Artificial Intelligence 215 (2014) 1–23

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Fractals and Ravens

Keith McGregor*, Maithilee Kunda, Ashok Goel

Design & Intelligence Laboratory, School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA

ARTICLE INFO

Article history:
Received 4 June 2013
Received in revised form 14 April 2014
Accepted 13 May 2014
Available online 20 May 2014

Keywords:
Analogy
Visual reasoning
Fractal representations
Computational psychometrics

ABSTRACT

We report a novel approach to visual analogical reasoning, one afforded expressly by fractal representations. We first describe the nature of visual analogies and fractal representations. Next, we exhibit the Fractal Ravens algorithm through a detailed example, describe its performance on all major variants of the Raven's Progressive Matrices tests, and discuss the implications and next steps. In addition, we illustrate the importance of considering the confidence of the answers, and show how ambiguity may be used as a guide for the automatic adjustment of the problem representation. To our knowledge, this is the first published account of a computational model's attempt at the entire Raven's test suite.

1. Introduction

Despite references to its role as core to cognition [33,35], analogy defies a singular definition [58]. In one way, analogy may be seen as a process of transference, a mapping of knowledge from one situation to another, based upon a judgment of the similarity between the two situations [12,19,20,22–27,37,40,57,67]. Gentner [24], for example, proposed that analogies entail transfer of relations from a source case to a target problem and that judgment of similarity between the target and the source depends on the correspondence between the structure of their representations. Holyoak and Thagard [37] proposed that judgments of similarity between a source case and a target problem depend on multiple criteria: structural correspondence, semantic similarity, and pragmatic constraints. Keane [40] proposed incremental mapping between the source case and the target problem. Dunbar [19] found a paradox in that humans appear to exhibit significant spontaneous use of analogies in their natural workflow but less so in laboratory settings. Kokinov and Petrov [42] describe several constraints to facilitate the integration of analogue retrieval from memory and analogical transfer. Holyoak and Hummel [36] similarly examine findings for recruiting memory, reasoning and learning in the service of analogy making. Clement [12] describes several processes of analogy such as the use of intermediate representations (or bridging analogies). Nersessian [57] similarly describes the use of generic mechanisms in scientific analogies. Our own work on *model-based analogy* [6,28] has focused on the use of semantic similarity and pragmatics constraints for evaluating similarity between source cases and target problems, and identification and abstraction of generic mechanisms for transfer from a source case to a target problem in creative design [6,28,29].

In contrast, case-based reasoning [1,43,49,63] views within-domain analogical reasoning as a memory task in which memory supplies a source case containing an almost correct solution to the target problem. Hammond [32], for example, describes retrieval of plans based on semantic similarity to the target problem and modification of the retrieved plan to meet the target goal. Ashley and Rissland [3] describe the use of case-based reasoning in law. Smyth, Cunningham and Keane [65] describe hierarchical case-based reasoning, and Aha, Breslow and Munoz-Avila [2] describe conversational case-based

[*] Corresponding author.
E-mail addresses: keith.mcgregor@gatech.edu (K. McGregor), mkunda@gatech.edu (M. Kunda), goel@cc.gatech.edu (A. Goel).

http://dx.doi.org/10.1016/j.artint.2014.05.005
0004-3702/© 2014 Elsevier B.V. All rights reserved.

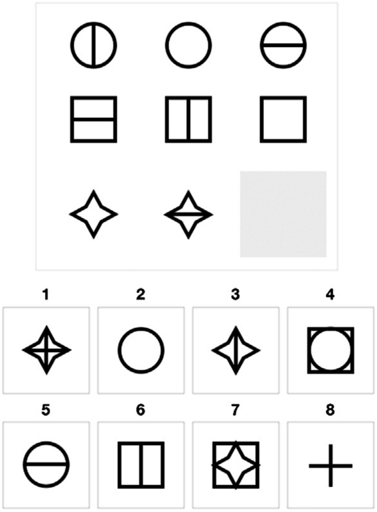


Fig. 1. Problem similar to those of the Raven's Standard Progressive Matrices test.

reasoning. Our own work on case-based reasoning has focused in integration of case-based and model-based reasoning for adaptive design [30].

Another line of research views analogy as a mechanism of perception, where one situation is recognized in terms of another [10,34,56] or as a mechanism of learning, where one situation is interpreted in terms of another [39]. Yet another line of research on analogy pertains to visual analogy [13–17,69,70]. In visual analogy, the source case and the target problem contain only modal, visual knowledge, and causality is (at most) implicit. For example, Yaner and Goel [69] describe a technique for retrieving design drawings from memory that are similar to a target drawing; Davies, Goel and Yaner [16] describe the technique of *constructive analogy* for incrementally transferred from the source drawing to the target drawing; and Yaner and Goel [70] describe the technique of *compositional modeling* that builds a causal model of the target drawing by analogy to the causal model of the source drawing.

Each of these schools of thought emphasizes the importance of certain aspects of analogy making, and, in turn, establishes certain criteria that must be achieved by any associated methodologies through some combination of mechanism and representation. Our work concerns visual analogy, the act of forming analogies based upon purely visual perceptions, or, more formally, upon a purely visual perceptual history. We propose a new representation – the fractal representation – and corresponding mechanism for addressing a class of visual analogies that occur in computational psychometrics, and in particular, on the Raven's Progressive Matrices test of intelligence. Although we focus our remarks on representation, visual analogy, and psychometrics, we expressly make no claims as to whether our model may be extended to provide for a cognitive account.

1.1. Computational psychometrics

AI research on computational psychometrics dates at least as far back as Evans' Analogy program [21], which addressed geometric analogy problems on the Miller Geometric Analogies test of intelligence. Bringsjord and Schimanski [7] have proposed computational psychometrics, i.e., AI that can pass psychometric tests of intelligence, as a possible mechanism for measuring and comparing AI.

Raven's Progressive Matrices Test suite is a set of standard and common tests of intelligence [61]. The standard version of the test consists of 60 geometric analogy problems. Fig. 1 illustrates a problem typical to those that appear on the test.[1] The task in the problem is to pick one of the eight choices in the bottom of the figure for insertion in that bottom-right element of the 3 × 3 matrix in the top of the figure. The chosen element should best match the patterns in the rows and columns of the matrix.

The Raven's Progressive Matrices (RPM) test paradigm is intended to measure eductive ability, the ability to extract and process information from a novel situation [61]. Eductive ability stands in contrast to reproductive ability, which is the ability to recall and use previously learned information.

The problems from Raven's various tests are organized into five sets. Each successive set is generally interpreted to be more difficult than the prior set. Some of the problem sets are 2 × 2 matrices of images with six possible answers; the remaining sets are 3 × 3 matrices of images with eight possible answers.

[1] Throughout this paper, we use example problems that are similar to those found on Raven's tests, due to copyright concerns and to ensure the integrity of the tests themselves. The results we report, however, are from the actual test problems.

The tests are purely visual: no verbal information accompanies the tests. The test-taker is asked to select from the available possible answers the single answer that best completes the matrix [61].

1.2. Prior approaches to RPM

Over the years, different models have proposed various combinations of representations and mechanisms for solving RPM problems. Hunt [38] gives a theoretical account of the information processing demands of certain problems from the Advanced Progressive Matrices (APM), in which he proposes two qualitatively different solution algorithms—"Gestalt," which uses visual operations on analogical representations, and "Analytic," which uses logical operations on conceptual representations.

Carpenter, Just, and Shell [9] describe a computational model that simulates solving RPM problems using propositional representations. Their model is based on the traditional production system architecture, with a long-term memory containing a set of hand-authored productions and a working memory containing the current state of problem solving (e.g. current goals). Productions are based on the relations among the entities in an RPM problem, for example, the location of the dark component in a row, which might be the top half in the top row of a problem, bottom-half in the bottom row, and so on. They did not test their system on the Standard Progressive Matrices (SPM), but two different versions of their system solved 23 and 32 out of 34 attempted problems on the APM.

Bringsjord and Schimanski [7] used a theorem-prover to solve selected RPM problems stated in first-order logic, though no results from this effort were reported.

Lovett, Forbus and Usher [51] describe a model that extracts qualitative spatial representations from visually segmented representations of RPM problem inputs and then uses the analogy technique of structure mapping to find solutions and, where needed to achieve better analogies, to regroup or re-segment the initial inputs to form new problem representations. Again, while visual information from the RPM problems is implicit in the final representations, the structure-mapping engine is applied to these representations without any commitment to the visual nature of the encoded information. This system was tested against sets B through E of the SPM and correctly solved 44 out of 48 attempted problems.

Cirillo and Ström [11] created a system for solving problems from the SPM that, like that of Lovett et al. [51], takes as inputs vector graphics representations of test problems and automatically extracts hierarchical propositional problem representations. Then, like the work of Carpenter et al. [9], the system draws from a set of predefined patterns, derived by the authors, to find the best-fit pattern for a given problem. This system was tested against Sets C through E of the SPM and solved 8, 10, and 10 problems, respectively.

Schwering et al. [64] describe a gestalt method for addressing RPM problems. Rasmussen and Eliasmith [60] used a spiking neuron model to induce rules for solving RPM problems. Ragni, Stahl and Fangmeier [59] analyze the cognitive complexity of RPM problems. However, none of these report results from the addressing problems on any set of RPM problems.

Our work on the RPM was initially motivated by trying to understand cognition in autism [45]: some individuals on the autism spectrum show RPM scores much higher than what would be predicted by their performance on other standardized IQ tests [18]. A recent neuroimaging study compared task-related brain activity between neurotypical individuals and individuals diagnosed with autism while solving RPM problems and found that, while both groups showed brain activation across a distributed network of regions, the individuals diagnosed with autism showed relatively greater activation in occipital regions, which are associated with visual processing, and relatively lower activation in prefrontal regions, associated with working memory and decision-making [66]. We analyzed behavioral and neuroimaging data across a range of cognitive tasks, including the RPM, and found that many empirical results cited in favor of other cognitive accounts of autism could be explained equally well by the hypothesis that certain individuals with autism are disposed towards thinking visually [45]. Thus, we hypothesized that it could be both feasible and useful to develop a computational model of RPM that used solely visual representations. Our first model, called the "affine" or "ASTI" model [46–48] operated directly on scanned image inputs from the test and used affine and set operations based on mental imagery (rotations, translations, image composition, etc.) to induce image transformations between images in the problem matrix and then predict an answer image based on the final induced transformation. We tested the first ASTI model on all 60 problems from the SPM and found that it correctly solves 38 of the problems.

1.3. Outline of the argument in this paper

We divide our argument for the use of fractal representations and reasoning for visual analogy into three parts. In the first part, we describe the nature of visual analogies and develop fractal representations. Next, we exhibit a computational model for fractal analogical reasoning and illustrate its use through a detailed example. Lastly, we describe the performance of our model on all major variants of the Raven's Progressive Matrices tests – SPM, APM, CPM, and SPM Plus – and discuss the implications and next steps. We show that the performance of the fractal computational model matches or exceeds that of any published account. We note that all the source code and images used in these examples are readily available on our research group's website.

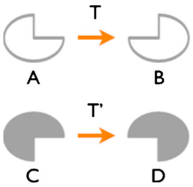


Fig. 2. An example of visual analogy.

2. Visual analogies and representations

Suppose we have a visual analogy, expressed symbolically as $A : B :: C : D$, with the symbols representing images, as shown in Fig. 2. We can interpret this as suggesting that some operation T exists which captures the relationship between image A and image B ("A is to B"). Likewise, some other operation T' is proposed which captures the relationship between image C and image D ("C is to D").

In this manner, we may see that the analogy in such a problem rests not with the images themselves, but in the degree to which the two operations T and T' are analogous. We can express the problem to make plain this distinction thus:

$$A : B :: C : D \longrightarrow T(A, B) :: T'(C, D)$$

2.1. Similarity between operations

The nature of this similarity may be determined by a number of methods, many of which might associate visual or geometric features to points in a coordinate space, and compute similarity as a distance metric. Tversky developed an alternate approach by considering objects as collections of features, and similarity as a feature-matching process [68].

We adopt Tversky's interpretation of similarity, and thus seek to express these operations T and T' in some representation which both is robust and affords sufficient feature production to permit feature matching [4]. A particular nuance of Tversky's approach, however, is that either the representation or the features derived from the representation must be formable into sets, as the calculation for similarity employed requires the counting of elements within sets (and their union and intersection).

Thus, we may revisit the typical visual analogy $A : B :: C : D$, where T and T' are now representations which meet Tversky's featural requirement. To make a comparison between the two representations, we first derive features from each, and then calculate a measure of similarity based upon those features.

2.2. Similarity metric

We desire a metric of similarity that is normalized so that the value 0.0 means entirely dissimilar and the value 1.0 means identical. Accordingly, we use the ratio model of similarity as described by Tversky [68], wherein the measure of similarity between the two representations T and T' is calculated thus:

$$S(T, T') = \mathcal{F}(T \cap T') / \left[\mathcal{F}(T \cap T') + \alpha \mathcal{F}(T - T') + \beta \mathcal{F}(T' - T)\right]$$

where the operator $\mathcal{F}(Y)$ derives the number of features in some set Y. The particular sets involved may be considered as indicating, respectively, those features the two representations share ($T \cap T'$), those features in T but not in T' ($T - T'$), and those features in T' but not in T ($T' - T$).

Tversky [68] notes that the ratio model for matching features generalizes several set-theoretical models of similarity proposed in the psychology literature (e.g. [8] and [31]), depending upon which values one chooses for the weights α and β. Later in this discussion, we shall revisit these weights, and illustrate the significance of their choice.

3. Fractals and the fractal representation

Benoit Mandelbrot coined the term "fractal" from the Latin adjective *fractus* and its corresponding verb (*frangere*, "to break" into irregular fragments), in response to his observation that shapes previously referred to as "grainy, hydralike, in between, pimply, pocky, ramified, seaweedy, strange, tangled, tortuous, wiggly, wispy, wrinkled, and the like" could be described by a set of compact, rigorous rules for their production [52]. Indeed, the computer graphics community has generated fractal imagery for several decades.

While the techniques for generating artificial fractal imagery are quite well known, as Mandelbrot noted many images of real-world artifacts appear to have "fractal" properties. If these images are "fractal" in some sense, then what formula may underlie these images?

3.1. The mathematical basis for fractals as operations

As we have noted previously [55], the mathematical derivation of fractal representation as an operation over images expressly depends upon the notion of real world images, i.e. images that are two dimensional and continuous [5]. Every image received by the human visual system may be construed as meeting this requirement, with the proviso that the notion of continuity has a resolution limit, and that limit plays a significant role in visual abstraction, as shall be discussed later in this paper.

Of particular relevance are two observations about images themselves, made by Mandelbrot [52]. The first is that all naturally occurring images appear to have similar, repeating patterns. The second observation is that no matter how closely one examines the real world, one may find instances of similar structures and repeating patterns. These observations suggested to Barnsley and Hurd [5] that it is possible to describe images in terms that capture the observed similarity and repetition alone, without regard to shape or traditional graphical elements.

3.2. Collage theorem

Computationally, the determination of the fractal representation of an image can be performed through the use of the fractal encoding algorithm. The collage theorem [5] at the heart of the algorithm can be stated concisely:

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

It is important to note that the collage theorem is describing a set of transformations that are derived by mapping an image into another. In other words, fractal encoding determines an iterated function system that is applied repeated to some source image, with the result that the encoded image emerges.

Suppose $F()$ is the fractal encoding of image B. Then, given any other image A:

$$F(A) = A_1, \quad F(A_1) = A_2, \quad F(A_2) = A_3 \ldots \quad \text{and so on, until} \quad F(A_\infty) \doteq B$$

$F()$ is itself a finite set of affine transformations T that describe how to modify portions of an image such that convergence is assured.

$$F() \equiv T \equiv \{T_1, T_2, T_3, \ldots, T_n\}$$

Each affine transformation may affect some or all of the given image, but it is the unordered union of their actions that comprises the resultant image. Thus:

$$F(A) = T(A) = \bigcup T_i(A), \quad 1 \le i \le n$$

3.3. Dependencies

There are several, interrelated dependencies implied in the Collage theorem. These are specificity, partitioning, and search. We shall describe each in turn now, in the context of the theorem, and later, in the context of the algorithm and the subsequent representation.

3.3.1. Dependency upon the specificity of the source and the destination

A fractal encoding is dependent not only upon the destination image, but also upon the source image, from which the set of affine transformations T is discovered. However, once the fractal encoding has been determined, the application of that encoding to any source image will result in the target image. This dependency is to suggest that, a priori any application of the encoding, a particular fractal encoding is determined uniquely by a particular source image.

3.3.2. Dependency upon the partitioning of the destination image

The cardinality of set of transformations is determined exactly and solely by the partitioning scheme chosen for the image being encoded. It is presumed that the image being encoded admits to being partitioned in some manner, however.

Images may be partitioned using a variety of methods. In computer vision, one typically seeks to segment an image into regions or shapes [62,71]. Another segmentation scheme would seek to segregate an image into two segments, a foreground and a background [41]. Other partitionings of images may be regular, such as the division of computer-based images into pixels, at some resolution. It must be noted that the choice of partitioning scheme affects the computational complexity of enacting the partitioning.

The Collage theorem imposes no constraint upon the choice of partitioning save one, and that is that the union of all partitions wholly covers the image to be encoded. From a topological perspective, the image B is treated as a set, and the partitioning $P()$ of that image into a finite collection of subsets is a cover of that set if:

```
First, choose a partitioning scheme $P$ to systematically divide the destination image $B$ into a set of images, such that $B \subseteq \{b_1, b_2, b_3, \ldots b_n\}$.
For each image $b_i$:

    · Search the source image $A$ for an equivalent image fragment $a_i$ such that an affine transformation of $a_i$ will likely result in $b_i$.
    · Collect all such transforms into a set of candidates $C$.
    · Select from the set $C$ that transform which minimally achieves its work, according to some predetermined metric.
    · Let $T_i$ be the representation of the chosen transformation associated with $b_i$.

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

Algorithm 1. Fractal encoding of B in terms of A.

$$P(B) = \{b_1, b_2, b_3, \ldots, b_n\}$$

$$B \subseteq \bigcup b_i, \quad 1 \le i \le n$$

3.3.3. Dependency upon the search of the source image

The essential implied step of the Collage theorem is that there is a match for each subimage of the destination, as determined by the partitioning, to be sought within the source image. Through this searching process, the affine transformation for that subimage is obtained. However, the quality and character of the match, as well as the computational complexity of the algorithm, depends upon the constraints selected for comparing the destination subimage with some portion of the source.

3.4. Fractal encoding algorithm

Given a target image B and a source image A, the fractal encoding algorithm seeks to discover this particular set of transformations T.

As can be seen in Algorithm 1, the fractal encoding of image B in terms of image A consists of two phases: partitioning and searching. We shall now discuss each phase.

3.4.1. Partitioning

The target image B is first partitioned into a set of other images. As we are dealing with computer images, we may safely assume that the image B will have some finite resolution, and thus a limit as to the smallest achievable partition. Practically, this smallest resolvable unit is a unitary pixel, which would denote both a spatial location and a photometric value.

As we noted above, the Collage theorem places a topological constraint on the partitioning scheme, and requires that the scheme should act as a topological cover over the image B. Such a constraint admits a wide variety of methods for partitioning the image, but many of these partitionings may prove computationally expensive.

In the interest of reducing computational complexity, we may impose two additional constraints: each subimage in the partition must be simply connected to at least one other subimage, and the union of all of the subimages must be exactly equivalent to the image B. Stated formally:

$$P(B) = \{b_1, b_2, b_3, \ldots, b_n\} \text{ is a valid partitioning iff:}$$

$$\forall i \exists j \neq i : \text{simplyconnected}(b_i \cup b_j), \quad \text{and}$$

$$B \equiv \bigcup b_i, \quad 1 \le i \le n$$

where

$$\text{simplyconnected}(X) \to \nexists x, y \subset X : x \cap y = \emptyset$$

One computationally inexpensive way to achieve such a constrained partitioning is to impose upon the image B a uniform, rectilinear grid, and select the subimages based upon some chosen grid size, as expressed in units of pixels.

Thus, a stronger specification of the fractal encoding T may be thought of as a function of three variables, the source image A, the target image B, and the partitioning scheme P:

$$T(A, B, P) = \{T_1, T_2, T_3, \ldots, T_n\}$$

where the cardinality of the resulting set is determined solely by the partitioning P. That is, each subimage b_i that P extracts from B will be represented by exactly one element of the set T.

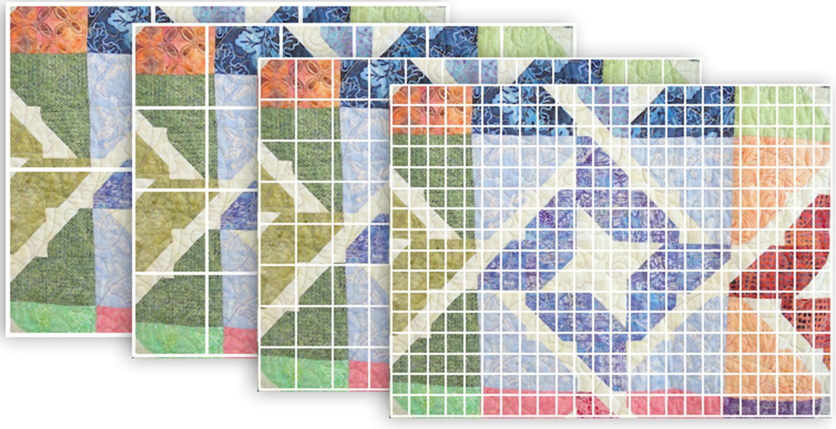


Fig. 3. Partitioning as levels of detail.

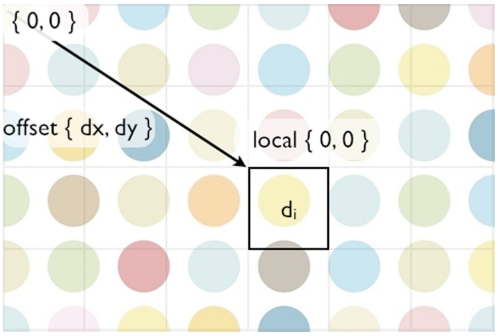


Fig. 4. Local and global coordinates.

3.4.2. *Partitioning and level of detail*

Choosing a partitioning determines the level of detail at which an image is encoded. Thus, the coarsest level of detail possible for an image is the partitioning into a single image (the whole image). The finest level of detail achievable is that set of images wherein each image is but a single pixel.

The choosing of a grid size, and of a partitioning in general, may be interpreted as an indication of the level of detail at which an image may be encoded. Fig. 3 illustrates the effect of partitioning an image into a variety of levels of detail, using a regular rectangular grid.

The ability to express level of detail as an artifact of partitioning, whether by controlling grid size, by altering the consistency of partition size, or by modification of the shape and nature of the underlying regions and their spatial arrangement (i.e. hexagonal versus rectilinear scaffolding, or polar versus Cartesian coordinates) is an important aspect of the encoding, and a key feature entailed by the fractal representation.

3.4.3. *Searching*

The partitioning scheme P extracts a set of images b_i from the target image B. The next step of the algorithm is to perform a systematic examination of the source image A for fragments of A that can be said to best match a particular image b_i. The method by which the search is conducted may be varied, as can the meaning of what is said to be a "best match."

3.4.3.1. Global and local coordinates An image b_i extracted by the partitioning scheme can be considered as a region containing a number of pixels that are addressable in some fashion. The addressability of these pixels may be viewed as a local coordinate system imposed upon the region. Additionally, the region described by the image b_i has a location and orientation within the image B, strictly determined by the partitioning scheme. Thus, we may consider the image b_i as an ordered set of pixels, having both a local (intrinsic) coordinate system and extent, and a position and orientation within a global (within image B) coordinate system. Fig. 4 illustrates the local and global coordinate systems.

However, the same partitioning scheme necessarily does not need to be applied to the source image. The entire source image A may be examined in any manner for a fragment that most closely matches b_i.

```
Let $C \leftarrow 0$.
For each pixel $x \in b_i$ and corresponding pixel $y \in a_k$:
    $C \leftarrow C + (\mathrm{Photometric}(x) - \mathrm{Photometric}(y))^2$
The value C is then the photometric correspondence between $a_k$ and $b_i$.
```

Algorithm 2. Photometric correspondence.

3.4.3.2. Discovering the "best match" The source image A is examined to determine which fragment of it, which we shall label a_k, can be said to "best match" the sought-for image b_i from the target image B. That is, the correspondence between a_k and b_i can be said to be "best" if it is the minimum value of the following function:

$$\mathrm{Correspondence}(a_k, b_i) = \mathrm{PhotometricCorrespondence}\big(\mathrm{Transform}(a_k, t), b_i\big)$$
$$\forall a_k \subset A, t \in \mathrm{AdmissibleTransformations}$$

where AdmissibleTransformations is a finite set of spatial transformations applied by the operator Transform() to the pixel values contained within a_k, and PhotometricCorrespondence() is a pixel comparison operation. We shall define both of these momentarily.

3.4.3.3. Photometric correspondence The photometric correspondence between the fragment a_k from the source image and b_i from the destination image is calculated to be the difference between the photometric values found in those fragments under a given alignment of their pixels. We wish to propose a metric to ensure that this difference would be 0 if the two fragments were identical photometrically. Such an algorithm to calculate the photometric correspondence is given by Algorithm 2.

The corresponding pixel in a_k is determined by imposing the same local coordinate system used in b_i upon a_k.

The Photometric value of a pixel used in this calculation may vary according to the nature of the image itself. For example, if the image is in full color, the photometric value may be a triplet of actual values; if the image is monochromatic, then the photometric value will be single valued. Since it is desired to calculate a photometric correspondence that is single-valued, a mapping from multivariate photometry to a single value is typically employed. This can be seen, globally, as mapping from one color space into another. For example, to reconcile traditional computer graphics images given in triplets of red, green, and blue values into single grayscale values, a formula such as this may be used, which seeks to equate the colorimetric luminance of the RGB image to a corresponding grayscale rendition [54]:

$$\mathrm{Photometric}\big(\langle R, G, B\rangle\big) = 0.3R + 0.59G + 0.11B$$

Careful consideration of the underlying photometric nature of the image being encoded therefore must be given, but only at this particular moment in the overarching algorithm for encoding. The choice of the Photometric() function determines the interrelationship of the image's colorimetry and its constituent importance to the matching function.

3.4.3.4. Affine transformations The fractal encoding algorithm seeks to find the best matching fragment in a source image that corresponds to a given image partitioned from the target image. As shown above, this matching is achieved by calculating the photometric correspondence function between two fragments, while considering all admissible transformations of the fragment from the source. The set of admissible transformations is a subset of affine transformations known as similitude transformations.

An affine transformation, in two dimensions, may be considered to be of the form:

$$W(x, y) = (ax + by + e, cx + dy + f)$$

where a, b, c, d, e, and f are all real numbers. This equation, which maps one point in a two-dimensional plane into another point in a two-dimensional plane, may be rewritten into matrix form like so:

$$W\big(\langle x, y\rangle\big) = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

In this way it can be seen that an affine transformation is a combination of a linear transformation followed by a translation.

Not all affine transformations are admissible for the fractal encoding transform, however. In particular, those that are admissible must be invertible [5]. Intuitively, this means that each point in space can be associated with exactly and only one other point in space. Mathematically, this means that the inverse has this form:

$$W^{-1}(x, y) = (dx - by - de + bf, -ex + ay + ce - af)/(ad - bc)$$

and the denominator must not be equal to zero to satisfy invertibility.

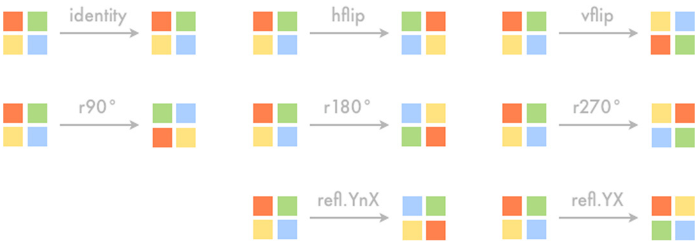


Fig. 5. Eight similitude transformations.

3.4.3.5. Similitude transformations An important group of affine transformations are those that are called similitudes [5].

A similitude transformation may be expressed in one of these two forms:

$$W(\langle x,y\rangle) = \begin{bmatrix} r\cos\theta & r\sin\theta \\ r\sin\theta & -r\cos\theta \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

$$W(\langle x,y\rangle) = \begin{bmatrix} r\cos\theta & -r\sin\theta \\ r\sin\theta & r\cos\theta \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

Thus, a similitude transformation is a composition of a dilation factor r, an orthonormal transformation (a rotation about the angle θ where $0 \le \theta < 2\pi$), and a translation (e, f). Similitude transformations are invertible except when $r = 0$.

3.4.3.6. Defining the AdmissibleTransformations set Given this formulation for similitude transformations, one can imagine having to consider a great many potential rotational angles to find the best match. Indeed, the computational complexity of the encoding would seem a function of the angles under consideration. In practice, we find that we may limit ourselves to considering only eight of these orthonormal transformations, as shown in Fig. 5.

Consider the smallest region of pixels for which orthonormal transformations upon those pixels would result in a visible change. The size of this region is an area two pixels wide by two pixels high. This small region has four lines of symmetry. Taking into account each line of symmetry, and reflecting the pixels in the region about each in turn, we find that there are eight possible outcomes.

Our implementation of the fractal encoding algorithm examines each potential correspondence under each of these possible transformations. These form the set of admissible transformations. The transformation from this set that yields the best photometric correspondence is noted by the search algorithm.

3.4.3.7. Translation arises from searching The searching process examines each potential fragment in a given source image for correspondence to a particular fragment of the target image. Let us presume that we may align the coordinate systems of the source and the target images such that their origins exactly coincide. Then, the relative location of a potential fragment in the source image can be mapped to a location within the target image. This mapping, from the potential fragment's local origin to the particular fragment's local origin, is a translation, and it is this mapping that forms the translation portion of the sought-for similitude transformation.

3.4.3.8. Dilation and fractals Taken together, the orthonormal transformation and the translation provide a sufficient means for describing self-similarity that may exist within an image. However, that self-similarity is not quite sufficient for describing how the similarity may occur at different levels of detail. The dilation factor, r, is used to invoke a contraction of space, whenever $r < 1.0$. The fractal encoding algorithm prescribes that the dilation factor to be used when searching may be conveniently set as $r = 0.5$. In practice, this entails that the source image, as a whole, may be scaled to one-half its original size, and then searched for photometrically corresponding fragments.

Mathematically, choosing $r < 1.0$ ensures that the encoding derived for the entire image, if applied successively and indefinitely to an image, will cause the resulting image to converge upon the desired destination image [5].

3.4.3.9. Colorimetric contraction As a final step, having located the best photometrically corresponding source fragment, the algorithm determines a rate at which the two regions may be brought into colorimetric harmony. To do this, the average colorimetric description of both regions is calculated, and the distance between the two is multiplied by a dilation. The formula our present implementation uses to calculate the colorimetric contraction is:

$$\text{colorContraction}(a_k, b_i) = 0.75 * \big(\text{colorMean}(b_i) - \text{colorMean}(a_k)\big)$$

where the colorMean of a region is the average of all colorimetric information available in that region, taking into account the multivariate nature of the underlying image as previously discussed. The derivation of the colorimetric dilation factor of 0.75 is given by Barnsley and Hurd [5], and is shown to be correlated to the spatial dilation factor of 0.5.

Table 1
Elements of a fractal code.

Spatial		Photometric	
s_x, s_y	Source fragment origin	C	Colorimetric contraction
d_x, d_y	Destination fragment origin	Op	Colorimetric operation
T	Orthonormal transformation		
S	Size/shape of the region		

3.4.3.10. Exhaustive searching The search over the source image A for a matching fragment is exhaustive, in that each possible correspondence a_k is considered regardless of its prior use in other discovered transforms. By allowing for such reuse, the algorithm affords the first Mandelbrot fractal observation, the notion of repetition.

3.4.3.11. Refining correspondence We note that there may be many fragments in the source image which may have identical photometric correspondence to the sought for fragment b_i (for example, when all of the values in the two fragments are identical). To break these potential ties, a further refinement of the correspondence function is necessary.

We compute a simple distance metric upon the images, and give it a weighting. Thus, the correspondence calculated between two fragments becomes:

$$\begin{aligned}\mathrm{Correspondence}(a_k, b_i) = {} & w_1 \mathrm{PhotometricCorrespondence}\big(\mathrm{Transform}(a_k, t), b_i\big) \\ & + w_2 \mathrm{Distance}(a_k, b_i) \\ \forall a_k \subset A, t \in {} & \mathrm{AdmissibleTransformations}\end{aligned}$$

where the weights w_1 and w_2 are chosen such that the calculation of correspondence is dominated by the value of the photometric correspondence. This can be ensured if the following relationship is held:

$$w_2 \text{ maximalDistance} \ll w_1 \text{ minimalJustNoticeablePhotometric}$$

where maximalDistance is the longest possible distance between the origins of b_i and any fragment in the corresponding source image, and minimalJustNoticeablePhotometric is the PhotometricCorrespondence which would be calculated if the photometric difference between b_i and any fragment were so small as to be indistinguishable. Practically, we set this value such that this is as small as possible yet not zero, given the color system used in the images. For example, for 8-bit greyscale images where the value 0 represents "black" and the value 255 represents "white," the minimalJustNoticeablePhotometric would be set to a value of 1.

3.4.4. Fractal codes

For each image b_i taken from a partitioning of the target image B, the fractal encoding algorithm locates, via exhaustive search over the source image A, a corresponding fragment a_k which the algorithm has deemed to be most minimally distant photometrically under a discovered transformation. The algorithm constructs a description of its discoveries, in a representation called a fractal code. A fractal code consists of six elements, as shown in Table 1.

Note that the dilation factor, for both spatial and photometric properties, is not represented here. This is for efficiency, as these dilations are presumed to be global.

Further efficiencies of expression also may be found by dropping the colorimetric operation (a way of describing how the colorimetric contraction value is to be combined into the region). Since the set of orthonormal transformations the search mechanism uses is finite, we may represent the transformation as a referent to that transformation's ordinal membership in the set. The size and shape of the region may be reduced itself, if the partitioning of the image is regular. In our implementation, we use a regular, uniform partitioning, which forms a grid. Thus, we can express the size and shape of the region with a single integer, which represents the width and height of the region in pixels.

3.4.5. Arbitrary selection of source

The choice of source image A is arbitrary. Indeed, the target image B may be fractally encoded in terms of itself, by substituting B for A in the above algorithm. Although one might expect that this substitution would result in a trivial encoding (in which all fractal codes correspond to an identity transform), this is not the case, a fractal encoding of B will converge upon B regardless of chosen initial image. For this reason, the size of source fragments considered is taken to be twice the dimensional size of the target image fragment, resulting in a contractive affine transform. Similarly, as shown above, color shifts are made to contract. This contraction, enforced by setting the dilation of spatial transformations at 0.5, provides the second key fractal observation, that similarity and repetition occur at differing scales.

3.4.6. Arbitrary ordinality of encoding

The ordinality of the set of fractal codes that comprise a fractal representation is similarly arbitrary. The partitioning P 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, so long as all are applied in any particular iteration.

3.4.7. Fractal representation is fractal encoding

The fractal encoding algorithm, while computationally expensive in its exhaustive search, represents the relationship between two images (or between an image and itself) as a much smaller set of fractal codes, an instruction set for reconstituting the relationship, with inherently strong spatial and photometric correspondence. It is through this encoding that the fractal representation of the relationship between those two images is derived. Indeed, the fractal representation is the fractal encoding.

3.5. Features from fractals

The fractal representation of an image is an unordered set of fractal codes, which compactly describe the geometric alteration and colorization of fragments of the source image that will collage to form the target 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 non-adjacent source material). Accordingly, we consider each of these fractal codes independently, and construct candidate fractal features from the individual codes themselves, and not from clusters of codes.

Each fractal code yields a small set of features, formed by constructing subsets of its underlying six-tuple. These features are determined in a fashion to encourage spatial- and photometric-agnosticism, as well as specificity. Our algorithm creates features from fractal codes by constructing subsets of each of the six members of the fractal code's tuple.

We further chose to represent each feature as a concatenated string in memory. We form these strings by attaching a character tag to each field in the fractal code and then converting that field into string format prior to concatenation, like so:

$$s_x, s_y \text{(source fragment origin)} \longrightarrow Ss_x s_y \text{(string representation)}$$

The choice of the particular tag is arbitrary, but tagging itself is not: tagging is necessary to avoid in-string matching between the different kinds of fields (e.g. a numerical value may appear in multiple fields of a fractal code). Doing so attributes a world grounding to each field, and collectively to the entire fractal code.

3.6. Mutuality

The analogical relationship between source and target 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 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 label the representation of the fractal transformation from image A to image B as T_{AB}. Correspondingly, we would label the inverse representation as T_{BA}. 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 fractal representations 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.

3.7. Extended mutuality

We note that the mutual fractal representation of the pairings may be employed to determine similar mutual representations of triplets, quadruplets, or larger groupings of images. As a notational convention, we construct these additional representations for triplets (M_{ijk}) and quadruplets (M_{ijkl}) in a like manner:

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

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

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

4. A strategy for visual analogies

As we noted at the paper's outset, we can interpret visual analogies as suggesting that some operation T exists that captures the relationship between image A and image B ("A is to B"). Likewise, some other operation T' is proposed which captures the relationship between image C and image D ("C is to D").

Let us now consider a class of visual analogy puzzles, an example of which is shown in Fig. 6.

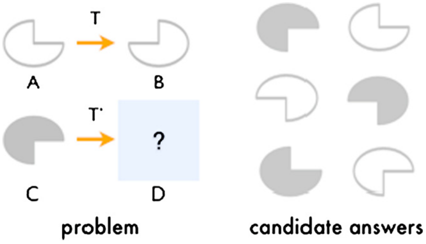


Fig. 6. A visual analogy puzzle.

In this problem, the image D is missing, and the challenge is to determine which of the offered candidate images would best fit into the matrix. That is, we must determine which of these candidate images, if selected as image D, would establish transformation T' as most analogous to transformation T.

Analogies in a general sense are based on similarity and repetition e.g. [34]. We would seek to employ a suitable representation, one that affords the capture of these qualities as well as sanctions reasoning over them. As we showed in the prior sections of this paper, fractals capture self-similarity and repetition at multiple scales [52], and we propose that fractal representations are an appropriate choice for addressing certain classes of analogy problems.

One method for solving this puzzle is this: from this set of candidates, we may form the fractal representations from the fractal encoding of the transformation of each candidate image X in terms of image C.

$$\forall X \in \{\text{candidate answers}\}, \quad T_x := \text{FractalEncode}(C, X)$$

$$\Omega = \{T_1, T_2, T_3, T_4, \ldots T_n\} \quad \text{and} \quad T' \in \Omega$$

This provides a set of possible transformations, which we shall label Ω, from which to seek the most analogous transformation T' and thereby find which candidate image was responsible for it.

4.1. *The generality of representations*

We wish to note that while in this paper we exclusively shall use fractal representations in our examples and subsequent discussion, our overall approach is agnostic with respect to representations, and may be used with any representation that affords the ability for objects thus represented to be decomposed into a set of features. The approach is distinguished from other analogical algorithms in that it presumes no explicit relationship between objects or between features of objects.

4.1.1. *Preliminary remarks*

Our approach compares each transform in the set Ω to the original transform T by means of recalling common features and calculating similarity metrics. First, however, we must make a few remarks on our present implementation. We chose data structures that facilitate the storage and retrieval of information based upon aspects of the data, specifically by using a hash table as a data structure surrogate for memory. As we will be hashing transformations into memory, we define two additional operators: $F()$, a method to generate a set of features from a given transformation; and $K()$, an injective hash function which operates solely over the domain of the features.

We made the commitment to a hash table for two reasons beyond that of wishing to use features. First, we note that it is desirous to find some overlap in the features that occur between two transformations, such that a perfect overlap would deem the transformations perfectly analogous. The hash function $K()$ may result in hashing multiple transformations to the same feature, and therefore $K()$ must operate only upon a given feature, and not take into consideration the transformation which gave rise to that feature. Second, $F()$, the method which generates features from a transformation, must do so in a manner such that each generated feature affords salience, or information content [68].

4.1.2. *Indexing features*

We wish to store each transformation in the hash table memory M. The set of possible analogous transformations Ω is combined with the original transformation T to form a new set Ω^*. The algorithm iterates over each member $\tau \in \Omega^*$, and from each member calculates a set of features using $F(\tau)$. For each feature $f_i \in F(\tau)$, the transformation is indexed as an ordered pair $(K(f_i), \tau)$. That there likely will be hash collisions at key value $K(f_i)$ is expected and desired.

4.1.3. *Calculating similarity*

We must determine a measure of similarity between the original transformation T and each possible analogous transformation $T_i \in \Omega$. Our choice of metric reflects similarity as a comparison of the number of features shared between candidate pairs taken in contrast to the joint number of features found in each pair member [30]. We desire a metric that is normalized with respect to the number of features under consideration. In our implementation, the measure of similarity between the target transform T and a candidate transform T_i is calculated using the ratio model [68]:

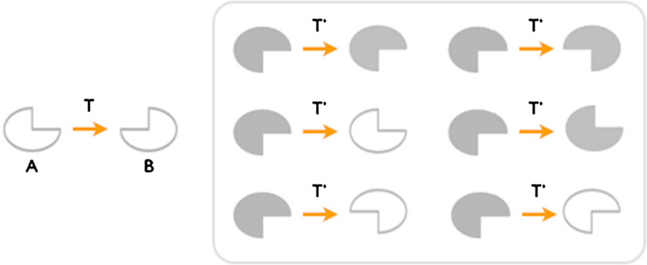


Fig. 7. The primary and candidate transformations.

$$S(T, T_i) = \mathcal{F}(T \cap T_i) / \big(\mathcal{F}(T \cap T_i) + \alpha \mathcal{F}(T - T_i) + \beta \mathcal{F}(T_i - T)\big)$$

and $\mathcal{F}(Y)$ is a function that determines the number of features that may be extracted from the set Y. We may calculate these values effectively, using hash table retrieval as a surrogate for distinguishing and counting common and distinct features within the sets $T \cap T_i$, $T - T_i$, and $T_i - T$ respectively.

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 [30]. We have found that significant discrimination between candidate answers may be found by using the Jaccard similarity; that is, by setting $\alpha \leftarrow \beta \leftarrow 1.0$, and thus favoring features from either transformation equally. As Tversky notes, by equating α and β, we ensure that the calculation of similarity is symmetric with respect to the transformations under comparison.

Once the algorithm has calculated the similarity function over all candidate transforms, it is a straightforward matter to determine which transformation has generated the maximal similarity. This transformation, T', is deemed by way of Tversky featural similarity, to be the most analogous to the original transformation T.

4.2. An example

We now present an example of using fractal representations and our strategy to solve the visual analogy puzzle shown in Fig. 3 above.

4.2.1. The primary and candidate transformations

In this example, the problem is to determine for which of the candidate images the transformation T' is made most analogous to transformation T. We first will represent T as a fractal representation, and then generate a set of candidate transformations Ω as shown in Fig. 7.

We arbitrarily may select any partitioning scheme, so long as that partitioning meets the criteria of coverage outlined above. For the purpose of this example, let us choose to partition each image into a series of 16 × 16 pixels, forming a regular grid. Each of our images is 134 pixels wide and 84 pixels high. Thus, each image is to be partitioned into 54 blocks.

Each of these blocks will be represented by a single fractal code. In our present implementation, each fractal code generates 63 features. Therefore, at this partitioning, each primary transformation will be indexed into memory using the $54 \times 63 \times 2 = 6804$ features generated from mutual fractal representation of that transformation.

4.2.2. Calculating similarities and selecting the most analogous

After the primary transformation T is indexed into memory using features derived from the fractal codes, we may calculate a similarity value for each of the candidate transformations in the set Ω, using the Tversky formula as noted. Table 2 illustrates the values calculated for each of the candidate transformations.

It may be seen that the fourth transformation is the most similar to the primary transformation T, with a value of 0.842. Therefore, for this puzzle, the answer is candidate answer 4.

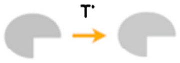
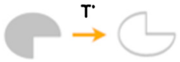
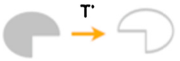
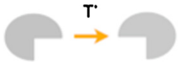
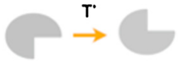
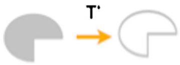
5. Tackling Ravens with fractals

With this strategy in mind, let us illustrate the use of fractal representations for solving Raven's matrices problems. We shall use by way of example the 3 × 3 matrix problem shown in Fig. 1.

5.1. Simultaneous relationships, multiple constraints

An aspect of any Raven's problem, whether 2 × 2 or 3 × 3, is that there exist simultaneous horizontal and vertical relationships which must be maintained by the selection of the most analogous answer. In a 2 × 2 problem, there is one horizontal relationship and one vertical relationship that constrain the selection. In a 3 × 3 problem, there are two horizontal and two vertical relationships. In our implementation, we represent these relationships as mutual fractal representations.

Table 2
Candidate transformation similarities.

Transformation	Similarity
	0.623750735
	0.517048795
	0.552910053
	0.84244562
	0.573780129
	0.535273369

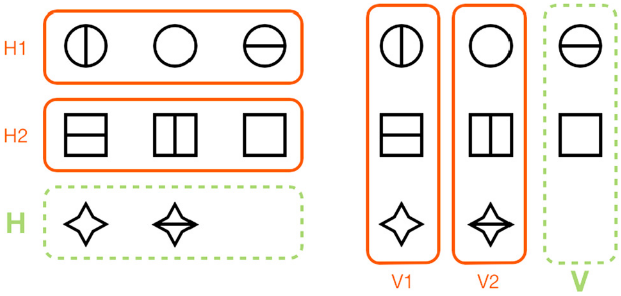


Fig. 8. The simultaneous relationships.

In Fig. 8, we illustrate these relationships using our example problem. As shown, relationships $H1$ and $H2$ constrain relationship H, while relationships $V1$ and $V2$ constrain relationship V. We note that there are other possible relationships that may be suggested by this problem: we have chosen to focus on these particular four relationships for clarity.

To solve a Raven's problem, one must select the image from the set of possible answers for which the similarity to each of the problem's relationships is maximal. For our example, this involves the calculation of a set of similarity values Θ_i for each answer A_i:

$$\Theta_i \leftarrow \{S(H1, H(A_i)), S(H2, H(A_i)), S(V1, V(A_i)), S(V2, V(A_i))\}$$
$$\forall i,\ 1 \le i \le 8$$

where $S(X, Y)$ is the Tversky similarity between two sets X and Y, and $H(A_i)$ and $V(A_i)$ denote the relationship formed when the answer image A_i is included.

5.1.1. Reconciling multiple analogical relationships

For each candidate answer, 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. Thus, for a 2 × 2 Raven's problem, the space is 2-dimensional; for a 3 × 3 Raven's problem, the space is 4-dimensional, using the relationships as shown above.

We would like to determine a single value for the similarity. To do so, we treat these multidimensional sets as a vector, and determine its length, using a Euclidean distance formula:

$$S_i \leftarrow \sqrt{\sum \Theta_{ij}^2} \quad \forall i,\ 1 \le i \le 8 \text{ and } \forall \Theta_{ij} \in \Theta_i$$

Thus, the longer the vector, the more similar; the shorter the vector, the more dissimilar.

Generally, 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. We note that giving preferential weighting to a relationship is a straightforward extension to the calculation above, but we also note that choosing which relationship to prefer may be non-trivial.

5.2. Confidence

As illustrated by the example above, a candidate representation can be found to be the most analogous by a straightforward calculation of its featural similarity, using the Tversky formula [68]. But a question quickly arises: how confident is the answer? That is, given the variety of answer choices, even though an answer may be selected based on maximal similarity, how may that choice be contrasted with its peers as the designated answer?

A potential path would be to assess the given similarity calculation as a member of the set of all such similarity calculations for the collection of potential answers. Let us suppose that for a given collection of answers, we use a strategy such as ours to calculate a corresponding set of similarity values:

$$\begin{aligned}&\text{strategy}\big(\{A_1, A_2, A_3, A_4, \ldots, A_n\}, \text{problem}\big) \rightarrow \{S_1, S_2, S_3, S_4, \ldots, S_n\}\\&\quad \forall S_i,\ 0.0 \le S_i \le 1.0\end{aligned}$$

We may then select

$$\zeta \leftarrow \max\big(\{S_1, S_2, S_3, S_4, \ldots, S_n\}\big)$$

as the maximal similarity value, and thereby deem the answer which generated that value as the most analogous. We may determine, additionally, how statistically distinct the value ζ is from its peers, by first calculating the mean, standard deviation, and standard error of the mean for the set of similarity values, and then, assuming a normal distribution, calculating the deviation of each of the values from the mean of the set:

$$\begin{aligned}&\mu = n^{-1}\sum S_i \quad \text{and} \quad \sigma = \sqrt{\Big[n^{-1}\sum (S_i - \mu)^2\Big]} \quad \forall i,\ 0 < i \le n\\&\sigma_\mu = \sigma/\sqrt{n}\end{aligned}$$

then

$$\begin{aligned}&\text{Deviations} \leftarrow \{D_1, D_2, D_3, D_4, \ldots, D_n\}\\&\quad \forall i,\ 0 < i \le n,\ D_i = (S_i - \mu)/\sigma_\mu\end{aligned}$$

where the set Deviations is a t-distribution of the similarity values. We surmise that the most analogical answer, the one corresponding to the maximal similarity value ζ, would have the largest positive deviation value under this reformulation:

$$\begin{aligned}&S_x = \zeta \leftarrow \max\big(\{S_1, S_2, S_3, S_4, \ldots, S_n\}\big) \quad \text{iff}\\&\quad \exists x = y, \quad D_y = \max\big(\{D_1, D_2, D_3, D_4, \ldots, D_n\}\big)\end{aligned}$$

This, then, suggests that the most analogical answer would in a sense “stand apart” from the rest of the answers. The degree to which it “stands apart” may be interpreted as a metric of confidence in selecting the answer. Indeed, assuming a normal distribution, we may calculate a confidence interval based upon the standard deviation, and score each of these values along such a confidence scale, where 0.0 would indicate no variation at all from the answer, and 1.0 would indicate an utterly apparent and distinct value. Thus, the problem of selecting the most analogous answer is transformed into a problem of distinguishing which of the possible answers is a statistical outlier.

Since the set of deviations determined is a t-distribution of the similarity values, the choice of a level of confidence in selecting the answer can be made using conventional statistics. If we assume a normal distribution of the deviations, then we may devise an interval such that with an expected confidence of C for any deviation d, the probability of d in that interval is C:

$$P(-z \le d \le z) = C$$

The set of deviations is a t-distribution of the similarity values. We know that if this is a normal distribution:

$$\begin{aligned}&P(-z \le d \le z) = \Phi(z) - \Phi(-z) = \text{erf}\big(z/\sqrt{2}\big)\\&\quad z = \sqrt{2}\,\text{erf}^{-1}(C)\end{aligned}$$

where $\Phi()$ is the cumulative normal distribution function and erf() is the error function. Therefore, given some value C, we can calculate the equation above to determine the boundary.

Let us suppose, as an example, that we wished to say with $C = 90\%$ that some answer X_i is the most analogous one. This would imply that the probability of that answer’s corresponding deviation D_i obeys this:

```
Given an image $P$ containing a Raven's problem, determine an answer.
Problem Segmentation

By examination, divide $P$ into two images, one containing the matrix and the other containing the possible answers. Further divide the matrix image into an ordered set of either 3 or 8 matrix element images, for $2 \times 2$ or $3 \times 3$ matrices respectively. Likewise, divide the answer image into an ordered set of its constituent individual answer choices.

Let $M := \{m_1, m_2, \ldots\}$ be the set of matrix element images.
Let $C := \{c_1, c_2, c_3, \ldots\}$ be the set of individual answer choices.
Let $\eta$ be an integer denoting the order of the matrix image (either 2 or 3, for $2 \times 2$ or $3 \times 3$ matrices respectively).

Relationship Designations

Let $R$ be a set of relationships, determined by the value of $\eta$ as follows:
If $\eta = 2$:
    $R \leftarrow \{H_1, V_1\}$ where
    $H_1 \leftarrow \mathrm{MutualFractal}(m_1, m_2)$
    $V_1 \leftarrow \mathrm{MutualFractal}(m_1, m_3)$
Else: (because $\eta = 3$)
    $R \leftarrow \{H_1, H_2, V_1, V_2\}$ where
    $H_1 \leftarrow \mathrm{MutualFractal}(m_1, m_2, m_3)$
    $H_2 \leftarrow \mathrm{MutualFractal}(m_4, m_5, m_6)$
    $V_1 \leftarrow \mathrm{MutualFractal}(m_1, m_4, m_7)$
    $V_2 \leftarrow \mathrm{MutualFractal}(m_2, m_5, m_8)$

Abstraction Level Preparation

Let $d$ be the largest pixel dimension for any image in the set $M * C$.
Let $A := \{a_1, a_2, \ldots\}$ represent an ordered range of abstraction values where
    $a_1 \leftarrow d$, and $a_i \leftarrow 1/2a_{i-1}\ \forall i, 2 \le i \le \lfloor \log 2\ d \rfloor$ and $a_i \ge 2$
The values within $A$ constitute the grid values to be used when partitioning the problem's images.
```

Algorithm 3. The Fractal Raven Algorithm, preparatory stage.

$$z = \sqrt{2}\,\mathrm{erf}^{-1}(0.90) = 1.644853$$

Thus, if the deviation D_i is larger than 1.644853, we can say with 90% confidence that the answer X_i is the most analogous. In our implementation, we chose a 95% confidence value, therefore we seek a deviation which is larger than 1.959964, approximately a 2-sigma signal.

In this manner, we have the ability to determine mathematically not only which value may have the highest similarity, but also the degree to which that value is statistically distinct. We require this, for our approach considers the analogy problem at an array of levels of abstraction. In addition, in this way may be seen the specific interplay of confidence and abstraction when solving problems of analogy, a novel aspect of our work.

6. The Fractal Raven Algorithm

We now present the Fractal Raven algorithm in pseudo-code form. We separate the algorithm into two parts: the preparatory stage (see Algorithm 3) and the execution stage (see Algorithm 4).

6.1. The Fractal Raven Algorithm: preparatory stage

In the first stage of our Fractal Raven Algorithm, an image containing the entire problem is first segmented into its component images (the matrix of images, and the possible answers). Next, based upon the complexity of the matrix, the algorithm determines the set of relationships to be evaluated. Then, a range of abstraction levels is determined.

As we have implemented it, the abstraction levels are determined to be a partitioning of the given images into gridded sections at a prescribed size and regularity.

6.2. The Fractal Ravens Algorithm: execution stage

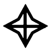
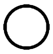
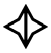
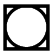
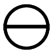
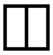
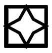
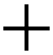
The algorithm concludes by determining the confidence in the answers at each level, stopping when ambiguity is sufficiently resolved. Thus for each level of abstraction, the relationships implied by the kind of Raven's problem (2×2 or 3×3) are re-represented into that partitioning. Then, for each of the candidate images, a potentially analogous relationship is determined for each of the existing relationships and a similarity value calculated. The vector of similarity values is reduced via a simple Euclidean distance formula to a single similarity. The balance of the algorithm, using the deviation from the mean of these similarities, continues through a variety of levels of abstraction, looking for an unambiguous answer that meets a specified confidence value.

```
Given $M, C, R, A$, and $\eta$ as determined in the preparatory stage, find the answer.
Preparatory
Let $E$ be a real number which represents the number of standard deviations beyond which a value's answer may be judged as "confident"
Let $S(X, Y)$ be the Tversky similarity metric for sets $X$ and $Y$
Execution
For each abstraction $a \in A$:
  • Re-represent each fractal representation $r \in R$ according to abstraction $a$
  • $S \leftarrow \emptyset$
  • For each answer image $c \in C$:
    • If $\eta = 2$:
        $H \leftarrow \mathrm{MutualFractal}(m_3, c)$ according to abstraction $a$
        $V \leftarrow \mathrm{MutualFractal}(m_2, c)$ according to abstraction $a$
        $\Theta \leftarrow \{S(H_1, H), S(V_1, V)\}$
    • Else: (because $\eta = 3$)
        $H \leftarrow \mathrm{MutualFractal}(m_7, m_8, c)$ according to abstraction $a$
        $V \leftarrow \mathrm{MutualFractal}(m_3, m_6, c)$ according to abstraction $a$
        $\Theta \leftarrow \{S(H_1, H), S(H_2, H), S(V_1, V), S(V_2, V)\}$
    • Calculate a single similarity metric from vector $\Theta$:
        $t \leftarrow \sqrt{\sum \theta^2}\ \forall \theta \in \Theta$
        $S \leftarrow S \cup \{t\}$
  • Set $\mu \leftarrow \mathrm{mean}(S)$
  • Set $\sigma_\mu \leftarrow \mathrm{stdev}(S)/\sqrt{n}$
  • Set $D \leftarrow \{D_1, D_2, D_3, D_4, \ldots, D_n\}$ where $D_i = (S_i - \mu)/\sigma_\mu$
  • Generate the set $Z := \{Z_i \ldots\}$ such that $Z_i \in D$ and $Z_i > E$
  • If $|Z| = 1$, return the answer image $C_i \in C$ which corresponds to $Z_i$
  • otherwise there exists ambiguity, and further refinement must occur.

If no answer has been returned, then no answer may be given unambiguously.
```

Algorithm 4. The Fractal Raven Algorithm, execution stage.

Table 3
Image deviations and confidences for the example problem.

Image	Deviations & confidences						
	0.175 13.84%	−2.035 −95.82%	−1.861 −93.72%	0.698 51.47%	−0.760 −55.29%	−0.610 −45.79%	−1.311 −81.02%
	−0.321 −25.17%	4.166 > 99.99%	2.783 99.46%	2.179 97.07%	0.681 50.4%	1.106 73.12%	0.480 36.86%
	6.390 > 99.99%	3.484 99.95%	2.930 99.66%	4.487 > 99.99%	3.961 99.99%	4.100 > 99.99%	4.006 > 99.99%
	0.495 37.97%	−3.384 −99.93%	−3.841 −99.99%	−4.848 < −99.99%	−4.958 < −99.99%	−5.454 −100%	−4.620 < −99.99%
	−1.741 −91.84%	−1.678 −90.67%	−2.148 −96.83%	−0.591 −44.56%	−2.825 −99.53%	−1.921 −94.52%	−2.775 −99.45%
	−0.321 −25.17%	1.560 88.12%	2.444 98.55%	−1.361 −82.64%	0.896 62.96%	0.643 47.99%	0.832 59.49%
	−1.741 −91.84%	0.254 20.02%	2.172 97.02%	−1.826 −93.22%	0.668 49.58%	0.213 16.85%	0.570 43.15%
	−2.935 −99.67%	−2.366 −98.20%	−2.479 −98.68%	1.262 79.31%	2.338 98.06%	1.922 94.54%	2.817 99.52%
grid size	200	100	50	25	12	6	3
μ	0.589	0.310	0.432	0.690	0.872	0.915	0.948
σ_μ	0.031	0.019	0.028	0.015	0.007	0.005	0.003
codes	6	24	96	384	1734	6936	26934
features	378	1512	6048	24 192	109 242	436 968	1 696 842

6.3. Addressing the example

Table 3 shows the results of running the Fractal Ravens algorithm on the example problem, starting at an original gridded partitioning of 200 × 200 pixels (the maximal pixel dimension of the images), and then refining the partitioning down to a grid of 3 × 3 pixels. The table gives the mean (μ), standard deviation (σ_μ), and number of features (f) for each level of abstraction (grid). The deviation and confidence for each candidate answer are given for each level of abstraction as well.

A confidence level of 95% is sought. In the table, we color a cell yellow if it exceeds the desired confidence level, and red if it does so unambiguously for the given grid partitioning.

6.4. *Discussion of the example results*

The deviations presented in Table 3 appear to suggest that if one starts at the very coarsest level of abstraction, the answer is apparent (image choice 3). Indeed, the confidence in the answer never dips below 99.83%, across all levels of abstraction. As we have shown, while that answer has the highest similarity value, statistically there are other alternatives.

We see evidence that operating with either too sparse a data set (at the coarsest) or with too homogeneous a data set (at the finest) may be problematic. The coarsest abstraction (200 pixel grid size) offers 378 features, whereas the finest abstraction (3 pixel grid size) offers more than 1.5 million features for consideration.

The data in the table continues to suggest the possibility of automatically detecting these boundary situations. We note that the average similarity measurement at the coarsest abstraction is 0.589, but then falls, at the next level of abstraction, to 0.310, only to thereafter generally increase. This constitutes further evidence for an emergent boundary for coarse abstraction.

We suspect that ambiguity exists for ranges of abstraction, only to vanish at some appropriate levels of abstraction, and then reemerges once those levels are surpassed. We see evidence of such behavior in this example, where there exists ambiguity at grid sizes 100, 50, 25, and 12, then the ambiguity vanishes for grid size 6, and then reemerges for grid size 3. This suggests that there are features within the image that are sufficiently discriminatory only at certain levels of abstraction.

7. Fractal Ravens results

We have tested our Fractal Ravens algorithm on all problems associated with the four main variations of the Raven's Progressive Matrices Tests: 60 problems of the Standard Progressive Matrices test, 48 problems of the Advanced Progressive Matrices test, 36 problems of the Coloured Progressive Matrices test, and 60 problems of the SPM Plus test. To our knowledge, this is the first published account of a computational model's attempt at the entire Raven's test. In this section, we present our results and discuss our findings.

7.1. *Inputs used for the test*

To create inputs for the Fractal Ravens algorithm, each page from the various Raven test booklets were scanned, and the resulting grayscale 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 technique for each problem. No further image processing or cleanup was performed, despite the presence of numerous pixel-level artifacts introduced by the scanning and minor inter-problem image alignment issues. Additionally, the fractal algorithm attempted to solve each problem independently: no information was carried over from problem to problem, or from test variant to test variant.

The code used in conducted these runs is precisely the same code as used in the earlier example. This code is available for download from our lab website. The images scanned, however, are copyrighted and thus are not available for download. However, we believe that the instructions for preparing the images provided above will allow for someone with access to the Ravens materials to reproduce our results.

7.2. *Levels of abstraction considered and calculations performed*

The images associated with each problem had a maximum pixel dimension of between 150 and 250 pixels. Accounting for variation within each test problem, and setting a minimum grid size of 4 pixels, we therefore calculated five or six levels of abstraction for each problem, using the formula described above for determining maximum grid size and using a strategy of halving the pixel dimension at each successively finer level of abstraction.

At each level of abstraction, we calculated the similarity value for each possible answer, as proscribed by the Fractal Ravens algorithm. For those calculations, we used the Tversky formula, and set alpha to 1.0 and beta equal to 0.0, conforming to values used in the coincidence model by Bush and Mosteller [8]. From those values, we calculated the mean and standard deviation, and then calculated the deviation and confidence for each answer. We made note of which answers provided a confidence above our chosen level, and whether for each abstraction level the answer was unambiguous or ambiguous, and if ambiguous, in what manner. In those cases where ambiguity was found, we explored several different data techniques to assist in the resolution. We describe those techniques after the presentation of the performance.

7.3. *Assessment of Fractal Ravens performance against human norms*

There are three main assessments that can be made following the administration of a Raven test to an individual: the total score, which is given simply as the number of correct answers; an estimate of consistency, which is obtained by

Table 4
SPM results.

SPM	Detected correct	Determined by strategy	Ambiguous between 2 or 3
Total	50	38	9
set *A*	12	11	1
set *B*	10	8	1
set *C*	11	8	3
set *D*	9	6	2
set *E*	8	5	3

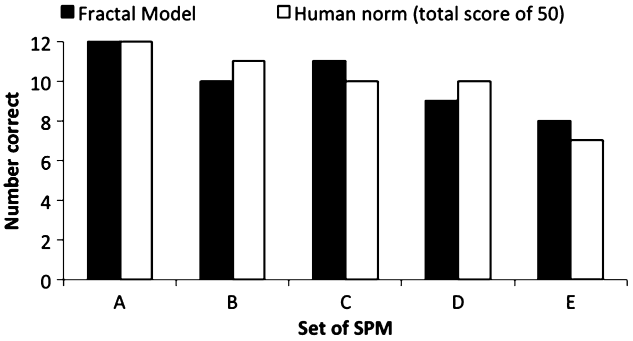


Fig. 9. Human norms comparison for SPM score of 50.

Table 5
APM results.

APM	Detected correct	Determined by strategy	Ambiguous between 2 or 3
Total	42	28	10
set *A*	10	6	4
set *B*	32	22	6

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 [61]. A score is "consistent" if the difference between the actual score and the expected score for any given set is no more than ±2 [61].

7.4. Performance on the Standard Progressive Matrices test

On the Raven's Standard Progressive Matrices test, the Fractal Ravens algorithm detected the correct answer at a 95% or higher level of confidence on 50 of the 60 problems. The number of problems with detected correct answers per set were 12 for set *A*, 10 for set *B*, 11 for set *C*, 9 for set *D*, and 8 for set *E*. Of the 50 problems where the correct answers detected, 38 were determinable by one or more of our ambiguity-resolution strategies. Of the remaining 12 problems noted answers, all but three were ambiguous between two or three particular answers. Table 4 provides a summarization of these results.

As shown in Fig. 9, the score differences for Fractal Ravens on each set were no more than ±1. For a human test-taker, this score distribution generally 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 for all sets.

Using norms from the United States, we note that a total score of 50 corresponds to the 95th percentile for children about 12 years old, the 75th percentile for children around 14 years old, and the 50th percentile for children older than 16 years old [61].

7.5. Performance on the Advanced Progressive Matrices test

On the Raven's Advanced Progressive Matrices test, the Fractal Ravens algorithm detected the correct answer at a 95% or higher level of confidence on 42 of the 48 problems. The numbers of problems with detected correct answers per set were 10 for set *A*, and 32 for set *B*. Of the 42 problems where the correct answers detected, 28 were determinable by one or more of our ambiguity-resolution strategies. Of the remaining 14 problems noted answers, all but four were ambiguous between two or three particular answers. Table 5 provides a summarization of the APM results.

The score differences for Fractal Ravens on both APM sets were no more than ±1, indicating consistency and that the results achieved by the algorithm fall well within typical human norms on the APM for both sets. We note that a total

Table 6
CPM results.

CPM	Detected correct	Determined by strategy	Ambiguous between 2 or 3
Total	30	24	6
set *A*	12	11	1
set *AB*	11	8	3
set *B*	7	5	2

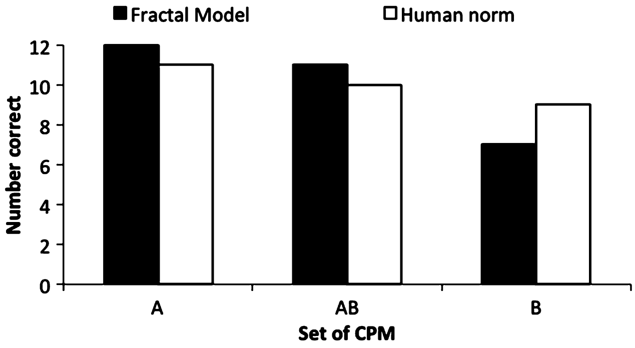


Fig. 10. Human norms comparison for CPM score of 30.

Table 7
SPM plus results.

SPM plus	Detected correct	Determined by strategy	Ambiguous between 2 or 3
Total	50	39	10
set *A*	10	9	1
set *B*	9	8	1
set *C*	9	5	4
set *D*	11	8	2
set *E*	11	9	2

score of 42 corresponds to the 95th percentile for adults between 50 and 60 years old, and exceeds the 75th percentile performance for adults of all measured ages [61].

7.6. Performance on the Coloured Progressive Matrices test

On the Raven's Coloured Progressive Matrices test, the Fractal Ravens algorithm detected the correct answer at a 95% or higher level of confidence on 30 of the 36 problems. The number of problems with detected correct answers per set were 12 for set *A*, 11 for set *AB*, and 7 for set *B*. Of the 30 problems where the correct answers detected, 24 were determinable by one or more of our ambiguity-resolution strategies. Of the remaining 6 problems noted answers, all were ambiguous between two or three particular answers. Table 6 provides a summarization of these results.

As shown in Fig. 10, the score differences for Fractal Ravens on each set were no more than ±2, indicating consistency. This score pattern also illustrates that the results achieved by the algorithm fall well within typical human norms on the CPM for all sets.

Using the United States norms, we note that a total score of 30 on the CPM test corresponds to the 95th percentile for children about 7 years old, the 75th percentile for children about 9 years old, and the 50th percentile for children about 11 years old [61].

7.7. Performance on the SPM Plus test

On the Raven's Standard Progressive Matrices Plus test, the Fractal Ravens algorithm detected the correct answer at a 95% or higher level of confidence on 50 of the 60 problems. The number of problems with detected correct answers per set were 10 for set *A*, 9 for set *B*, 9 for set *C*, 11 for set *D*, and 11 for set *E*. Of the 50 problems where the correct answers detected, 39 were determinable by one or more of our ambiguity-resolution strategies. Of the remaining 11 problems noted answers, all but one were ambiguous between two or three particular answers. Table 7 provides a summarization of these results.

Table 8
Comparing Fractal Ravens to other models.

Model/Set	SPM results							APM results			
	N	# att.	*A*	*B*	*C*	*D*	*E*	*N*	# att.	*A*	*B*
Carpenter et al. [9] "FairRaven"								23	34	7	16
Carpenter et al. [9] "BetterRaven"								32	34	7	25
Lovett et al. [50]	22	24	–	12	10	–	–				
Lovett et al. [51]	44	48	–	specifics unreported							
Cirillo & Ström [11]	28	36	–	–	8	10	10				
Kunda et al. [46–48] Affine	38	60	12	11	9	1	5	21	48	7	14
Fractal Ravens	50	60	12	10	11	9	8	42	48	10	32

7.8. *Comparison to other computational models*

While this is the first published account of a computational model's attempt at the entire suite of Raven's tests, as we noted in the introduction there are other computational models which have been used on some or all of certain tests. For those accounts that report scores, we compared their results with those achieved by the Fractal Raven algorithm. Table 8 documents the performance of the Fractal Ravens algorithm against those contemporaries.

Carpenter et al. [9] report results of running two versions of their algorithm (FairRaven and BetterRaven) against a subset of the APM problems. The subset of problems chosen by Carpenter et al. reflect those whose rules and representations were deemed as inferable by their production rule based system [9].

Lovett et al. [50,51] report results from their computational model's approach to the Raven's SPM test. In each account, only a portion of the test was attempted, but Lovett et al. project an overall score based on the performance of the attempted sections. The latest published account by Lovett et al. [51] reports a score of 44 out of 48 attempted problems from sets *B* through *E* of the SPM test, but does not offer a breakdown of this score by problem set. Lovett et al. [51] project a score of 56 for the entire test, based on human normative data indicating a probable score of 12 on set *A* given their model's performance on the attempted sets.

Cirillo and Ström [11] report that their system was tested against Sets *C* through *E* of the SPM and solved 8, 10, and 10 problems, respectively. Though unattempted, they predict that their system would score 19 on the APM (a prediction of 7 on set *A*, and 12 on set *B*).

Kunda et al. [46–48] report the results of running their Affine algorithm against all of the problems on both the SPM and the APM tests, with a detailed breakdown of scoring per test. They report a score of 38 for the SPM and a score of 21 on the APM. More recently, this approach has given better results on the SPM, and it also has been used to address problems on the CPM [44].

8. Conclusions

In this paper, we have presented a comprehensive account of our efforts to solve the entire Standard and Advanced Raven's Progressive Matrices tests using fractal representations. We developed the Fractal Ravens algorithm, a computational model which uses features derived from fractal representations to calculate Tversky similarities between relationships in the test problem matrices and candidate answers, and which uses levels of abstraction, through re-representing the fractal representation at differing resolutions, to determine overall confidence in the selection of an answer. Finally, we presented a comparison of the results of running the Fractal Ravens algorithm to all available published accounts, and showed that the Fractal Ravens algorithm's performance at detecting the correct answer met or exceeded those accounts. McGreggor [53] provides more details about our work.

Analogy initiates with an act of being reminded. We hold that fractally representing both that triggering percept as well as all prior percepts affords unprecedented similarity discovery, and thereby analogy-making. The Fractal Ravens algorithm, through its use of fractal representations and featural similarity evoked by recall, demonstrates these claims.

Although we have focused our attention on representation and computational psychometrics, and drawn comparisons between the Fractal Ravens performance and that of humans, we emphasize that what we developed within this paper are AI techniques and algorithms. While we expressly asserted no claims as to whether our model may be extended to provide a cognitive accounting, given the development of our computational model and the nature of our representation, it is our hope to spur others to explore additional computational and perhaps cognitive accounts of the intricate interplay between perception and analogy.

Acknowledgements

This work has benefited from many discussions with members of the Design and Intelligence Lab at Georgia Institute of Technology. We thank the US National Science Foundation for its support of this work through IIS Grant #1116541, entitled "Addressing visual analogy problems on the Raven's intelligence test."

References

[1] A. Aamodt, E. Plaza, Case-based reasoning: foundational issues, methodological variations, and system approaches, AI Commun. 7 (1) (1994) 39–59.
[2] D.W. Aha, L.A. Breslow, H. Muñoz-Avila, Conversational case-based reasoning, Appl. Intell. 14 (1) (2001) 9–32.
[3] K.D. Ashley, E.L. Rissland, A case-based approach to modeling legal expertise, IEEE Expert 3 (3) (1988) 70–77.
[4] F.G. Ashby, D.M. Ennis, Similarity measures, Scholarpedia 2 (12) (2007) 4116.
[5] M. Barnsley, L. Hurd, Fractal Image Compression, A.K. Peters, Boston, MA, 1992.
[6] S. Bhatta, A. Goel, An analogical theory of creativity in design, in: D. Leake, E. Plaza (Eds.), in: Lecture Notes in Computer Science, vol. 1266, 1997, pp. 565–574.
[7] S. Bringsjord, B. Schimanski, What is artificial intelligence? Psychometric AI as an answer, in: International Joint Conference on Artificial Intelligence, 2003, pp. 887–893, Citeseer.
[8] R.R. Bush, F. Mosteller, A stochastic model with applications to learning, Ann. Math. Stat. (1953) 559–585.
[9] P.A. Carpenter, M.A. Just, P. Shell, What one intelligence test measures: a theoretical account of the processing in the Raven Progressive Matrices Test, Psychol. Rev. 97 (1990) 404–431.
[10] D.J. Chalmers, R.M. French, D.R. Hofstadter, High-level perception, representation, and analogy: a critique of artificial intelligence methodology, J. Exp. Theor. Artif. Intell. 4 (3) (1992) 185–211.
[11] S. Cirillo, V. Ström, An Anthropomorphic Solver for Raven's Progressive Matrices, Chalmers University of Technology, Göteborg, Sweden, 2010.
[12] J.J. Clement (Ed.), Creative Model Construction in Scientists and Students: The Role of Imagery, Analogy, and Mental Simulation, Springer, Netherlands, 2008.
[13] D. Croft, P. Thagard, Dynamic imagery: a computational model of motion and visual analogy, in: L. Magnini, N. Nersessian (Eds.), Model-Based Reasoning, 2002, pp. 259–274.
[14] J. Davies, J. Glasgow, T. Kuo, Visuo-spatial case-based reasoning: a case study in prediction of protein structure, Comput. Intell. 22 (3/4) (2006) 194–207.
[15] J. Davies, A. Goel, N. Nersessian, A computational model of visual analogies in design, Cogn. Syst. Res., Special Issue on Analogies: Integrated Cognitive Abilities 10 (2009) 204–215.
[16] J. Davies, A. Goel, P. Yaner, Proteus: a theory of visual analogies in problem solving, Knowl.-Based Syst. 21 (7) (2008) 636–654.
[17] J. Davies, N. Nersessian, A. Goel, Visual models in analogical problem solving, Found. Sci. 10 (1) (2005) 133–152.
[18] M. Dawson, I. Soulières, M.A. Gernsbacher, L. Mottron, The level and nature of autistic intelligence, Psychol. Sci. 18 (8) (2007) 657–662.
[19] K. Dunbar, The analogical paradox: why analogy is so easy in naturalistic settings yet so difficult in the psychological laboratory, in: D. Gentner, K.J. Holyoak, B.N. Kokinov (Eds.), The Analogical Mind: Perspectives from Cognitive Science, MIT Press, Cambridge, MA, 2001, pp. 313–334.
[20] K. Dunbar, I. Blanchette, The in vivo/in vitro approach to cognition: the case of analogy, Trends Cogn. Sci. 5 (2001) 334–339.
[21] T.G. Evans, A heuristic program to solve geometric-analogy problems, in: Proceedings of the April 21–23, 1964, Spring Joint Computer Conference, ACM, 1964, pp. 327–338.
[22] B. Falkenhainer, K. Forbus, D. Gentner, The structure-mapping engine: algorithms and examples, Artif. Intell. 41 (1) (1989) 1–63.
[23] K. Forbus, D. Gentner, K. Law, MAC/FAC: a model of similarity-based retrieval, Cogn. Sci. 19 (2) (1995) 141–205.
[24] D. Gentner, Structure-mapping: a theoretical framework for analogy, Cogn. Sci. 7 (2) (1983) 155–170.
[25] D. Gentner, A. Markman, Structure mapping and analogy and similarity, Am. Psychol. 52 (1) (1996) 45–57.
[26] M. Gick, K.J. Holyoak, Analogical problem solving, Cogn. Psychol. 12 (1980) 306–355.
[27] M. Gick, K.J. Holyoak, Schema induction and analogical transfer, Cogn. Psychol. 15 (1983) 1–38.
[28] A. Goel, S. Bhatta, Use of design patterns in analogy-based design, Adv. Eng. Inform. 18 (2) (2004) 85–94.
[29] A. Goel, B. Chandrasekaran, Case-based design: a task analysis, in: C. Tong, D. Sriram (Eds.), Artificial Intelligence Approaches to Engineering Design, Volume II: Innovative Design, Academic Press, San Diego, CA, 1992, pp. 165–184.
[30] A. Goel, S. Bhatta, E. Stroulia, An early case-based design system, in: M. Maher, P. Pu (Eds.), Issues and Applications of Case-Based Reasoning, Erlbaum, 1992, pp. 87–132.
[31] R. Gregson, A comparative evaluation of seven similarity models, Br. J. Math. Stat. Psychol. 29 (2) (1976) 139–156.
[32] K. Hammond, Case-Based Planning: Viewing Planning as a Memory Task, Academic Press, San Diego, CA, 1987.
[33] D.R. Hofstadter, Analogy as the core of cognition, in: D. Gentner, K.J. Holyoak, B.N. Kokinov (Eds.), The Analogical Mind: Perspectives from Cognitive Science, MIT Press, Cambridge, MA, 2001, pp. 499–538.
[34] D.R. Hofstadter (Ed.), Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought, Basic Books, New York, 1996.
[35] D.R. Hofstadter, E. Sander, Surfaces and Essences: Analogy as the Fuel and Fire of Thinking, Basic Books, 2013.
[36] K.J. Holyoak, J.E. Hummel, Toward an understanding of analogy within a biological symbol system, in: The Analogical Mind: Perspectives from Cognitive Science, 2001, pp. 161–195.
[37] K.J. Holyoak, P. Thagard, Mental Leaps: Analogy in Creative Thought, MIT Press, Cambridge, MA, 1996.
[38] E. Hunt, Quote the Raven? Nevermore, in: L.W. Gregg (Ed.), Knowledge and Cognition, Erlbaum, Hillsdale, NJ, 1974, pp. 129–158.
[39] B. Indurkhya, On the role of interpretive analogy in learning, New Gener. Comput. 8 (4) (1991) 385–402.
[40] M.T. Keane, Analogical Problem Solving, Halsted Press, 1988.
[41] K. Kim, T.H. Chalidabhongse, D. Harwood, L. Davis, Real-time foreground–background segmentation using codebook model, Real-Time Imaging 11 (3) (2005) 172–185.
[42] B. Kokinov, A. Petrov, Integrating memory and reasoning in analogy-making: the AMBR model, in: D. Gentner, K.J. Holyoak, B.N. Kokinov (Eds.), The Analogical Mind: Perspectives from Cognitive Science, MIT Press, Cambridge, MA, 2001, pp. 59–124.
[43] J. Kolodner, Case-Based Reasoning, Morgan Kaufmann, San Mateo, CA, 1993.
[44] M. Kunda, Visual problem solving in autism, psychometrics, and AI: the case of the Raven's Progressive Matrices intelligence test, Doctoral dissertation, School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, 2013.
[45] M. Kunda, A. Goel, Thinking in pictures as a cognitive account of autism, J. Autism Dev. Disord. 41 (2011) 1157–1177.
[46] M. Kunda, K. McGreggor, A. Goel, Two visual strategies for solving the Raven's Progressive Matrices intelligence test, in: Proceedings of 25th AAAI Conference on Artificial Intelligence, 2011.
[47] M. Kunda, K. McGreggor, A. Goel, Reasoning on the Raven's Advanced Progressive Matrices test with iconic visual representations, in: Proceedings of 34th Annual Meeting of the Cognitive Science Society, Sapporo, Japan, 2012.
[48] M. Kunda, K. McGreggor, A. Goel, A computational model for solving problems from the Raven's Progressive Matrices intelligence test using iconic visual representations, Cogn. Syst. Res. 22–23 (2013) 47–66.
[49] D. Leake (Ed.), Case-Based Reasoning: Experiences, Lesson and Future Directions, MIT Press, Cambridge, MA, 1996.
[50] A. Lovett, K. Forbus, J. Usher, Analogy with qualitative spatial representations can simulate solving Raven's Progressive Matrices, in: Proceedings of the 29th Annual Conference of the Cognitive Society, Nashville, Tennessee, 2007.
[51] A. Lovett, K. Forbus, J. Usher, A structure-mapping model of Raven's Progressive Matrices, in: Proceedings of the 32nd Annual Conference of the Cognitive Science Society, Portland, Oregon, 2010, pp. 2761–2766.

[52] B.B. Mandelbrot, The Fractal Geometry of Nature, W.H. Freeman, San Francisco, CA, 1982.
[53] K. McGreggor, Fractal analogical reasoning, Doctoral Dissertation, School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, 2013.
[54] K. McGreggor, C.M. Yerga, D. Van Brink, Color processing system, U.S. Patent No. 5,963,201 U.S. Patent and Trademark Office, Washington, DC, 1999.
[55] K. McGreggor, M. Kunda, A. Goel, Fractal analogies: preliminary results from the Raven's test of intelligence, in: Proceedings of the International Conference on Computational Creativity, Mexico City, Mexico, 2011, pp. 27–29.
[56] M. Mitchell, Analogy Making as Perception: A Computer Model, MIT Press, Cambridge, MA, 1993.
[57] N. Nersessian, Creating Scientific Concepts, MIT Press, Cambridge, MA, 2008.
[58] H. Prade, G. Richard (Eds.), Computational Approaches to Analogical Reasoning: Current Trends, Springer, 2014.
[59] M. Ragni, P. Stahl, T. Fangmeier, Cognitive complexity in matrix reasoning tasks, in: Proceedings of the European Conference on Cognitive Science, NBU Press, Sofia, 2011.
[60] D. Rasmussen, C. Eliasmith, A neural model of rule generation in inductive reasoning, Top. Cogn. Sci. 3 (1) (2011) 140–153.
[61] J. Raven, J.C. Raven, J.H. Court, Manual for Raven's Progressive Matrices and Vocabulary Scales, Harcourt Assessment, San Antonio, TX, 2003.
[62] S. Ray, R.H. Turi, Determination of number of clusters in *k*-means clustering and application in colour image segmentation, in: Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques, 1999, pp. 137–143.
[63] C.K. Riesbeck, R.C. Schank, Inside Case-Based Reasoning, Lawrence Erlbaum Associates, Hillsdale, NJ, 1989.
[64] A. Schwering, U. Krumnack, K.U. Kühnberger, H. Gust, Using gestalt principles to compute analogies of geometric figures, in: Proceedings of the 19th Meeting of the Cognitive Science Society, 2007.
[65] B. Smyth, P. Cunningham, M. Keane, Hierarchical case-based reasoning integrating case-based and decompositional problem solving techniques, IEEE Trans. Knowl. Data Eng. 13 (2001) 792–812.
[66] I. Soulières, M. Dawson, F. Samson, E.B. Barbeau, C.P. Sahyoun, G.E. Strangman, T.A. Zeffiro, L. Mottron, Enhanced visual processing contributes to matrix reasoning in autism, Hum. Brain Mapp. 30 (12) (2009) 4082–4107.
[67] P. Thagard, K.J. Holyoak, G. Nelson, D. Gochfeld, Analog retrieval by constraint satisfaction, Artif. Intell. 46 (1990) 259–310.
[68] A. Tversky, Features of similarity, Psychol. Rev. 84 (4) (1977) 327–352.
[69] P. Yaner, A. Goel, Visual analogy: viewing retrieval and mapping as constraint satisfaction, Appl. Intell. 25 (1) (2006) 91–105.
[70] P. Yaner, A. Goel, Analogical recognition of shape and structure in design drawings, Artif. Intell. Eng. Des. Anal. Manuf. 22 (2) (2008) 117–128.
[71] S.C. Zhu, A. Yuille, Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 18 (9) (1996) 884–900.