Efficient Learning of Qualitative Descriptions for Sketch Recognition

Andrew Lovett Morteza Dehghani Kenneth Forbus
{andrew-lovett, morteza, forbus}@northwestern.edu
Qualitative Reasoning Group, Northwestern University
2133 Sheridan Road, Evanston, IL 60201 USA

Abstract

We are trying to solve the problem of learning to recognize objects in an open-domain sketching environment. Our system builds generalizations of objects based upon previous sketches of those objects and uses those generalizations to classify new sketches. We represent sketches qualitatively because we believe qualitative information provides a level of description that abstracts away details that distract from classification, such as exact dimensions. Bayesian reasoning is used in the process of building up representations to deal with the inherent uncertainty in the perception problem. Qualitative representations are compared using SME, a computational model of analogy and similarity that is supported by psychological evidence from studies of perceptual similarity. We produce generalizations based on the common structure found by SME in different sketches of the same object. We report on the results of testing the system on a corpus of sketches of everyday objects, drawn by ten different people.

1. Introduction

The problem of sketch recognition has received a lot of attention in recent years because sketching provides a convenient, natural interface for transferring information from a person to a computer. However, this problem can be extremely difficult because everyone sketches differently and a given person will often sketch the same thing in a different way each time. The key is to identify the properties that remain constant across each sketch of a given object. In order to deal with this quandary, many programs use a narrow domain containing a small set of possible sketch objects (e.g., circuit diagrams: Liwicki and Knipping 2005; simple symbols: Anderson, Bailey, and Skubic 2004; architectural objects: Park and Kwon 2003). Thus, the programmers can go through ahead of time and either hand-code the classifiers themselves or train the classifiers on a large body of data (700 images for Liwicki and Knipping 2005). Even systems designed to work in multiple domains require a certain amount of preprogramming for each particular domain (Alvarado, Oltmans, and Davis 2002). While these types of systems have certainly proven useful, they limit the communication between the person and the computer. Only information that is expected to be encountered in domains that the programmers expect the system to work in can be transmitted.

We believe the key to recognition in the absence of domain expectations is efficient, on-line learning. This means that while a user works with the system, it should be learning from the sketches the user produces, so that when the user sketches an object that has been sketched a few times in the past, it will recognize that object. Such a system has a couple of key requirements. Firstly, there must be some simple way for the user to tell the system what a sketched object is supposed to be. We satisfy this requirement by using sKEA, the *sketching Knowledge Entry Associate*, which is described in greater detail in the next section. Secondly, an algorithm that can learn a new category based on only a few examples is required. It is difficult to learn a category representation for a basic object, such as a house, with only a few examples in the training set. This problem is particularly difficult if one is relying on quantitative information like lengths of edges and angles between edges because this information can vary significantly from one sketch to another. Therefore, we believe qualitative sketch representations are necessary.

There have been several papers in the past that have examined building qualitative representations of images, although few of them have dealt with raw, user-drawn sketches. Museros and Escrig (2004) worked on the problem of comparing closed shapes. Their representations contained descriptions of basic features of the curves and angles in the shapes. Using their representations, they were able to compare two shapes and determine whether one was a rotation of the other. Because their representations were qualitative, they were able to match shapes that differed along irrelevant dimensions, such as absolute size.

Ferguson and Forbus (1999) built a system called GeoRep that generated qualitative representations based on a line-drawing program that allowed users to make perfect lines and perfect curves. GeoRep applied a low-level relational describer to each drawing to find domain-independent qualitative information, such as relative orientation of and connections between lines. GeoRep also used high-level relational describers to extract domain-dependent information from the low-level description. It could perform tasks such as recognizing objects in particular domains and was used as an input for Ferguson's (1994) MAGI, which detects symmetry and regularity, including finding axes of symmetry.

Veselova and Davis (2004) built a system that produced a qualitative representation of hand-drawn sketches. Their representational vocabulary overlapped a fair amount with Ferguson and Forbus'. Their system used several cognitively motivated grouping rules to determine the relative weights of different facts in the representation. The system was designed to produce a representation that could be used to classify other sketches, although the learning and classification stages have not, to the best of our knowledge, been integrated.

We believe the three systems described above provide evidence for the effectiveness of using qualitative information to represent and compare sketches. However, these systems lack the ability to learn robust categories of sketches based on more than one example. In the following sections, we will describe our system, which we believe takes a step towards accomplishing this goal.

2. The Sketching Environment

sKEA is an open-domain sketch understanding system (Forbus et al. 2004). It is able to reason about user-drawn sketches without any domain expectations of what a user is likely to sketch because it is not dependent on sketch recognition. Rather, it is based on the idea that when people communicate through sketching, their communication is a multi-modal process. People verbally describe what they are sketching as they create it. Similarly, sKEA allows users to label each glyph, or object in a sketch, with categories from its knowledge base. sKEA computes a number of spatial relations between glyphs in a sketch, and it uses this information along with its knowledge about the categories of the glyphs to reason about a sketch, or to compare two sketches. sKEA has been used in a number of experiments that involve spatial reasoning in different domains (Tomai et al. 2005; Lockwood, Forbus, and Usher 2005; Klenk et al. 2005).

Of course, sKEA's performance does not match human reasoning about sketches. While humans do often describe what they are sketching, they also expect others to recognize some objects without having to be told what they are, particularly if those objects have already been discussed, or sketched, in the past. Thus, it is not surprising that sKEA's requirement that every glyph be labeled can become onerous at times, especially if the user is performing a task that requires the same object to be sketched and labeled many times. This concern leads to the question of whether some type of sketch recognition can be added to sKEA without sacrificing its domain independence.

The key to domain-independent recognition is learning. When a user begins using sKEA to perform some task, sKEA should have no expectations about what the user will sketch. However, over time, if the user sketches the same object more than once, sKEA ought to learn to recognize that object. Thus, the fourth time the user draws, say, a building, sKEA could generate a guess as to what that object is most likely to be. If that guess is wrong, the user can always perform the usual glyph labeling task to correct it, just as a person would correct another person who misunderstood part of a sketch. We see any sketching session as an opportunity for sKEA to learn to recognize objects in parallel with the user's sketching of those objects.

In order for sKEA to learn to recognize objects, three other components are required: a system for building representations of sketched shapes, a system for learning representations of categories of shapes, and a system for comparing a new shape's representation to the category representations in order to classify it. We will describe the comparison and learning components in the next section.

3. Comparison and Generalization

We compare representations using the Structure-Mapping Engine (SME) (Falkenhainer, Forbus, and Gentner 1989). SME is a computational model of similarity and analogy based on Gentner's (1983) structure-mapping theory. According to structure-mapping, we draw analogies between two cases by aligning their common structure. Each case's representation contains entities, attributes of entities, and relations. Structure is based on the connections between elements in the representation. A simple relation between two entities has a small amount of structure, whereas a more complex relation between other relations in the representation has a deeper structure.

SME takes as input two cases: a base case and a target case. It finds possible correspondences between entities, attributes, and relations in the two cases. It combines consistent correspondences to produce mappings between the cases. SME attempts to find mappings which maximize systematicity, the amount of structural depth in the correspondences. SME also produces candidate inferences about the target by identifying attributes and relations in the base that lack corresponding elements in the target.

Our system learns categories of objects using SEQL (Kuehne, Forbus, and Gentner 2000), a model of generalization built on SME. The idea behind SEQL is that people form a representation of a category by abstracting out the common structure in all the exemplars of that category. In its default mode, SEQL works in the following way: when it encounters a new case, it uses SME to compare that case to the known generalizations. If the new case aligns with a sufficient amount of the structure in one of the generalizations, the case is added to that generalization. Any part of the generalization's structure that does not align with the new case is removed, so that the generalization continues to represent only the structure found in all of its exemplars.

A recent update to SEQL associates a probability with each fact in the generalization, allowing for greater flexibility (Halstead and Forbus 2005). When a new case is added to a generalization, those parts of the generalization that do not align with the case are not automatically removed, but instead have their probability reduced. When a fact's probability falls below a threshold, it is removed from the generalization.

SEQL is capable of quickly learning new generalizations. Even a generalization based on a pair of exemplars may be sufficient for classifying new cases. Each additional exemplar further refines the generalization by allowing SEQL to remove facts that are not important for membership in the category.

4. Perceptual Elements

We now come to the problem of building representations from sketches. Our system works by decomposing a sketch into a set of primitive perceptual elements. There are two types of primitive elements: segments and terminations. These elements align with elements of the raw primal sketch in Marr's (1982) theory of vision. Segments may be straight or curved. Terminations, which exist at the endpoints of segments, may be classified as corners, meaning there is a corner between two segments; connections, meaning they connect two collinear segments; or neither. Once the primitive elements are found, they can be grouped to form more complex elements. Thus, there is an element hierarchy. This strategy of segmenting an image into primitive elements and then grouping them has been used successfully in the sketching domain by Saund et al. (2002). Thus far, there is only one level to the hierarchy. Segments and their terminations can be grouped to form edges. While there are rules for grouping edges, there are no explicit structures for more complex perceptual elements at this time.

Our system begins with the raw output from sKEA, consisting of a list of polylines. Each polyline is a list of points corresponding to a line drawn by the user. The system does not assume that the endpoints of polylines match endpoints of edges in the shape sketched by the user. Rather, it begins by joining together polylines with adjacent endpoints, provided there is no third adjacent polyline to create ambiguity.

The system then searches for discontinuities in the slope of each polyline, representing potential corners. Discontinuities are a key concept at every level in Marr's (1982) model, and they provide vital information about the location of terminations. In our system, evidence for a discontinuity includes both changes in the overall orientation and high values for the derivative of the slope of the curve, as calculated by Lowe (1989). Polylines are divided into segments which are linked by termination points anywhere there is a sufficiently salient discontinuity.

The system also finds potential corners and connections between segments from separate polylines whose endpoints are not adjacent. Two segments may have a corner between them if extending the lines beyond their endpoints would result in an intersection at some point in space. They may have a connection between them if they are collinear.

Once the system has located termination points and gathered evidence, the termination points must be classified. Previous systems have used Bayesian Networks (BNets) to deal with uncertainty in perception (Bokor and Ferguson 2004; Alvarado and Davis 2005). Following this precedent, we use BNets to determine whether termination points are corners, connections, or neither. Our system uses Recursive Conditioning (RC) (Darwiche 2001) to perform exact inference on the network and to calculate the probabilities. RC is an any-space algorithm which works by recursively partitioning the network into smaller networks using conditioning and solving each subnetwork as an independent problem.

Once termination points have been classified, segments can be grouped together to form edges. Edges consist of maximal lists of unambiguously connected segments. Segments are unambiguously connected if there is a termination between them that has been classified as a connection and if the connected endpoints of the two segments are not linked by connections or corners to any other segments. The threshold for connection detection is lowered if the segments to be grouped form a compatible curve.

Edges inherit connection information from the segments upon which they are built. Thus, edges whose segments were connected will themselves be connected. This connection information is used by the system to group edges into *cyclic edge groups*. A cyclic edge group is a list of sequentially connected edges in which the first and the last edge are connected. These edge groups represent closed shapes in the sketch. For example, a square would be a cyclic edge group containing four edges. Once the edges and edge groups have been computed, the system is ready to build a qualitative representation of the sketch.

5. Qualitative Vocabulary

An appropriate qualitative vocabulary is vitally important for any kind of comparison between sketches. If the vocabulary fails to capture the key properties of each sketch, there will be no way to determine whether two sketches are similar. In building our qualitative vocabulary, we began by examining the vocabularies used by Ferguson and Forbus (1999), Museros and Escrig (2004), and Veselova and Davis (2004), who shall henceforth be referred to as FF, ME, and VD. We then sought out additional features that would be useful for representing the full range of objects that people might choose to sketch.

Most predicates (attributes and relations) convey information about only one or two objects and contain relatively little structural depth. Because SME uses structure to match two representations, it is difficult to find corresponding entities using only these predicates, particularly when there is a large number of them in each representation. Thus, it is helpful to have *anchoring* relations. These relations, which convey information that we believe is particularly salient in the match, contain greater structural complexity. Because of SME's systematicity bias, they are generally the first relations SME matches up. Thus, they anchor the rest of the mapping.

Basic Elements

Before relations between elements in a sketch can be encoded, it is necessary to encode the basic elements themselves. In our case, we begin by creating entities for the edges found in the sketch. These entities must be classified according to the type of edge. FF, ME, and VD all draw a distinction between straight and curved edges, with FF allowing for three types of entities: lines (**straight** edges), arcs (**curved** edges), and **ellipses** (closed shapes consisting of a single curved edge). We follow FF in classifying our edges as one of these three types. In addition, we follow VD in asserting **vertical** or **horizontal** attributes for straight lines that align with the y or x axes, as axis-aligned edges appear to be particularly salient.

Pairwise Relations

There are a number of pairwise relations between edges that can provide useful information about a shape. FF and VD in particular use a large number of these. We have implemented relations for the relative position (**left-of** or **above**), relative length (**same-length** or **longer-than**), and relative orientation (**parallel** or **perpendicular**) of pairs of edges. Like FF, we distinguish between parallel edges with and without common extent (i.e., parallel edges that do or do not overlap along the axis traveling parallel to them).

One major concern with pairwise relations is determining the pairs of edges for which relations will be asserted. Asserting relations between every pair of edges in a shape results in an overly complex representation with a large number of redundant or irrelevant facts. There are several heuristics for determining appropriate pairs of edges. FF use simple proximity between the edges. VD use a few more complicated grouping strategies. We follow VD in grouping edges together by adjacency. That is, pairwise relations are asserted between two edges if there are no other edges blocking the path between those two edges. We use a Voronoi diagram (Edwards and Moulin 1998) to find all the adjacent edges in the glyph. We further limit the relative length relations by only asserting relative length for pairs of edges that are parallel or perpendicular, as these orientations make the relative length more salient.

Connections & Closure

FF, ME, and VD all make use of connections between edges in their representations. Connections between edges, and particularly corners between edges (connections that occur at the edges' endpoints), are key to recovering the spatial structure of most shapes. We use a general **connected** relation any time two edges are connected to allow connections of different types to potentially align. However, we also classify the connections into three types: **corner**, **connects-to** (where one edge's endpoint touches the middle of another edge), and **intersection** (when two edges meet between both their endpoints). Thus, due to the structural mapping process, matches between connections of the same type are much stronger than matches between connections of different types.

The concept of closure, wherein a group of connected edges together make a full cycle, thus creating a closed shape, plays an important role in FF's representation. As FF point out, there is evidence that humans compute closure early on in perceptual processing (Treisman and Paterson 1984). FF make assertions about polygons and about the convexity of the corners in a polygon. ME, whose representations always describe closed shapes, also make assertions about the convexity of corners. Our system uses cyclic edge groups to compute the **convexity** of any corners that make up part of a cycle. We also assert special relations for any **three-sided** or **four-sided closed shapes** in a sketch. Because these relations include information about corners between several different edges, they contain much more structure than any of the relations discussed above. Thus, they become anchoring relations in SME's matching process.

Junctions

All of the predicates described thus far are useful for processing two-dimensional shapes. However, our system also needs to be able to process three-dimensional shapes, should a user choose to sketch such shapes. One feature that has proven useful for comprehending three-dimensional shapes is junctions. Junctions are points where the endpoints of two or more edges meet. The layout of the junction can provide important cues to the relative orientations of surfaces on a three-dimensional object. Clowes (1971) classified junctions into four types, three of which were junctions between three edges, and showed how the relations between junctions could be used to recover shape information in simple line drawings.

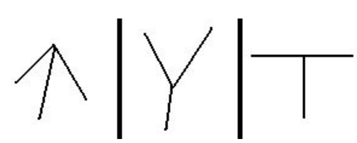

Figure 1. From left to right: an arrow junction, a fork junction, and a tee junction

We assert **junction** relations for points in a sketch where exactly three edges met. We classify junctions into three types which align with the types described by Clowes (see Figure 1): **arrow junctions**, **fork junctions**, and **other junctions**, which could include tee junctions or other similar junctions. Because groups of related junctions, rather than individual junctions, are necessary for recovery the shape of an object, we also assert positional relations between junctions (**above** and **left-of**). These relations are structurally deep, and so they also act as anchoring relations.

Organization of Facts

Unfortunately, we found that when complex shapes were analyzed, the representations based on the vocabulary described above became far too large for SME and SEQL to handle (600+ facts). We concluded that it was necessary to limit the number of facts that would actually be encoded. Therefore, we order the facts in our representation according to priority. We give the highest priority to facts about external edges, i.e., edges that touch the outer bounds of the glyph itself. These edges seem to be the most important for encoding and recognizing an object because of the role they play in determining the overall shape of the object. We give the second-highest priority to edges connected to external edges by corners and the third-highest priority to other edges that are part of external edge groups, sequences of connected edges in which at least one edge is external. We do not assert relations for edges that are not part of external edge groups. These internal edges were deemed unimportant for recovering the overall shape of the sketch. Once facts have been appropriately ordered, we can cut off the list of facts in a representation at different points depending on how large we want to allow the representations to grow.

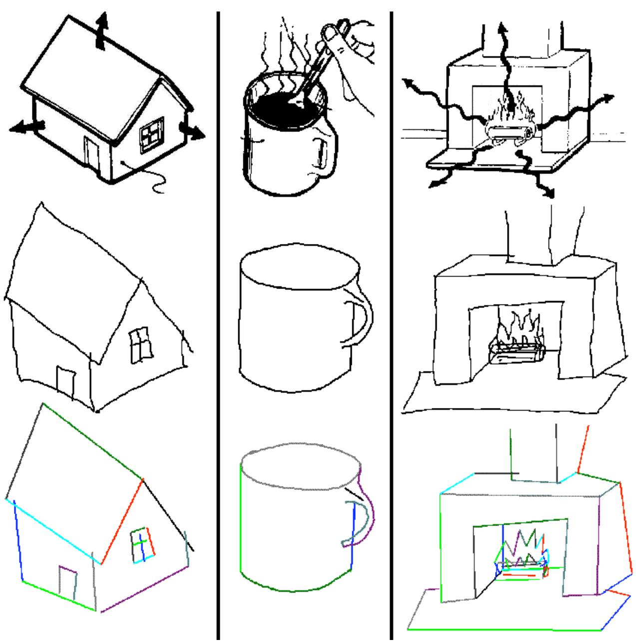

Figure 2. Sample objects. The top row are the illustrations of the object. The second row are sketches produced by subjects. The third row are the edges found by the system.

6. Experiment

We evaluated our system by testing its ability to build generalizations of sketches of 8 everyday objects: a house, a fireplace, a brick, a cup, an oven, a cylinder, a refrigerator, and a bucket.. The objects were selected from *Sun Up to Sun Down* (Buckley 1979), a book that uses simple drawings to illustrate physical processes such as heat transfer. 10 subjects were instructed to sketch each object using the drawings from the book as guides (see Figure 2). The drawings were provided so that the general features and orientations of the sketches would be similar. However, subjects were told that they needed only sketch those parts of the object that they believed were necessary for a person to recognize it. See Figure 3 for four sketches of the fireplace.

Subjects sketched the objects in sKEA using a Wacom tablet, a pen-based device that takes the place of a mouse. Of the 10 subjects, 5 had previous experience working with sKEA, and 6 had previous experience with a pen-based device. After subjects sketched the objects, each object was labeled by the experimenter using sKEA's interface.

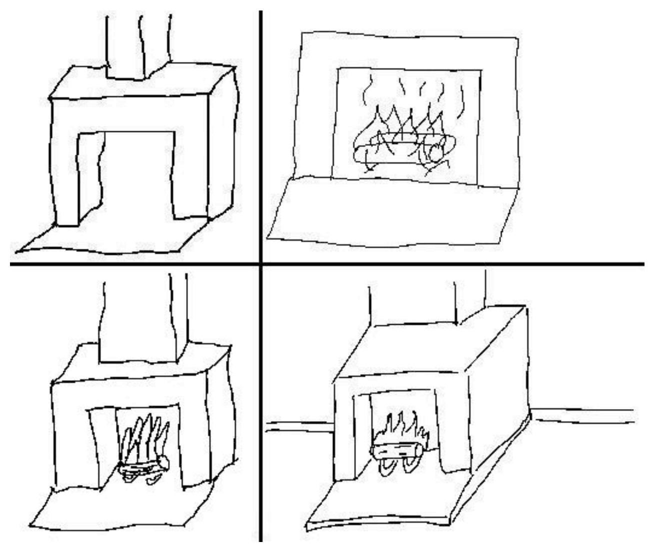

Figure 3. Four fireplace sketches drawn by subjects

We chose to throw out one subject's set of sketches because the subject failed to follow the instructions. The remaining 72 sketches were used to test the system. In each test run, generalizations for the 8 objects were built based on sketches by 5 of the subjects (the training set). Although SEQL can determine generalizations automatically, our system forced SEQL to build exactly one generalization for the 5 training sketches of each object.

After the generalizations were built, they were used to classify the objects in the other 4 sketches (the test set). A given object was classified by comparing its representation to each of the 8 generalizations and returning the generalization with the closest match. We determined empirically that classification results were the most accurate when the number of candidate inferences was used to evaluate the strength of a match. We used SME to match a generalization to the new case and compute candidate inferences about the new case based on the generalization. These candidate inferences represented identifiable differences between the generalization and the new case, in particular facts found in the generalization but not in the new case. Matches with fewer candidate inferences were considered better. Thus, the strength of a match was based on how well the new case covered everything that was expected to be found in instances of the generalization. Because some generalizations were larger than others and thus likely to produce more candidate inferences, we divided the number of candidate inferences by the total number of expressions in the generalization. This heuristic had an effect of approximately normalizing the match scores.

We validated our results by averaging the scores over 20 test runs. In each run, the sketches were randomly divided into training and test sets. Our training algorithm rebuilt the generalizations from scratch using each run's training set. Because we were unsure how limiting the number of facts in a representation would affect the results, we ran the test four times with four different limits on the number of facts.

Preliminary tests indicated that many of the classification mistakes made by the system involved a failure to distinguish between the three cylindrical objects: cups, cylinders, and buckets. This is hardly surprising, as these three objects have similar shapes with few distinguishing characteristics. Therefore, we used two criteria in reporting our results. According to the strict criterion, only an exact match between an object's actual type and its classified type was considered a correct classification. According to the weak criterion, a classification in which the two types did not match was still considered correct when both were cylindrical types.

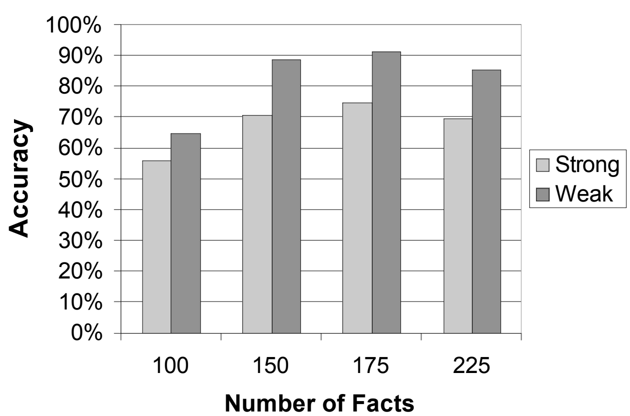

Figure 4. Classification results

The results can be found in Figure 4. The best performance was achieved when representations were limited to 175 facts. With this limit, the strong criterion was met 74.5% of the time and the weak criterion was met 91.1% of the time. Note that chance performance with the strong and weak criteria would be 12.5% and 21.9% respectively. Performance dropped when either more facts or fewer facts were used.

7. Discussion

We believe we have demonstrated the effectiveness of our system in learning to classify sketches of simple, everyday objects. While the number of types of objects for classification was not large, the objects varied significantly in terms of shape and complexity. Most importantly, the system worked with no prior knowledge of the object classes for which it learned generalizations. Based only on the 5 sample objects for each type, it was able to build generalizations that were sufficiently robust to classify new objects about 75% of the time. Furthermore, when confusion between the different types of cylindrical objects was discounted, the system was able to classify objects more than 90% of the time.

We were somewhat concerned that limiting the number of facts that could be included in a representation would hamper performance. Based on the results achieved, this does not appear to be the case. Performance with 175 facts was actually better than performance with a larger number of facts. This is an interesting result that deserves further exploration. Given the ordering of facts in the representations, those facts beyond the first 175 were probably less relevant for recovering the shape of the sketch. As such, these facts may have varied more between different subjects' renditions of the same object. Thus, they may have added noise to the representations, resulting in a more difficult matching process.

Of course, one would expect the optimal number of facts to vary depending on the complexity of the shape being represented. However, given the range of the shapes used for this experiment, with the number of facts for a shape ranging from 60 to 600, we believe the results support 175 being a good general cutoff for the qualitative representation scheme used in this experiment.

8. Future Work

Our system, as used in this experiment, assumes that each new sketch must match one of the previously learned generalizations. Obviously, this will not always be the case. The ability to recognize that a new object is novel instead of forcing it into a category would be useful. This recognition could be based on a threshold for structural evaluation scores in the SME matches between new cases and previous generalizations. If the score between the new case and a generalization fell below that threshold, the system would not accept the match even if it scored better than the matches to all the other generalizations.

There is evidence that basic closed shapes, such as triangles, are detected early in the human perceptual process and that these shapes are themselves treated as properties of an image (Treisman and Paterson 1984). In the present study, we represented triangles and quadrilaterals only at the qualitative stage. However, we are currently in the process of developing more complex perceptual elements, so that we can represent closed shapes earlier in the perceptual pipeline. This will allow us to utilize BNets for recognizing basic shapes. The BNets will be used in a hierarchical architecture, where the probabilities for basic elements such as corners and connections are calculated, and then these values along with other evidence are fed to the nets used for recognizing shapes.

Another expansion we are considering is the ability to classify multiple objects in a single sketch. The present system assumes that the entire sketch represents only a single object, but obviously this is not always the case. sKEA allows users to manually distinguish between objects in a sketch by drawing them as separate glyphs, but like manual glyph labeling, this adds to the user workload.

Automatic object segmentation in a sketch depends on intelligent rules for grouping edges together. Obviously, one rule for grouping edges could be based on grouping connected edges together, as our system already does. This rule would be sufficient provided all the edges of each object in a sketch were connected, and none of the edges of different objects were connected. A further refinement might be to include edges located inside the bounds of connected edge groups.

Finally, we plan to incorporate our system with sKEA so that it will be running in the background while users are sketching. Thus far, we have only shown that the system works in an experimental setting. We hope to demonstrate that it is actually useful to sKEA users while running in real time. The interaction between the user, the system, and sKEA will create an environment in which we believe open-domain sketch recognition can become a reality.

Acknowledgments

This research was supported by a grant from the Computer Science Division of the Office of Naval Research.

References

Alvarado, C., Oltmans, M., and Davis, R. 2002. A Framework for Multi-Domain Sketch Recognition. In *2002 AAAI Spring Symposium on Sketch Understanding*. Palo Alto, CA.

Alvarado, C., and Davis, R. 2005. Dynamically Constructed Bayes Nets for Multi-Domain Sketch Understanding. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 1407-1412. Edinbourgh, Scotland.

Anderson, D., Bailey, C., and Skubic, M. 2004. Hidden Markov Model Symbol Recognition for Sketch-Based Interfaces. In *Making Pen-Based Interaction Intelligent and Natural*, 15-21. Arlington, VA: AAAI Press.

Bokor, J. L., and Ferguson, R. W. 2004. Integrating Probabilistic Reasoning into a Symbolic Diagrammatic Reasoner. In *Proceedings of the 18th International Workshop on Qualitative Reasoning (QR'04)*. Evanston, IL.

Buckley, S. 1979. *Sun Up to Sun Down*. McGraw Hill: New York.

Darwiche, A. 2001. Recursive Conditioning. *Artificial Intelligence* 126: 5-41.

Edwards, G., and Moulin, B. 1998. Toward the Simulation of Spatial Mental Images Using the Voronoi Model. In Oliver, P. and Gapp, K. P. (Eds.), *Representation and Processing of Spatial Expressions*. Mahwah, NJ: LEA.

Falkenhainer, B., Forbus, K. and Gentner, D. 1989. The Structure-Mapping Engine: Algorithms and Examples. *Artificial Intelligence* 41: 1-63.

Ferguson, R. W. 1994. MAGI: Analogy-based Encoding Using Regularity and Symmetry. In *Proceedings of the 16th Annual Conference of the Cognitive Science Society*, 283-288. Atlanta, GA

Ferguson, R. W., and Forbus, K. D. 1999. GeoRep: A Flexible Tool for Spatial Representations of Line Drawings. In *Proceedings of the 13th International Workshop on Qualitative Reasoning (QR'99)*, 84-91. Loch Awe, Scotland.

Forbus, K., Lockwood, K., Klenk, M., Tomai, E., and Usher, J. 2004. Open-Domain Sketch Understanding: The nuSketch Approach. In *AAAI Fall Symposium on Making Pen-based Interaction Intelligent and Natural*, 58-63. Washington, DC: AAAI Press.

Gentner, D. 1983. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science 7*: 155-170.

Halstead, D., and Forbus, K. 2005. Transforming between Propositions and Features: Bridging the Gap. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI'05)*. Pittsburgh, PA: AAAI Press.

Klenk, M., Forbus, K., Tomai, E., Kim, H., and Kyckelhahn, B. 2005. Solving Everyday Physical Reasoning Problems by Analogy Using Sketches. In *Proceedings of 20th National Conference on Artificial Intelligence* (*AAAI'05*). Pittsburgh, PA: AAAI Press.

Kuehne, S., Forbus, K., Gentner, D. and Quinn, B. 2000. SEQL: Category Learning as Progressive Abstraction Using Structure Mapping. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, 770-775. Philadelphia, PA.

Liwicki, M., and Knipping, L. 2005. Recognizing and Simulating Sketched Logic Circuits. In *Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information & Engineering Systems*, 588 – 594. Melbourne, Australia: LNCS.

Lockwood, L., Forbus, K., and Usher, J. 2005. SpaceCase: A Model of Spatial Preposition Use. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society*, 1313-1318. Stresa, Italy.

Lowe, D. G. 1989. Organization of Smooth Image Curves at Multiple Scales. *International Journal of Computer Vision* 3(2): 119-130.

Marr, D. 1982. *Vision*. W.H. Freeman and Company: New York.

Museros, L., & Escrig, M. T. 2004. A Qualitative Theory for Shape Representations and Matching. In *Proceedings of the 18th*

International Workshop on Qualitative Reasoning (QR'04). Evanston, IL.

Park, J., and Kwon, Y-B. 2003. Main Wall Recognition of Architectural Drawings Using Dimension Exension Line. In *Proceedings of the Fifth IAPR International Workshop on Graphics Recognition (GREC'03)*, 116-127. Barcelona, Spain: Springer.

Saund, E., Mahoney, J., Fleet, D., Larner, D., & Lank, E. 2002. Perceptual Organization as a Foundation for Intelligent Sketch Understanding. In *2002 AAAI Spring Symposium on Sketch Understanding*, 118-125. Palo Alto, CA.

Tomai, E., Lovett, A., Forbus, K., and Usher, J. 2005. A Structure Mapping Model for Solving Geometric Analogy Problems. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society*, 2190-2195. Stresa, Italy.

Treisman, A., and Patterson, R. 1984. Emergent Features, Attention, and Object Perception. *Journal of Experimental Psychology: Human Perception and Performance* 10(1): 12-31.

Veselova, O., and Davis, R. 2004. Perceptually Based Learning of Shape Descriptions. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI'04)*, 482-487. San Jose, CA: AAAI Press.