A Fractal Analogy Approach to the Raven's Test of Intelligence

Keith McGregor, Maithilee Kunda, and Ashok Goel

Design & Intelligence Laboratory, School of Interactive Computing,
Georgia Institute of Technology, Atlanta, GA 30332, USA
{keith.mcgreggor, mkunda}@gatech.edu, goel@cc.gatech.edu

Abstract

We present a fractal technique for addressing geometric analogy problems from the Raven's Standard Progressive Matrices test of general intelligence. In this method, an image is represented fractally, capturing its inherent self-similarity. We apply these fractal representations to problems from the Raven's test, and show how these representations afford a new method for solving complex geometric analogy problems. We present results using the fractal algorithm on all 60 problems from the Standard Progressive Matrices version of the Raven's test.

Introduction

Psychometrics entails the theory and technique of the quantitative measurement of psychological variables such as intelligence and aptitude. Research on "computational psychometrics" dates at least as far back as Evan’s (1968) ANALOGY program, which addressed geometric analogy problems on the Miller Geometric Analogies test of intelligence. Bringsjord & Schimanski (2003) have proposed psychometric AI, i.e., AI that can pass psychometric tests of intelligence, as a possible mechanism for measuring and comparing AI.

Extant computational theories of geometric analogy problems that appear on typical psychometric tests are united in their use of propositional representations. In this paper, we present a computational technique that uses fractal representations (Mandelbrot 1982) to address geometric analogy problems from the Raven’s test. In particular, the technique uses fractal image representations that rely only on the grayscale pixel values of input images and are grounded in the theory of fractal image compression (Barnsley & Hurd 1992).

State of the Art

The Raven’s Progressive Matrices tests (Raven, Raven, & Court 1998) are a collection of standardized intelligence tests that consist of visually presented, geometric analogy problems in which a matrix of geometric figures is presented with one entry missing, and the correct missing entry must be selected from a set of answer choices. Figure 1 shows an example of a problem that is similar to one of the problems in the Standard Progressive Matrices (SPM). The SPM consists of 60 problems divided into five sets of 12 problems each (sets A, B, C, D & E), roughly increasing in difficulty both within and across sets.

Computational Accounts of the Raven’s Test

There have been several attempts to provide a computational account of solving the Raven’s test, and with one exception, all previous proposed accounts have been propositional in nature, i.e. visual inputs from the test are converted into propositional descriptions before any problem solving is carried out. Carpenter, Just, & Shell (1990) used a production system that took as inputs hand-coded propositional descriptions of Raven's problems from the Advanced Progressive Matrices (APM) and selected an appropriate rule to solve each problem. The rules were generated by the authors from inspection of the APM

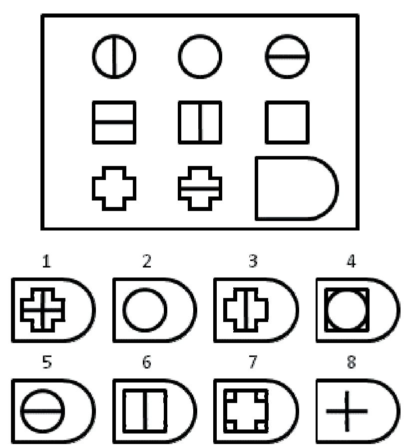

Figure 1. Example problem similar to one in the Standard Progressive Matrices (SPM) test.

beforehand and validated in experimental studies of subjects taking the test with verbal reporting of solution strategies. Their program correctly solved 32 of 34 problems on the APM, scoring as well as high-performing college students did on the test.

Bringsjord and Schimanski (2003) used a theorem-prover system to solve selected Raven's problems stated in first-order logic, although the system appears never to have been tested on any of the complete Raven's tests. Lovett, Forbus, & Usher (2007) developed a system that combined automated sketch understanding with the structure-mapping technique for analogy to solve problems from the Standard Progressive Matrices (SPM) test. This system took as inputs problem entries sketched in Powerpoint as segmented shape objects and translated these shapes into propositional descriptions of each entry. Then, a two-stage structure-mapping process was used to select the answer which most closely fulfilled inferred analogical relations in the matrix entries. This system was tested on sets B and C from the SPM and solved 12 and 10 of the 12 problems in each set, respectively. (It is not clear how this performance might compare to that of humans without results from the other three sets.)

In all of these computational methods, the problem solving strategy is propositional, regardless of whether the system takes propositional inputs to begin with or takes visual inputs and automatically converts them to propositions. However, insofar as we can tell, there is little in the Raven's tests that requires a purely propositional account for all problems. In fact, Hunt (1974) gave a theoretical account of the information processing demands of certain problems from the Advanced Progressive Matrices (APM), in which he proposed that two qualitatively different solution algorithms—"Gestalt," which uses visual representations and perceptually based operations, and "Analytic," which uses feature-based representations and logical operations—could yield identical results on at least portions of the test. However, these two algorithms were never implemented.

Computational Methods for Visual Analogies

Outside computational psychometrics, Gross & Do (2000) describe a heuristic program that retrieves design drawings from a library similar to a sketch drawn by hand. Yaner & Goel (2006) describe a computer system that uses partial constraint satisfaction to retrieve design drawings from a library similar to an input drawing generated by a vector-graphics tool. Croft & Thagard (2002) describe a computer program that generates analogical mappings between visuospatial representations of target and source problems. Davies, Goel, & Yaner's (2009) Proteus system not only generates mappings between visuospatial representations of target and source problems, but also transfers problem-solving procedures from the source to the target problem. Hofstadter & McGraw's (1995) Letter Spirit program takes a stylized seed letter as input (e.g., f but with the crossbar suppressed) and outputs an entire font, from a to z, in the same style as the seed. Again, the extant computational models of visual analogy are united in their use of propositional representations.

Prior Work on Visual Analogies

We are unaware of any previous work on using fractal representations to address either geometric analogy problems of any kind or other intelligence test problems. However, there has been some work in computer graphics on image analogies for texture synthesis in image rendering. Consider the problem in which, given an image A and its filtered rendering A′, the goal is to render a different image B in the same style as A′. For this class of problems, Hertzmann et al. (2001) describe a technique that abstracts the textural filter between A and A′ and then applies it to B to generate the desired image B′. The technique uses a Gaussian pyramid technique to construct a multi-scale image and then determines features for individual pixels in this image by examining luminance distributions in pixel neighborhoods.

Both the class of problems of we address and the fractal technique we have designed differ significantly from those of Hertzmann, et al. We explicitly represent each image fractally and use this representation to determine its transfer. As a result, the underlying pixels, while significant in determining a match, become irrelevant for transfer, because the collage theorem (see below) guarantees convergence into the target irrespective of the source. Further, our method determines similarity between two images via systematic recall and construction of feature sets. Finally, the features our method extracts from the fractal encodings are indicative of both geometric and colorimetric properties.

Our Current Work

The main goal of our work was to evaluate whether the Raven's Standard Progressive Matrices test could be solved using purely visual representations, i.e. without converted the image inputs into propositional descriptions during any part of the reasoning process. We use fractal representations, which encode transformations between images, as our primary non-propositional representation of problem information.

Our system operates on problem inputs that have been scanned directly from a hard copy of the Raven's test and contain the usual rough alignments and pixel-level artifacts. Problem entries are converted to fractal representations as described in the next section, and relationships among these fractal representations are used to choose the best answer. At no point during the solution process are inputs converted to any kind of propositional form in terms of shapes, colors, lines, edges, or any other visually segmented entity; only the raw RGB pixel values are used. Now, we describe in detail the fractal solution algorithm and results showing the performance of this system on all 60 problems from the SPM test.

Earlier (McGreggor, Kunda, & Goel 2010) we had proposed the use of fractal analogies to address problems on the Raven's test. In this paper, for the first time, we describe the technique we have developed and present actual results for problems on the Raven's test.

Fractal Representations

Consider the general form of an analogy problem as:

$$A : B :: C : ?$$

For visual analogy, we can presume each of these analogy elements to be a single image. Some unknown transformation T can be said to transform image A into image B, and likewise, some unknown transformation T' transforms image C into the unknown answer image.

The central analogy in the problem may then be imagined as requiring that T is analogous to T'. In other words, the answer will be whichever image X yields the most analogous transformation. Using fractal representations, we shall define the most analogous transform T' as that which shares the largest number of fractal features with the original transform T.

Mathematical Basis

The mathematical derivation of fractal image representation expressly depends upon the notion of real world images, i.e. images that are two dimensional and continuous (Barnsley & Hurd, 1992). A key observation is that all naturally occurring images we perceive appear to have similar, repeating patterns. Another observation is that no matter how closely you examine the real world, you find instances of similar structures and repeating patterns. These observations suggest that it is possible to describe the real world in terms other than those of shapes or traditional graphical elements—in particular, terms which capture the observed similarity and repetition alone.

Computationally, determining fractal representation of an image requires the use of the fractal encoding algorithm. The collage theorem (Barnsley & Hurd, 1992) at the heart of the fractal encoding algorithm can be stated concisely, in terms of a source image S and a destination image D:

For any particular real world image D, there exists a finite set of affine transformations T which, if applied repeatedly and indefinitely to any other real world image S, will result in the convergence of S into D.

We now present the fractal encoding algorithm in detail.

The Fractal Encoding Algorithm

Given a destination image D, the fractal encoding algorithm seeks to discover the set of transformations T. The algorithm is considered “fractal” for two reasons: first, the affine transformations chosen are generally contractive, which leads to convergence, and second, the convergence of S into D can be shown to be the mathematical equivalent of considering D to be an attractor (Barnsley & Hurd, 1992).

The steps for encoding an image D in terms of another image S are shown in Algorithm 1. The decomposition of D into smaller images can be achieved through a variety of methods. In our present implementation, we merely choose to subdivide D in a regular, gridded fashion, typically choosing a grid size of either 8x8 or 32x32 pixels. Alternate decompositions could include irregular subdivisions, partitioning according to some inherent colorimetric basis, or levels of detail.

Decompose destination image D into a set of *N* smaller images $\{d_1, d_2, d_3, \ldots, d_n\}$. These individual images are sets of points.

For each image d_i:

- Examine the entire source image S for an equivalent image s_i such that an affine transformation of s_i will result in d_i. This affine transformation will be a 3x3 matrix, as the points within s_i and d_i under consideration can be represented as the 3D vector <x, y, c> where c is the (grayscale) color of the 2D point <x,y>. Collect all such transforms into a set of candidates C.
- Select from the set of candidates the transform which most minimally achieves its work, according to some predetermined, consistent metric.
- Let T_i be the representation of the chosen affine transformation of s_i into d_i.

The set $T = \{T_1, T_2, T_3, \ldots, T_n\}$ is the fractal encoding of the image D.

Algorithm 1. Fractal Encoding

Searching and Encoding

The search of the source image S for a matching fragment is exhaustive, in that each possible correspondence s_i is considered regardless of its prior use in other discovered transforms. Also, for each potential correspondence, each transformation under a restricted set of similitude transformations is considered. A similitude transformation is a composition of a dilation, orthonormal transformation, and translation. Our implementation presently examines each potential correspondence under identity (I), horizontal (HF) and vertical (VF) reflections, and 90° (R90), 180° (R180), and 270° (R270) orthonormal rotational transformations. We fix our dilation at a value of either 1.0 or 0.5, depending upon whether the source and target image are identical. The translation is found as a consequence of the search algorithm.

Once a transformation has been chosen, we construct a compact representation of it called a fractal code. A fractal code T_i is a 5-tuple, $< <s_x, s_y>, d_x\text{-}s_x, d_y\text{-}s_y, k, c >$, where $<s_x, s_y>$ is the location of the leftmost and topmost pixel in s_i; $<d_x, d_y>$ is the location of the left most and top most pixel in d_i; $k \in \{I, HF, VF, R90, R180, R270\}$ indicates which affine transformation is to be used; and $c \in [\,-255, 255\,]$

indicates the overall color shift to be added uniformly to all elements in the block.

Note that the choice of source image S is arbitrary. Indeed, the image D can be fractally encoded in terms of itself, by substituting D for S in the algorithm. Although one might expect that this substitution would result in a trivial encoding (in which all fractal codes correspond to an identity transform), in practice this is not the case, for we want a fractal encoding of D to converge upon D regardless of chosen initial image. For this reason, the size of source fragments considered is taken to be twice the dimensional size of the destination fragment, resulting in a contractive affine transform. Similarly, color shifts are made to contract. The fractal encoding algorithm, while computationally expensive in its exhaustive search, transforms a real world image into a much smaller set of fractal codes, which form, in essence, an instruction set for reconstituting the image.

Determining Fractal Features

As we have shown, the fractal representation of an image is a set of specific affine, similitude transformations, i.e. a set of fractal codes, which compactly describe the geometric alteration and colorization of fragments of the source image that will collage to form the destination image. While it is tempting to treat contiguous subsets of these fractal codes as features, we note that their derivation does not follow strictly Cartesian notions (e.g. adjacent material in the destination might arise from strongly non-adjacent source material). Accordingly, we consider each of these fractal codes independently, and construct candidate fractal features from individual codes.

In our present implementation, each fractal code $< <s_x, s_y>, d_x\text{-}s_x, d_y\text{-}s_y, k, c >$ yields a small set of features, generally formed by constructing subsets of the tuple. These features are determined in a fashion to encourage both position-, affine-, and colorimetric-agnosticism, as well as specificity. In the present implementation of our algorithm, we generate $P(5,2)+P(5,3)+P(5,4) = 85$ distinct features for each fractal code, where $P(n, m)$ refers to the permutation equation.

A Fractal Process for Geometric Analogy

To find analogous transforms, our algorithm first visits memory to retrieve a set of candidate solution images X to form candidate solution pairs in the form <C, X>. For each candidate pair of images, we generate the fractal encoding of the candidate image X in terms of the former image C. As we illustrated earlier, from this encoding we are able to generate a large number of fractal features per transform. We store each transform in a memory system, indexed by and recallable via each associated fractal feature.

To determine which candidate image results in the most analogous transform to the original problem transform *T*, we first fractally encode that relationship between the two images A and B. Next, using each fractal feature associated with that encoding, we retrieve from the memory system those transforms previously stored as correlates of that feature (if any). Considering the frequency of transforms recalled, for all correlated features in the target transform, we then calculate a measure of similarity.

This metric reflects similarity as a comparison of the number of fractal features shared between candidate pairs taken in contrast to the joint number of fractal features found in each pair member (Tversky 1977). In our present implementation, the measure of similarity S between the candidate transform *T′* and the target transform *T* is calculated using the ratio model:

$$S(T,T') = f(T \cap T') / [f(T \cap T') + \alpha f(T-T') + \beta f(T'-T)]$$

where $f(Y)$ is the number of features in the set Y. As we progressed in our experiments, we found that significant discrimination between candidate answers could be found by setting $\alpha = 1.0$ and $\beta = 0.0$. Our final formula for similarity thus becomes:

$$S(T,T') = f(T \cap T')/ f(T)$$

This calculation determines the similarity between unique pairs of transforms. However, the Raven's test, even in its simplest form, poses an additional problem in that many such pairs may be formed.

Reconciling Multiple Analogical Relationships Roughly 40% of the Raven's Standard Progressive Matrices (SPM) test contains 2x2 matrix problems; the remaining 60% contains 3x3 matrices. In 2x2 problems, there are two apparent relationships for which analogical similarity must be calculated: the horizontal relationship and the vertical relationship. Closer examination of such problems, however, reveals two additional relationships which must be shown to hold as well: the two diagonal relationships. Furthermore, not only must the "forward" version of each of these relationships be considered but also the "backward" or inverse version. Therefore for a 2x2 Raven's problem, we must determine eight separate measures of similarity for each of the possible candidate solutions.

The 3x3 matrix problems from the SPM introduce not only more pairs for possible relationships but also the possibility that elements or subelements within the images exhibit periodicity. The number of potential analogical relationships blooms significantly beyond that of the simpler 2x2 form. In the present implementation of our system, we consider 48 of these relationships concurrently.

Relationship Space For each candidate solution, we consider the similarity of each potential analogical relationship as a value upon an axis in a large “relationship space.” The dimensionality of this space is determined by the problem at hand: for 2x2 problems, the space is 8 dimensional; for 3x3 problems, the space is (at least) 48 dimensional. We currently do not favor any particular relationship; that is, we do not, as an example, weight more decisively those values we find upon the horizontal relationships over those upon the vertical relationships.

Treating Maximal Similarity as Distance To specify the overall fit of a candidate solution, we construct a vector in this multidimensional relationship space and determine its length, using a Euclidean distance formula. The candidate solution with the longest vector length is chosen as the solution to the problem.

Example

Figure 2 shows an example problem with four candidate answers. The fractal algorithm begins by determining the fractal encoding of each of the exhibited relationships in the problem, along with their possible analogous answer pairing. For example, we have a pairing of the upper left and the upper right images (one block, two blocks), for which we would examine an analogous pairing of the lower right image (one circle) with each of the four answer choices (A, B, C, and D).

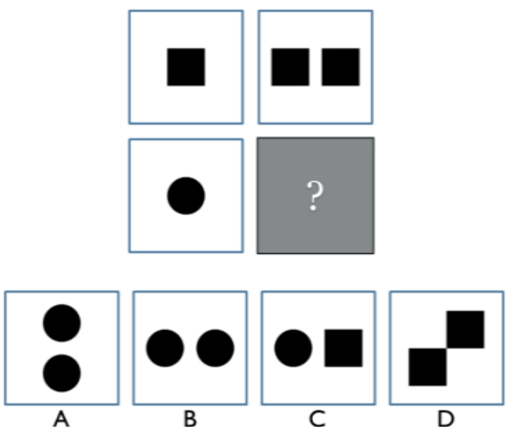

Figure 2. Example problem to illustrate fractal algorithm.

The steps for determining the fractal representation are the same for each of the pairings. For this example, we consider how the lower left image (one circle) might be encoded as a pairing with answer choice B (two circles).

First we decompose the target image into blocks. Each image is 231x231 pixels, and we shall break the image into 16x16 pixel blocks, padding the image as necessary to make for a whole number of blocks. This results in 15x15 = 225 blocks that completely tessellate the target image.

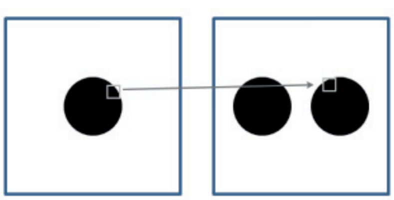

Figure 3. Illustration of block matching across images.

Next, for each of the target blocks, we examine the source image for the best possible matching similitude transformation. In this example, block #85 most closely matches a block in the source image which has <<140, 89>, <144, 80>, R90, +1> as its fractal code (meaning "move the source area from <140,89> to a target area at <144,80>, rotating 90°, and adjust the average grayscale value of each pixel by +1). The fractal representation of the transformation from "one circle" to "two circles" is the set of all 225 of these fractal codes (one fractal code for each block in the target image).

For each pairing, we create features from permutations of aspects of each of the 225 fractal codes, and we use each of these features as a retrieval key, storing the pairing into a memory system. Note that while we generate 85 distinct features from each of the codes, a particular key might be redundant, so the number of overall keys (that is, features) with which we index a pairing is likely to be less than the expected maximum of 225x85 = 19125 possible features. As an example, for the "one circle" / "two circle" pairing, we find that there are 1563 unique features.

To answer the problem, we first calculate the similarity metric for each of the candidate pairings, within each of the eight relationships present in this problem. For simplicity, we consider here only the top horizontal transformation ("one block" becomes "two blocks") and how it corresponds to the transformation of the lower right image ("one circle") into each of the candidate answers. We shall label this comparison between these transformations "Relation1." We have these calculated similarity metrics for the candidates, for Relation1:

A: 584/(584 + 1.0*799 + 0.0*927) = 0.422
B: 778/(778 + 1.0*605 + 0.0*785) = 0.563
C: 842/(842 + 1.0*541 + 0.0*638) = 0.609
D: 607/(607 + 1.0*776 + 0.0*961) = 0.439

As an example, the top horizontal transformation has 1383 features and Answer A's Relation1 has 1511 features. 584 of these features the two share in common. Thus, the intersection feature count is 584, and we divide that by the feature count of the horizontal transformation (1383), to arrive at the Relation1 similarity value of 0.422.

We continue to calculate similarities for each answer, for each of the eight relationships, which leads to a 8-tuple similarity vector in relationship space for each answer:

A:<0.422, 0.601, 0.527, 0.713, 0.427, 0.408, 0.418, 0.462>
B:<0.563, 0.639, 0.762, 0.693, 0.489, 0.446, 0.568, 0.670>
C:<0.609, 0.639, 0.797, 0.687, 0.484, 0.449, 0.608, 0.701>
D:<0.439, 0.609, 0.529, 0.738, 0.436, 0.431, 0.439, 0.464>

We then calculate the Euclidean distance for each vector:

A overall similarity = 1.436
B overall similarity = 1.731
C overall similarity = 1.784
D overall similarity = 1.474

We thus determine that the most similar answer, and the fractal algorithm's solution to the problem, is answer C.

Results on the Raven's Test

To create inputs for the fractal algorithm, each page from the SPM test booklet was scanned, and the resulting greyscale images were rotated to roughly correct for page alignment issues. Then, the images were sliced up to create separate image files for each entry in the problem matrix and for each answer choice. These separate images were the inputs to the fractal algorithm for each problem; no futher image processing or cleanup was performed, despite the presence of numerous pixel-level artifacts and minor alignment issues. The fractal algorithm attempted to solve each SPM problem independently, i.e. no information was carried over from problem to problem.

Performance on the SPM

There are three main assessments that can be made following the administration of the SPM to an individual: the total score, which is given simply as the number of correct answers; an estimate of consistency, which is obtained by comparing the given score distribution to the expected distribution for that particular total score; and the percentile range into which the score falls, for a given age and nationality (Raven, Raven, & Court 1998).

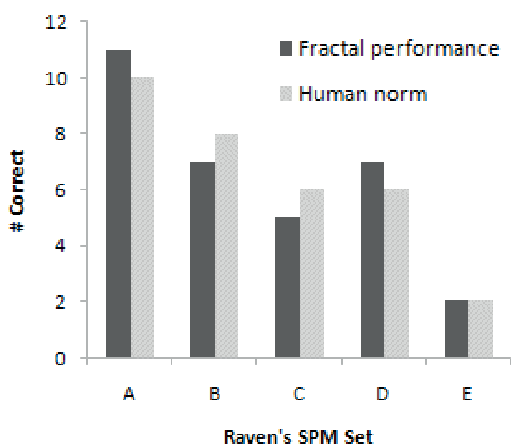

Figure 4. SPM scores ordered by set for fractal algorithm (dark) and human norms for given total score (light).

The total score obtained by the fractal algorithm was 32 correct out of 60 problems. The score breakdown by set, along with the expected score composition for a total score of 32 are shown in Figure 4. A score is "consistent" if the difference between the actual score and the expected score for any given set is no more than +/- 2 (Raven, Raven, & Court 1998). Inconsistent scores may result from test takers not understanding the test instructions, randomly guessing, or trying to choose incorrect answers to artificially lower the total score, for example. The score differences for the fractal algorithm on each set were no more than +/-1. For a human test-taker, this score distribution would indicate that the test results do provide a valid measure of the individual's general intellectual capacity. This score pattern illustrates that the results achieved by the algorithm fall well within typical human norms on the SPM.

Finally, the total score can be compared to age-group and national norms to determine percentile rankings. Using norms from the United States, we see that a total score of 32 corresponds to the 95th percentile for children around 7 years old, the 50th percentile for children around 9.5 years old, and the 5th percentile for children around16.5 years old (Raven, Raven, & Court 1998).

Conclusions

We have described a fractal technique for addressing geometry analogy problems of the kinds that appear on many psychometric tests of human intelligence. This technique works directly on the visual inputs, without any need to extract propositional representations from them. We have applied the technique to the entire Raven's Standard Progressive Matrices test, and found that it successfully solves 32 of 60 problems. The performance of our program would place it at the 50th percentile for 9-10 year old children. We believe that the fractal technique described above can be enhanced significantly and we anticipate improved results in the near future.

Fractal representations are analogical representations in that they are structurally isomorphic to the images they represent. The collage theorem provides a rigorous characterization of this structural isomorphism: The tessellation of the destination image of each of the transforms and the subsequent search and discovery of a "best-fit" companionable partition of the source image for each tessellate can be viewed mathematically as a bijection mapping between the images. This mapping is accomplished via a similitude transform.

Similarity and analogy often have been viewed as central to theories of intelligence (e.g., Evans 1968). Hofstadter (1995), among others, has posited that analogy forms the core of human cognition. Fractal representations add the powerful idea of self-similarity. In fact, the nature of these representations places a strong emphasis on discovering self-similarity. In the fractal technique, each discovered self-similar mapping is rendered into feature fragments without bias or preference. The contractive nature of the transformations on the fractal representations guarantees convergence.

While the use of fractal representations is central to our technique, the emphasis upon visual recall in our solution afforded by features derived from those representations is also important. We take the position that placing candidate transformations into memory, indexed via fractal features, affords a new method of discovering image similarity. That images, encoded either in terms of themselves or other images, may be indexed and retrieved without regard to shape, geometry, or symbol, suggests that the fractal representation bears further exploration not only as regards solutions to problems akin to the RPM, but also to those of general visual memory and recall.

Acknowledgments

This research has been supported by an NSF grant (IIS Award #0534266) entitled "Multimodal Case-Based Reasoning in Modeling and Design," by ONR through an NDSEG fellowship, and by the NSF GRFP fellowship program.

References

Barnsley, M. F., & Hurd, L. P. 1992. Fractal Image Compression. Boston, MA: A.K. Peters.

Bringsjord, S., & Schimanski, B. 2003. What is artificial intelligence? Psychometric AI as an answer. IJCAI, 18, 887–893.

Carpenter, P., Just, M., & Shell, P. 1990. What one intelligence test measures: a theoretical account of the processing in the Raven Progressive Matrices Test. Psychol. Review, 97, 404-31.

Croft, D., & Thagard, P. 2002. Dynamic imagery: A computational model of motion and visual analogy. In L. Magnani & N. J. Nersessian (Eds.), Model-based reasoning: Science, technology, values (pp. 259-274).

Davies, J., Goel, A., & Yaner, P. 2008. Proteus: A theory of visual analogies in problem solving. Knowledge-Based Systems, 21, 636–654.

Evans, T. 1968. A program for the solution of a class of geometric-analogy intelligence-test questions. In M. Minsky (Ed.), Semantic Information Processing (pp. 271-353). Cambridge, MA: MIT Press.

Gross, M. D., & Do, E. Y. 2000. Drawing on the Back of an Envelope: A framework for interacting with application programs by freehand drawing. Computers & Graphics, 24(6), 835-849.

Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., & Salesin, D. 2001. Image analogies. In SIGGRAPH (pp. 327–340).

Hofstadter, D., & Fluid Analogies Research Group (Eds.). 1995. Fluid concepts & creative analogies: Computer models of the fundamental mechanisms of thought. New York: Basic Books.

Hofstadter, D., & MacGraw, G. 1995. LetterSpirit. In D. Hofstadter & Fluid Analogies Research Group (Eds.), Fluid concepts & creative analogies: Computer models of the fundamental mechanisms of thought. New York: Basic Books.

Hunt, E. 1974. Quote the raven? Nevermore! In L. Gregg (Ed.), Knowledge and cognition (pp. 129–158). Hillsdale, NJ: Erlbaum.

Lovett, A., Forbus, K., & Usher, J. 2007. Analogy with qualitative spatial representations can simulate solving Raven's Progressive Matrices. Proc. 29th Annual Conf. Cognitive Science Society (pp. 449-454).

Mandelbrot, B. 1982. The fractal geometry of nature. San Francisco: W.H. Freeman.

McGreggor, K., Kunda, M., & Goel, A. 2010. A fractal approach towards visual analogy. In Proc. International Conference on Computational Creativity, Lisbon, Portugal, January 9-11.

Raven, J., Raven, J. C., & Court, J. H. 1998. Manual for Raven's Progressive Matrices and Vocabulary Scales. San Antonio, TX: Harcourt Assessment.

Tversky, A. 1977. Features of similarity. Psychological Review, 84(4), 327-352.

Yaner, P. W., & Goel, A. K. 2006. Visual analogy: Viewing analogical retrieval and mapping as constraint satisfaction problems. Applied Intelligence, 25(1), 91–105.