Dynamically Constructed Bayesian Networks for Sketch Understanding

Christine Alvarado CALVARAD@AI.MIT.EDU
MIT Computer Science and Artificial Intelligence Laboratory, 200 Technology Square, Cambridge MA, 02139 USA

1. Introduction

People sketch to express their early design ideas in many domains, but current computer tools offer few advantages to designers during this sketching phase. Our goal is to construct a general recognition architecture that can be applied to a number of domains that is capable of parsing the user's strokes (in real time) and interpreting them as depicting objects in the domain of interest without limiting the designer's drawing freedom. Such an interpretation engine will enable the creation of powerful and natural early-stage computer aided design tools.

The problem of two-dimensional recognition is to determine which set of known patterns best describes the input the system receives. Because sketched objects rarely appear in their canonical representations (for example, an arrow may point in any direction, and may vary in size), recognizing an object in a sketch involves recovering the underlying shape of the object in the face of a large number of legal transformations, or poses. The process of searching over the possible poses for a shape is often infeasible in real-time, even over a pre-segmented portion of the image.

In contrast, rather than naively matching all possible poses of an object or set of objects to the user's input, our system uses a two-stage generate-and-test method to identify shapes in the user's drawing. In the first stage, our system relies on a rough processing of the user's strokes to generate zero or more likely shape models, each in a set pose, that might explain a portion of the drawing. In the second stage, the system uses a novel application of dynamically constructed Bayesian networks to determine how well each model fits the data, then uses this evaluation to guide further hypothesis generation. This document focuses specifically on our technique for hypothesis evaluation.

2. Approach

We use a hierarchical shape description language to describe the shapes in a domain. In Figure 1, the arrow is an example of a *compound shape*, i.e., one composed of *subshapes* (labelled "Components") fit together according to *constraints*. A line is a *primitive shape*—one that cannot be decomposed further.

As the user draws, the system generates candidate interpretations for the user's strokes based on a rough estimate of the low-level shapes and constraints in the drawing. For example, if strokes a and b both look like lines, and are roughly connected, the system might propose that they are the head and shaft of an arrow. If stroke a is significantly longer than stroke b the system would propose the hypothesis in which a is the shaft, but would not propose a hypothesis in which b is the shaft.

Once hypotheses are generated, the system interprets how well each describes the data. Our approach differs from previous constraint satisfaction approaches (e.g. (Grimson, 1991)) for two reasons. First, our technique must be able to evaluate partially filled hypotheses (e.g. an arrow with no shaft) because we wish to interpret drawings as they develop. Second, because sketches are noisy, we cannot set a hard threshold on whether or not a constraint is satisfied. For example, two lines that appear to connect to form a corner of a square may not actually be connected, and the same data may not appear connected in a different context.

Graphical models handle both of these issues. Missing data can be treated as unobserved nodes in a Bayesian network, while the system assesses likely interpretations for the strokes that have been observed thus far. Furthermore, the system's belief in low-level shapes and constraints can be influenced by both the stroke data and the context in which those shapes or constraints appear.

Time-based graphical models (e.g. Dynamic Bayesian Networks) are not well-suited to our task because we must model shapes based on two-dimensional constraints (e.g. intersects) rather than on temporal constraints, and because our models cannot simply unroll in time as data arrives (we cannot necessarily predict drawing order). The network's fundamental structure must be changed to reflect each new stroke. To allow for the change in structure, we specify Bayesian network fragments that correspond to shape and domain pattern descriptions (e.g. Figure 1). Our fragment representation is similar to the Probabilistic Relational Models proposed in (Getoor et al., 1999).

As the recognition system produces interpretations for the user's strokes, new fragments are instantiated and linked

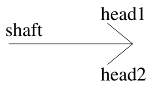
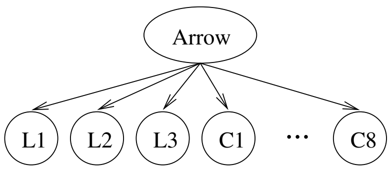
Shape	Description	Network Fragment
	DEFINE ARROW (**Components** *L1,L2,L3*: (Line shaft head1 head2)) (**Constraints** *C1*: (connects shaft.p1 head1.p1) *C2*: (connects shaft.p1 head2.p2) *C3*: (= head1.length head2.length) *C4*: (< head1.length shaft.length) *C5*: (< (angle head1 shaft) 80) *C6*: (< (angle shaft head2) 80) *C7*: (> (angle head1 shaft) 0) *C8*: (> (angle shaft head2) 0))	

Figure 1. The description of the shape "arrow" and the corresponding Bayesian Network fragment.

together to form a complete Bayesian network. Note that a particular fragment may be instantiated any number of times, and each time it is instantiated it refers to a specific hypothesis with a specific mapping from stokes to subcomponents of the object. Nodes representing shapes and constraints are binary; their probability at any given time represents how strongly the system believes that interpretation.

For each primitive shape and constraint, we define a corresponding feature that can be measured from a given stroke or set of strokes and create a node to represent this variable.[1] For example, for lines, the observation feature is the normalized squared error between the stroke and best-fit line. Each primitive shape and constraint, S_i, in the network has a child node, F_i, for its corresponding feature. By collecting low-level data, we have estimated the distribution $p(F_i|S_i)$ for each i. Data enters the network through observations at the feature level.

3. Related Work

Others have used Bayesian networks for image interpretation (Jepson & Mann, 1999; Frey & Jojic, 2000). Our task differs in that sketches are highly stylized, so the problem of locating low-level shapes is lessened, allowing us to use a more efficient hypothesis generation scheme. An approach to sketch interpretation by Shilman *et. al.* takes an approach similar to ours, relying on a Bayesian formulation of the structure of the shapes to be recognized (Shilman et al., 2002). Our work differs in the way the system parses the user's strokes.

4. Current Status and Future Work

We have applied an early implementation of our system to the domain of mechanical engineering and obtained proof-of-concept results. Our system is capable of using context to recover from low-level interpretation errors without blindly trying all interpretations for each stroke. For example, a stroke that does not appear to be a line in isolation can be reinterpreted as a line if it is interpreted as the shaft of an arrow, but this interpretation will be considered only if there are other strokes in the vicinity that roughly meet the constraints for the head of the arrow.

To perform inference in real time, our system system regularly prunes unlikely interpretations. As we expand the system, we will experiment with the pruning threshold to ensure that the system does not prune correct hypotheses.

We are currently expanding the system so that we can collect more substantial performance results and test the system with designers working on realistic tasks. In the longer term, we will explore how to use our recognition system to build a natural early-stage design tool. This exploration will involve determining what type of feedback to display to the user, the amount of error the user is willing to tolerate from the recognition system, and how the system should allow the user to correct those errors.

[1] To increase conditional independence, constraints are calculated in such a way that their value is not dependent on the true interpretation for a stroke, but instead is calculated from the stroke data directly.

References

Frey, B., & Jojic, N. (2000). Learning graphical models of images, videos and their spatial transformations. *Proceedings of UAI '00*.

Getoor, L., Friedman, N., Koller, D., & Pfeffer, A. (1999). Learning probabilistic relational models. *IJCAI* (pp. 1300–1309).

Grimson, W. E. L. (1991). The combinatorics of heuristic search termination for object recognition in cluttered environments. *IEEE Trans. on PAMI*, *13*, 920–935.

Jepson, A., & Mann, R. (1999). Qualitative probabilities for image interpretation. *Proceedings of IEEE ICCV*.

Shilman, M., Pasula, H., Russell, S., & Newton, R. (2002). Statistical visual language models for ink parsing. *AAAI 2002 Spring Symposium on Sketch Understanding* (pp. 126–132).