Keith McGreggor, Maithilee Kunda, & Ashok Goel. A Fractal Approach Towards Visual Analogy. In *Proceedings of the First International Conference on Computational Creativity*, Lisbon, Portugal, January 9-11, 2010.

A Fractal Approach Towards Visual Analogy

Keith McGreggor, Maithilee Kunda & Ashok Goel

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

Abstract. We present a preliminary computational model of visual analogy that uses fractal image representations that rely only on the grayscale pixel values of input images, and are mathematical abstractions quite rigorously grounded in the theory of fractal image compression. We have applied this model of visual analogy to problems from the Raven's Progressive Matrices intelligence test, and we describe in detail the fractal solution strategy as well as some preliminary results. Finally, we discuss the implications of using these fractal representations for memory recall and analogical reasoning.

1 Cognition, Computation, and Creativity

We may study creative behavior at many levels of aggregation and abstraction ranging from environmental and genetic to neural and cognitive to social and cultural. Our work on computational creativity focuses on the cognitive level. Although at present there is little agreement in the cognitive sciences about the proper characterization of creativity, nevertheless there is broad and deep consensus that analogy is a core cognitive process of creative behavior, e.g. [1]–[8]. Indeed, some cognitive scientists have argued that analogy is a core process not only of creativity, but all cognition, including perception [6][9]. Thus, understanding the computational processes of analogy seems critical to understanding and developing computational models of creativity.

We may classify computational theories of analogy into two broad categories. In one category are theories that propose general-purpose mechanisms for mapping and transfer of relations from one problem to another, e.g. production systems [10], structure mapping [11][12], schema induction [13], and constraint satisfaction [14]. These *mechanism theories* typically make few commitments about the contents of knowledge; indeed, their mechanisms are general-purpose because they are agnostic towards knowledge contents. In the other category are computational theories that describe contents of knowledge that drive analogies in creative tasks such as story understanding [15], diagram understanding [16][17], scientific problem solving [18][19], innovative design [20][21][22], and calligraphy [23]. These *content theories* of analogy, too, describe computational processes, but their processes are driven by the contents of knowledge (and corresponding vocabularies for capturing the knowledge contents). The extant content and mechanism computational theories of analogy nevertheless are united in their use of *propositional representations*: From the perspective of the content theories of analogy, the mechanism theories offer potential substrates for computational implementation.

We present a preliminary computational theory of visual analogy that is fundamentally different from both content and mechanism theories because it uses *fractal representations* instead of propositional ones. Visual analogy is a topic of longstanding interest in computational creativity because of its central role in many creative tasks such as intelligence tests [24]. In particular, in this paper, we focus on geometric analogy problems that are a type of visual analogy problems: geometric analogy problems focus solely on shapes, sizes and locations of the shapes, and spatial relations among the shapes. Fractal image representations rely only on the grayscale pixel values of input images, and are mathematical abstractions quite rigorously grounded in the theory of fractal image compression [25].

Below, first we describe how fractal representations can be used for computing features and similarity among distinct images, and how these operations can be combined to form a technique for creating visual analogies using purely pictorial inputs. Second, we present initial experimental results from using this method on geometric analogy problems that occur on the standardized intelligence test called the Raven's Progressive Matrices test [26]. Third, we discuss how this fractal model of visual analogy may be considered to be a mechanism for creativity.

2 Fractal Image Representation

Consider the general form of an analogy problem as:

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

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

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

2.1 Mathematical Basis

The mathematical derivation of fractal image representation expressly depends upon the notion of real world images, i.e. images that are two dimensional and continuous [25]. A key observation is that all naturally occurring images we perceive appear to have similar, repeating patterns. Another observation is that no matter how closely you examine the real world, you find instances of similar structures and repeating patterns. These observations suggest that it is possible to describe the real world in terms other than those of shapes or traditional graphical elements—in particular, terms which capture the observed similarity and repetition alone.

Computationally, to determine the fractal representation of an image requires the use of the fractal encoding algorithm. The collage theorem [25] at the heart of the fractal encoding algorithm can be stated concisely:

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

We now shall present the fractal encoding algorithm in detail.

2.2 The Fractal Encoding Algorithm

Given an image *D*, the fractal encoding algorithm seeks to discover the set of transformations *T*. The algorithm is considered "fractal" for two reasons: first, the affine transformations chosen are generally contractive, which leads to convergence, and second, the convergence of *S* into *D* can be shown to be the mathematical equivalent of considering *D* to be an attractor [25].

Here are the general steps for encoding an image *D* in terms of another image *S*:

1) Decompose *D* into a set of *N* smaller images $\{d_1, d_2, d_3, ..., d_n\}$. These individual images are sets of points.
2) For each image d_i:
    a) Examine the entire source image *S* for an equivalent image s_i such that an affine transformation of s_i will result in d_i. This affine transformation will be a 3x3 matrix, as the points within s_i and d_i under consideration can be represented as the 3-D vector $<x, y, c>$ where *c* is the (grayscale) color of the 2-D point $<x,y>$. Collect all such transforms into a set of candidates *C*.
    b) Select from the set of candidates that transform which most minimally achieves its work, according to some predetermined and consistent metric.
    c) Let T_i be the representation of the chosen affine transformation of s_i into d_i.
3) The set $T = \{T_1, T_2, T_3, ..., T_n\}$ is the fractal encoding of the image *D*.

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

2.3 Searching and Encoding

The search of the source image *S* for a matching fragment is exhaustive, in that each possible correspondence s_i is considered regardless of its prior use in other discovered transforms. Also, for each potential correspondence, each transformation under a restricted set of affine, similitude transformations is considered. Our implementation presently examines each potential under identity (*I*), horizontal (*HF*) and vertical (*VF*) reflections, and 90° (*R90*), 180° (*R180*), and 270° (*R270*) rotational transformations.

The metric we employ to evaluate candidate transformations prefers those transformations which induce the least translation, rotation or reflection, and color manipulation. We use this to ensure that areas which are aligned in a Cartesian sense are chosen. We shall revisit this important metric in a later section.

Once a transformation has been chosen, we construct a compact representation of it, called a fractal code. A fractal code T_i is a 3-tuple, $\langle \langle s_x, s_y \rangle, \langle d_x, d_y \rangle, k, c \rangle$, where $\langle s_x, s_y \rangle$ is the location of the leftmost and topmost pixel in s_i; $\langle d_x, d_y \rangle$ is the location of the leftmost and topmost pixel in d_i; $k \in \{ I, HF, VF, R90, R180, R270 \}$ indicates which affine transformation is to be used; and $c \in [-255, 255]$ indicates the overall color shift to be added uniformly to all elements in the block.

Note that the choice of source image *S* is arbitrary. Indeed, the image *D* can be fractally encoded in terms of itself, by substituting *D* for *S* in the algorithm. Although one might expect that this substitution would result in a trivial encoding (wherein all of the chosen fractal codes correspond to an identity transform), in practice this is not the case, for we want a fractal encoding of *D* to converge upon *D* regardless of chosen initial image. For this reason, the size of source fragments considered is taken to be twice the dimensional size of the destination fragment, resulting in a contractive affine transform. Similarly, color shifts are made to contract.

The fractal encoding algorithm, while computationally expensive in its exhaustive search, transforms a real world image into a much smaller set of fractal codes, which form, in essence, an instruction set for reconstituting the image. This resulting fractal representation of an image *D* vis-à-vis another given image forms the basis of our investigation into solutions for visual analogy problems.

2.4 Determining Fractal Features

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

In our present implementation, each fractal code $\langle \langle s_x, s_y \rangle, \langle d_x, d_y \rangle, k, c \rangle$ yields a small set of features, generally formed by constructing subsets of the tuple:

$\langle \langle s_x, s_y \rangle, \langle d_x, d_y \rangle, k, c \rangle$:	a specific feature;
$\langle \langle d_x - s_x, d_y - s_y \rangle, k, c \rangle$:	a position agnostic feature;
$\langle \langle s_x, s_y \rangle, \langle d_x, d_y \rangle, c \rangle$:	an affine transform agnostic feature;
$\langle \langle s_x, s_y \rangle, \langle d_x, d_y \rangle, k \rangle$:	a color agnostic feature;
$\langle k, c \rangle$:	an affine specific feature;
$\langle c \rangle$:	a color shift specific feature.

As shown, the features derived may be categorized into either specific or agnostic aspects. Our implementation of the fractal encoding algorithm employs a metric which minimizes the transformative work (either geometrically or colorimetrically) expressly due to our desire to exploit specific feature matches.

With this basis of fractal representation and feature discovery, we now may address the problem of visual analogy.

2.5 A Fractal Process of Visual Analogy

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

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

This metric reflects similarity as a comparison of the number of fractal features shared between candidate pairs taken in contrast to the joint number of fractal features found in each pair member [27]. In our present implementation, the measure of similarity *S* between the candidate transform *T′* and the target transform *T* is calculated using the following formula, also known as the ratio model:

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

where $f(X)$ is the number of features in the set X. For our initial work, we have chosen values of $\alpha = \beta = 1.0$, which, according to [27], results in this simplification:

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

The final solution is taken as the candidate image from memory that results in the highest measured similarity according to this measure.

3 The Raven’s Progressive Matrices Test

In this section, we describe the Raven’s Progressive Matrices test and present some preliminary results for a solution algorithm that uses the fractal algorithm for visual analogy described in the previous section.

The Raven's Progressive Matrices (RPM) test is a standardized intelligence test that consists of visually presented, geometric-analogy-like problems in which a matrix of geometric figures is presented with one entry missing, and the correct missing entry must be selected from a set of answer choices. Figure 1 shows an example of a problem that is similar to one of the problems in the Standard Progressive Matrices (SPM).

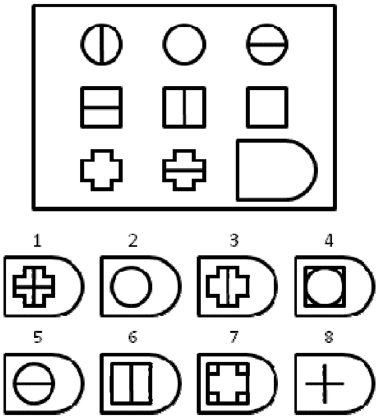

Fig. 1. Example problem similar to one in the Standard Progressive Matrices (SPM) test.

Although the test is supposed to measure only eductive ability, or the ability to extract and understand information from a complex situation [26], the RPM's high level of correlation with other multi-domain intelligence tests have given it a position of centrality in the space of psychometric measures [28], and it is therefore often used as a test of general intelligence.

Despite its widespread use, neither the computational nor the cognitive characteristics of the process of solving the RPM are well understood. Hunt 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 representations and perceptually based operations, and "Analytic," which uses feature-based representations and logical operations [29].

Existing AI systems for problem solving on the RPM, in contrast to Hunt's early work, use propositional representations [30][31]. Carpenter, Just, and Shell describe a computational model that simulates solving RPM problems using propositional representations [30]. Their model is based on the traditional production system architecture, with a long-term memory containing a set of 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 a 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. Lovett, Forbus, and Usher describe a model that extracts qualitative spatial representations from visually segmented representations of RPM problem inputs and then uses the technique of structure mapping to find solutions [31]. While, as we mentioned earlier, production systems and structure mapping offer alternative mechanisms for implementing content theories of analogies, from the perspective of our work on fractal representations, the two are similar in that both use propositional representations.

3.1 Preliminary Results from the Fractal Method of Visual Analogy

A RPM problem can be viewed as a sequence of images (ordered in rows and columns), where some unknown transformation *T* can be said to transform one image into a corresponding adjacent image. In a typical 2x2 RPM problem, there are four such transformations, as shown in Figure 2. (RPM problems can also have three-by-three matrices, which we do not address in this paper.)

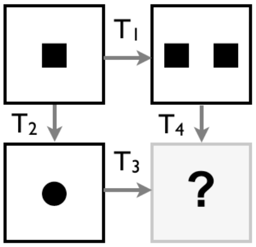

Fig. 2. Illustration of four image transformations implicit in a 2x2 RPM problem matrix.

RPM problems are formulated to suggest that these transformations are pairwise analogous (i.e. the two row transformations are analogous to one another). A 2x2 RPM problem generally has six candidate solutions presented. Following the algorithm for visual analogy described above, we seek to solve an RPM problem by determining which of the candidate solutions yields the most analogous transformations. In particular, we perform the recall and similarity calculation described in Section 2.3 independently over all row and column relationships present in the RPM problem. The final solution is taken as the candidate that results in the highest measured similarity for any such relationships.

As an example, we shall use the "arrow" problem shown in Figure 3. Also, while the complete fractal method examines each of the problem's analogies, we shall restrict this detailed discussion to just one of these transformations (T_1, as labeled in Figure 2).

The initial transformation T_1 is the fractal encoding of the transformation from the upper left arrow into the upper right arrow. When encoded using a block size of 32 x 32 pixels, this encoding generates 39 distinct fractal features.

Each candidate answer is encoded likewise, from the upper left arrow into the candidate, each resulting in between 27 and 45 distinct features. For the arrow problem, using a 32 x 32 block size, the similarity measures for each answer C_i are:

$S(T,C_1) = 21 / (21+18+24) \cong 0.333333$
$S(T,C_2) = 15 / (15+24+30) \cong 0.217391$
$S(T,C_3) = 16 / (16+23+11) \cong 0.32$
$S(T,C_4) = 14 / (14+25+29) \cong 0.205882$

The answer with the highest calculated similarity is deemed correct. Therefore, the fractal method chooses as its answer #1.

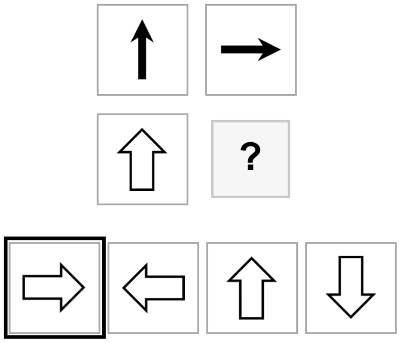

Fig. 3. The “arrow” problem, in the format of a 2x2 problem matrix with four answer choices.

4 Implications of Fractal Representations for Analogical Reasoning and Memory Recall

Benoit Mandelbrot coined the term “fractal” from the Latin adjective *fractus* and its corresponding verb (*frangere*, which means “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 [32]. The approach we outline in this paper seeks to define a class of visual representations in a similar, fractal manner, where images are taken as encountered and transformed into a fractal representation either with regard to other images or to themselves.

The very nature of this fractal representation places an immediate emphasis on similarity discovery. These discoveries are used to advantage in solving problems that place equivalent emphasis on visual analogy, such as the Raven’s Progressive Matrices test. While the use of fractal representation is important, the emphasis upon visual recall in our solution afforded by features derived from those representations bears further discussion.

The importance of individual discovered features presents itself strongly in our calculation of the similarity metric from [27]. The weights α and β may be skewed toward or away from equality in order to favor feature inclusion or exclusion. We note that while a 2x2 RPM problem offers little opportunity for determining feature importance, the class of RPM problems which are 3x3 in nature require it, for the additional row and column provide an in-domain manner in which to report visual evidence for the kinds of features to expect within the eventual successful candidate. Using this additional evidence would allow for the weights of feature inclusion (α) and exclusion (β) to become covariant with features. We are actively exploring this as we prosecute solutions for 3x3 RPM problems.

We take the position that placing candidate transformations into memory, indexed via those discovered fractal features, affords a new method of discovering image similarity. Indeed, this method of solving the RPM, by being reminded of similar transformations, bears close kinship to various methods and theories of case-based reasoning [33], although we exploit recalled transformations toward a significantly different end. That images, encoded either in terms of themselves or other images, may be indexed and retrieved without regard to shape, geometry, or symbol, suggests that the fractal representation bears further exploration not only as regards solutions to problems akin to the RPM, but also to those of general visual recall and analogy based creativity.

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

References

1. Christensen, B. T., & Schunn, C. D. The relationship of analogical distance to analogical function and pre-inventive structure: The case of engineering design. Memory & Cognition, 35(1), 29-38. (2007)
2. Clement, J. Creative Model Construction in Scientists and Students: The Role of Imagery, Analogy, and Mental Simulation. Dordrecht: Springer. (2008)
3. Darden, L. (ed). Reasoning in Biological Discoveries. Cambridge University Press. (2006)
4. Dunbar, K. The Analogical Paradox. In Gentner, D., Holyoak, K.J., & Kokinov, B.N. (Eds.) The Analogical Mind: Perspectives from Cognitive Science, MIT Press. (2001)
5. Goel, A. K. Design, Analogy, and Creativity. IEEE Expert 12(3): 62-70. (1997)
6. Hofstadter, D. Godel, Escher, Bach: An Eternal Golden Braid. NY: Basic Books. (1979)
7. Holyoak, K., & Thagard, P. Mental Leaps: Analogy in Creative Thought. Cambridge, MA: MIT Press. (1995)
8. Nersessian, N.J. Creating Scientific Concepts. Cambridge, MA: MIT Press. (2008)
9. Hofstadter, D. (editor). Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought. NY: Basic Books. (1995)
10. Anderson, J.R., & Thompson, R. Use of Analogy in a Production System Architecture. In Vosniadou, S. & Ortony, A. (Eds.), Similarity and analogical reasoning, pp. 267-297, London: Cambridge University Press. (1989)
11. Gentner, D. Structure Mapping. Cognitive Science, 7: 155-170. (1983)
12. Falkenhainer, B., Forbus, K., and Gentner, D. The Structure-Mapping Engine: Algorithms and Examples. Artificial Intelligence, 41-1:63. (1989)
13. Gick, M., & Holyoak, K.J. Schema Induction and Analogical Transfer. Cognitive Psychology, 15(1):1-38. (1983)

14. Holyoak, K., & Thagard, P. Analogical Mapping by Constraint Satisfaction. Cognitive Science, 13: 295-355. (1989)

15. Winston, P. Learning & Reasoning by Analogy. CACM, 23(12): 689-703. (1979)

16. Gross, M., & Do, E. Drawing on the Back of an Envelope. Computer Graphics and Applications. (2000)

17. Yaner, P., & Goel, A.K. Understanding Drawings by Compositional Analogy. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI-2007),* Hyderabad, India. January 2007, pp. 1131-1137. (2007)

18. Davies, J., Nersessian, N., & Goel, A.K. Visual Models in Analogical Problem Solving. Foundations of Science, 10(1):133-152, 2005. (2005)

19. Griffith, T., Nersessian, N.. & Goel, A.K. Function-follows-Form: Generative Modeling in Scientific Reasoning. In Proc. 22nd Cognitive Science Conference. (2000)

20. Casakin, H., & Goldschmidt, G. Expertise and the Use of Visual Analogy: Implications for Design Education. Design Studies, 20(2): 153-179, 1999.

21. Goel, A. K., & Bhatta, S. Use of Design Patterns in Analogy-Based Design. Advanced Engineering Informatics, 18(2):85-94. (2004)

22. Davies, J., Goel, A.K. & Nersessian, N. A Computational Model of Visual Analogies in Design. Cognitive Systems Research, 10:205-215. (2009)

23. Hofstadter, D., & MacGraw, G. LetterSpirit. In Fluid Concepts and Creative Analogies, Hofstadter (editor), NY: Basic Books. (1995)

24. Evans, T. G. A program for the solution of a class of geometric-analogy intelligence-test questions. Semantic Information Processing, pp. 271–353, MIT Press. (1968)

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

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

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

28. Snow, R., Kyllonen, P., & Marshalek, B. The topography of ability and learning correlations. In Sternberg, R. ed. Advances in the Psychology of Human Intelligence 2: 47-103. Hillsdale, NJ: Erlbaum. (1984)

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

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

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

32. Mandelbrot, B. The Fractal Geometry of Nature. San Francisco: W.H. Freeman. (1982)

33. Kolodner, J.L. Case-Based Reasoning. CA: Morgan Kaufmann Publishers. (1993)