paper-like interface for creative design. In *Proceedings of UIST 96*, pages 183–192, 1996.

[3] Mark D. Gross. The electronic cocktail napkin - a computational environment for working with design diagrams. *Design Studies*, 17:53–69, 1996.

[4] S. W. Hsiao and C. H. Chen. A semantic and shape grammar based approach for product design. *Design studies*, 18(13), 1997.

[5] V. P. Kumar and U. B. Desai. Image interpretation using bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):74–77, 1996.

[6] James A. Landay and Brad A. Myers. Sketching interfaces: Toward more human interface design. *IEEE Computer*, 34(3):56–64, March 2001.

[7] Jianming Liang, Finn V. Jensen, and Henrik I. Christensen. A framework for generic object recognition with bayesian networks. In *Proceedings of the First Interational Symposium on Soft Computing for Pattern Recognition*, 1996.

[8] T. Stahovich, R. Davis, and H.Shrobe. Generalting multiple new designs from a sketch. *Artificial Intelligence*, 104(1-2):211–264, 1998.

[9] George Stiny. Introduction to shape and shape grammars. *Environment and Planning B: Planning and Design*, 7:343–351, 1980.

[10] George Stiny and James Gips. Shape grammars and the generative specification of painting and sculpture. In C V Freiman, editor, *Information Processing 71*, pages 1460–1465. North-Holland, 1972.

[11] M. Tapia. A visual implementation of a shape grammar system. *Environment and Planning B: Planning and Design*, 26(1):59–73, 1999.

[12] K. Tombre. Analysis of engineering drawings: State of the art and challenges. *Lecture Notes in Computer Science*, 1389:257–??, 1997.

[13] M. E. Westling and L. S. Davis. Interpretation of complex scenes using Bayesian networks. *Lecture Notes in Computer Science*, 1352, 1997.