A Retargetable Framework for Interactive Diagram Recognition

Edward H. Lank
Computer Science Department
San Francisco State University
1600 Holloway Avenue
San Francisco, CA, USA, 94132
lank@cs.sfsu.edu

Abstract

The design of new diagram recognition systems remains a challenging problem. Ideally, recognition systems should accept real-world input, perform robustly, fail gracefully, and be implemented in a timely manner. In reality, the intricacy involved in implementing recognition systems for diagram notations makes this a challenging open problem. One solution to these challenges is the design of middleware to speed the development of robust applications. Middleware takes the form of a framework or toolkit for the creation of applications. This paper describes a retargetable framework which can be used to speed the development of robust interactive sketch recognition systems. The system includes a drawing surface to capture interactively created drawings, a set of generic segmentation routines, a character recognizer, and a common interface for integrating domain-specific components. The framework has been used to construct systems for the recognition of UML, math, and molecular diagrams. Work is on-going on the design of additional generic recognizers of logical structure and spatial layout of diagrams.

1. Introduction

Diagram recognition can be separated into two areas: off-line recognition based on document image analysis and on-line or interactive recognition based on the interpretation of digital ink strokes. In both off-line and on-line domains, researchers face the challenge of designing robust recognition systems that work on real input data in a timely manner. In both the off-line and on-line domain, the development of toolkits to support the design of recognition systems would represent a significant step toward accomplishing this task.

This paper describes the implementation of a retargetable framework for interactive diagram and sketch recognition. The framework can be augmented with domain specific recognition components. Recognition systems have been designed to analyze UML diagrams, math notation, and molecular diagrams.

In this paper, previous work in designing retargetable frameworks for the off-line and on-line domain is described in section 2. Section 3 introduces the retargetable framework which has been implemented and describes its targeting to UML, math, and chemical diagrams. Section 4 evaluates the success of the retargetable framework. Finally, section 5 outlines our on-going research.

2. Background

2.1. Off-Line Retargetable Kernels

Figure 1 depicts a model of retargetable systems for diagram image analysis that has been commonly used in the literature. A standard set of libraries is configured via a control module or configuration file to interpret a scanned image of a diagram into a logical representation of the information content in the diagram. The system incorporates a knowledge base which is used by the control structure to guide the application of the generic routines.

Bayer [3] describes a system in which a toolbox of algorithms, a document model and an inference control algorithm work to match the scanned image of a document to the domain specific model of the document structure. The control structure controls and mediates the application of generic routines and the document model to the scanned image. Lam [6] describes a small variation on Bayer's system. In Lam's architecture a knowledge base guides the control structure in its application of generic image processing tools. Lam's system produces a set of object descriptions of the diagram in a database. The database can be queried to obtain attributes of the objects.

Liang et al. [8] describe a toolbox for document image analysis. Their system is an experimental environment to build and test systems. Diagram recognition routines are contained in a toolbox. A configuration file applies these

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 **IEEE**

routines to create information on entities in the diagram image.

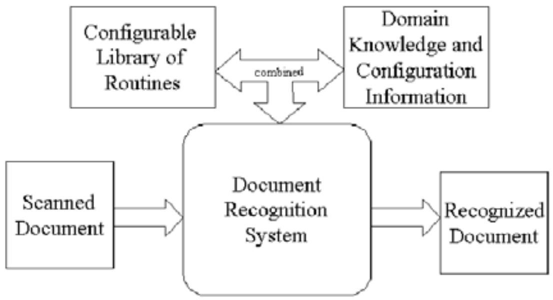

Figure 1 Configurable library view of retargetable frameworks.

Satoh, Mo and Sakuchi describe a rule interpretation kernel for retargetable diagram recognition in [10]. The system allows rules to be specified. The rules are then applied to a scanned diagram image by a rule processor to produce a recognized diagram.

Pasternak describes a recognition kernel called the Adaptive Drawing Interpretation Kernel (ADIK) [9]. The system obtains lines and arcs from a vectorization tool and allows the construction of more complex objects from these primitives on a blackboard. The kernel also contains an interpretation engine for a specification language used to construct the knowledge base of complex objects. The specification language uses geometric predicates and relations to constrain objects, and allows the definition of variables to create "functional links" between objects.

An alternative to configurable libraries for off-line recognition is a system, ScanScribe [11]. This system uses Perceptual Organization, a concept from early stages in human vision concerned with alignment, size similarity, and smooth continuation. Perceptual organization is used to group and analyze scanned sketches. ScanScribe uses this grouping to allow a user to perform intelligent sketch editing.

2.2. Informal Sketch Recognition

Research in the area of informal sketch recognition includes systems designed by Gross and Do, Sezgin et al., and Davis and Alvarado.

The Electronic Cocktail Napkin, designed by Gross and Do, is a sketch based system which supports the creation of and interaction with informal free-form content [5]. The goal is to create a system which allows users to draw as freely as they would on the back of a cocktail napkin, and to create constraints between entities which can then allow natural interaction with the sketch. While the free-form nature of the Electronic Cocktail Napkin is attractive, the constraints that are used by the system are spatial constraints inspired by structures found in architectural drawings.

Sezgin, Stahovitch and Davis explore the general purpose recognition of sketches [12]. The system recognizes various shapes such as rectangles and circles. The recognized shapes can then be beautified by the system, producing a sketch which looks more refined. In many ways, this work resembles that of Arvo and Novins in text entry [2], where they explore a combination of recognition and morphing to produce refined text from hand drawing characters.

Finally, Alvarado, Oltmans and Davis have proposed building a sketch recognition system which supports recognition of a broad subset of diagrams [1]. Their proposed system will be organized as a blackboard system. Knowledge sources, some of which will presumably be domain independent, will recognize aspects of the sketch. Their system is most similar to Pasternak's Adaptive Drawing Interpretation Kernel (ADIK) [9]. Like Pasternak, they propose the use of a shape description language to define complex glyphs for symbol recognition. To target the framework, the user of the framework will presumably use this language to define more complex shapes and groups of shapes. One open problem which will be encountered by their system is managing the complexity of the targeting language.

3. A Retargetable Framework for Interactive Diagram Recognition

We have designed a retargetable framework for the recognition of hand drawn sketches. The framework includes a drawing area, routines for glyph segmentation, a character recognizer, and a common interface for the incorporation of domain specific recognition components. The retargetable framework has been applied to the recognition of hand drawn UML diagrams, hand drawn molecular diagrams, and math notation.

Image capture and the depiction of recognition alternatives are important in the recognition of hand drawn sketches. The retargetable framework includes a standard sketching area which can either be used in stand alone applications or embedded as a panel in other applications. For example, one could imagine creating a stand alone UML recognition system for interpreting hand drawn UML sketches. On the other hand, one could imagine incorporating the drawing area into a software engineering tool such as Exceed, which allows the editing of code and the creation of UML diagrams.

The final task performed by the drawing area is the display of recognition results. The display of recognition results is complicated by two factors. First, there is the need to display domain-specific recognition results in a generic drawing area. Second, recognition results include the results of several different processes, including segmentation, symbol recognition, and syntactic analysis. All these processes contribute to the overall semantic interpretation of the diagram. Symbol recognition may be perfect, but, due to an error in segmentation or the recognition of structure, the diagram may be incorrectly recognized.

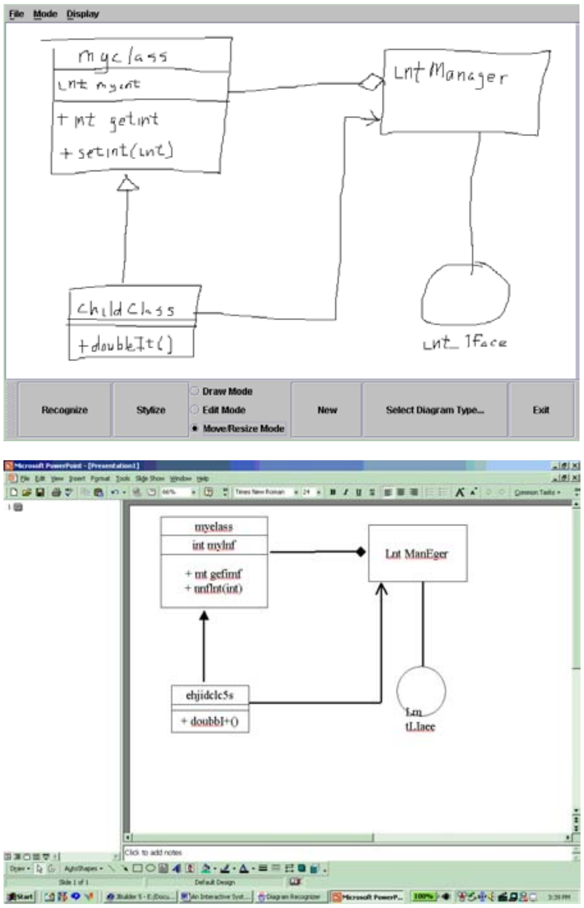

Figure 2 Input UML diagram and PowerPoint output

The second component of our retargetable framework is a set of generic segmentation routines. Glyph segmentation occurs in two phases. In the first phase, the ends of strokes are extended by a small, programmer-specified amount. In free-hand drawing tasks, users do not always draw neatly. It is often the case that users will leave gaps between two strokes which comprise the same glyph. Stroke extension allows the capture of this near-intersection without forcing a user to make an effort to correct their drawing. In the second phase, strokes are examined for intersection. Intersecting strokes are considered part of the same glyph.

Our retargetable framework also incorporates a character recognizer. The character recognizer used is part of the Caltech Interactive Toolkit (CIT), publicly available at http://www.caltech.edu/.

Finally, some mechanism for incorporating domain specific recognition components is needed. To accomplish this, we make two assumptions. First, we assume that there are domain-specific aspects to the sketch. A sketch, it is assumed, is either created with a specific diagram notation (e.g. UML, math, molecular diagrams), is in some way associated with a given domain (e.g. physics free body diagrams, mechanical engineering sketches), or is an informal artifact created during discussion (e.g. hand drawn meeting notes). In the first two cases, context information is required to make sense of the diagram. As well, particularly in the first case, symbols specific to the domain may be present. Finally in the third case, tasks such as note-taking have structure defined by their layout which provides an indication as to meaning. How domain semantics is brought to bear varies according to domain. We provide an interface which can then be implemented by domain-specific recognition components. This allows the system to invoke domain specific components without being aware of their structure or requiring any domain knowledge. These domain specific components return their recognition results in the form of n-best lists which can be used to correct errors in recognition.

Currently in our system, all interaction with third party applications is handled by domain-specific components. The interaction with other applications is highly dependent on domain and on the specific application with which the system is trying to communicate.

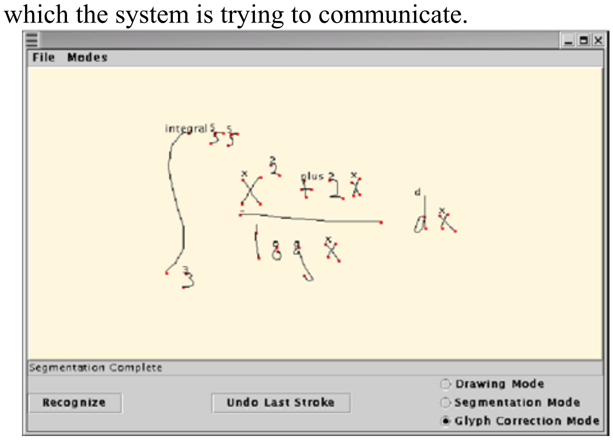

Figure 3 Math recognition system

Using the retargetable framework, recognition systems have been designed for UML diagrams, math notation, and molecular diagrams. The UML recognition system is the most complete. It includes the ability to analyze the structure of UML class, sequence, and use case diagrams and to export an XML description of the UML diagrams. The system also interfaces with Windows applications such as PowerPoint using the Microsoft COM interface. The UML system is shown in figure 2.

The Math recognition system demonstrates the integration with third party recognition systems. A math

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 IEEE

parser, developed by Zanibbi, is added to the system [13]. The parser accepts a list of characters and symbols with spatial positions. The retargetable framework produces input for the parser, and the math expressions are transformed into a LaTeX string by the parser. The tex2gif application is used to produce an image from the LaTeX string. Figure 3 shows output of this system.

Finally, the molecular diagram recognition system provides an initial proof of concept in the recognition of hand drawn molecular diagrams. Figure 4 shows sample output produced by this recognizer.

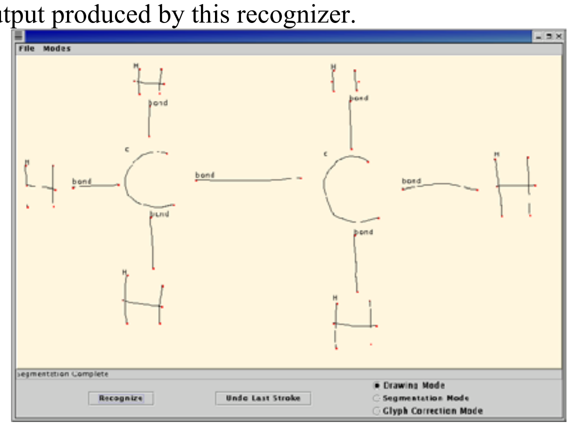

Figure 4 Molecular diagram recognition system

4. Evaluating the Retargetable Framework

Evaluation of middleware is a challenging problem. Examining individual systems created using the middleware is one option. However, this tells us little about the success or failure of the middleware itself.

There are three factors which can be used to determine whether there is an improvement in sketch recognition systems developed using the retargetable framework. These are:

- The development time of new recognition systems;
- The robustness of systems created with the kernel;
- The complexity of systems created with the kernel (are the resulting systems more or less understandable and programmable).

Whether development time is shortened or lengthened depends on a number of factors. We believe that system development proceeds faster with the framework. Measuring this belief is, however, difficult. The framework allows developers to focus on domain-specific aspects of a notation, and the common framework handles those aspects of the recognition problem which are generic across domains. While this should speed development of new recognition systems, it is possible that the cost of understanding the kernel with the goal of using it is higher than the cost of re-implementing the features found in the kernel. Perhaps a description of features is sufficient, and the implemented framework is not an aid in constructing recognition systems.

Robustness faces a similar problem. While claims can be made about the robustness of individual systems (for example, the UML recognition system was used by hundreds of users when demonstrated at the IBM Center for Advanced Studies Research Conference, CASCON, in 2000), claims about relative robustness of systems created with the framework vs. without cannot be made. Is a system created with the framework more (or less) robust because of the framework or in spite of it?

Finally, there are measures of software complexity. Due to a lack of identical systems made with and without the framework, relative claims cannot be made. It is possible, however, to measure the increase in system complexity resulting from the addition of domain-specific components. Table 1 compares the recognition systems to the kernel in terms of lines of code and overall cyclomatic complexity. Note that the math recognition system's measurements include only information on integrating the parser and framework. The complexity of the math parser is not included.

Table 1 Comparison of recognition systems

Application	Lines of Code	Cyclomatic Complexity
Framework	2000	277
UML recognition system	5500	705
Math recognition system	2075	285
Molecular diagram recognition system	2700	350

One question this table raises is whether it is worth pursuing the creation of a retargetable framework. If very simple programming (code with low cyclomatic complexity) is found in the retargetable framework, and the domain-specific recognition components result in a large increase in code complexity, then the retargetable framework provides only limited benefit. The complex, time-consuming tasks involved in development still must be performed.

When examining table 1, we see that for the most complete recognition system, the UML system, the framework comprises only a very small component of the final recognition system. To produce a working recognition system, a significant amount of engineering is required even when using the retargetable framework. On the other hand, the kernel only includes a limited number of features. As the kernel is extended with complex grouping and structuring algorithms, kernel complexity will increase. One would hope to observe a corresponding decrease in recognition system complexity due to the kernel extensions.

5. Discussion and Future Work

To effectively construct a retargetable framework for sketch recognition, it is essential to construct generic recognition components. The challenge in constructing these components is that there is no established idea of what might constitute generic information in a diagram or sketch. Some previous systems, such as Pasternak's kernel [9], have built symbols out of generic primitives using a type of symbol description language. This approach works well for constructing domain-specific symbol recognizers. However, symbol recognition constitutes only part of the process of recognizing the meaning contained in a diagram. Spatial arrangement is very important to meaning in math equations [13]. Logical structure such as containment, connectivity, and alignment create meaning in many informal diagrams [7]. Finally, in diagrams such as UML sequence diagrams, symbols, relationships between symbols (i.e. grouping, containment and connectivity), and spatial arrangement all contribute to meaning. There is a need for generic recognition techniques to represent different aspects of the information contained in a diagram. The generic output can then be mined during domain specific recognition.

Our current work focuses in this area. We are working on perceptual organization to recognize grouping, connectivity, and containment [11] [7]. We are also exploring positional grammars as a representation of the spatial layout of a diagram [4]. It is our hope that higher level representations of the information in a diagram can be constructed prior to applying domain constraints. Domain specific recognition can then make use of the generic information.

Another challenging area in interactive diagram recognition involves the correction of recognition errors. The recognition results contain spatial information, structural information, and symbol recognition. How best to represent recognition results and allow user interaction to correct those results remains an open problem.

This paper describes a retargetable framework for recognizing hand drawn sketches and diagrams. The system currently incorporates algorithms to acquire diagrams, perform some basic segmentation, recognize text characters, and interface with domain-specific recognition components. Work is on-going on the development of more enhanced generic recognizers applicable to a wide variety of hand drawn diagrams and sketches.

Acknowledgements

The author would like to thank Dr. Dorothea Blostein for valuable feedback, and Jeb Thorley and Sean Chen for work on the UML system. This work was supported in part by the Natural Science and Engineering Research Council of Canada, the Center for Excellence for Communication and Information Technology of Ontario, and San Francisco State University.

References

[1] C. Alvarado, M. Oltmans and R. Davis, "A Framework for Multi-Domain Sketch Recognition", Proceedings of the AAAI 2002 Spring Symposium on Sketch Understanding, Stanford, March 25-27, 2002.

[2] J. Arvo and K. Novins, "Fluid Sketches: Continuous Recognition and Morphing of Simple Hand-Drawn Shapes", Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technlogy, San Diego, California, November 2000, pp. 73-80.

[3] T. Bayer, "Understanding Structured Text Documents by a Model Based Document Analysis System", Proc. Second Intl. Conf. Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp. 448-453.

[4] G. Costagliola, S. K. Chang "Parsing Linear Pictorial Languages by Syntax-Directed Scanning", Languages of Design, vol. 2, no. 3, Oct 1994, Elsevier Science Publishers, pp. 229-248.

[5] Mark D Gross and Ellen Yi-Luen Do, "Drawing on the Back of an Envelope", Computers and Graphics, Vol. 24 (6) (2000) pp. 835-849.

[6] S. Lam, "An Adaptive Approach to Document Classification and Understanding", Proc. IAPR Workshop on Document Analysis Systems, Kaiserslautern, Germany, Oct. 1994, pp. 231-251.

[7] E. Lank, R. Augustine, S. Chen, H. Kane, S. Ngo and J. Tutt, "Controlling You Environment Through Sketching", in Adjunct Proceedings of the Conference on Ubiquitous Computing (UBICOMP 2002), Gothenberg, Sweden, Sept. 2002, pp. 13-14.

[8] J. Liang, R. Rogers, R. M. Haralick and I. T. Phillips, "UW-ISL Document Image Analysis Toolbox: An Experimental Environment", Proc. Fourth Intl. Conf. on Document Analysis and Recognition, Ulm, Germany, August, 1997, pp. 984 - 988.

[9] B. Pasternak, "Processing Imprecise and Structural Distorted Line Drawings by an Adaptable Drawing Interpretation Kernel," Proc. IAPR Workshop on Document Analysis Systems, Kaiserslautern, Germany, Oct. 1994, pp. 349-363.

[10] S. Satoh, H. Mo, and M. Sakauchi, "Drawing Image Understanding System with Capability of Rule Learning", Proc. Second Intl. Conf. Document Analysis and Recognition, Tsukuba, Japan, Oct. 1993, pp. 119-124.

[11] E. Saund, J. Mahoney, D. Fleet, D. Larner, and E. Lank, "Perceptual Organization as a Foundation for Intelligent Sketch Editing", Proceedings of the AAAI 2002 Spring Symposium on Sketch Understanding, Stanford, March 25 - 27, 2002.

[12] T. M. Sezgin, T. Stahovich and R. Davis, "Sketch Based Interfaces: Early Processing for Sketch Understanding", Workshop on Perceptive User Interfaces, November, 2001, ACM Digital Library.

[13] R. Zanibbi, D. Blostein and J. R. Cordy. "Baseline Structure Analysis of Handwritten Mathematics Notation", In Proc. Sixth Intl. Conference on Document Analysis and Recognition, Seattle, Washington, Sept. 2001, pp. 768-773.