CS7637 Project 2 Reflection

Chenxi Gao
cga071@gatech.edu

INTRODUCTION:

Before I started project 2, I planned to use a similar approach in Project 1, assuming the problems can be solved mostly by simple transformations (e.g., unchanged, rotation, and reflection). However, after I closely examine multiple problems in Problem Set C, I quickly realized that solving 3x3 RPMs is clearly more complicated than solving 2x2 RPMs because of the increasingly many cases our agent has to consider other than simple transformations. For example, some problems (e.g., Basic C-03) requires our agent to be able to count the number of shapes in a figure.

Unlike solving 2x2 RPMs, where our agent only needs to look for the type of transformation from A to B or A to C, then applying the same transformation to find a match, the amount of cases it needs to compare grows exponentially. Unlike solving 2x2 RPMs, the following are the requirements and expectations that our agent needs to accomplish in this project.

1. Our agent needs to find the relationships between each row, each column, and the diagonal if necessary.
2. Because it is not practical to compare all the figures in the same row/column at a glance, our agent needs to find the relationships between many couples of images. For example, our agent needs first to compare A and B, B and C, and A and C in the first row to get a result.
3. Without checking the diagonal entries, the total number of comparisons is 3x3x2 = 18 without counting the number of comparisons when finding the answer in the answer set. This is computationally much more expensive than solving the 2x2 RPMs, which is 2x2 = 4.

To save computation time, I assume that it is not necessary to compare all rows and columns. In most cases, if one relation is found in a row or column, we can apply the same relation to obtain the answer. I will also discuss later the pros and cons of this approach. Below are the journals that include my detailed implementation of this project.

Because Project 3 requires an entirely visual approach to solve the 3x3 RPMs, I decided to solve the problems visually as preparation for Project 3. In the journals and the conclusion, I will explain the advantages and disadvantages of using a purely visual approach.

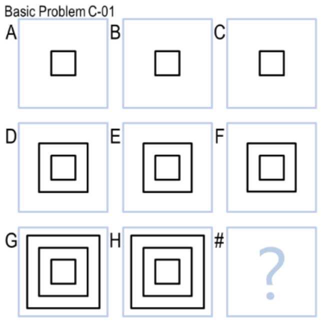
JOURNAL 1:

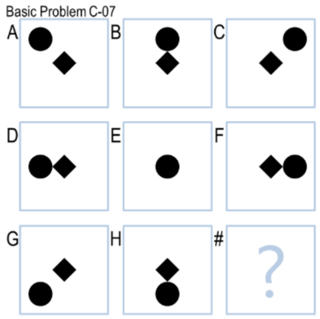
This is my first submission at 12:13 p.m., 10/22. Below is a summary of how I made some changes to Project 1 to adapt Project 2.

In Project 1, simple transformation (unchanged, reflection, rotation) is a rule of thumb to solve most problems. In project 2, it is still practical to solve some problems shown below. For example, in Basic Problem C-01, all pictures in the same row are identical. In Basic Problem C-07, the pictures in the third row (or column) can be viewed as the reflection of images in the first row (or column). I borrow the code of simple transformation from project 1. However, the bigger challenge is I have to modify the method of comparing images to find relations.

I managed my agent to compare AC and BC in the same row or AG and DG in the same column. If my agent finds the simple transformation between the two images, my agent will apply the same transformation to G or C, respectively, and then try to find a match in the answer set.

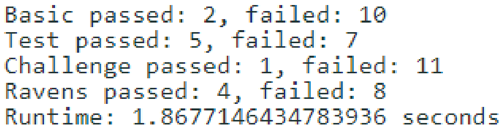
I think it is unnecessary to compare the relationship between AB in the first row or AD in the first column because finding such a relation does not affect my agent's ability to choose the correct answer. For instance, imagine that B is obtained by a rotation of 90 degrees clockwise, and C is obtained by vertically reflecting B. We do not need to apply the same transformation from G to H because H is given to us in the problem. We only have to apply the relation to H, a vertical reflection, and then fetch the answer in the answer set.





In addition, my agent skips comparing the second row and column to save computation time. It turns out that skipping does not affect my agent performance to get the correct answer.

Same as I have expected, my agent answered two answers correctly in the Basic Problems. Surprisingly, it answered five correctly in Test problems and four correctly in Raven problems. I suspect that the test may contain more examples of simple transformations, or my agent was just lucky to guess them correctly. Additionally, the overall runtime is less than two seconds. A summary is provided below:



```
Basic passed: 2, failed: 10
Test passed: 5, failed: 7
Challenge passed: 1, failed: 11
Ravens passed: 4, failed: 8
Runtime: 1.8677146434783936 seconds
```

Comparing my agent's performance to what I would generally solve 3x3 RPMs of simple transformations, I think:

1. It is similar to my conclusion in Project 1 that the way of finding and applying transformation is close to humans.
2. The method of comparing figures in 3x3 RPMs is similar to what I would do. For example, if I can find a relation in the first row, I would skip the second row and directly apply the same relation in the third row to get the answer. If it failed, I would try to find other relations.
3. The way of solving Basic Problem C-07 is slightly different. As humans, I would quickly choose answer 2 because it seems to complete the image set. Because my agent cannot view the complete problem simultaneously, it uses simple transformations to solve this problem, where reflection is prioritized to rotation, as I discussed in the Project 1 report.

JOURNAL 2:

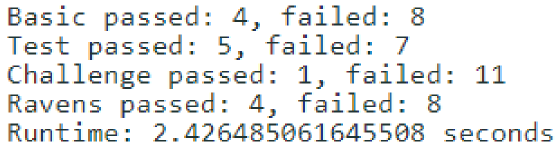
This is my second submission at 3:12 p.m., 10/22. Below is a summary of how I implemented an operation compare the differences between images in the same row or column.

After I have implemented the simple transformations, I noticed that many problems involve counting and comparing the number of shapes or proportions in the problem. To reduce the notion of "shapes", which is widely used in a verbal approach, to a purely visual method of solving the problem, I came up with the idea to quantize the difference, find the amount of difference between images and, compare it to the amount of difference in a different image pair. For example, in Basic Problem C-05, the number of circles around the star increases by one from left to right and from top to bottom.

Below are the details of my implementation:

1. Find the difference between AB and AC or AD and DG. (use *Image.Chops.difference*)
2. Quantize the difference by counting the amount of non-black pixels.
3. If the difference between the two images is the same:
    a. Find the amount of difference between GH or CF
    b. For each image in the answer set, find the quantity of difference between the image and H or F.
    c. If the amount is equal to step 3.a, select the answer.

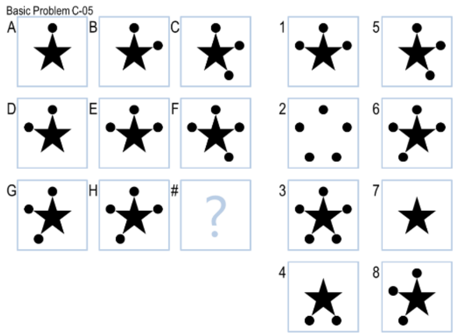
In this case, I reduce the counting problem to an utterly visual approach. Below is the result of my submission:



```
Basic passed: 4, failed: 8
Test passed: 5, failed: 7
Challenge passed: 1, failed: 11
Ravens passed: 4, failed: 8
Runtime: 2.426485061645508 seconds
```

I notice two problems that affect my agent’s performance

1. This method only works when the pictures in the problem set are aligned. It does not work if the shapes change their position. (e.g., Basic Problem C-03)
2. Although my agent can quantify the difference, it does not know the difference is added or deleted.



I will address the above two problems in my following journals.

Comparing my agent’s performance to humans at this point, I think it acts very differently to humans because humans usually use a verbal approach to solve these problems by counting and comparing the number of shapes. However, the idea of “shapes” is missing in the visual approach.

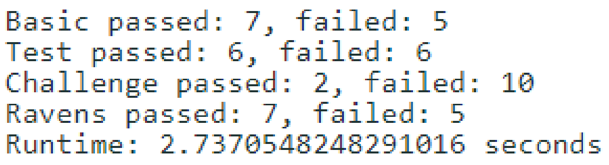
JOURNAL 3:

This is my third submission at 10:10 p.m., 10/22. Below is a summary of how I quantize the differences when the pictures are not aligned.

In Basic Problem C-03, the number of black rhombuses increases by one from left to right and from top to bottom, but the rhombuses’ position changes in each picture. In this case, I use the method of computing the change of black pixels’ ratios between images to count and compare the number of shapes.

1. Count the number of black pixels in A, B, C, or A, D, G.
2. Compare the ratio of back pixels in AB and AC or AD and AG.
3. If the ratio of B/A and C/A or D/A and G/A.
4. If C/A == B/A + 1 or D/A == G/A + 1
    a. Find the number of black pixels of GH or CF.
    b. Compute H/G or F/C.
    c. Find the answer in the answer set where the number of black pixels of the answer / G or C, equals H/G or F/C, respectively.

Below is the summary of the result:



```
Basic passed: 7, failed: 5
Test passed: 6, failed: 6
Challenge passed: 2, failed: 10
Ravens passed: 7, failed: 5
Runtime: 2.7370548248291016 seconds
```

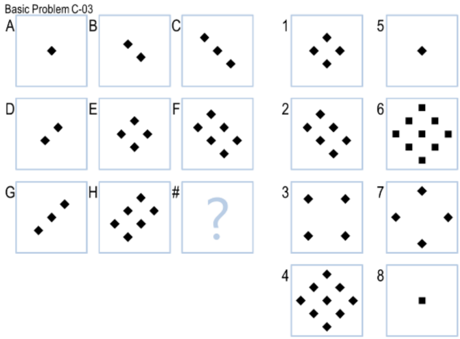
Again, my agent performs differently to humans for the same reason I mentioned above. Additionally, the counting process is replaced by comparing the ratios between images. At this moment, I think my agent is going further away from human cognition because we discard the notion of “shapes” in a visual approach.

JOURNAL 4:

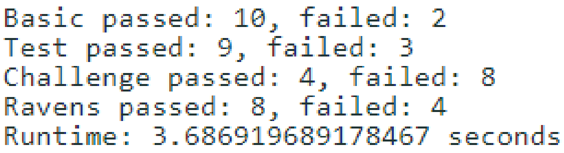
This is my fourth submission at 2:21 p.m., 10/23. Below is a summary of my final improvement of my agent's performance in comparing differences.

As I mentioned before, my agent does not know the difference is added or deleted, although it can quantify the difference. In the example above (Basic Problem C-05), my agent would choose answer 1 because the absolute difference between GH and H1 is the same. I fixed this problem in this submission by adding a sign to the difference. For example, if the number of black pixels increases, no change will be made. If the number decreases, I will change the sign of difference. Therefore, when comparing the absolute value of differences between the two images, my agent can distinguish 1 and 3 in the example above.

Before I submitted, I thought about using a visual approach to solve Basic Problem C-02 and C-04. However, they both rely heavily on the verbal method. C-02 is the change of outer shapes, and C-04 is merging and counting the number of circles. I decided to skip these two problems. I noticed that all the answers but one appear in C-04. Therefore, I implemented another function to select the answer that does not appear in the problem set if all the other methods failed.



Below is the summary:



```
Basic passed: 10, failed: 2
Test passed: 9, failed: 3
Challenge passed: 4, failed: 8
Ravens passed: 8, failed: 4
Runtime: 3.686919689178467 seconds
```

At this moment, my agent can solve the majority of the problems. However, my agent's visual approach is still very different from humans. Since my agent cannot see the image in the visual approach, it only knows to compare the differences of pixels between images rather than counting the shapes that humans would usually do.

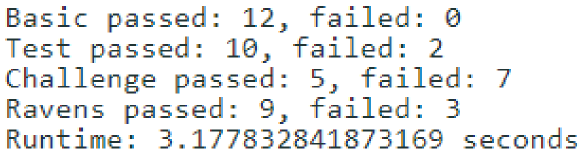
JOURNAL 5:

This is my final submission at 5:05 p.m., 10/24. Below is a summary of my improvement to solve Basic Problem C-12 and my attempts to solve Basic Problem C-2.

C-12 is similar to C-05 above except that the answer is identical to F and H. I think the interpretation behind this is it is gradually flood-filling the first row and first column. The correct answer is identical to F and H because the first row and first column are already flood-filled. Therefore, I included a special case. If the changes from A to B and B to C or A to D and D to G are the same, and if F and H are the same, choose the answer that matches F or H.

Then, I tried using a visual approach to solve C-02, but none of the approaches is helpful. One of them I have been attempting is to compare the addition of the images. However, it does not work because many of the answers set figures would satisfy the constraints because each pixel's addition is capped at 255.

Below is the summary:



```
Basic passed: 12, failed: 0
Test passed: 10, failed: 2
Challenge passed: 5, failed: 7
Ravens passed: 9, failed: 3
Runtime: 3.177832841873169 seconds
```

In this submission, I think the way I improve my agent is similar to the process of incremental concept learning we studied in class. Because 8 is the correct answer, my agent generalizes the model a little bit by choosing the answer that is identical to F or H if all other criteria are satisfied.

CONCLUSION:

I believe the result is fair, but not the best. I kind of cheated to get 2 of the questions in the Basic Problem set C correctly because I did not figure out a visual approach to solve these problems. in the end, my agent can tackle simple transfromations and comparing addition/deletion by the amount of difference or ratio.

I would characterize the overall process of designing my agent to be targeting one type of problem at a time. Since I have implemented simple transformation in Project 1, I modified it in Project 2. Then I started targeting addition/deletion when the pictures are aligned (C-05). Then I worked on the cases when the pictures are proportional and not aligned.

Comparing my agent's approach to my approach to solve the RPMs, my conclusion is similar to what I concluded in the journals and from Project 1. The similarity is my agent mimics human's knowledge and reasoning by using a library to transform and find similarities between images. The differences are:

1. Instead of comparing the pictures by recognizing shapes and patterns, my agent compares images by finding the relationship between pixels in a greater precision.
2. Humans tend to solve RPMs verbally, but my agent solves them visually. Because these are two fundamentally different approaches, how my agent addresses the problem is different from humans, especially when the problems are more complicated.
3. In a completely visual approach, my agent cannot distinguish shapes and find the relations between shapes like what humans would do.
4. My agent "counts" differently. It compares the pixels while humans count by shapes.
5. My agent is not able to learn or create knowledge like a human.

There are some limitations that impact my agent's overall performance:

1. In a completely visual approach, my agent cannot distinguish shapes and find the relations between shapes like what humans would do.
2. My agent would fail to compare the differences if the pictures are not perfectly aligned.
3. My agent is unable to crop a portion of the figure or move a shape to a different location. Therefore, it cannot count the number of circles in the image when they are linked to each other. (e.g., C-04)

If I had more time or computational resources, I would spend time cropping the image and find the relationship within the image (e.g., C-09). Additionally, I would try to figure out a visual approach to solve problems that rely heavily on a verbal approach like C-02 and C-04.