AI Agent to Solve 2x1, 2x2, and 3x3 Visual Analogy Problems in Image Format

Magahet Mendiola
(mmendiola3@mail.gatech.edu)

April 12th, 2015

Contents

Introduction

We will explore the design of an AI agent built to solve 2x1, 2x2, 3x3 visual analogy problems given images rather than object descriptions. This will include details on the agent's strengths, weaknesses, efficiency, and the how it's methodology compares to that of human cognition.

In the spirit of full disclosure, this report includes much of the original content from the design report for our previous RPM agent. The agent's core problem solving methodology has not changed. Significant changes include experimentation into advanced voting methods and applying our visual approach to 3x3 problems.

Visual vs. Propositional Approaches

Processing visual input rather than descriptive data provides both challenges and opportunities for our agent. On one hand, propositional reasoning is made more difficult, as we 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.

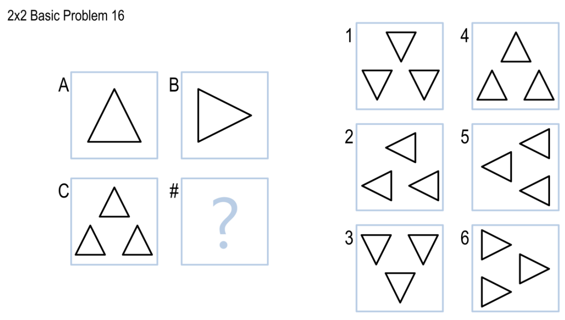

Figure 1: 2x2BasicProblem16

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 than the propositional approach. 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

We will use the problem in figure 2 to illustrate our agent's heuristic algorithm.

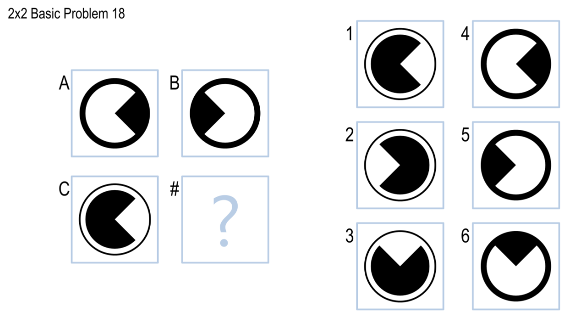

Figure 2: 2x2BasicProblem18

The process of quantifying the visual data is shown in the following output:

```
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
```

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 two numbers that follow these vectors 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 useful 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.

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 9/20 2x1, 13/20 2x2, and 14 3x3 basic test questions. It performed very well on problems with whole frame rotations and reflections; problems where object spatial relationships changed between frames; and problem where objects were added and removed between frames. It performed well across a variety of problems that were quite difficult to solve using propositional methods, including our first example in figure 1.

Interestingly, our agent performed poorly on fairly simple problems including the one in figure 3.

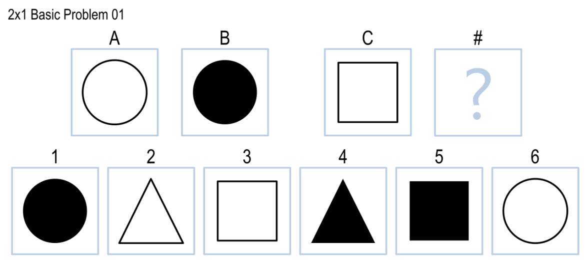

Figure 3: 2x1BasicProblem01

In this instance, and a few others, the fact that we compare the un-normalized vectors confused the agent into choosing answers with the closest match in terms of exact pixel changes in each quadrant. In this example, we would have been better served by comparing the proportional change in pixels rather than the pixel change count. However, experiments to add proportional change metrics resulted in a decrease in correctly chosen answers, even when used in combination with total change metrics.

It remains an area of improvement to find a process to utilize proportional change metrics, or even completely different visual methods, on problems our agent knows the default method would perform poorly on. For this specific problem, we may have been able to decide to use proportional metrics based on the fact that each quadrant's pixel count and change are symmetric. We could add a rule that in such cases, the agent should utilize a reasoning method with a better chance at success.

In cases where our agent was unable to find a truly distinct answer choice, it was still able to narrow the candidate solutions to just a few. In these instances, a randomized guess among two answer choices has a much higher probability of success. This highlights the importance of utilizing multiple combined methods to rank solutions and collectively make decisions. It also illustrates another connection to human reasoning. Our agent does not get stuck if it cannot find a perfect match. It is robust and will, as a human would, venture an educated guess after rejecting obviously incorrect answers.

Alternative Visual Methods

A number of visual methods for solving the RPM problem sets were tested along with the final heuristic method described above. The results of these experiments were mixed, but overall proved to be less successful. We'll explore these alternative approaches in the following sections.

Affine Surprisingly, the Affine method of finding and applying image transformations performed rather poorly. Our agent was able to solve 9/40 on Basic Problems and very few challenge questions. The agent performed the following process:

```
for transform in [rotations, mirror, flip]:
    apply transform to frame A
    compute similarity(transform(A), B)
choose best transform based on similarity
for answer in choices:
    compare transform(C) to answer
choose best answer based on similarity
```

This method was further refined to rank answers found by finding and applying transforms both horizontally and vertically in the RPM grid and form a consensus on an answer. However, this method continued to get led astray by problems such as in figure 4.

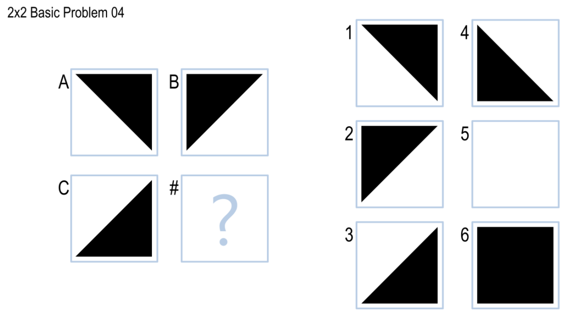

Figure 4: 2x2BasicProblem04

Here, the agent would find a rotation or would flip the image, leading it to choose answers 1 and 2. We can see the Affine algorithm run here:

```
2x2 Basic Problem 04

# transforms and resulting S(T(A), B) scores
rotate 270 0.405911635671
rotate 90 0.982131328488
flip 0.405600189036
no change 0.648440341549
mirror 0.982069410748
rotate 180 0.645021345732

# best transform
rotate 90 0.982131328488
```

```
# S(T(C), X) scores for each answer choice
1 0.981612994321
3 0.64950401923
2 0.638670997616
5 0.700274692817
4 0.40145764414
6 0.553490128611
```

We see in this example that each affine transformation was applied and both rotate90 (counterclockwise) and mirror resulted in figures with very high similarity scores to the example destination frame. Rotate90 was chosen, as it had a slightly higher match. This transform was then applied to frame C. Then similarity scores were computed for each candidate answer. We see from the result that Answer 1 provided the closest match.

Although the procedure was valid and resulted in a strong answer choice, the agent incorrectly identified the initial transition as rotate90 instead of mirror. In contrast, our quadrant-based pixel change heuristic method had no trouble discerning the frame relationship in this problem. To improve this method in the future, we could assign preferences to each transform and set a threshold for how much better a match a lower ranked transform would have to be to consider it the better choice.

Other Visual Heuristics Other visual heuristic were tested as well. These are some of the methods tested:

- 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).

Voting/Bagging 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.

In addition to testing voting among multiple heuristics methods, training was also done against combinations of frame relationships in 3x3 problems. This was done to find the optimal set of frame relationships to use in creation of the voting pool. The following illustrates this experimentation:

```
Each relationship tuple is defined as:
    (example source, example destination, target)
    # example:
    # get A to C change and compare that to H to X change
    # (A, C, H)

For each combination of relationship tuples:
    # example: ((A, C, H), (A, H, C), (A, E, E))
    Get ranked answers based on each tuple
    Use first consensus election to select a final answer
```

This process was performed on every possible combination of relationship tuples. The set of tuples that resulted in the highest number of correct answers was the following:

```
transition_sets = (
    ('A', 'C', 'G'),
    ('D', 'F', 'G'),
    ('E', 'F', 'H'),
    ('B', 'H', 'C'),
    ('E', 'H', 'F'),
    ('A', 'E', 'E'),
)
```

Used in combination, votes from each of these frame relationships resulted in an improvement of 50% over a single relationship comparison.

2x2 Problems only have two such relationship tuples. Although (A, B, C) and (A, C, B) tuples resulted in the same number of correct answers, our agent uses them both in voting with the expectation that accuracy on untested problems may be improved.

Computational Complexity

Our agent runs 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 the basic visual heuristic method of solving RPMs quite efficient compared to propositional methods, which can quickly grow in complexity as you consider correspondence combinations between various frames.

Another performance benefit to a strictly visual method is that translation to propositional models is not required. This saves our agent from having to perform complex object detection and spatial reasoning. Although some of the visual methods require image transforms, the simple heuristic method that performed best in this experiment required no transformations or processing of any kind. This reduced the computational complexity of our agent to simply counting pixels and performing basic arithmetic.

The only additional computation was introduced with voting based on the different frame relationships comparisons. In 2x2 problems, this doubles the running time. In 3x3 problems, running time is six times longer. However, running time for all 87 provided problems still completes in under three seconds. This is orders of magnitude faster than the previous version of our RPM agent that solved problems using propositional methods.

Conclusion

The process of developing a visual heuristic based RPM agent has highlighted the robust and elegant way in which humans approach visual analogy problems. We observed that with simplistic methods and little prior understanding of object types and relationships, we were able to mimic human intuition in developing visual relational models.

We've seen that multiple heuristics are utilized in human reasoning and how we can implement metacognition into our agent to improve flexibility across diverse problem types. Our experiments illustrate the need for strong consensus development and tie-breaking processes to insure that the best reasoning method is being applied to a given problem.

Finally, we observed the efficiency and effectiveness in utilizing visual heuristic methods over complex propositional modeling and reasoning. The simplicity, robustness, and speed of our agent's approach makes solving RPMs with visual heuristics a promising choice for future experiments.