Visual Analogical Mapping

Paul Thagard, David Gochfeld, and Susan Hardy
Cognitive Science Laboratory
Princeton University
221 Nassau St., Princeton, NJ 08542[1]

Abstract

This paper describes some results of research aimed at understanding the structures and processes required for understanding analogical thinking that involves images and diagrams. We will describe VAMP.1 and VAMP.2, two programs for visual analogical mapping. VAMP.1 uses a knowledge representation scheme proposed by Janice Glasgow that captures spatial information using nested three-dimensional arrays. VAMP.2 overcomes some limitations of VAMP.1 by replacing the array representation with a scheme inspired by Minsky's Society of Mind and connectionism.

Introduction

Part of analogical thinking involves finding correspondences between structures that represent analogous problems. Various computational models of how mapping between analogs can be conducted have been proposed (SME: Falkenhainer, Forbus, and Gentner 1989; Gentner 1983; ACME: Holyoak and Thagard 1989).[2] Like the vast majority of AI programs, analogy programs such as SME and ACME represent analogs propositionally rather than visually.

But many analogies have a strong visual component. Consider the Duncker tumor problem that has been widely used in psychological experiments (Gick and Holyoak 1980, 1983). Subjects are told to try to figure out how to use an x-ray machine to destroy a tumor inside a patient without damaging the patient's flesh. The solution is to use a number of x-ray sources producing rays of diminished intensity that converge on the tumor and destroy it. Subjects are aided in coming up with this solution if they are told of a general whose strategy for attacking a fortress involved dispersing his army and having them converge on the fortress from different directions. Although all the information necessary can be represented propositionally, it is natural to produce a diagram or mental picture that shows the army and the rays converging on the tumor and the fortress from different directions. Gick (1985) and Beveridge and Parkins (1987) found that the use of diagrams improved subjects problem solving effectiveness on this problem.

But it is difficult to model the visual aspect of analogical reasoning using the knowledge representation techniques that have been most common in AI. Ideally, visual representations should serve to make mapping between analogs much easier than propositional representations, which require considerable work to place appropriate predicates and arguments in correspondence. If, for example, we had a visual representation of the Duncker problem, we could map the tumor problem to the fortress problem by simply superimposing an image of the one onto the other and identify by inspection the objects that correspond to each other, such as the tumor and fortress. We cannot expect the visual representation to do all the work of analogical mapping, since many predicates such as **cause** will not lend themselves to visual representation, but visual representation should help greatly with aspects of the problems that are easily pictured. Finke (1989) provides a convenient summary of the large body of psychological experimentation that supports the contention that human thinking involves an important visual component.[3]

[1] This research was supported by contract MDA903-89-K-0179 from the Basic Research Office of the U.S. Army Research Institute for the Behavioral and Social Sciences. Paul Thagard's current address is: Department of Philosophy, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1. Email: prthagard@logos.waterloo.edu.

[2] Mapping is also implicit in computational models of case-based reasoning (e.g. Riesbeck and Schank 1989). See also Mitchell and Hofstadter (1990).

[3] Finke and others distinguish between visual information (how things look) and spatial information (how things relate to each other) but I shall include both of these under the heading "visual."

This paper describes some results of research aimed at understanding the structures and processes required for understanding analogical thinking that involves images and diagrams. We will describe VAMP.1 and VAMP.2, two programs for visual analogical mapping. VAMP.1 uses a knowledge representation scheme proposed by Janice Glasgow that captures spatial information using nested three-dimensional arrays. VAMP.2 overcomes some limitations of VAMP.1 by replacing the array representation with a scheme inspired by Minsky's Society of Mind and connectionism.

VAMP.1

VAMP.1 (Visual Analogical Mapping Program) is based on the knowledge representation scheme for computational imagery that Glasgow and her colleagues have been developing (Glasgow 1990; Glasgow and Papadias in press; Papadias and Glasgow 1991). In the earlier computational model of Kosslyn (1980), quasi-pictorial images were represented by a configuration of points in a matrix; an image is displayed by selectively filling in cells of the matrix. An image, then, is construed as a two-dimensional array, with each entry like a pixel that is either on or off. Glasgow's scheme is more complex in two key respects. First, it takes images to be inherently three-dimensional, although two-dimensional projects can also be handled as a special case. Greater dimensionality obviously makes possible representation of more complex images such as those required for mental rotation. Second, the entries in the three-dimensional arrays can be encoded hierarchically, in that each entry is represented symbolically by a entry that can have a subimage. For example, a house can be represented by the array shown in Figure 1, with each symbolic entry such as "window" providing a pointer to another array. In sum, Glasgow's representational scheme takes images to be three-dimensional symbolic hierarchical arrays. Numerous important visual operations can be defined on Glasgow's arrays, including constructing symbolic arrays from propositional representations, comparing images using array information, and moving and rotating images.[4]

roof	roof	roof
window		
	door	

Figure 1: Array representation of a house.

We have developed a Common LISP implementation of parts of Glasgow's scheme and extended it to produce VAMP.1, a visual analogical mapping program. Given two arrays, VAMP.1 can do simple analogical mapping, putting the elements of the two arrays in correspondence with each other. VAMP.1 first checks to see if the arrays are the same size. If not, it scales them up to the size of the least common multiple of their sizes. For example, to compare a 4x4x4 array and a 6x6x6 array, VAMP.1 converts both arrays to 12x12x12 arrays. When both arrays are equal in size, VAMP.1 superimposes them and gives a list of all parts which are in corresponding cells.

As described in Thagard and Hardy (1992), VAMP.1 has been used to model the use by John Dalton (1808) of an analogy between the structure of the atmosphere involving molecules and a pile of shot. It is natural to construct a mental image of a pile of cannon balls with one ball nesting on four below which nest on nine below, and then transform this into a picture of the atmosphere consisting of atoms surrounded by heat similarly nesting. The representation of the pile of balls is not just the various slices shown, but the whole array which encapsulates a very large amount of spatial information. This encapsulation makes creating a visual analog trivial: all we have to do to produce a representation of the structure of the atmosphere is to replace each entry of BALL with an entry of ATOM. The hierarchical nature of the representation scheme is important because it allows us to substitute a complex of atom and heat, as Dalton recommended, rather than just atom.

Glasgow's knowledge representation scheme is very useful in suggesting how visual/spatial information can be stored and used. But it has some clear limitations. Arrays are too "boxy" to capture more complex spatial arrangements than *left, right, above, below*: a cannon ball sits above four others at roughly 60 degree angles, not directly above. Also not naturally represented in Glasgow's scheme are relations of containment. In the tumor problem, for example, the patient's flesh contains the tumor: there are not distinct objects of flesh filling all the adjacent boxes. From the perspective of processing, the Glasgow scheme has advantages in making the appropriate maps readily identifiable when the arrays coincide,

[4] Other computational models of visual thinking have been developed by Funt (1980), Shrager (1990), and Chandrasekaran and Narayanan (1990).

but does not suggest how partial maps might be found. In addition, using the array structures seems potentially inefficient, since they will contain various empty cells and have to be worked with in monolithic fashion. Accordingly, we have tried to retain some of the advantages of Glasgow's scheme while producing more flexible mappings.

VAMP.2

According to Marvin Minsky's provocative "Society of Mind" theory, each mind is made up of many small processes he calls *agents*. Minsky says (1986, p. 17): "Each mental agent by itself can only do some simple thing that needs no mind or thought at all. Yet when we join these agents in societies - in certain very special ways - this leads to true intelligence." We propose to reconceptualize Glasgow's scheme by imagining that corresponding to each box in the 3-D array there is a simple agent that can communicate with other agents representing other boxes. Each agent knows what other agents are adjacent to it in various directions. The agents can process information in parallel to provide answers to simpler questions. For example, if you want to know what is above the door in a visual representation, you can query all agents until you find one that has the door, then have that agent ask the agent above it what it has. This corresponds to simply looking at the door and then looking up above it.

Once you have a set of agents each of which has knowledge of the adjacent agents, you no longer need the array structure at all. The same information captured by the boxes in the 3-D array can be captured more locally by what the individual agents know about themselves and the adjacent agents. Moreover, much more flexible spatial structures can be used than simply left, right, above, and below as in the array: an agent can know that there is an agent above it and to the left at a particular angle. Agents can also possess another important kind of spatial information: what agents contain them or are contained by them.

For visual analogical mapping, each analog can be represented by a set of agents, and the computational problem is to put agents from different sets in communication with each other in such a way that the appropriate correspondences are found. For example, the agent for **tumor** in the Dunker problem must be put in contact with the agent for **fortress** in the other problem. Think of two competing baseball teams whose members shout at each other to find the players in corresponding positions; after an initial noisy display, the shortstops on each team will find each other, and so on. This example shows that the mapping may not be simple, since more than one pitcher on each team may correspond to more than one pitcher on the other.

VAMP.2 is a program that implements this kind of analogical mapping. It is written in the Common LISP Object System, for it is natural to encode society-of-mind ideas using object-oriented programming. Each thing in an analogy is represented by an agent, implemented as a CLOS object. For mapping purposes, we want to avoid the complexity of having every agent in one analog try to correspond to every agent in the other analog, so visual similarity used to screen for agents of mutual relevance: two agents only begin a relationship if the things they represent have similar appearance or containment relations. But once such a relationship is established between agents S1 from the source analog and T1 from the target analog, the agents adjacent to S1 can be put in correspondence with agents adjacent to T1 even if they have different shapes and spatial relations. We want, for example, to have S2 which is to the right of S1 establish a connection with T2 which is to the right of T1. Matters obviously become much trickier when the target contains more than one agent that is similar in shape to S1 so that we cannot tell right away which one should correspond to it.

To solve this problem, we have used connectionist techniques of parallel constraint satisfaction that worked well in earlier models of analogical mapping and retrieval, ACME and ARCS (Holyoak and Thagard 1989; Thagard, Holyoak, Nelson and Gochfeld 1990). ACME and ARCS showed that analogs can be retrieved from memory and mapped by satisfying a combination of semantic, structural, and pragmatic constraints. These constraints are represented in a connectionist network of units with excitatory and inhibitory links, and a simple settling process selects out what correspondences best satisfy the constraints. VAMP.2 uses constraints specific to visual representations that can however be viewed as special cases of constraints in the more general programs. We want to encourage mappings between things of similar appearance, encourage mappings between things with similar adjacencies and containment relations, and discourage one-many and many-one mappings.

For each pair of agents who establish a relationship for appearance or containment relations, VAMP.2 creates a mapping unit that represents the plausibility of their being in correspondence: we will write the mapping unit that pairs S1 and T1 as S1=T1. Mapping units form packages that tend to go together. If S1 is adjacent to S2, and T1 is adjacent to T2 in the same way, then the unit S2=T2 will be formed. We want the mappings S1=T1 and S2=T2 to go together, so a symmetric excitatory link is established between these two units. Similarly, if S1 contains S2 and T1 contains T2, then we want the mappings S1=S2 and T1=T2 to encourage each other, so an excitatory link is established between those units. A special unit that is always active is used to encourage mappings between agents representing things that are visually similar. VAMP.2 is given a verbal description of things and uses this to compute visual similarity, or visual similarity is specified by the programmer.[5] If S1 and T1 are visually similar, then an excitatory link is established between S1=T1 and the special unit. To discourage mappings that are not one-to-one, an inhibitory link will be created between S1=T1 and any units S*=T1 and S1=T* representing other ways of mapping S1 and T1. In VAMP.2, all links are symmetric. Once these networks are created, a simple connectionist settling algorithm is used to adjust the activation of units in parallel until they all settle and the winning and losing units are apparent. The appendix contains a precise description of the algorithms used by VAMP.2.

Now let us look at a simple example of VAMP.2 in operation. Holyoak and Koh (1987) did experiments using the Duncker tumor problem with another problem that is more isomorphic to it than the fortress problem. The filament problem requires finding a way to use a laser to fuse a broken filament inside a glass bulb without breaking the glass. Our representation of the two problems is portrayed in figure 2. The tumor is contained in flesh which is contained in a hospital room along with an x-ray source and the rays, which are to the left of the the patient. To provide a greater challenge for VAMP.2, the representation of the other problem has the laser and beam to the right of the the filament and glass, which are contained in a laboratory. VAMP.2 is given the information that there is some visual similarity between the beam and ray and between the laser and x-ray source.[6] It therefore creates the units BEAM=RAY and LASER=SOURCE. Similarity in containment relations leads to creation of units LAB=ROOM, GLASS=FLESH, and several others. Figure 3 shows all the units created by VAMP.2 along with their inhibitory links. After 44 cycles of updating, the units all achieve stable activations and the appropriate mappings, BEAM=RAY, LASER=SOURCE, FILAMENT=TUMOR,

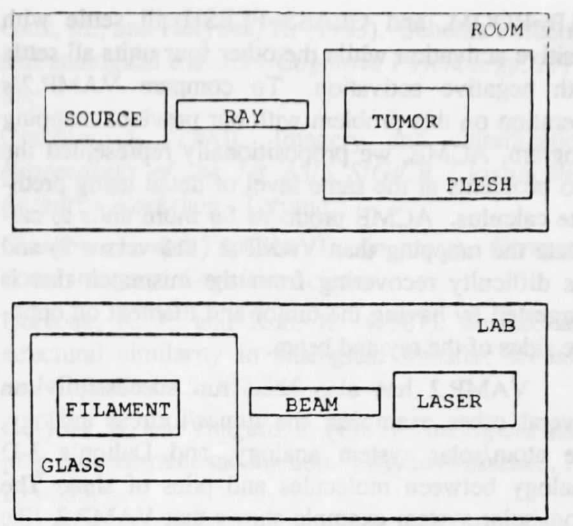

Figure 2. Diagrammatic representation of tumor and filament problems.

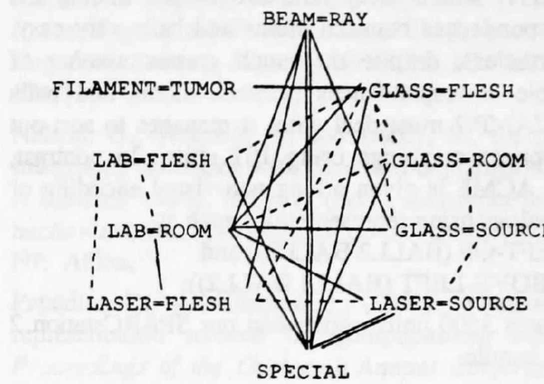

Figure 3. Network created to map tumor and filament representations.

Links to the special unit based on visual similarity are shown with double solid lines. Other links to the special unit based on adjacency and containment relations are shown by single lines, as are links between other units based on adjacency and containment relations. The dotted lines indicate inhibitory links that serve to discourage mappings that are not one-to-one.

[5] Ideally, the program would make this sort of judgment itself on the basis of a pictorial representation.

LAB=ROOM, and GLASS=FLESH all settle with positive activation while the other four units all settle with negative activation. To compare VAMP.2's operation on this problem with our previous mapping program, ACME, we propositionally represented the two problems at the same level of detail using predicate calculus. ACME produces far more units to calculate the mapping than VAMP.2 (115 versus 9) and has difficulty recovering from the mismatch that is suggested by having the tumor and filament on opposite sides of the ray and beam.

VAMP.2 has also been run successfully on several other examples: the tumor/fortress analogy, the atom/solar system analogy, and Dalton's 3-D analogy between molecules and piles of shot. The atom/solar system example shows that VAMP.2, like ACME but unlike SME, can perform one-many mappings when it is appropriate to do so. Given representations of a hydrogen atom with one electron and a solar system with several planets, VAMP.2 correctly maps the electron to each of the planets. The Dalton analogy is much trickier for VAMP.2 than for VAMP.1, where array structure makes finding the correspondences between atoms and balls very easy. Nevertheless, despite the much greater number of possible correspondences between atoms and balls that VAMP.2 must deal with, it manages to sort out appropriate mappings using 107 units. In contrast, when ACME is given a long non-visual encoding of the analogs using representations such as

(LEFT-OF (BALL2 BALL3)) and
(ABOVE-LEFT (BALL1 BALL2)),

it creates 3500 units, more than our SPARCstation 2 could handle.

VAMP.2 is by no means the final word on visual analogical mapping. While it has a much more flexible scheme for knowledge representation than VAMP.1, it still is limited in how well it can represent such visually complex matters as how rays converge at a point. Moreover, it does not address the crucial question of visually representing dynamic information of the sort that might be found in a movie-like mental image of rays shooting out and converging. The atom/solar system analogy can most effectively be conveyed by imagining electrons and planets in moving orbits. Finally, VAMP.2 performs mapping by visual representations alone, ignoring many cues that might be provided by proposition-based mapping. A powerful integrated mapping scheme could be built by having VAMP.2 work in concert with a program like ACME, with each program passing partial results back and forth, getting the most out of the different kinds of representation available. Our new system CARE already integrates analogical mapping with retrieval and rule based reasoning, and it should be possible to fit VAMP.2 into it gracefully (Nelson, Thagard, and Hardy, in press). VAMP.2 is already capable of modeling some of what is involved in transferring a solution to a source problem into one for the target problem: given a description of the filament problem that includes its solution with convergent beams, it maps part of the solution back to the target tumor problem.

[6] If this information is omitted, VAMP.2 still creates the units BEAM=RAY and LASER=SOURCE because of similar containment relations and number of adjacencies.

Many analogies in ordinary life and in science have a substantial visual component. We have shown that it is possible to start to model visual aspects of analogy without having to simulate the entire human perceptual system. While structured array representations have many attractive features, visual analogical mapping of complex examples requires a more flexible representation such as that inspired by Minsky's Society of Mind theory. The price of this flexibility is that additional mechanisms of parallel constraint satisfaction are needed to accomplish the mapping.

Appendix: VAMP.2 Algorithms

A. Map visually similar things.

For each thing S in the source image, and any thing T in the target image such that

a) T is the same type of thing as S,

b) T has been declared to be visually similar to S, or,

c) both S and T contain something,

create a mapping unit "S=T" and add this to M, the list of mapping units. Make an excitatory link between this unit and the visual special unit.

B. Map adjacencies of previously mapped things.

Copy M into N, and then repeat the following steps until there are no mapping units left in N:

1) Let N1 be the first unit in N.

2) Let S be the source thing that is mapped in N1, and T be the target thing mapped in N1.

3) Let A(S) be the list of things adjacent to S, and A(T) be the things adjacent to T.

4) For each thing AS1 in A(S), and any thing AT1 in A(T) which is the same direction from T as AS1 is from S, create a mapping unit "AS1=AT1", if it does not already exist.

5) Add AS1=AT1 to N and M, and create an excitatory link between it and unit N1.

6) Remove N1 from N, and repeat the above steps until N is empty.

C. Map contents of previously mapped things.

Again copy M into N and repeat the following step:

1) Let N1 be the first unit in N.

2) Let S be the source thing that is mapped in N1, and let T be the target thing that is mapped in N1.

3) Let C(S) be the list of things directly contained in S, and C(T) be the things directly contained in T.

4) For each thing CS1 in C(S), and any thing CT1 in C(T) that has the same number of adjacencies as does CS1, create a mapping unit "CS1=CT1", if it does not already exist.

5) Add CS1=CT1 to N and M. Create an excitatory link between this and N1.

6) Remove N1 from N, and repeat the above steps until N is empty.

D. Inhibit multiple mappings by creating inhibitory links.

E. Update activation of units until network settles.

The algorithms for D and E are the same as in ACME; see Holyoak and Thagard (1989), p. 314.

References

Beveridge, M., and Parkins, E. (1987). Visual representation in analogical problem solving. *Memory and Cognition, 15:* 230-237.

Chandrasekaran, B., and Narayanan, N. (1990). Integrating imagery and visual representations. *Proceedings of the 12th Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum, 670-677.

Dalton, J. (1808). *A new system of chemical philosophy.* London: Bickerstaff.

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

Finke, R. (1989). *Principles of mental imagery.* Cambridge, MA: MIT Press.

Funt, B., (1980). Problem solving with diagrammatic representations. *Artificial Intelligence, 13*, 201-230.

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

Gick, M. (1985). The effect of a diagram retrieval cue on spontaneous analogical transfer. *Canadian Journal of Psychology, 39:* 460-466.

Gick, M., and Holyoak, K., (1980). Analogical problem solving. *Cognitive Psychology, 12*, 306- 355.

Gick, M., and Holyoak, K. (1983). Schema induction and analogical transfer. *Cognitive Psychology, 15*, 1-38.

Glasgow, J. (1990). Imagery and classification. *Proceedings of the 1st ASIS SIG/CR Classification Research Workshop.* Toronto.

Glasgow, J., and Papadias, D. (in press). Computational imagery. *Cognitive Science.*

Holyoak, K. J., and Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory and Cognition, 15*, 332-340.

Holyoak, K. and Thagard, P. (1989) Analogical mapping by constraint satisfaction. *Cognitive Science, 13*, 295-355.

Kosslyn, S. (1980). *Image and mind.* Cambridge: Harvard University Press.

Minsky, M. (1986). *The society of mind.* New York, Simon and Schuster.

Mitchell, M., and Hofstadter, D. (1990). The right concept at the right time: How concepts emerge as relevant in response to context-dependent pressures. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Erlbaum, 174-181.

Nelson, G., Thagard, P., and Hardy, S., (in press). Integrating analogies with rules and explanations. In J. Barnden and K. Holyoak (Eds.), *Advances in Connectionism*, vol. 2, *Analogical Connections*, Norwood, NJ: Ablex.

Papadias, D., and Glasgow, J., (1991). A knowledge representation scheme for computational imagery. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Erlbaum, 49-54.

Riesbeck, C., and Schank, R. (1989). *Inside case-based reasoning.* Hillsdale, NJ: Erlbaum.

Thagard, P., Holyoak, K., Nelson, G., and Gochfeld, D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence, 46*, 259-310.

Thagard, P., and Hardy, S. (1992). Visual thinking in the development of Dalton's atomic thinking. *Proceedings of the Ninth Canadian Conference on Artificial Intelligence*, Vancouver.