Incremental Learning of Perceptual Categories for Open-Domain Sketch Recognition

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

Abstract

Most existing sketch understanding systems require a closed domain to achieve recognition. This paper describes an incremental learning technique for open-domain recognition. Our system builds generalizations for categories 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 building representations to deal with the inherent uncertainty in perception. Qualitative representations are compared using SME, a computational model of analogy and similarity that is supported by psychological evidence, including studies of perceptual similarity. We use SEQL to 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 much attention in recent years because sketching provides a convenient and natural interface for transferring information from a person to a computer. This problem can be extremely difficult because everyone sketches differently and a single 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 *et al*., 2004; architectural objects: Park and Kwon, 2003]. Thus, the programmers can examine the domain 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 *et al.,* 2002]. While these types of systems have certainly proven useful, they limit the communication between the person and the computer. Only information based 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 in the past, it will recognize that object. Such a system has a couple of key requirements. Firstly, there must be a simple way for the user to tell the system what a sketched object is supposed to be. Secondly, an algorithm that can learn a new category based on only a few examples is required. This is difficult if one is relying on quantitative information about lengths and angles because this information can vary significantly from one sketch to another. Therefore, we believe qualitative sketch representations are necessary.

Several efforts have examined building qualitative representations of images, although few have dealt with raw, user-drawn sketches. Museros and Escrig [2004] worked on comparing closed shapes. Their representations contained descriptions of basic features of the curves and angles in the shapes. They were able to compare two shapes and determine whether one was a rotation of the other.

Ferguson and Forbus' [1999] GeoRep generated qualitative representations based on a line-drawing program that allowed users to make perfect lines and 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 was used for recognizing objects in particular domains and identifying axes of symmetry.

Veselova and Davis [2004] built a system that produced a qualitative representation of hand-drawn sketches. Their representational vocabulary overlapped somewhat 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 representations for classification, 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 multiple examples. Here, we describe our system, which we believe takes a step towards accomplishing this goal. We begin with the sketching environment in which our system operates. Next, we give a brief overview of the applications we use for comparing sketches and constructing generalizations. We then describe how our system decomposes rough sketches into perceptual elements and how those elements are represented using a qualitative vocabulary. Finally, we discuss the results from an experiment designed to test our system and consider areas for future work.

2 The Sketching Environment

Our system uses sketches drawn in sKEA, the *sketching Knowledge Entry Associate.* 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.

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. 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 objects 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 domain independence.

Our approach to domain-independent recognition is based on incremental 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 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 objects, a system for learning perceptual categories of objects, and a system for comparing a new object's representation to the category representations in order to classify it. We will describe the comparison and learning components in the next section.

3 Comparisons and Generalization

We compare representations using the Structure-Mapping Engine (SME) [Falkenhainer *et al.,* 1989]. SME is a computational model of similarity and analogy based on Gentner's [1983] structure-mapping theory. According to structure-mapping, humans 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 has a deeper structure.

SME takes as input two cases: a base and a target. 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.

Our system learns categories of objects using SEQL [Kuehne *et al*., 2000; Halstead and Forbus, 2005], a model of generalization built on SME. SEQL is based on the theory that people form a representation of a category by abstracting out the common structure in all the exemplars of that category. SEQL uses SME to compare new cases to the known generalizations. If a new case aligns with a sufficient amount of the structure in one of the generalizations, the case is added to that generalization. SEQL associates probabilities with each expression in a generalization, representing the proportion of the instances of that generalization that include that particular expression. When a new case is added to the generalization, if its structure does not align with an expression in the generalization, that expression's probability is decremented.

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.

4 Perceptual Elements

Our system decomposes a sketch into a set of primitive perceptual elements. There are two types of primitive elements: segments and the endpoints of segments. These elements align with elements of the raw primal sketch in Marr's [1982] theory of vision. Segments may be straight or curved. Endpoints may be classified as corners, meaning

there is a corner between two segments; connections, meaning they connect two collinear segments; or terminations, meaning the endpoint does not connect to another segment. Once the primitive elements are found, they can be grouped to form more complex elements, creating an element hierarchy. So 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 currently no explicit structures for more complex perceptual elements.

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. 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 segment endpoints. 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 endpoints 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 endpoints and gathered evidence, the endpoints must be classified. Previous systems have used Bayesian Networks (BNets) to deal with uncertainty in perception [Bokor and Ferguson, 2004; Alvarado and Davis 2005]. We utilize BNets which use the evidence gathered about an endpoint to classify it as a corner, connection, or termination

After endpoints 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 an endpoint 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 *connected edge groups*, lists of sequentially connected edges. A *cyclic edge group* is a connected edge group 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 uses this information to build a qualitative representation of the sketch.

5 Qualitative Representation

An appropriate representational vocabulary is crucial 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. Our qualitative vocabulary draws on the work of Ferguson and Forbus [1999], Museros and Escrig [2004], and Veselova and Davis [2004].

The terms in our vocabulary can be divided into three types: *attributes*, *pairwise relations*, and *anchoring relations*. Attributes convey information about a single edge in the sketch. Pairwise relations describe a relationship between two edges. Because these first two types of terms can apply to only one or two entities in the representation, they contain relatively little structural depth. SME uses structure to match two representations, so it is difficult to find corresponding entities using only these predicates, particularly when there is a large number of them in each representation. Thus, anchoring relations are necessary. Anchoring relations, which convey information that we believe is particularly salient in the match, refer to more than two edges, and 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.

Attributes describe an edge's type. An edge can be classified as **straight**, **curved**, or **elliptical**, where an elliptical edge is a curved edge that closes on itself, such as a circle. In addition, straight edges that align with the x or y axes are assigned the **horizontal** or **vertical** attributes.

Pairwise relations describe the relative position (**left-of** or **above**), relative length (**same-length** or **longer-than**), or relative orientation (**parallel** or **perpendicular**) of pairs of edges. 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 sketch results in an overly complex representation with a large number of redundant or irrelevant facts. We follow Veselova and Davis [2004] in only asserting pairwise relations between adjacent edges. We further limit the relative length relations between straight edges by only asserting relative length for pairs of edges that are parallel or perpendicular.

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). We also use cyclic edge groups to compute the **convexity** of any corners that make up part of a closed shape.

We assert two types of anchoring relations. Firstly, we use the cyclic edge groups to find any **three-sided** or **four-sided closed shapes**, i.e., triangles or quadrilaterals. These shapes are important because they often make up the surfaces of three-dimensional objects, and because there is evidence that humans identify basic shapes early on in perceptual processing [Ferguson and Forbus, 1999].

Secondly, we assert **junction** relations for points in a sketch where exactly three edges meet. Clowes [1971] demonstrated that junctions between edges provide useful information for recovering shape information from line drawings. We classify junctions into three types described by Clowes: **arrow junctions**, **fork junctions**, and **tee junctions**, as well as a fourth, **other** type. We also assert positional relations between junctions.

5.1 Organization of Facts

Unfortunately, we found that when complex shapes were analyzed, the representations based on the vocabulary described above became unmanageably large (600+ facts). Consequently we limit the number of facts that are allowed in a representation. We order the facts in our representation according to a ranking system. 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.

Facts are ranked based on both the qualitative term and the edges being described. Among qualitative terms, anchoring relations are ranked above other relations due to their importance in the mapping. Among edges, the highest ranking is given to external edges, those that reach the outer bounds of the entire sketch. These edges are considered the most important because the outer edges of an image convey vital information about the shape which the image represents [Hoffman and Richards, 1984]. The lowest ranking is given to purely internal edges, those that are not part of any connected edge group containing an external edge. Presently we do not assert relations between internal edges.

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], which uses simple drawings in teaching about solar energy. 10 subjects were instructed to sketch each object using the drawings from the book as guides. 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. On examining the sketches drawn by subjects, we found significant cross-subject differences in the sketches, although most of the sketches of each object shared a core set of similarities (see Figure 1).

Subjects sketched the objects in sKEA. Of the 10 subjects, 5 had previous experience working with sKEA. After subjects sketched the objects, each object was labeled by the experimenter using sKEA's interface.

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 a subset of the 9 users (the training set). Although SEQL can determine generalizations automatically, our system forced SEQL to build exactly one generalization from the training sketches of each object.

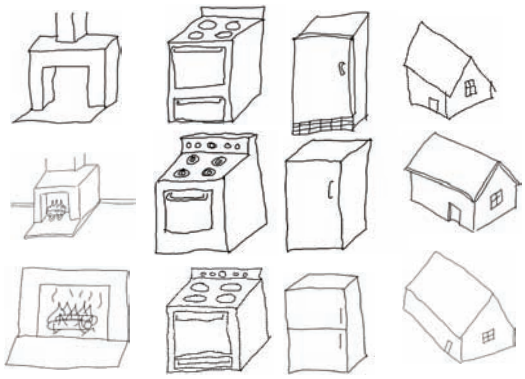

Figure 1. Examples of sketches drawn by subjects

After the generalizations were built, they were used to classify the objects in the sketches by a subset of the remaining users (the test set). A given object was classified by comparing its representation to each of the 8 generalizations and returning the generalization with the strongest match. The strength of a match was calculated based on coverage. After SME was used to identify the common structure in a generalization (the base) and a new instance (the target), the base or target coverage could be calculated by determining the percentage of expressions in the base or target that were a part of the common structure. For example, if every expression in the base matched something in the target, the match would have 100% base coverage. We found that both base and target coverage provided useful information about the strength of a match. Therefore, the system calculates the match's strength by taking the average of the two.

We validated our results by averaging the scores over 80 test runs. In each run, the sketches were randomly divided into training and test sets. Because we were unsure how limiting the number of facts in a representation would affect the results, we ran the test with four different limits on the number of facts. In addition, because we were interested in the incremental effect of adding more cases to a generalization, we ran the test multiple times with different training set sizes. We varied the training set size from two to six cases, while keeping the test set size constant at three.

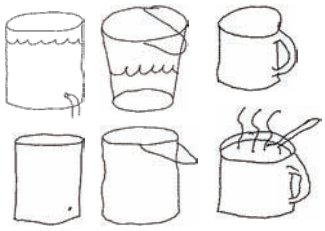

Figure 2. Cylinders, buckets, and cups drawn by subjects

Preliminary tests indicated that many of the classification mistakes made by the system involved a failure to distinguish between the three cylindrical objects: cylinders, buckets, and cups. This is hardly surprising, as these three objects have similar shapes, with nearly as much variation within category as across categories (see Figure 2). Therefore, we used two criteria in reporting our results. According to the strong 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.

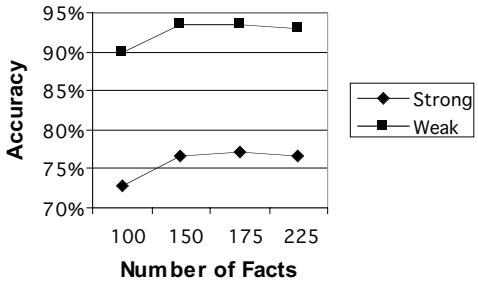

Figure 3. Results when the fact limit is varied

6.1 Results

The results achieved when the limit on the number of facts in each representation was varied are found in Figure 3. These results were based on a training set size of 5. The best results were achieved with a limit of 175 facts. With this limit, the strong criterion was met 77% of the time, and the weak criterion was met 93.5% of the time. Note that chance performance with the strong and weak criteria would be 12.5% and 21.9%, respectively. A t-test was used to look for statistical differences in the results. We found that the increase in performance when the number of facts went from 100 to 150 was statistically significant ($p < .01$) for both the strong and weak criteria. There were no significant differences between the results for 150, 175, and 225 facts.

The results for different training set sizes, shown in Figure 4, were collected with a fact limit of 175 facts. The results for both the strong and weak criteria consistently improved as the training set size increased. With only two cases in each generalization, the results were 71% and 88.5% for the strong and weak criteria. With six cases in each generalization, the results were 77.5% and 94.2%. While there was a clear overall improvement, the differences between adjacent pairs of results were generally small. Increasing the training set size from 3 to 4 resulted in significant performance improvements with both the strong and weak criteria ($p < .01$), but no other differences between adjacent pairs were statistically significant.

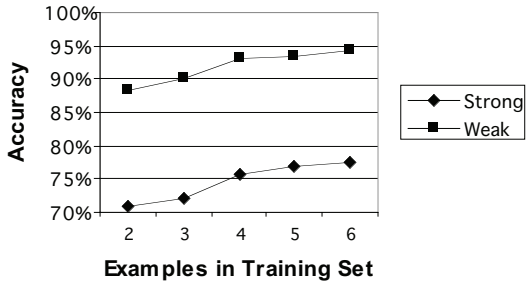

Figure 4. Results when the training set size is varied

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 on only two sample objects for each type, it was able to build generalizations that were sufficiently robust to classify new objects into one of eight categories 71% of the time, and into one of six categories 88.5% of the time. As expected, classification became more accurate as the number of sketches in the training set increased, suggesting that each additional sample allowed the generalizations to be refined and improved. The relatively narrow range of improvement is most likely due to a ceiling effect: the system achieved near-optimal performance with generalizations based on two examples, so there was not a great deal of room for improvement.

We were concerned that limiting the number of facts that could be included in a representation might hamper performance. However, we found no significant differences between performance with 150, 175, or 225 facts. This result suggests that, given our ordering of the facts, the first 150 or 175 facts were sufficient for recovering the shape of the sketch. Of course, one would expect the necessary 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 over 600, we believe the results support 175 being a good general cutoff for the current qualitative representation scheme.

One assumption our system makes that limits our ability to generalize from these results is that relations between interior edges are not needed. If the system is to be able to learn to distinguish between more similar objects, it may be necessary to include these relations in the representations.

One limitation of our experiment is that subjects were given a guiding illustration of each object, rather than drawing objects from their own imagination. However, we believe the results represent an important step towards solving the problem of sketch perception. To the best of our knowledge, no other recognition system has been tested on cross-subject sketches of the complexity and variability used in this experiment.

8 Future Work

Our system currently assumes that each new sketch must match one of the previously learned generalizations. 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.

Thus far we have only shown that our system works in an experimental setting. In the future, we plan to incorporate the system with sKEA so that it will be running in the background while users are sketching. The interaction between the user, the system, and sKEA will create an environment in which we believe open-domain sketch recognition will become a possibility.

Acknowledgments

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

References

[Alvarado *et al.*, 2002] Alvarado, C., Oltmans, M., and Davis, R. A framework for multi-domain sketch recognition. In *2002 AAAI Spring Symposium on Sketch Understanding*, Palo Alto, CA, 2002.

[Alvarado and Davis, 2005] Alvarado, C., and Davis, R. Dynamically constructed Bayes nets for multi-domain sketch understanding. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 1407-1412, Edinbourgh, Scotland, 2005.

[Anderson *et al.*, 2004] Anderson, D., Bailey, C., and Skubic, M. Hidden Markov model symbol recognition for sketch-based interfaces. In *Making Pen-Based Interaction Intelligent and Natural*, pages 15-21, Arlington, VA, 2004. AAAI Press.

[Bokor and Ferguson, 2004] Bokor, J. L., and Ferguson, R. W. Integrating probabilistic reasoning into a symbolic diagrammatic reasoner. In *Proceedings of the 18th International Workshop on Qualitative Reasoning (QR'04)*, Evanston, IL, 2004.

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

[Falkenhainer *et al.*, 1989] Falkenhainer, B., Forbus, K. and Gentner, D. The Structure-Mapping Engine: Algorithms and examples. *Artificial Intelligence,* 41: 1-63, 1989.

[Ferguson and Forbus, 1999] Ferguson, R. W., and Forbus, K. D. GeoRep: A flexible tool for spatial representations of line drawings. In *Proceedings of the 13th International Workshop on Qualitative Reasoning (QR'99)*, pages 84-91. Loch Awe, Scotland, 1999.

[Forbus *et al.*, 2004] Forbus, K., Lockwood, K., Klenk, M., Tomai, E., and Usher, J. Open-domain sketch understanding: The nuSketch approach. In *AAAI Fall Symposium on Making Pen-based Interaction Intelligent and Natural*, pages 58-63, Washington, DC, 2004. AAAI Press.

[Gentner, 1983] Gentner, D. Structure-mapping: A theoretical framework for analogy. *Cognitive Science, 7*: 155-170, 1983.

[Halstead and Forbus, 2005] Halstead, D., and Forbus, K. Transforming between propositions and features: Bridging the gap. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI'05)*, pages 777-782, Pittsburgh, PA, 2005. AAAI Press.

[Hoffman and Richards, 1984] Hoffman, D. D., and Richards, W. A. Parts of recognition. *Cognition*, *18*: 65-96, 1984.

[Kuehne *et al.*, 2000] Kuehne, S., Forbus, K., Gentner, D. and Quinn, B. SEQL: Category learning as progressive abstraction using structure mapping. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, pages 770-775, Philadelphia, PA, 2000.

[Liwicki and Knipping, 2005] Liwicki, M., and Knipping, L. Recognizing and simulating sketched logic circuits. In *Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information & Engineering Systems*, pages 588 – 594, Melbourne, Australia, 2005.

[Lowe, 1989] Lowe, D. G. Organization of smooth image curves at multiple scales. *International Journal of Computer Vision,* 3(2): 119-130, 1989.

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

[Museros and Escrig, 2004] 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, 2004.

[Park and Kwon, 2003] Park, J., and Kwon, Y-B. Main wall recognition of architectural drawings using dimension extension line. In *Proceedings of the Fifth IAPR International Workshop on Graphics Recognition (GREC'03)*, pages 116-127, Barcelona, Spain, 2003. Springer.

[Veselova and Davis, 2004] Veselova, O., and Davis, R. Perceptually based learning of shape descriptions. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI'04)*, pages 482-487, San Jose, CA, 2004. AAAI Press.