

A Structure Mapping Model for Solving Geometric Analogy Problems

Emmett Tomai Andrew Lovett Kenneth D. Forbus Jeffrey Usher

Qualitative Reasoning Group
Northwestern University
1890 Maple Avenue,
Evanston, IL, 60201 USA
etomai@northwestern.edu

Abstract

Evan's 1968 ANALOGY system was the first computer model of analogy. This paper demonstrates that structure-mapping, when combined with high-level visual processing and qualitative representations, can do the same kinds of problems with hand-drawn sketched inputs. Importantly, the bulk of the computations are not particular to the model of this task but are general purpose: we sketch the problems using our existing sketch understanding system, sKEA, which computes visual structure that is used by our existing analogical matcher, SME. We show how SME can be used to facilitate high-level visual matching and how *second-order analogies* over differences computed via analogies between sketches provide a more elegant model of this task.

1. Introduction

One of the mysteries of human cognition is how we make sense of the world around us. We have powerful visual systems, and it appears that part of their job is to compute descriptions of visual structure (cf. [15,20]) which can be used for recognition and understanding. We have argued previously that qualitative spatial reasoning plays an important role in medium and high-level visual processing [8]. Qualitative spatial representations provide a bridge between vision and cognition, since they seem to be computed via visual processes, but take functional constraints into account. We have been exploring this idea by research on sketching. Understanding sketches is a useful approach to understanding qualitative visual structure because starting with digital ink lets us focus on processes of perceptual organization and ignore image processing issues. Previously we have described techniques for imposing human-like visual structure on sketches and how that structure enables our software to better model human similarity judgments [9]. Building on that work, we describe here a computational model of the classic Miller Geometric Analogies task. We use the same set of problems used by Evans in his pioneering work on the system ANALOGY [3]. Figure 1 provides an example. The form of the question is, "A is to B as C is to ___?".

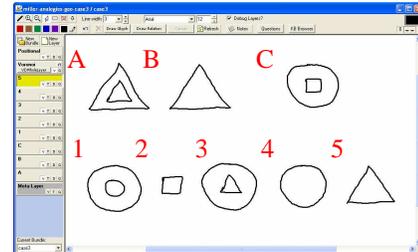


Figure 1: Sketched input of a GMAT problem

Like Evans, we view these problems as non-trivial and useful for the exercising of internal descriptions. Evans was starting from scratch. Our goal is to show that the progress in analogical processing, qualitative spatial reasoning, and other areas of cognitive science allows us to use general-purpose simulation models to construct a model that solves this same task.

We start by briefly reviewing the essentials of SME, our model of analogical matching, followed by a summary of the *sketching Knowledge Entry Associate* (sKEA) [10], the open-domain sketch understanding system used in these experiments, focusing on representation of sketches and the visual structures computed over them. This includes the use of SME to help recognize visual rotations and reflections, a novel extension. We then describe how we use *second-order analogies* between differences found via analogies between sketches to solve these problems. We walk through an analysis of one example in detail to illustrate the processing and then summarize how our model performs on the Evans corpus. Finally, we discuss plans for future work.

2. Overview of SME

We rely on the *Structure-Mapping Engine* (SME) [4], an implementation of Gentner's structure mapping theory [12] to provide human-like analogical processing. SME takes a pair of structured representations, the *base* and the *target*, and returns a set of directed mappings between them. Each mapping consists of *entity correspondences* between entities in the two representations, *expression correspondences* that form the multi-level structure of support for the entity matches and *candidate inferences* that project unmatched relationships and features from the base into the target [4]. The mappings are assigned a

numerical score that reflects the systematicity of the structure in support of the mapping.

3. Overview of nuSketch and sKEA

We view sketching as a form of multimodal interaction, where participants combine drawing, language, and their world knowledge to provide high-bandwidth communication. Our *nuSketch architecture* [8] provides an approximation of such capabilities, by capturing digital ink for visual processing, and enabling people to identify sketched entities as instances of concepts drawn from a large background knowledge base¹. Our ongoing work with sKEA is aimed at creating a sketch understanding system whose visual and conceptual understanding is deep enough to participate in sketching as fluently as people do.

A sketch in sKEA consists of one or more *glyphs*. A glyph has its *ink* and *content*. The ink of a glyph is represented by a set of one or more poly-lines. The content of a glyph is the entity that it represents. By marking entities as instances of KB contents, axioms associated with those concepts are available for reasoning. sKEA incorporates computational geometry algorithms that provide an approximation of visual processing. Our concern currently is producing human-like visual representations, rather than modeling in detail the particular processes that construct them.

Since the visual representations sKEA produces are crucial for this task, we summarize what visual structure sKEA produces. These computations are carried out incrementally, while someone is interactively sketching, using a process outlined in [9].

sKEA starts by computing qualitative topological relationships, using Cohn's RCC8 relational vocabulary [1]. These distinctions include whether two glyphs are inside each other, touching each other, and so on. This information is used in turn to automatically compute two kinds of visual groups: *contained glyph groups* and *connected glyph groups*. A contained group consists of a single container glyph and the set of glyphs that are inside of it. The contained group does not include glyphs that are contained within other glyphs in the group. A connected glyph group consists of a set of glyphs that overlap ink strokes with one another. Articulation points can be computed over connected glyph groups, and tangentially connected pairs of glyphs can be noted as such. The algorithms used for computing glyph groups are detailed in [9].

¹ Our knowledge base contents are a 1.2 million fact subset of Cycorp's Cyc KB, containing just over 39,000 different concepts, over 8,500 relationships and 5,000 functions. This includes some augmentations to support qualitative and analogical reasoning. Northwestern's FIRE system is used for the KB implementation and reasoning.

Positional relationships are expressed in a viewer-oriented coordinate system of left/right and above/below. They are only computed between pairs of glyphs that are adjacent, as determined via a Voronoi diagram, or intersecting, as long as one glyph does not completely contain the other. Sketches can be further structured into layers, analogous to drawing on acetate overlays, and positional relationships are only computed between glyphs that are on the same layer.

sKEA assigns each glyph a qualitative size value, one of *tiny*, *small*, *medium*, *large* or *huge*. Sizes are based on the area of a glyph's axis-aligned bounding box, a coarse but empirically useful approximation. Glyph areas are normalized with respect to either the area of the bounding box around all glyphs on all layers, or the area of the user's view port, whichever is larger. The normalized areas are then clustered into qualitative size values based on a logarithmic scale of the square root of the area. Informal experimentation suggests that this is a reasonable method for the varieties of sketches we have examined thus far.

4. Visual shape matching

sKEA previously did not analyze the internal structure of a glyph's ink, focusing only on relationships between glyphs. For the Evans task this is not enough, since it is important to recognize shape similarity and cases wherein one shape is a rotation or reflection of another (cf. Figure 2).

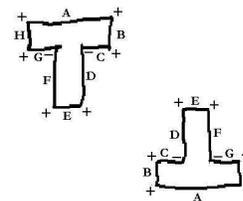


Figure 2: Recognizing rotated/reflected shapes is important

We accomplish this by first decomposing every glyph's ink into a set of connected edges. We use a greedy algorithm that grows edges from segments of uniform orientation, looking for corners². The edge sets are organized into *cycles* and *segments*. A cycle is an ordered sequence of connected edges in which the first and last edge are identical. A segment is a maximal ordered sequence of connected edges containing no cycles. Cycles and/or segments that represent the shape of the glyph as a

² While sufficient for GMAT-style problems, this part of our algorithm is still more sensitive to the way that glyphs are drawn than we would like.

whole – *bounding edge sets* – are gathered to be used in shape matching.

The second step is determining if there is a good mapping between each pair of glyphs by matching their bounding edge sets. After augmenting them with qualitative angular relationships (convex and concave, illustrated by + and – in Figure 2, respectively), we use SME to compare them, producing a set of candidate mappings (up to five). These mappings, based on essentially qualitative criteria, are then evaluated via several quantitative criteria. Overall shape factors, e.g., convex/concave angles and acute/obtuse angles, receive the most weight, whereas factors such as relative edge length and whether an edge is axis-aligned receive less weight. Any mappings that receive sufficiently high scores are kept in consideration for possible rotations or reflections.

Mappings are scrutinized to determine if they represent rotations or reflections. For rotations, the system examines the differences in each corresponding edge pair's orientations, and if the disparities are sufficiently similar over all pairs, it returns the average difference as the rotation. The mapping with the smallest angle of rotation is considered the most salient. Reflections are handled similarly, by checking to see if the orientations of all corresponding edges are reflected over the same axis. If no consistent rotation or axis is found, the match is a failure. Otherwise, appropriate relationships are asserted between the two glyphs.

5. Solving Miller Analogy Test problems

Consider again the example of Figure 1. The correct answer is the one that best completes the analogy “A is to B as C is to ?”. sKEA provides a natural means of entering these geometry problems. We use the layer facility to create eight layers named A, B, C and 1 – 5 which will contain the glyphs that make up each respective drawing. Object segmentation within each drawing is determined by the user who decides what comprises a glyph. In all these cases simple shapes, symbols and groups of connected lines are treated as individual glyphs.

To solve the problems, we use a two-stage structure mapping process, depicted in Figure 3. The first stage is the computation of mappings from picture A to picture B and from picture C to each of the answer pictures 1 – 5. This generates six mappings (the example mapping AB and the potential answer mappings C1 – C5) that represent the similarities and differences between their respective pairs. The second stage takes those mappings as input and computes the prescribed analogy from AB to each of the answer mappings C1 – C5. The strongest result from the second stage indicates the correct answer.

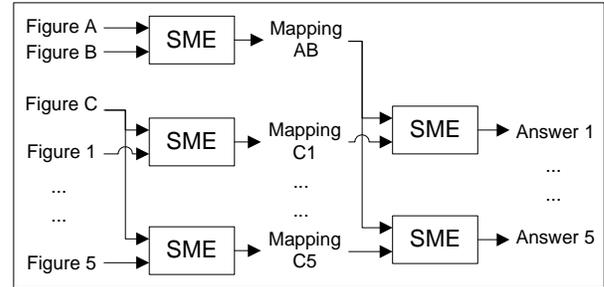


Figure 3: Two-stage structure mapping

Let us examine this process in more detail. In addition to the visual structure usually computed by sKEA, the input to the first stage includes the shape identity, rotation, and reflection relationships computed as per Section 4. In some cases a non-symmetric shape can display both reflection and rotation possibilities. This is fine except in situations where both facts match from the example pair to an answer pair, thus carrying twice the weight the feature should. To avoid this, we allow assertion of both in the example pair, but allow only one of the two in the answer pairs. Following Evans' lead, rotation is preferred over reflection.

Because it is possible that more than one legitimate mapping might exist between a given pair of pictures, we run SME twice for each input pair in the first stage. The structural evaluation scores produced by SME are then used to judge the relative strength of the second, less optimal mapping to the first. In cases where the second mapping scores nearly as high as the first (within 5%) both mappings are considered valid. This results in twice as many second stage comparisons of which the best single answer is still taken.

Normally SME only computes candidate inferences from the base to the target. For our purposes, it is just as important to detect novel relationships and attributes in the target that are not present in the base. We therefore used an extension to SME that computes candidate inferences in the reverse direction as well, using the same algorithm used to generate standard candidate inferences but with swapped arguments [17].

The first stage of comparison works through the similarities between pairs of pictures. Descriptions of differences arise out of comparisons [13]. Because the alignable differences computed as forward and reverse candidate inferences by the first stage are already grounded in the similarities, those differences provide all the necessary information for this task. In our experiments with the twenty problems from Evans' work we have passed only the alignable differences to the second stage; our results have shown this to be sufficient.

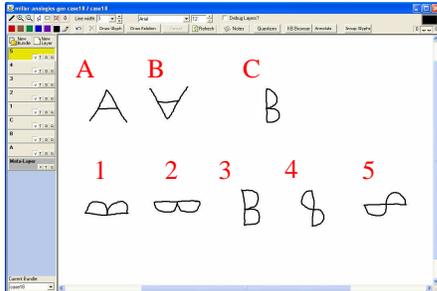


Figure 4: Sketch of Problem 18

Symmetric shapes display reflection in many orientations, creating a large number of redundant facts. We disallow these by default to keep the system focused. However, there are times (cf. Figure 4) where no suitable answer can be found and it is necessary to use a non-obvious reflection. When a first stage mapping returns a judgment of no difference at all, the system backtracks and reevaluates it with those reflections allowed.

In the first stage mappings, attributes must match identically. Circles must match with circles and left must match with left or there is simply too much flex in the system for meaningful conclusions. But when comparing differences in the second stage, we relax this constraint, allowing for instance circles in one case to be consistently mapped to squares in another. Similarly, a *rightOf* relationship in one pair might correspond to an *above* relationship in another answer pair, a 90 degree rotation might be analogous to a 45 degree rotation, or (esoterically) a change in position might correspond to a change in shape. Clearly some of these possibilities are better than others, so we use information from the knowledge base to compute preferences. Identical relations are still preferred, e.g., two 90 degree rotations match better than two rotations of different degrees. Attributes or relationships that are closer, in the conceptual hierarchy of the KB, are preferred as well. For instance, matching *leftOf* with *rightOf* is preferred to matching *leftOf* with *above*, since the former are both horizontal positional relationships. The system elaborates the results of each first-stage mapping by querying the KB concerning the attributes in the mapping and the relationships that hold between them. In cases where an unmapped glyph exists in either the base or target layer, SME generates a skolem representation in the candidate inferences. We augment this representation with the attributes of the glyph it maps to. These elaborated descriptions become the input for the second stage. We consider this a significantly more general and powerful approach than Evans' alternate rules [3] wherein non-matching predicates were substituted for alternatives of like type until an answer was found.

Possible answers are evaluated by combining SME's structural evaluation score for the second-stage mappings with a *difference score*. The structural evaluation score

indicates only how similar the differences are. The difference score penalizes answers that have additional differences (aka *leftovers*). For example, in Figure 1, answer 4 could be seen as the removal of a glyph while answer 5 would be seen as the removal combined with a shape change. Clearly the example pair AB shows only the removal. In spite of this, these two answers receive the same structural evaluation score since they both reflect the removal. The shape change is a leftover and should be penalized.

The difference score is a linear weighted sum, based on the types of leftovers, which is subtracted from the structural evaluation score. Leftovers involving unmatched glyph additions or removals are penalized the most strongly, since they are unlikely to be caused by any errors in the visual processing. Leftover relationship expressions, indicating a relationship appearing or disappearing in the answer but not in the example, are next highest penalized. The lowest penalty is given to attribute leftovers, which indicate a spurious difference in features such as shape, rotation, reflection, or size, since these might arise due to noisy perception.

6. A detailed example

To illustrate the system's operation and the issues raised by it, we walk through the problem depicted in Figure 1. The correct answer is 4. The difference between A and B is the lack of the smaller, inner triangle. C likewise has a contained small square that is lacking in 4.

Our first step is to draw the sketch in sKEA. Each shape is drawn as an individual glyph in the proper layer. sKEA's spatial processing then computes size grouping for each glyph. The larger glyphs are all determined to be of medium size while the smaller are small. Contained glyph groups are asserted in A, C, 1 and 3; no connected glyph groups are found. There are no adjacent glyphs within any of the layers and thus no positional relationships are generated.

The first stage structural mapping between A and B maps together the two larger triangles on the strength of their size and similarity of shape and generates a candidate inference proposing that the triangle in B should have another glyph inside of it. No reverse candidate inferences are formed. The first stage mappings from C to each of the five answers return notably similar results, showing differences in shape and removal of the inner and outer glyphs, as one would expect looking at the problem.

The second stage mappings correctly identify 4 as the answer. Answer mappings from 1 and 3 generate candidate inferences and reverse candidate inferences indicating difference in the shape of the inner glyphs. These fail to map with anything in the example mapping resulting in null scores for both. The answer mapping for 2 generates a candidate inference indicating the lack of the outer glyph. This fails to map with the lack of an inner

glyph, again resulting in a null score. Answer mappings for 4 and 5 receive identical structural evaluation scores for reflecting the removal of the inner glyph. However, 5 is penalized for having a leftover, the difference in the shape of the outer glyph, and 4 is selected as the answer.

7. Experimental results

Due to space concerns we cannot include the sketches and analysis of the twenty problems. In eighteen of the problems, our system selects the same answer that ANALOGY did. Our system also solves Evans' problem 10 which ANALOGY did not. For problem 12, discussed below, our system selects a different answer than ANALOGY, but we believe our answer is just as consistent and consider it correct as well.

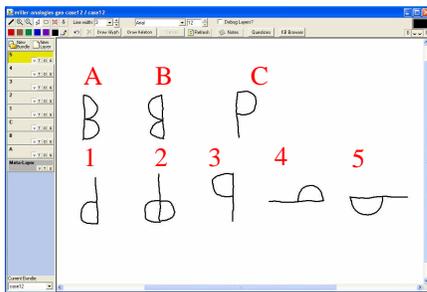


Figure 5: Sketch of Problem 12

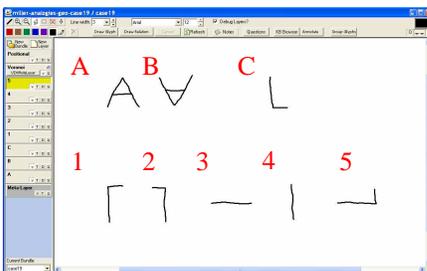


Figure 6: Sketch of Problem 19

In discussing Problem 12, Evans reports 3 as the correct answer, a case of vertical axis reflection. In Problem 19, he reports 2 as the correct answer, noting that his system is biased to prefer rotation to reflection (answer 1 shows horizontal axis reflection). Taken together, these do not seem to be consistent. Our system is currently biased to prefer rotation, as Evans reported his to be, and thus selects answer 1 for Problem 12 and answer 2 for Problem 19. If the bias were switched to prefer reflection, answers 3 and 1 would be selected, respectively. But based on the information Evans provides in [3], we conclude that our answers are satisfactory.

8. Other related work

Evans' classic work was the first to illustrate that machines could do analogy. To fit his program into the punch-card computer available at the time, the geometric

processing was done as a separate module, taking coordinates as input and producing symbolic descriptions. Due to limitations in this part of the program, half of the examples reported in [3] actually use hand-coded inputs instead. Subsequent attempts to build on Evans' work use hand-generated symbolic inputs as starting points (e.g., [19]). By contrast, our model exploits sKEA's built-in qualitative visual structure computing abilities to generate representations from ink input, capabilities which are part of a general-purpose architecture for sketch understanding. sKEA's visual processing evolved from Ferguson's work on GeoRep [5], which was first to show that structure mapping could be used to identify geometric similarity. Another significant difference is that Evans construed the problem as creating transformation rules between pairs of figures, which led to ambiguities due to the need to consider alternate possible rules in some cases. Our model illustrates that computing explicit rules is unnecessary: comparing the similarities and differences is sufficient to explain human behavior on the task.

Tight interleaving of the construction of representations with matching is a hallmark of systems from Hofstadter's group, including Mitchell's Copycat program [15] and French's TableTop [11]. Unfortunately, each of these systems only operates in the single domain it was designed for, letter-strings for Copycat and table settings for TableTop. The kinds of comparisons that can be made are hand-wired into the system (the Slipnet). Similarly, Galatea [2] has a built-in specialized language of spatial entities and transformations that must be used in posing problems to it. By contrast, SME has been used in a wide variety of domains, and automatically figures out what kinds of things can be matched [7].

9. Discussion and future work

By solving this set of classic visual analogy problems without sacrificing the generality of our system, we believe that we have made several significant contributions. We have shown that qualitative representations are a significant element of doing geometric analogies of a kind that has commonly been used in intelligence testing. We have also shown that the set of representations we are working with are a reasonable subset of the representations needed for solving said problems. We have shown that structure mapping can be effectively used to identify geometric rotation and reflection in addition to similarity. Finally, we have shown that a two-level analogical processing scheme can capture the phenomena without searching over transformation rules as Evans did.

Future work will include continued research on visual structure as well as conceptual relationships. We plan to extend our visual processing to more sophisticated positional relations and incorporate Ferguson's MAGI model of symmetry [5]. We also plan to improve the noise tolerance of our visual processing. Ongoing work in

this area has proposed a mixture of interactive and automated techniques (cf. [14,21]). In this work we have used a very basic system of backtracking and reinterpretation which we intend to flesh out into a full-fledged model. Finally, we plan to apply our system to more tasks, including a more advanced geometric intelligence test.

Acknowledgements

This research was supported by a ROLE grant from the National Science Foundation, by the Cognitive Science Program of the Office of Naval Research, and by the Information Processing Technology Office of the Defense Advanced Projects Research Agency.

References

1. Cohn, A.G. 1996 Calculi for Qualitative Spatial Reasoning. In Calmet, J., Campbell, J.A., and Pfalzgraph, J. (Eds.) *Artificial Intelligence and Symbolic Mathematical Computation*. LNCS 1138. Springer Verlag, 124-143.
2. Davies, J. and Goel, A. 2003. Representation issues in visual analogy. *Proceedings of CogSci03*.
3. Evans, T. (1968) A Program for the Solution of a Class of Geometric-Analogy Intelligence-Test Questions, *Semantic Information Processing*, 1968, MIT Press.
4. Falkenhainer, B., Forbus, K., Gentner, D. (1989) The Structure-Mapping Engine: Algorithm and examples. *Artificial Intelligence*, 41, pp 1-63.
5. Ferguson, R. W. (1994). MAGI: Analogy-based encoding using symmetry and regularity, *Proceedings of CogSci94*.
6. Ferguson, R. and Forbus, K. (2000). GeoRep: A flexible tool for spatial representation of line drawings. *Proceedings of AAAI-2000*. Austin, Texas.
7. Forbus, K., Gentner, D., Markman, A. and Ferguson, R. (1997). Analogy just looks like high-level perception: Why a domain-general approach to analogical mapping is right. *Journal of Experimental and Theoretical Artificial Intelligence (JETI)*, 4, 185-211.
8. Forbus, K., Ferguson, R. and Usher, J. (2001). Towards a computational model of sketching. *IUI'01*, January 14-17, 2001, Santa Fe, New Mexico
9. Forbus, K., Tomai, E., and Usher, J. (2003). Qualitative Spatial Reasoning for Visual Grouping in Sketches. *Proceedings of the 17th International Workshop on Qualitative Reasoning*, Brasilia, Brazil, August 2003.
10. Forbus, K. and Usher, J. (2002). Sketching for knowledge capture: A progress report. *IUI'02*, January 13- 16, 2002, San Francisco, California.
11. French, R. (1995). *The subtlety of similarity*. Cambridge, MA: The MIT Press.
12. Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7, pp 155-170.
13. Gentner, D., & Markman, A. B. (1997). Structure mapping in analogy and similarity. *American Psychologist*, 52, 45-56.
14. Mahoney, JV, & Fromherz, MPJ (2002). Three main concerns in sketch recognition and an approach to addressing them. *AAAI Spring Symposium on Sketch Understanding*, pp 105—112, March 25-27 2002.
15. Marr, D. (1983). *Vision: A computational investigation into the human representation and processing of visual information*. W. H. Freeman
16. Mitchell, M. (1993). *Analogy-making as perception: A computer model*. The MIT Press.
17. Nicholson, S. and Forbus, K. (2002). Answering comparison questions in SHAKEN: A progress report. *Proceedings of the AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*, Palo Alto.
18. Nielsen, P. (1988). A qualitative approach to rigid body mechanics. *Technical Report No. UIUCDCS-R-88-1469*, Department of Computer Science, University of Illinois at Urbana-Champaign.
19. O'Hara, S. and Indurkha, B. (1995). Adaptation and redescription in the context of geometric proportional analogies. *AAAI-95 Fall Symposium on Adaptation of Knowledge for Reuse*, pp 80-86.
20. Palmer, S. (1999). *Vision Science: Photons to Phenomenology*. MIT Press.
21. Saund, E., and Moran, T, (1995). Perceptual Organization in an Interactive Sketch Editing Application. *ICCV '95*.