AI Agent to Solve 2x1 and 2x2 Visual Analogy Problems in Image Format

Magahet Mendiola
(mmendiola3@mail.gatech.edu)

March 19th, 2015

Contents

describing the way your algorithm works, its relative strengths, its relative weaknesses, its efficiency, its relationship with human cognition, and the different challenges and opportunities in solving problems visually rather than verbally.

In addition to your agent, you should also write and submit a design report of roughly 1500 words. This design report should do a number of things. First, it should describe the reasoning your agent uses. How does it work? Second, it should describe how the agent comes to some of its correct answers. Third, it should describe why your agent makes some of the mistakes it does. Fourth, it should describe what could be done to improve the agent if you had more time, resources, or processing power. Fifth, it should describe the unique challenges and opportunities in visual reasoning compared to verbal reasoning. Your design report can include diagrams in addition to the ~1500 words.

Introduction

We will describe the design of an AI agent built to solve 2x1 and 2x2 visual analogy problems given images rather than object descriptions. This will include details on the agent's strengths, weaknesses, efficiency, and relationship with human cognition.

Visual vs. Propositional Approaches

Given visual input rather than descriptive data provides both challenges and opportunities for our AI agent. On one hand, propositional reasoning is made more difficult as we would first have to translate pixel-based information into a descriptive model of object types, sizes, and inter-relationships.

On the other hand, visual input allows us to broaden our agent's approach to understanding the inter-figure relationships and simplifies the application of transformations that should be applied to the entire figure. For example, figure 1 shows a difficult problem to solve with propositional reasoning. First, we would have to reason that the rotation of the single triangle in frames A and B should be applied to all three triangles in C. Second, we would need to model how rotating all three triangles would affect their spatial relationships. Third, we would have to address the correspondence problem between comparing triangles in frame C and the answer frames.

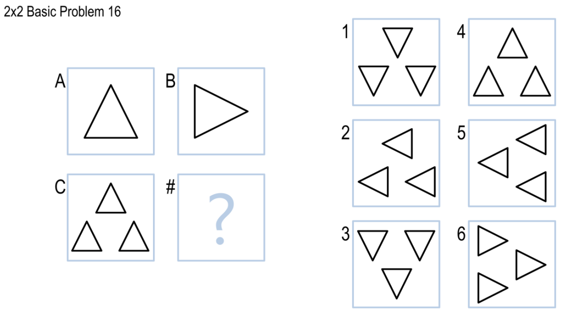

Figure 1: 2x2BasicProblem16

These complexities are avoided altogether with a visual approach. Once we observe that the transformation from A to B is a 90° rotation, it is trivial to apply the same rotation to frame C and compare the result to the answer choices. There is no need to teach the agent the concepts of triangles, changes in spatial relationships, or object correspondence.

The visual approach, for this example, seems to match much more closely to human cognition. We observe the rotation in the example frames and apply it, effortlessly, to frame C. We do not need to enumerate the spatial relationship dynamics between the three triangles in order to rotate the image.

This example illustrates the comparative strengths of the visual approach to solving RPMs. In the following sections, we will examine a visual heuristic algorithm that solves better than half of the provided RPM test cases with simplistic reasoning and no prior understanding of shapes, spatial relationships, or transformation models.

Visual Heuristic and Answer Selection

Our agent performs the following steps for each problem:

1. Load each figure as bilevel (black and white) images.
2. Create a vector for each image of the black pixel count in each quadrant (top-left, top-right, bottom-left, bottom-right).
3. Get the pairwise difference and absolute difference between example source and destination frame vectors.
4. Get the pairwise difference between the target frame and each answer choice frame vectors.
5. Select the answer frame with the smallest euclidean distance (l-2 norm) between the example and candidate transition vectors.

Example

Figure 2 illustrates our agent's heuristic algorithm.

The process of quantifying the visual data is shown in the following output. The first vector represents the black pixel change from frame A to B. Each vector after that shows the black pixel change from frame C to each answer choice. The following two numbers are the distance from the A:B change vector to the C:X transition vector. The first number is the distance between the signed vectors and the second is the distance between the unsigned vectors. Having both signed and unsigned changes is used in certain edge cases, including this example. We see from the figure that although a given quadrant in frame C should have the same number of pixels change, the color change is opposite that of the A to B change.

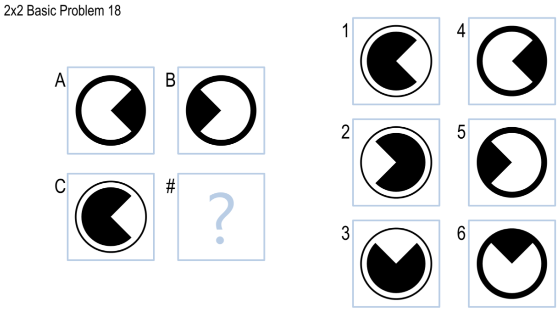

Figure 2: 2x2BasicProblem18

```
2x2 Basic Problem 18
t [ 1588 -1558  1678 -1644]
1 [-45  33  16  22] 3276 3177
3 [-1619    24   -13  1616] 5125 2264
2 [-1637  1597 -1597  1657] 6478 103
5 [-684 -626 -576 -631] 3483 1981
4 [-2271   978 -2244   966] 6596 1258
6 [ -646   944 -2240  -646] 5253 1605
```

We see that the A to B transition is quantified in this way:

```
# of pixels changed
top-left quadrant: 1588
top-right quadrant: -1558
bottom-left: 1678
bottom-right: -1644
```

From this data we can observe that the top two quadrants have both changed approximately the same number of pixels. The same is true of the bottom two quadrants. Intuitively, we interpret this as a horizontal reflection. Quantitatively, we see from the output that the lowest distance was found between the absolute differences between frame C and frame 2.

Performance

Ability to Solve RPMs

The quadrant-based pixel change heuristic was able to solve 22 of 40 2x1 and 2x2 basic test questions. It performed very well on problems with whole frame rotations and reflections. It performed poorly on problems where

A number of visual methods for solving the RPM problem sets were tested along with the final heuristic method described here. The results of these experiments were mixed, but overall proved to be less successful. Surprisingly, the Affine method of finding and applying image transformations performed rather poorly: 9/40 on Basic Problems. Other visual heuristic were tested as well:

- Pixel count change between figures.
- Pixel change ratio between figures.
- Key point (corners) count change between figures.
- Key point change ratio between figures.
- Key point change within each quadrant between figures.
- Pixel count change within smaller image subdivisions (16 to 256 sub-regions).

Along with each of these heuristics, combinations of each were also attempted. This was accomplished by having each method rank the answer choices and various election methods were used to form a consensus. In the end, the combination of the two quadrant-based pixel change metrics provided the best method for identifying correct answers.

Computational Complexity

The agent ran with minimal resource usage and in constant time for a given problem type. Since each problem does a finite and fixed number of computations, there is no variance in the runtime between problems. This makes this basic visual heuristic method of solving RPMs quite efficient compared to propositional methods which quickly grow in complexity as you consider correspondence combinations between various frames.