Fractally Finding the Odd One Out: An Analogical Strategy For Noticing Novelty

Keith McGreggor and Ashok Goel

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

Abstract

The Odd One Out test of intelligence consists of 3x3 matrix reasoning problems organized in 20 levels of difficulty. Addressing problems on this test appears to require integration of multiple cognitive abilities usually associated with creativity, including visual encoding, similarity assessment, pattern detection, and analogical transfer. We describe a novel fractal strategy for addressing visual analogy problems on the Odd One Out test. In our strategy, the relationship between images is encoded fractally, capturing important aspects of similarity as well as inherent self-similarity. The strategy starts with fractal representations encoded at a high level of resolution, but, if that is not sufficient to resolve ambiguity, it automatically adjusts itself to the right level of resolution for addressing a given problem. Similarly, the strategy starts with searching for fractally-derived similarity between simpler relationships, but, if that is not sufficient to resolve ambiguity, it automatically shifts to search for such similarity between higher-order relationships. We present preliminary results and initial analysis from applying the fractal technique on nearly 3,000 problems from the Odd One Out test.

Keywords: computational creativity, visual representations, analogical reasoning and inference, cognitive strategy, psychometrics

Computational Psychometrics

Psychometrics entails the theory and technique of quantitative measurement of intelligence, including factors such as personality, aptitude, knowledge, creativity, and academic achievement. AI 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. Recently, 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.

Visual analogies are common on standardized intelligence tests that measure personality, aptitude, knowledge, creativity, and academic achievement, such as the Miller's Geometric Analogies test, the Wechsler's intelligence test (Wechsler 1939), and the Raven's Progressive Matrices test (Raven et al. 1998). Some tests of intelligence, such as the Wechsler's, contain a mix of verbal and visual questions. Others, such as the Raven's Progressive Matrices test, consist of purely visual analogy problems. There is general agreement that addressing the matrix reasoning problems on tests such as Raven's requires integration of multiple abilities including visual encoding, pattern detection, rule integration, similarity assessment, analogical transfer, and problem solving.

Most computational models of the Raven's intelligence test rely on hand-coded propositional representations of the visual inputs (e.g., Bringsjord & Schimanski 2003; Carpenter, Just, & Shell 1990; Lovett, Forbus, & Usher 2007). Given that the inputs and outputs of all problems on the Raven's test are visual, recently there have been have some attempts at building computational models based on (purely) visual representations and reasoning (reference suppressed). The standard Raven's test however consists of only 60 problems.

In this paper, we present preliminary results and initial analysis from applying our technique to the much larger corpus of problems from the Odd One Out test of intelligence containing nearly 3000 problems (Hampshire 2010). Like our prior work on Raven's (McGreggor et al. 2011; Kunda et al. 2010), our technique here uses fractal representations of the problems on the test. However, our work on the Odd One Out test develops also develops fractal representations into a cognitive strategy. Two of the advantages of our new strategy are that (1) the technique starts with fractal representations encoded at a high level of resolution, but, if that is not sufficient to resolve ambiguity, it automatically adjusts itself to the right level of resolution for addressing a given problem, and (2) the strategy starts with searching for similarity between simpler relationships, but, if that is not sufficient to resolve ambiguity, it automatically shifts its searches for similarity between higher-order relationships.

Visual Analogies

In the context of the Raven's test of intelligence, Carpenter, Just, & Shell (1990) used a production system that took as inputs hand-coded propositional descriptions of 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 beforehand and validated in experimental studies of subjects taking the test with verbal reporting of solution strategies. Bringsjord and Schimanski (2003) used a theorem-prover system to solve selected Raven's problems stated in first-order logic. Lovett, Forbus, & Usher (2007) developed a system that combined automated sketch understanding with the structure-mapping technique for analogy to solve a small subsrt of 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. Once again, in all these extant computational models, 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.

Hunt (1974) gave a theoretical account of the information processing demands of certain problems from Raven's Advanced Progressive Matrices test, in which he proposed that two qualitatively different solution algorithms—"Gestalt," which uses imagistic 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, the algorithms were never implemented.

The recently developed Odd One Out test of intelligence (Hampshire 2010) was algorithmically designed to examine brain activity while solving problems at various levels of difficulty. It consists of 3x3 matrix reasoning problems organized in 20 levels of difficulty. Like the Raven's test, addressing the Odd One Out test requires multiple abilities ranging from visual encoding to pattern detection, from rule invocation to problem solving, from similarity assessment to analogical transfer. Unlike the Raven's test, the Odd One Out test seeks to discover dissimilarity, an ability at the core of attention and visual search.

Visual Analogies and Fractals

The main goal of our work is to evaluate whether visual analogy problems may be solved using purely imagistic representations, without converting the images into propositional descriptions during any part of the reasoning process. We use fractal representations, which encode transformations between images, as our primary non-propositional imagistic representation. The representation is imagistic (or analogical) in that it has a structural correspondence with the images it represents.

Analogies in general are based on similarity and repetition. Fractal representations capture self-similarity and repetition at multiple scales. Thus, we believe fractal representations to be a good choice for addressing analogy problems. Our fractal technique is grounded in the mathematical theory of general fractals (Mandelbrot 1982) and specifically of fractal image compression (Barnsley & Hurd, 1992). 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 (Hertzmann et al. 2001).

Fractal Representations

Consider the general form of an analogy problem as being A : B :: C : D. One can interpret this visually as shown in the example in Figure 1.

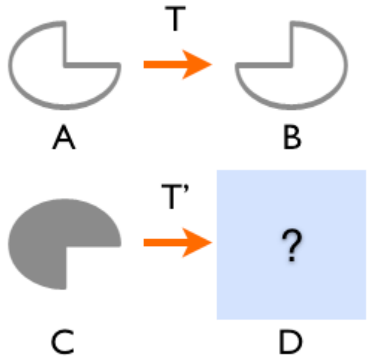


Figure 1. An Example of Visual Analogy.

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.

Analogy, Similarity, and Features

The central analogy in such a visual problem may then be imagined as requiring that T be analogous to T′; that is, the answer will be whichever image D yields the most analogous transformation. That T and T' are analogous may be construed as meaning that T is in some fashion similar to T'. The nature of this similarity may be determined by a number of means, many of which associate visual or geometric features to points in a coordinate space, and compute similarity as a distance metric (Tversky 1977). Tversky developed an alternate approach by considering objects as collections of features, and similarity as a feature-matching process. We adopt Tversky's interpretation, and 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:

> *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.*

Although in practice one may begin with a particular source image and a particular destination image and derive the encoding from them, it is important to keep in mind that once the finite set of transformations T has been discovered, it may be applied to any source image, and will converge onto the particular destination image. The collage theorem defines fractal encoding as an iterated function system (Barnsley & Hurd, 1992).

The Fractal Encoding Algorithm

Given an 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).

Partition D into a set of smaller images, such that D = {d_1, d_2, d_3, … }.

For each image d_i:

- Examine the entire source image S for an equivalent image fragment s_i such that an affine transformation of s_i will likely result in d_i.
- Collect all such transforms into a set of candidates C.
- Select from the set C that transform which most minimally achieves its work, according to some predetermined metric.
- Let T_i be the representation of the chosen transformation associated with d_i.

The set T = {T_1, T_2, T_3, … } is the fractal encoding of the image D.

Algorithm 1. Fractal Encoding

The steps for encoding an image D in terms of another image S are shown in Algorithm 1. The partitioning 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. Alternatives could include irregular subdivisions, partitioning according to some inherent colorimetric basis, or levels of detail.

We note that the fractal encoding of the transformation from a particular source image S into a destination image D is tightly coupled with the partitioning P of the destination image D. Thus, a stronger specification of the fractal encoding T may be thought of as a function:

$$T(S, D, P) = \{T_1, T_2, T_3, \dots \}$$

where the cardinality of the resulting set is determined solely by the partitioning P.

Searching and Encoding

As mentioned, a chosen partitioning scheme P extracts a set of smaller images d_i from the destination image D. In turn, the entire source image S is examined for a fragment which most closely matches that fragment d_i. The matching is performed by first transforming di, as described below, and then comparing photometric (or pixel) values. A simple Euclidean distance metric is used for the photometric similarity between the two fragments.

The search over 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. By allowing for such reuse, the encoding algorithm captures the important notion of repetition, one of the two key observations driving fractal encoding.

Similitude Transformations

For each potential correspondence, the transformation of d_i via 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 eight transformations, specifically dihedral group D4, the symmetry group of a square.

We fix our dilation at a value of either 1.0 or 0.5, depending upon whether the source and target image are dissimilar or identical, respectively. The translation is found as a consequence of the search algorithm.

Fractal Codes

Once a transformation has been chosen for a fragment of the destination image, we construct a compact representation of that transformation called a fractal code.

A fractal code T_i is a six-tuple:

$$Ti = < <o_x,o_y>,<d_x,d_y>, k, s, c, Op >$$

with each member of the tuple defined as follows:

$<o_x,o_y>$ is the origin of the source fragment;
$<d_x,d_y>$ is the origin of the destination fragment d_i;
k $\in$ { I,HF,VF,R90,R180,R270, RXY, RNXY }, one of the eight transformations;
s is the block sized used in the given partitioning;
c $\in$ [-255, 255] indicates the overall color shift to be used uniformly to all elements in the block; and
Op is the pixel-level operation to be used when combining the color shift c onto pixels in the destination fragment d_i.

A fractal code thus collects the spatial and photometric properties necessary to transform a portion of a source image into a destination.

Arbitrary selection of source

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. This contraction, enforced by setting the dilation of spatial transformations at 0.5, provides the second of the key fractal observations, that similarity and repetition occur at differing scales.

Arbitrary nature of the encoding

The cardinality of the resulting set of fractal codes which constitute the fractal encoding is determined solely by the partitioning of the destination image. However, the cardinality of that set is arbitrary. The partitioning may be traversed in any order during the matching step of the encoding algorithm. Similarly, once discovered, the individual codes may be applied in any order at all, so long as all of the codes are applied in any particular iteration, to satisfy the constraint of the Collage Theorem.

The fractal encoding algorithm, while computationally expensive in its exhaustive search, transforms any 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 an unordered set of specific 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 the individual codes themselves.

Each fractal code yields a small set of features, formed by constructing subsets of the underlying six-tuple. These features are determined in a fashion to encourage both position-, affine-, and colorimetric-agnosticism, as well as specificity. Our algorithm creates features from fractal codes by constructing almost all possible subsets of each of the six members of the fractal code's tuple (we ignore singleton sets as well as taking the entire tuple as a set). Thus, in the present implementation of our algorithm, we generate C(6,2)+C(6,3)+C(6,4)+C(6,5) = 106 distinct features for each fractal code, where C(n, m) refers to the combination formula ("from n objects, choose m").

Mutuality

The analogical relationship between source and destination images may be seen as mutual. That is, the source is to the destination as the destination is to the source. However, the fractal representation which entails encoding is decidedly one-way (e.g. from the source to the destination). To capture the bidirectional, mutual nature of the analogy between source and destination, we now introduce the notion of a mutual fractal representation.

Let us call the representation of the fractal transformation from image A to image B as T_{AB}, as shown in Figure 2. Correspondingly, we would label the inverse representation as T_{BA}.

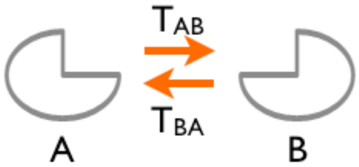


Figure 2. Mutual relationships.

We shall define the mutual analogical relationship between A and B by the symbol M_{AB}, given by this equation:

$$M_{AB} = T_{AB} \cup T_{BA}$$

By exploiting the set-theoretic nature of T_{AB} and T_{BA} to express M_{AB} as a union, we afford the mutual analogical representation the complete expressivity and utility of the fractal representation.

Thus, in a mutual fractal representation, we have the necessary apparatus for reasoning analogically about the relationships between images, in a manner which is dependent upon only features which describe the mutual visual similarity present in those images.

The Odd One Out Problems

The Odd One Out test of intelligence (Hampshire 2010) consists of 3x3 matrix reasoning problems organized in 20 levels of difficulty. In the test, a participant must decide which of the nine abstract figures in the matrix does not belong (the so-called "Odd One Out"). Figure 3 shows a sampling of the problems, illustrating the nature of the task, and several levels of complexity.

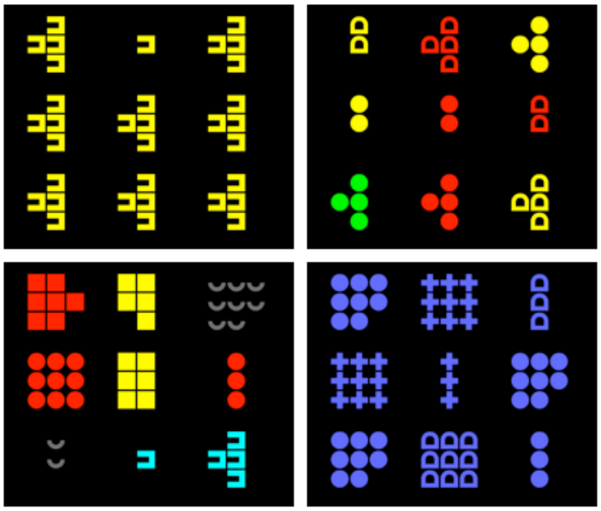


Figure 3. Odd One Out problems.

From the computational perspective, one drawback of computationally modeling a visual analogy task such as the Raven's test is that the algorithm for generating the problems on the test is not known; human examiners generate the test problems based on historical and empirical data. In contrast, problems on the Odd One Out test are generated using a complex set of algorithms (Hampshire 2010). Thus, many tens of thousands of novel problems may be generated.

Finding the Odd One Out, Fractally

We now present our algorithm for tackling the Odd One Out problem, using the mutual fractal representation as a basis for visual reasoning. The algorithm consists of three phases: segmentation, representation, and reasoning.

The segmentation phase.

First, we must segment the problem image P into its nine constituent subimages, I_1 through I_9. In the present implementation, the problems are given as a 478x405 pixel JPEG image, in the RGB color space. The subimages are arrayed in a 3x3 grid within the problem image. At this resolution, we have found that each subimage fits well within a 96x96 pixel image, as may be seen in Figure 4.

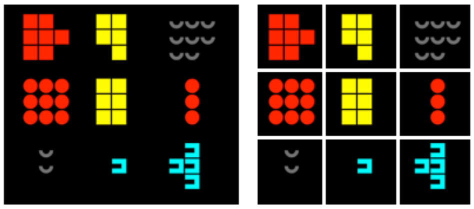


Figure 4. Segmentation of an Odd One Out problem

The representation phase.

We next must transform the problem into the domain of fractal representations. Given the nine subimages, we group subimages into pairs, such that each subimage is paired once with the other eight subimages. Thus, we form 36 distinct pairings. We then calculate the mutual fractal representation M_{ij} for each pair of subimages I_i and I_j. We determine the fractal transformation from I_i to I_j in the manner described in (McGreggor et al. 2011), then form the union of the sets of codes from the forward and backward fractal transformation to construct M_{ij}.

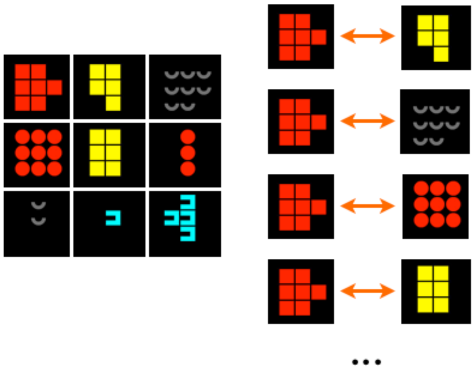


Figure 5. Pairing Subimages

The block partitioning we use initially is identical to the largest possible block size (in this case, 96x96), but subsequent recalculation of M_{ij} may be necessary using finer block partitioning (as proscribed in the reasoning phase). In the present implementation, we conduct the finer partitioning by uniform subdivision of the images into block sizes of 48x48, 24x24, 12x12, 6x6, and 3x3.

Extended Mutuality

At this phase, we note that the mutual fractal representation of the pairings may be employed to determine similar mutual representations of triplets or quadruplets of images. These subsequent representations may be required by the reasoning phase. As a notational convention, we construct these additional representations for triplets (M_{ijk}) and quadruplets (M_{ijkl}) in this manner:

$$M_{ijk} = M_{ij} \cup M_{jk} \cup M_{ik}$$

$$M_{ijkl} = M_{ijk} \cup M_{ikl} \cup M_{jkl} \cup M_{ijl}$$

Visually, we may interpret these extended mutual representations as shown in Figure 6.

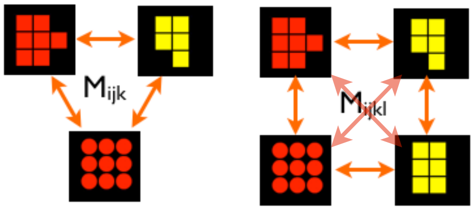


Figure 6. Mutuality in Pairs, Triplets, and Quadruplets

The reasoning phase.

We shall determine the odd one out solely from the mutual fractal representations, without reference or consideration to the original imagery. We start by considering groupings of representations, beginning with pairings, and, if necessary, advance to consider other groupings.

Reconciling Multiple Analogical Relationships

For a chosen set of groupings, G, we must determine how similar each member is to each of its fellow members. We first derive the features present in each member, as described above, and then calculate a measure of similarity as a comparison of the number of fractal features shared between each pair member (Tversky 1977).

We desire a metric of similarity which is normalized with respect to the number of features under consideration, and where the value 0.0 means entirely dissimilar and the value 1.0 means entirely similar. Accordingly, in our present implementation, we use the ratio model of similarity as described in (Tversky 1977). According to the ratio model, the measure of similarity S between two representations A and B is calculated thusly:

$$S(A,B) = f(A \cap B) / [f(A \cap B) + \alpha f(A-B) + \beta f(B-A)]$$

where $f(X)$ is the number of features in the set X. Tversky notes that the ratio model for matching features generalizes several set-theoretical models of similarity proposed in the psychology literature, depending upon which values one chooses for the weights α and β. To favor features from either image equally, we have chosen to set $\alpha = \beta = 1$.

Relationship Space

As we perform this calculation for each pair A and B taken from the grouping G, we determine for each member of G a set of similarity values. We consider the similarity of each analogical relationship as a value upon an axis in a large "relationship space" whose dimensionality is determined by the size of the grouping: for pairings, the space is 36 dimensional; for triplets, the space is 84 dimensional; for quadruplets, the space is 126 dimensional.

Treating Maximal Similarity as Distance

To arrive at a scalar similarity score for each member of the group G, we construct a vector in this multidimensional relationship space and determine its length, using a Euclidean distance formula. The longer the vector, the more similar two members are; the shorter the vector, the more dissimilar two members are. As the Odd One Out problem seeks to determine, literally, "the odd one out," we seek to find the shortest vector, as an indicator of dissimilarity.

Distribution of Similarity

We have determined a score for the grouping G, but have not yet arrived at individual scores for the subimages. To determine the subimage scoring, we distribute the similarity equally among the participating subimages. For each of the nine subimages, a score is generated which is proportional to its participation in the similarity of the grouping's similarity vectors. If a subimage is one of the two images in a pairing, as an example, then the subimage's similarity score receives one half of the pairing's calculated similarity score.

Once all of the similarity scores of the grouping have been distributed to the subimages, the similarity score for each subimage is known. It is then a trivial matter to identify which one among the subimages has the lowest similarity score. As it turns out, this is not yet sufficient for solving the problem, as ambiguity may be present.

Ambiguity

Similarity scores for the subimages may vary widely. If the score for any subimage is unambiguously smaller than that of any other subimage, then the subimage is deemed "the odd one out." By unambiguous, we mean that there is no more than one score which is less than **ε**, which we may vary as a tuning mechanism for the algorithm, and which we see as a useful yet coarse approximation of the boundary between the similar and the dissimilar in feature space. In practice, we calculate the deviation of each similarity measure from the average of all such measures, and use confidence intervals (as calculated from the standard deviation) as a means for indicating ambiguity.

Refinement strategy

However, if the scoring is inconclusive, then there are two readily available mechanisms at the algorithm's disposal: to modify the grouping such that larger sets of subimages are considered simultaneously (from pairs to triplets, or from triplets to quadruplets), or to recalculate the fractal representations using a finer partitioning. In our present implementation, we attempt bumping up the elements considered simultaneously as a first measure. If after reaching a grouping based upon quadruplets the scoring remains inconclusive, then we consider that the initial representation level was too coarse, and rerun the algorithm using ever finer partitions for the mutual fractal representation. If, after altering our considerations of groupings and examining the images at the finest level of resolution the scores prove inconclusive, the algorithm quits, leaving the answer unknown.

Example

We now present an example of the algorithm, selected for its illustrative power, and not for its difficulty. We do note that the answer to this particular problem may be at once obvious to the reader.

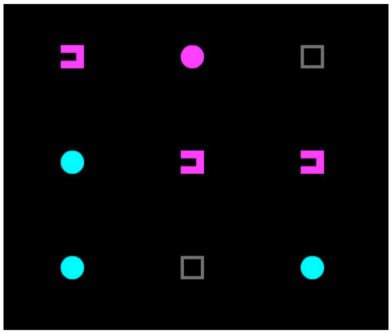


Figure 7. An example problem from the Odd One Out test.

The algorithm begins by segmenting the image into the nine subimages. For convenience, let us label the images A through I, as shown in Figure 8. Once segmented, fractal representations are formed for each possible pairing of the subimages, for a total of 36 distinct representations. The initial partitioning of the subimages for fractal encoding shall be at the coarsest possible level, 96x96 pixels.

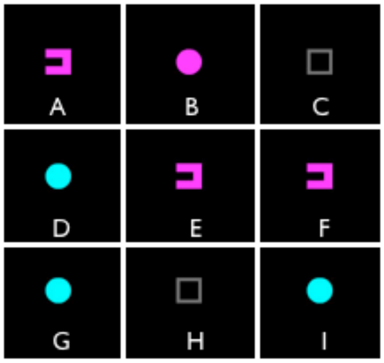


Figure 8. The example, segmented and labelled

In this example, it is quite clear to the reader that there are pairings which are identical (e.g. {A,E}, {E, F}, {A, F}, {C, H}, {D, G}, {D, I}, and {G, I}). The fractal representation of each of these pairings, at this coarsest level of partitioning (96x96) will yield the Identity transformation, with zero photometric correction. Thus the similarity between these particular transform pairs will be 1.0. These pairings we shall deem therefore to be perfectly analogous. However, not all of the representations will be similar. For example, the pairing of subimage C to any image other than subimage H will result in a substantially different fractal encoding than the {C,H} pairing.

For each subimage, we calculate the similarities of all eight possible pairings of that subimage against all other unique pairings. We next construct a similarity vector in 36-space for each pairing.

We derive the length of the 36-tuple similarity vector, normalize the result, and distribute this value to each of the subimages involved in the pairing by summing. In this example, the length of the similarity vector for the pairing {A,B} is found to be 4.55, we divide by 6 (the length of a 36-tuple with all entries 1.0), for a value of 0.7583. This value is added to the current summation for subimages A and B. At the close of this process, each subimage will have a score, representing the distributed similarity scores for all of the pairings in which it played a part.

The algorithm then examines the set of scores for all of the subimage, looking for ambiguity. Our present implementation defines ambiguity as the data having more than one item which deviates from the mean by a value greater than the standard deviation of the data. If this holds true, then the result is deemed ambiguous.

Table 1 illustrates the ambiguity found in using a 96x96 partitioning of the subimages, with two values having a deviation of 1.713σ. Thus, the algorithm must proceed to a finer partitioning in order to produce an unambiguous answer. Using a 48x48 partitioning, produces a single unambiguous result, with a deviation of 2.117σ. Accordingly, subimage B is selected as the Odd One Out.

96x96		
0.727	0.727	0.674
0.716	0.727	0.727
0.716	0.674	0.716
μ : 0.711, σ : 0.022		

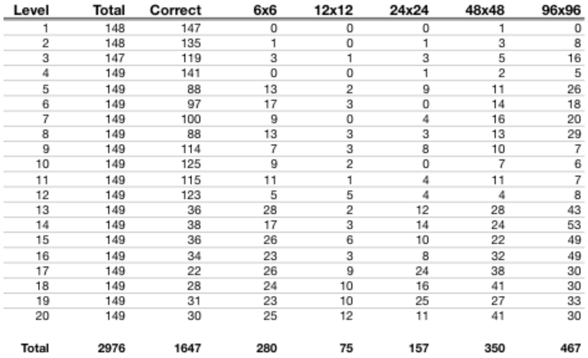
48x48		
0.534	0.413	0.468
0.510	0.534	0.534
0.510	0.468	0.510
μ : 0.499, σ : 0.041		

Table 1. Similarity scores for 96x96 and 48x48 partitions.

Results, Preliminary Analysis and Discussion

We have run our algorithm against 2,976 problems of the Odd One Out. These problems were randomly selected from a span of difficulty from the very easiest (level one) up to the most difficult (level 20). The example problem presented in Figure 7 is a level 11 problem.

We restricted the algorithm to attend only to pairings of subimages, and to progress from an initial partitioning of 96x96 blocks (essentially, the entire subimage) to no further refinement of partitioning than 6x6. We made these restrictions in order to fully exercise the strategic shifting in partitioning, to assess the similarity calculations, and to judge the effect of mutuality, at a tradeoff in execution time. The results are presented in Table 2.



Level	Total	Correct	6x6	12x12	24x24	48x48	96x96
1	148	147	0	0	0	1	0
2	148	135	1	0	1	3	8
3	147	119	3	1	3	5	16
4	149	141	0	0	1	2	5
5	149	88	13	2	9	11	26
6	149	97	17	3	0	14	18
7	149	100	9	0	4	16	20
8	149	88	13	3	3	13	29
9	149	114	7	3	8	10	7
10	149	125	9	2	0	7	6
11	149	115	11	1	4	11	7
12	149	123	5	5	4	4	8
13	149	36	28	2	12	28	43
14	149	38	17	3	14	24	53
15	149	36	26	6	10	22	49
16	149	34	23	3	8	32	49
17	149	22	26	9	24	38	30
18	149	28	24	10	16	41	30
19	149	31	23	10	25	27	33
20	149	30	25	12	11	41	30
Total	**2976**	**1647**	**280**	**75**	**157**	**350**	**467**

Table 2. Scores for pairings of the OddOneOut.

We note that there are quite clear degrees of performance variation generally grouped according to sets of levels (levels 1-4, 5-8, 9-12, 13-16, and 17-20). This is consistent with the (unknown both to us and to the algorithm) knowledge that the problems at these levels were generated using varying rules. Our algorithm at present does not carry forward information between its execution of each problem, let alone between levels of problems. However, that the output illustrates such a strong degree of performance shift provides a further research opportunity in the areas of reflection, abstraction and meta-reasoning, in the context of the original fractal representations.

The rightmost five columns of the results data provide a breakdown of errors made at differing partitioning levels. Immediately the reader will note that the majority of errors occur when the algorithm stops at quite high levels of partitioning (96x96 or 48x48). We interpret this as strong evidence that there exists levels-of-detail (or partitioning) which are too gross to allow for certainty in reasoning. Indeed, the amount of data upon which decisions are made at these levels are three orders of magnitude less than that which the finest partitioning affords (roughly 100 features at 96x96 versus more than 107,000 features at 6x6). We find an opportunity for a refinement of the algorithm to assess its certainty (and therefore, its halting) based upon a naturally emergent artifact of the representation.

A temptation might be to reverse the partitioning process, beginning at the finest partition (6x6) and progress upward until ambiguity appears due to insufficient level of detail. In an earlier test of this notion, using a random sampling of problems across a span of difficulty levels, we found that ambiguity existed at both small and large levels of detail; that is, that ambiguity exists at either too fine or too large a level of detail, and that an unambiguous answer arose once some sufficiency in level of detail was realized. It is important to note that the sufficient level of detail was discoverable by the algorithm, emerging from the features derived from the fractal representation.

The errors which occurred at the finest level of partitioning (6x6) are caused not due to the algorithm reaching an incorrect unambiguous answer (though this is so in a few cases) but rather that the algorithm was unable to reach a sufficiently convincing or unambiguous answer. As we noted, these results are based upon calculations involving considering shifts in partitioning only, using pairwise comparisons of subimages. Thus, there appear to be Odd One Out problems for which considering pairs of subimages shall prove inconclusive (that is, at all available levels of detail, the results will be found to be ambiguous). It is this set of problems which we believe implies that a shift in grouping (from pairs to triplets, or from triplets to quadruplets) must be undertaken to reach an unambiguous answer.

Conclusion

We have described a new strategy employing fractal representations for addressing visual analogy problems on the Odd One Out test of intelligence. This strategy uses a precise characterization of ambiguity which emerges from similarity measures derived from fractal features. When the fractal representation at a given scale results in an ambiguous answer to an Odd One Out problem, our algorithm automatically shifts to a finer level of resolution, and continues this refinement step until it reaches an unambiguous answer. If the answer remains ambiguous through all of the possible levels of resolution, then our algorithm shifts toward considering groups of triplets, and then quadruplets, working through each grouping from coarsest to finest resolution as necessary.

Our analysis, while intriguing, is preliminary, and our observations are based upon results of running the algorithm against a large (2,976) corpus of Odd One Out problems.

Fractal representations are imagistic (or analogical) in that they have a structural correspondence with the images they represent. Like other representations, fractal representation support inference and composition. As we mentioned in the introduction, we have earlier used a different fractal technique to address a subset of problems on Raven's Standard Progressive Matrices Test (McGreggor et al. 2011; Kunda et al. 2010). That work, and the algorithm and results presented here, suggests a degree of generality to fractal representations for addressing visual analogy problems.

The fractal representation captures detail at multiple scales. In doing so, it sanctions an iterative problem solving strategy. The twin advantages of the strategy are (1) to start at high level of resolution, but, if that is not sufficient to resolve ambiguity, to automatically adjust to the right level of resolution for addressing a given problem, and (2) to start searching for similarity between simpler relationships, but, if that is not sufficient to resolve ambiguity, to automatically search for similarity between progressively higher-order relationships. Cognitively, this is an illustration of what Davis (et al. 1993) referred to as the deep, theoretic manner in which representation and reasoning are intertwined. This powerful strategy is made possible by using fractal representations of visual images.

References

Barnsley, M., & Hurd, L. 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. A., Just, M. A., & Shell, P. What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices Test. Psychological Review 97 (3): 404-431. (1990).

Davis, R. Shrobe H., and Szolovits, P. "What is a Knowledge Representation?" AI Magazine 14.1 (1993):17-33.

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.

Hampshire, A. 2010 The Odd One Out Test of Intelligence. <http://www.cambridgebrainsciences.com/browse/reasoning/test/oddoneout>

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

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

Kunda, M., McGreggor, K. & Goel, A. Taking a Look (Literally!) at the Raven's Intelligence Test: Two Visual Solution Strategies. In Proc. 32nd Annual Meeting of the Cogntive Science Society, Portland, August 2010.

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. 2011. Fractal Analogies: Preliminary Results from the Raven's Test of Intelligence. In Proc. International Conference on Computational Creativity, Mexico City, Mexico, April 27-29.

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.

Wechsler, D. (1939). Wechsler-Bellevue intelligence scale. New York: The Psychological Corporation.