Evaluating Qualitative Models of Shape Representation

Andrew Lovett, Subu Kandaswamy, Matthew McLure, & Kenneth D. Forbus

Qualitative Reasoning Group, Northwestern University
andrew-lovett@northwestern.edu

Abstract

Creating intelligent sketch-based educational software requires representing sketches the way a person would. Here, we evaluate several qualitative models of shape representation to determine which one best matches human judgments. The models differ in a) the way they segment an object's contour into meaningful edges; and b) their level of detail for describing a shape. Our analysis indicates our improved edge-segmentation algorithm produces better results than an older algorithm, and a representation focusing on the edges in each shape best matches human judgments.

Introduction

Qualitative features play a key role in human perception and comparison. For example, it is easier to see that two objects are different when there is a qualitative change between them (e.g., the angle between the object's parts changes from **perpendicular** to **oblique**) than when there is a purely quantitative change (between two **oblique** angles) (Rosielle & Cooper, 2001). Similarly, it is easier to determine that an object is asymmetric when there is a qualitative change in its vertices (from **convex** to **concave**) (Ferguson, Aminoff, & Gentner, 1996). Some researchers (Biederman, 1987; Hummel & Biederman, 1992) have argued that object recognition particularly relies on qualitative representations. Because qualitative features such as relative orientation and relative size remain constant across most transformations, objects may be recognized even when seen from novel viewpoints (but see Tarr et al., 1997).

Many domains, especially in science, technical, engineering, and mathematical (STEM) fields, are highly spatial. Sketching is a natural modality that people use to communicate about spatial topics. Creating intelligent sketch-based educational software that can interact via sketching could potentially revolutionize education. Doing that, in turn, requires using human-like qualitative shape representations. For example, Figure 1 depicts a cross-sectioning problem from the Santa Barbara Solids Test (Cohen & Hegarty, 2007), which has been used to examine students' ability to visualize cross-sections of objects. We are using problems from this test to help students improve their cross-sectioning ability. Each problem is recast as a *sketch worksheet* (Yin et al., 2010). In a sketch worksheet, students are given a problem for which they sketch a solution. At any time they can ask for feedback from a coach built into the software. The coach operates by comparing the student's work against a correct sketch drawn by the instructor or a domain expert, and generating advice based on the differences. Here, the students are to sketch the two-dimensional cross-section of the objects intersected by the plane. Figure 1B shows the correct solution, and Figure 1C shows a student sketch. To identify the important differences, it is crucial for the software to represent both sketches in human-like ways.

Our research approach is to develop computational models of human reasoning and intelligent sketch-based educational software in parallel (Forbus et al., 2011). The

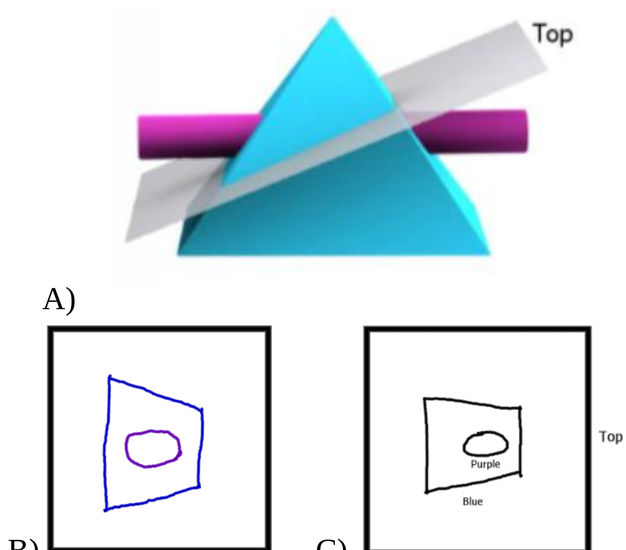

Figure 1. A cross-sectioning problem. A: The assignment. B: The instructor solution. C: An example student sketch.

Lovett, A., Kandaswamy, S., McLure, M., & Forbus, K. (2012). Evaluating qualitative models of shape representation. Proceedings of the 26th International Workshop on Qualitative Reasoning. Los Angeles, CA.

models are designed based on psychological evidence about representation and comparison in humans. They are evaluated by conducting simulations of visual problem-solving tasks and comparing their performance against that of humans (Lovett et al., 2009; Lovett & Forbus, 2011a). Once they are sufficiently vetted, they are incorporated into educational software.

We are currently integrating a model of qualitative shape representation into sketch worksheets, in order to support problems like the one described above. This project presents two significant research challenges. Firstly, the modeling work used perfect, noise-free line drawings from psychological experiments and tests. In contrast, sketch-based educational software must handle rough student-drawn sketches. To address this challenge, we have developed a more robust edge-segmentation algorithm than was used in the modeling work. This algorithm attempts to segment a shape's contour into meaningful edges (e.g., the four edges in a rectangle) by identifying discontinuities in the contour's curvature. It must distinguish between discontinuities from intentional corners and those resulting from random noise. Until now, we have lacked evidence on how this new algorithm compares to the original.

Secondly, the models are designed to flexibly move between several possible shape representations, depending on the task demands (Lovett & Forbus, 2011b). For example a rectangle could be represented as four straight edges. Alternatively, it could be represented as a closed shape with two axes of symmetry. It is unclear which shape representation is most useful for providing feedback on the cross-sectioning task.

Here, we address these questions by evaluating the model on actual student sketches of cross sections. We generated 307 shape pairs based on a previous cross-sectioning experiment. We had human raters judge each pair as "same shape" or "different shape." In our analysis, we compare the human ratings to the model's similarity judgments. We test the old and new edge-segmentation algorithms and consider several possible shape representation schemes. Overall, this evaluation suggests a) our new edge-segmentation algorithm improves shape comparison considerably; and b) a representation focusing on the edges within each object provides the most accurate measures of shape similarity.

In the following section, we present our model of shape representation. We contrast the old and new edge-segmentation algorithms, and we summarize the set of possible shape representation schemes. We then describe our experiment, in which we compared each possible shape representation to the human judgments. Afterwards, we discuss related systems and consider future work.

Model

Our model is built within the CogSketch sketch understanding system (Forbus et al., 2011). CogSketch is a general platform for developing cognitive models and intelligent sketch-based educational software. (In addition to sketch worksheets, a software coach for helping engineering design students learn to communicate by sketching is also being developed (Wetzel & Forbus, 2010).) Cogsketch generates qualitative representations from a user-drawn sketch. Rather than fully performing vision, CogSketch requires the user to provide additional information about their sketch. Firstly, the user manually segments the sketch into separate objects—this can be done by pressing a button when the user finishes drawing a new object, or by manually re-segmenting the ink. Secondly, the user can indicate what each object represents via *conceptual labeling*, selecting a concept from the OpenCyc[1] knowledge base to describe the object.

Given a set of sketched objects with (optional) conceptual information, CogSketch automatically computes spatial information about a sketch. This includes qualitative spatial relations between the object (describing relative position and topology), as well as attributes for each individual object. The spatial and conceptual features combine to produce a structural description of the sketch. This description can serve as input to cognitive simulations or educational software.

In particular, sketch descriptions can be compared using the Structure-Mapping Engine (SME) (Falkenhainer, Forbus, & Gentner, 1989). SME is a computational model of comparison based upon Gentner's (1983) structure-mapping theory, which proposes that people compare representations by aligning their common relational structure. Given two structural descriptions in predicate calculus, SME computes one to three mappings between them. Each mapping contains: 1) correspondences between the entities, attributes, and relations in the two descriptions; 2) *candidate inferences*, inferences drawn about one description based on elements in the other that failed to align; and 3) a similarity score, based on the breadth and depth of aligned structure.

Because SME is based on a psychological theory, its similarity scores represent predictions about human similarity judgments. Provided the items being compared (e.g., object shapes) are represented in a way that captures the features salient to humans, the similarity scores should indicate how similar people find those items to be.

Below, we describe our extension to CogSketch, which attempts to generate human-like representations of space and shape. This extension depends critically on the system's ability to segment an object's contour into

[1] http://www.opencyc.org/

meaningful edges. Therefore, we also discuss our original and improved edge-segmentation algorithms.

Representing Space and Shape

Our cognitive model builds on CogSketch's default representations. Given a set of objects, it generates a hierarchical representation, describing the objects at three levels: groups, individual objects, and edges within each object. At a given level, there are qualitative attributes for each element and qualitative spatial relations between elements. For example, at the edge level, the attributes indicate whether an edge is curved or straight, and whether it is long or short (relative to the other edges), while the relations describe connectedness, relative orientation and whether a corner between edges is concave or convex. At the object level, the attributes indicate whether a shape is open or closed, whether its edges share common attributes (e.g., straightness), and its symmetry type—an isosceles triangle has one axis of symmetry, while a rectangle has two perpendicular axes. See Lovett and Forbus (2011b) for the full set of qualitative terms.

Importantly, the hierarchy provides two ways of representing an object's shape: at the edge level, as a set of relations between edges; or at the object level, as a set of attributes for a single object. The edge level provides more detail because it refers to each edge in the shape. However, the sparser object-level representation may provide sufficient detail for shape comparison, and because it is not concerned with every edge, it is less likely to be distorted by noise in the student drawings (McLure et al., 2011).

At each level in the hierarchy, the model can generate an *orientation-specific* or an *orientation-invariant* representation. The orientation-specific representation (e.g., Tarr et al., 1997) is tied to the particular orientation of the object. For example, an edge-level rectangle representation might indicate that the longer edges are vertical while the shorter edges are horizontal. Alternatively, the sparser orientation-invariant representation (e.g., Biederman, 1987) excludes such information. Thus, it would allow two rectangles drawn at different orientations to match, without there being noticeable differences.

In the case of the cross-section worksheets, it seems likely that orientation-specific representations would be more useful. If an object is drawn at the wrong orientation, this generally indicates a mistake by the user. However, again, there may be an advantage to using a sparser representation. In the analysis below, we evaluate both types. For simplicity, we use the term *oriented* for orientation-specific representations.

Edge-Segmentation Algorithms

CogSketch represents an object as several lists of points, one for each stroke drawn by the user. However, there is no guarantee that these strokes will correspond to an object's edges. A user might draw all four edges of a rectangle with a single stroke, or they might carefully draw multiple strokes for each edge. An edge-segmentation algorithm is used to identify the object's meaningful edges.

For simple closed shapes, such as those often drawn by students on cross-section worksheets, the user-drawn strokes can be easily grouped together to form a single contour, a list of points tracing along the shape's exterior. Thus, in this paper, we focus on the next step: segmenting a contour into meaningful edges. Below, we first describe our original edge-segmentation algorithm, built for modeling experiments. We then present our improved algorithm, optimized for handling hand-drawn student sketches.

Original Segmentation Algorithm

The original algorithm seeks out corners by computing the curvature and the derivative of the curvature at each point along the contour, using a convolution with a Gaussian kernel and its derivative (Lowe, 1989). A corner is a point where the contour's orientation changes sharply. Thus, the curvature should be high. However, there is also a discontinuity in the rate of change in curvature at a corner. Thus, the curvature derivative should be high in the vicinity.

If there is a local maximum in curvature with an accompanying high derivative, a *candidate corner* is created. However, this local evidence for a corner is insufficient. It is possible for minor noise to create a local discontinuity in what is otherwise a straight line. Therefore the system gathers global evidence by segmenting the contour at the candidate corner, creating two *candidate edges*. These edges can then be compared to determine whether a corner between them is meaningful. For example, if the edges are straight, a corner is meaningful unless the edges have nearly identical orientations. If the edges are curved, and they curve in the same direction, then the corner almost certainly is *not* meaningful. More likely it is local noise, as smooth curves are harder to create than straight lines. If the global evidence supports the local evidence, the contour is segmented at the corner.

Improved Segmentation Algorithm

The improved algorithm utilizes the same basic principles as the algorithm above. However, whereas the original algorithm was home-brewed, based on the needs of the modeling experiments, this algorithm systematically applies established edge-segmentation techniques.

We use a modification of the *Curvature Scale Space (CSS) corner detector* (Mokhtarian & Suomela, 1998). Like the original approach, the detector computes the

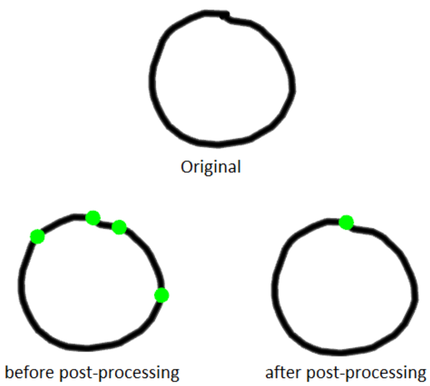

Figure 2. Improved edge-segmentation, with and without post-processing.

curvature at each point along the contour. The contour is segmented into corners at local maxima in the curvature.

We perform post-processing to filter out corners based only on local noise. For every sequence of three corners along the contour, the system evaluates whether the middle corner could be removed. It does this by measuring the segment's closeness of fit to a geometric primitive, either a straight line or an arc. If the segment closely fits one of these primitives, then the corner in the middle of the segment is not meaningful, and it can be removed. Closeness of fit is measured using *Hausdorff distance* (Huttenlocher, Klanderman, & Rucklidge, 1993) with a match probability (Field et al., 2011). In Figure 2, post-processing removes most of the false corners in the circle. Unfortunately, it fails to remove one false corner.

The improved algorithm contains one additional refinement, which only applies to object-level representations. A perfect circle should consist of only a single edge which closes on itself; however, a poorly drawn circle may be segmented into multiple edges, based on local noise. To address this problem, the system applies a *circle detector* to an object's contour before computing the object-level representation. As above, this detector measures the Hausdorff distance between the contour and a perfect circle. If the distance is small enough, the object is treated as a circle for representation purposes (e.g., it receives the attributes **2D-Shape-Ellipse** and **Fully-Symmetric-Shape**; see Lovett & Forbus, 2011b).

Experiment

We evaluated our shape representations on a corpus of shape pairs taken from a cross-sectioning experiment. In that experiment, participants were shown images such as Figure 1A and asked to draw the two-dimensional cross-sections. For example, one participant drew Figure 1C. We generated our corpus by pairing each student-drawn object with the corresponding object in the solution sketch (e.g., Figure 1B). This produced 307 shape pairs.

Three human participants rated the similarity of the shape pairs. The raters were given an explanation of the cross-sectioning task and told to act as graders, indicating whether each student-drawn object was the same shape as the solution-sketch object. For each pair, they clicked on a button to indicate "Same," "Different," or "Not Sure."

We also computed the similarity of the shape pairs using our shape representation models. We varied segmentation algorithm (old/new) *x* representation level (edges/objects) *x* orientation information (invariant/specific) = eight models. For each model, we used the Structure-Mapping Engine to compute a similarity score. Similarity scores were normalized, so they ranged from 0 to 1.

Behavior Results

One human rater's scores were thrown out because they failed to follow instructions. The other two raters agreed on 92.5% of the shape pairs. Their inter-rater reliability score Kappa was 0.81. In the analysis below, we exclude the 7.5% on which the raters disagreed. Among the 284 remaining pairs, all responses were either "Same" or "Different."

Modeling Results

Segmentation

We first evaluated each edge-segmentation algorithm using a simple heuristic: number of edges. If two objects have the same shape, the algorithm should identify the same number of meaningful edges for them (four for a rectangle, one for a circle, etc). Note that this heuristic does not fully determine shape similarity: two objects with the same number of edges may have very different shapes.

Table 1 gives the percentage of shape pairs with matching numbers of edges, according to each segmentation algorithm. Let us first consider the "Same" shape pairs (i.e., the pairs deemed the "Same" by human raters). Clearly, the improved algorithm outperforms the original algorithm. It computes matching numbers of edges 82% of the time, compared to the original's 69%.

The results for "Different" shape pairs are harder to interpret. If two objects have different shapes, they may or may not have the same number of edges, so it is unclear which algorithm is more accurate. However, it is interesting to note that the original algorithm appears far more permissive for "Different" pairs: it is considerably

	"Same"	"Different"
Original	.69	.58
Improved	.82	.38

Table 1. Percentage of edge-number matches for the original and improved edge-segmentation algorithms.

Original Segmentation

	"Same"	"Different"
Edge	.79	.54
Edge, Oriented	.81	.55
Shape	.79	.68
Shape, Oriented	.79	.68

Improved Segmentation

	"Same"	"Different"
Edge	.84	.40
Edge, Oriented	.85	.40
Shape	.92	.75
Shape, Oriented	.91	.74

Table 2. Average similarity scores for each shape representation. High values are preferred in the "Same" column while low values are preferred in the "Different" column.

more likely to compute the same number of edges. We must consider other results to determine if this disrupts the system's performance.

Similarity

We next consider our eight different shape representations. For each of these, we computed the average SME similarity score for "Same" and "Different" shape pairs. To match human judgments, a model should produce high scores for "Same" pairs and low scores for "Different" pairs.

Table 2 gives the results for each representation. Overall, the edge-level, oriented representation with improved edge-segmentation performed best: its "Same" pair scores were .45 higher than its "Different" pair scores.

There are several general observations we can make. Firstly, the object-level representations appear to be ineffective. The "Different" similarity scores are quite high, suggesting these representations provide insufficient detail for distinguishing between different shapes.

Secondly, the original segmentation algorithm is also ineffective. Again, the "Different" similarity scores are high, even for edge-level representations. The Table 1 results may help explain this finding. This algorithm appears to be too permissive in finding the same number of edges for "Different" shape pairs.

Thirdly, there is no apparent advantage for oriented representations. This suggests that students in the cross-sectioning study rarely drew objects at an incorrect orientation. When objects were different, it was because of their overall shape, not because of their orientation. Thus, the additional information in the oriented representations would provide little assistance in distinguishing between shapes.

Original Segmentation

	"Same"	"Different"
Edge	.60	.33
Edge, Oriented	.61	.25
Object	.46	.29
Object, Oriented	.46	.29

Improved Segmentation

	"Same"	"Different"
Edge	.74	.14
Edge, Oriented	.71	.09
Object	.66	.43
Object, Oriented	.66	.42

Table 3. Proportion of Same classifications for each shape representation. High values are preferred in the "Same" column while low values are preferred in the "Different" column.

Classification

Finally, we evaluated whether the models could classify pairs as Same or Different. This was done by applying a threshold to the similarity scores; pairs above the threshold would be classified as Same, while those below would be Different. We experimented with a range of thresholds and found that .90 produced the best results[2]. Table 3 shows each representation's performance with this threshold.

Again, the edge-level, oriented representation with improved edge-segmentation performed best. It classified 71% of the human-rated "Same" pairs as Same, but only 9% of the "Different" pairs as Same. While the false positive rate (9%) was quite low, the false negative rate was higher: 29% of the "Same" pairs were misclassified as Different. We discuss this further in the next section.

The first two observations about similarity scores also apply to classification. Again, the object-level representations often cannot distinguish between "Different" pairs, leading to high false positive rates. The original segmentation algorithm also produces many false positives. However, the oriented representations *do* have a small advantage here, at least for edge-level representations: for both the original and improved segmentation algorithms, there are fewer false positives. This suggests our classification measure may be more sensitive than our similarity measure.

Discussion

Our results show that the improved edge-segmentation algorithm consistently outperforms the original. For pairs judged the "Same" by human raters, it produces: a) more matching numbers of edges; b) higher similarity scores; and c) more similarity scores above a .90 threshold. For pairs judged "Different," it produces: a) fewer matching numbers of edges; b) lower similarity scores; and c) fewer similarity scores above a .90 threshold. Interestingly, the

[2] The edge-level representation with improved edge-segmentation out-performed other representations at every threshold considered (0.65 - .95), although there was not always an advantage for oriented.

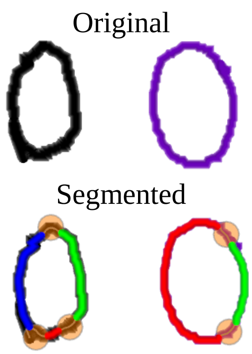

Figure 3. Error segmenting an ellipse with improved edge-segmentation.

second set of effects are more pronounced. That is, the new algorithm is particularly good at distinguishing between "Different" shapes.

An inspection of the results suggests this may be because the original algorithm is too permissive in grouping curves together. This is due to the global evidence phase, where a candidate corner will be eliminated if the curved edges on either side of it are remotely consistent. This can result in a single, elliptical edge where there should actually be multiple edges. In contrast, the new approach's Hausdorff distance computation may produce more accurate information on when curves should be grouped together.

The results also show that our object-level representations provide insufficient detail for distinguishing between "Different" shapes. Focusing on the improved edge-segmentations, the object-level representations provide higher similarity scores and more false positives on the "Different" pairs. This suggests that for the cross-sectioning task, edge-level information is necessary to distinguish between shapes. Note that this does not mean every sketch worksheet should use edge-level representations. The object level may provide sufficient detail for other tasks.

The edge-level, oriented representations with improved edge-segmentation provide the best overall results. However, there is room for improvement. In the classification evaluation, only 71% of "Same" shape pairs were classified as Same. This is because the current algorithm, even with post-processing, is still sensitive to noise. For example, the left ellipse in Figure 3 has a hook at its lower left point, inducing the system to introduce a false corner. Thus, the system classifies these shapes as Different, whereas the humans rated them as "Same."

Related Work

Museros and Escrig's (2004) system used qualitative descriptions to identify rotations between polygons. Ferguson & Forbus' (1999) GeoRep constructed qualitative descriptions of line drawings that could be used as input for tasks such as recognition and identifying symmetry. Neither of these systems worked on hand-drawn input. Veselova and Davis's (2004) system learned qualitative constraints describing a hand-drawn object which could be used to recognize other sketches of that object. Their system was designed specifically for sketch recognition, whereas we use a task-general approach for representing shape and space.

Conclusions

Building intelligent sketch-based educational software is a difficult challenge. It requires software that can predict human similarity judgments for image and shape comparisons. Here we have shown that edge-level shape representations, developed for cognitive simulations, can approximate human judgments. When humans judge two shapes as "Different," the model agrees 91% of the time. However, when they judge two shapes as "Same," the model only agrees 71% of the time.

A major reason for the false negative rate is the difficulty of accurate edge-segmentation. While our improved algorithm, based on established segmentation techniques, outperforms the original, it fails to achieve human levels of noise-tolerance.

We believe we can increase the algorithm's noise-tolerance through greater reliance on geometric primitives. Recall that these primitives provide global information, used to filter out local noise. At present, the algorithm uses two primitives (lines and arcs) at the edge level, and one (circles) at the object level. We would like to apply all primitives at the edge level, so that circular edges can be recognized. We would also like to add a fourth primitive, *ellipses*, to better process elliptical shapes like those in Figure 3.

Much of human noise-tolerance may come from our ability to use global context in interpreting shapes. For example, when comparing ambiguous shapes, people changed their interpretations to make the shapes more similar (Medin, Goldstone, & Gentner, 1993). We would like to implement a similar capability. We term this *comparison-based re-segmentation*: during comparison, the system would add or remove corners in one shape to make it better match the other. For example, the shapes in Figure 4 may have different numbers of edges when viewed in isolation. However, during comparison, the system would recognize that a corner could be added to the left shape or removed from the right shape. Each potential corner would have a confidence level, based on the curvature, and thus each re-segmentation would have an associated cost. This re-segmentation cost could provide a more precise quantitative measure of shape similarity.

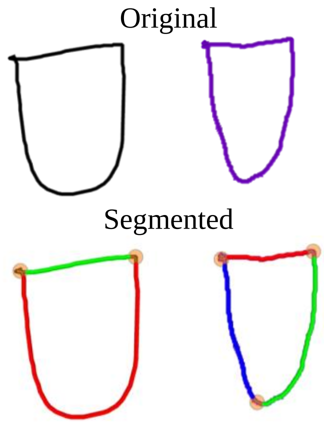

Figure 4. These shapes may require comparison-based re-segmentation.

Ackowledgements

This work was supported by NSF SLC Grant SBE-0541957, the Spatial Intelligence and Learning Center, and by the Office of Naval Research.

References

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review, 94*, 115-147.

Cohen, C. A., & Hegarty, M. (2007). Sources of difficulty in imagining cross sections of 3D objects. *Proceedings of the 29th Annual Conference of the Cognitive Science Society*.

Falkenhainer, B., Forbus, K., & Gentner, D. (1989). The structure mapping engine: Algorithm and examples. *Artificial Intelligence*, *41*, 1-63.

Ferguson, R. W., Aminoff, A., & Gentner, D. (1996). Modeling qualitative differences in symmetry judgments. *Proceedings of the 18th Annual Meeting of the Cognitive Science Society*.

Ferguson, R. W., & Forbus, K. (1999). GeoRep: A flexible tool for spatial representations of line drawings. *Proceedings of the 13th International Workshop on Qualitative Reasoning.*

Field, M., Valentine, S., Linsey J., & Hammond, T. (2011). Sketch recognition algorithms for comparing complex and unpredictable shapes. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence.*

Forbus, K., Usher, J., Lovett, A., Lockwood, K., & Wetzel, J. (2011). CogSketch: Sketch understanding for cognitive science research and for education. *Topics in Cognitive Science, 3*(4). 648-666.

Hummel, J. E., & Biederman, I. (1992). Dynamic binding in a neural network for shape recognition. *Psychological Review, 99*, 480-517.

Huttenlocher, D. P., Klanderman, G. A., & Rucklidge, W. J. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 15*(9), 850-863.

Lovett, A., & Forbus, K. (2011a). Cultural commonalities and differences in spatial problem-solving: A computational analysis. *Cognition, 121*(2), 281-287.

Lovett, A. & Forbus, K. (2011b). Organizing and representing space for visual problem-solving. *Proceedings of the 25th International Workshop on Qualitative Reasoning.*

Lovett, A., Tomai, E., Forbus, K., & Usher, J. (2009). Solving geometric analogy problems through two-stage analogical mapping. *Cognitive Science, 33*(7), 1192-1231.

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

McLure, M., Friedman, S., Lovett, A., & Forbus, K. (2011). Edge-cycles: A qualitative sketch representation to support recognition. *Proceedings of the 25th International Workshop on Qualitative Reasoning.*

Medin, D. L., Goldstone, R. L., & Gentner, D. (1993). Respects for similarity. *Psychology Review, 100*(2), 254-178.

Mokhtarian, F., & Suomela, R. (1998). Robust image corner detection through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(12), 1376-1381.

Museros, L., & Escrig, M. T. (2004). A qualitative theory for shape representations and matching. *Proceedings of the 18th International Workshop on Qualitative Reasoning.*

Rosielle, L. J., & Cooper, E. E. (2001). Categorical perception of relative orientation in visual object recognition. *Memory & Cognition, 29*(1), 68-82.

Tarr, M. J., Bülthoff, H. H., Zabinski, M., & Blanz, V. (1997). To what extent do unique parts influence recognition across viewpoint? *Psychological Science, 8*(4), 282-289.

Veselova, O., & Davis, R. (2004). Perceptually based learning of shape descriptions. *Proceedings of AAAI '04.*

Wetzel, J., & Forbus, K. (2010). Design Buddy: Providing feedback for sketched multi-modal causal explanations. *Proceedings of the 24th International Workshop on Qualitative Reasoning.*

Yin, P., Forbus, K., Usher, J., Sageman, B. & Jee, B. (2010). Sketch worksheets: A sketch-based educational software system. *Proceedings of the 22nd Annual Conference on Innovative Applications of Artificial Intelligence.*