# Project 1 (Summer 2020)

Anshuta Awasthi
aawasthi32@gatech.edu

## 1 INTRODUCTION

As part of Project1 we are trying to solve the Problem Set B of RPM problems, which consists of 2x2 matrices only . In the matrix, 3 out of 4 images are known and we have to design and implement the agent to predict the 4th image by selecting the most optimal answer from the 6 choices given.

After going through the suggested readings (Maithilee Kunda and Goel, 2011)and (Maithilee Kunda and Goel, 2012) , I decided to go with the "Affine Method" of the visual approach first . As per Affine method, we try to solve the matrix by applying the common transformations on the image , observe if the transformation fits the pattern and then verify against the available answer choices . This approach is based on the "Generate and Test" method in a way that first we "Generate" the image by applying the transformation and then we "Test" it against adjacent images to identify the transformation pattern. The "Affine method" makes 2 assumptions:

1. Elements within a row or column in an RPM matrix are related by some transform.
2. Same transformation relationship exists within the elements of parallel rows or columns.

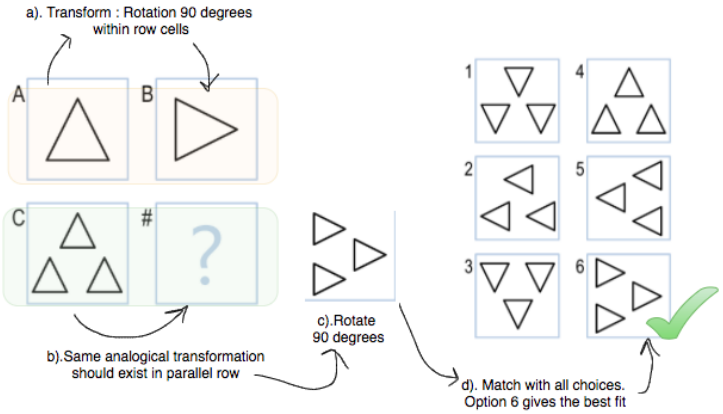


*Figure 1*—Affine Method

Affine method enables the agent to decide the stronger relationship between 2 choices a). Adjacent cells in rows(horizontal relationship) b).Adjacent cells in column(vertical relationship)

On close observation of the problems , it was evident that only a few problems out of 12 basic problems could be solved by these pre-defined transformations. For the problems which could not be solved by this method, later "Heuristics" and “Verbal” approaches are also implemented.

## 2 JOURNAL ENTRIES

### 2.1 Journal Entry 1:

#### 2.1.1 *Date of Submission*

2020-05-23 21:34:43 UTC

#### 2.1.2 *Changes in this version*

This is the first submission of the Agent implemented using “Affine Method”.The Implementation is done in following steps.

1. Define a few basic relationships as - a).Same image b). Image rotated 90 degrees c).Image rotated 180 degrees d).Image rotated 270 degree e) Image flipped horizontally f) . Image flipped vertically
2. Apply these transformations on image A using in-built functions from the “Pillow” library in python and produce resulting images.
3. Each resulting image(TiA) will be compared against the adjacent image in the row first(image B)and then adjacent image in the column(image C).
4. The transformation with minimum difference of all obtained in step( iii) ( where the difference is less than a certain threshold value) is considered to be “Best Fit”.
5. If the minimum difference is obtained between image A and C(elements in a column).This “Best Fit” transform is applied to image B to predict the image D.
6. Else if The “best fit” transform is obtained between image A and B(elements in a row).This transform is applied to image C to predict the image D.
7. The predicted image D is now compared with the 6 available option images . The image with minimum difference is returned as the answer.

For comparing the images “ImageChops.difference” method from “pillow” library is used which returns pixel by py pixel difference between the 2 images . The mean value of statistics obtained using “ImageStat.stat” is taken as a measure for Image comparison. The “Best Fit” is then obtained as -” lower the mean of the difference of 2 images - better the match “.

### 2.1.3 *Agent Vs. Human Approach*

The above approach is more likely the same way humans also approach the problem. Here agent tries to simulate the human problem solving ability by observing change in rotation , flip , reflection of image and moreover it successfully establishes the direction of pattern too i.e. horizontal or vertical.

But the Agent is still inefficient in so many ways . The ability of the agent to establish the relationship is limited to 6 transformations mentioned above. While humans use the previously acquired knowledge of shapes , darkness,size and many more to establish patterns . Agent ‘s ability to compute the difference in images relies on the threshold value of normalized pixel wise difference. This is a very quantitative way of finding the best match and does not provide the information regarding the spatial distribution of an image’s pixel values. If there are multiple different images that share the same normalized pixel wise difference , the Agent will not be able to differentiate while humans can easily identify spatial distributions.

### 2.1.4 *Outcome ,Efficiency and Performance*

***Table 1***—Autograder Results

| **Basic** | **Raven’s** | **Test** | **Challenge** |
|---|---|---|---|
| 7/12 | 4/12 | 5/12 | 5/12 |

**Execution Time** : 6.2591 Secs

As mentioned above , the agent has a very limited ability to solve problems that only qualify the 6 mentioned transformations . The agent does not differentiate between the filled shape or not filled shape . This agent does not recognize the different shapes and cannot detect the pattern where the sides of the shapes are varying.The agent also failed to establish the pattern where the image has multiple shapes and relationship involves the addition or removal of the shapes.

### 2.2 Journal Entry 2:

#### 2.2.1 *Date of Submission*

2020-06-03 02:00:30 UTC

#### 2.2.2 *Changes in this version*

After the previous iteration , I was targeting to get a few easy wins on test problems , so decided to improve the agent with incremental changes in each iteration. In this version, I tried to implement

- The ability of recognizing the pattern of "filled shape/ non filled shape".It turned out to be a little challenging as my agent still does not identify the shapes .I tried different approaches like finding edges or contours.
- The rotation of 45 degrees . "Image.transpose" class in PIL does not have an option of 45 degree rotation.On applying the 45 degree rotation using the "image.rotate " method was leading to black corners. This was solved by filling the black corners resulting after rotation with white colour.

#### 2.2.3 *Agent Vs. Human Approach*

The issue my agent struggled with in this iteration is to identify the shape and filling it . My agent is of no match yet with the human cognitive ability of identifying different shapes in an image

#### 2.2.4 *Outcome ,Efficiency and Performance*

*Table 2*—Autograder Results

| Basic | Raven's | Test | Challenge |
|---|---|---|---|
| 7/12 | 4/12 | 5/12 | 6/12 |

**Execution Time** : 6.7 Secs

My agent improved the score from the previous iteration by solving 1 more challenge problem , which involved 45 degree rotation of image . The approach to identify the shape and filling didn't work and the test set score is still unchanged from the previous iteration. I was trying to find the edges and contours using PIL , but it did not turn out to be good idea to identify shapes.

### 2.3 Journal Entry 3:

#### 2.3.1 *Date of Submission:*

2020-06-04 13:09:34 UTC

#### 2.3.2 *Changes in this version:*

As part of this iteration I implemented 2 changes :

1. I Changed my approach to identify and fill the shape in the image.
   - Converted the image to RGB mode and used a PIL method to flood fill the area outside the shape with green color .
   - Converted the image to numpy array
   - Used pixel intensities to fill green areas with white and non-green areas with black colour.
   - Converted the image back to "grayscale" PIL format.
2. Switched to verbal approach for identifying the number of shapes . The different shapes in the figure are objects that are enumerated as 'a' ,'b'... The number of different shapes in the figure is obtained by counting the number of objects in the problem figure. This comparison helped me to solve a few more basic and challenge problems where the relationship is either addition of one shape or elimination of one shape.

#### 2.3.3 *Agent Vs. Human Approach:*

After this iteration , although my agent can fill a shape in the figure and identify the number of shapes in the figure . It is still failing in so many areas where humans can identify the pattern very easily. My agent will fail where the image consists of many different filled or not filled shapes and it has to identify , which shapes to fill and which to leave.

In "verbal" approach - we are already given the set of attributes defining the structural characteristics of figures. I think it is not a good idea to compare the verbal method based agent with the human thought process of solving the problem. The reason being - in a verbal approach , the agent has the problem representation already given to it in a standard format. Which is similar to how humans will perceive the different objects and shapes .In my opinion , the Agent following "verbal" approach has already been given the directions to behave like humans.

### 2.3.4 *Outcome ,Efficiency and Performance:*

*Table 3*—Autograder Results

| Basic | Raven's | Test | Challenge |
|---|---|---|---|
| 10/12 | 7/12 | 6/12 | 6/12 |

**Execution Time** : 7.95 Secs

As we see above that the changes in this iteration made a significant improvement in overall scores. Although I believe that the agent is not a very generalized one. In the unseen test set , it is still under performing.Currently my agent is able to fill only when there is one shape in the image and fails if multiple shapes are there.Also , I noticed that execution time has been increased by 1.2 sec, that might be due to image intensive operation like floodfill.

## 2.4 Journal Entry 4:

### 2.4.1 *Date of Submission:*

2020-06-05 00:43:52 UTC

### 2.4.2 *Changes in this version:*

I failed to identify the different shapes in the image using visual approach. In this iteration I planned to address the simple problem from the challenge set where the relationship between the figures is based on the "No. of sides" . I chose to follow the verbal method for this and selected the images with 1 shape , defined the mapping between image shape and associated number of sides (ex. shape : "Square" , sides: 4).My agent is reading the shape of the figure from ProblemData.txt and if the shape of the adjacent cell has more or less number of edges than image A - it tries to establish this transformation following the steps of "Affine Method" described above i.e. add or subtract the same number in the number of edges and try to match it with the given solutions.

### 2.4.3 *Agent Vs. Human Approach*

As already mentioned above that I used a verbal approach to solve this and in this method, the agent uses all the attributes of the image (stored in ProblemData.txt file), as we perceive and differentiate as human . The performance of the agent mostly depends on the technical implementation to access the attributes of different objects , compare them and infer meaningful relationships or trans-

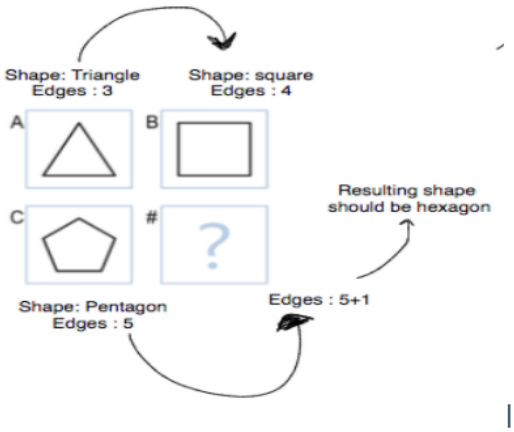


*Figure 2*—Transformation : Challenge Problem B-8

formations. I have not gone deep into the verbal approach to solve problems of greater difficulty which may bring a different perspective than a human's approach. But the "verbal method" to solve this problem is similar the way humans approach this problem.

### 2.4.4 *Outcome ,Efficiency and Performance*

*Table 4*—Autograder Results

| Basic | Raven's | Test | Challenge |
|---|---|---|---|
| 10/12 | 7/12 | 6/12 | 6/12 |

**Execution Time** : 7.14 Secs

I implemented this solution hoping that the test set may have one such problem which is based on the number of edges that this approach could solve. It did not improve the test score and since it is based on the verbal approach , it did not change the challenge problem set score either. Tackling one type of problem at a time is not improving the score. The Agent is getting specific to the given problems and the solution is not a generalized one.

## 2.5 Journal Entry 5 and 6:

### 2.5.1 *Date of Submission:*

2020-06-05 13:46:40 UTC and 2020-06-05 13:53:33 UTC

#### 2.5.2 *Changes in this version:*

The $5^{th}$ attempt of bonnie submission resulted in following error:
*"Execution": "Your code caused RavensProject to crash."*
This is due to the python version issue on Bonnie. I installed python 3.7 and created a new virtual environment to resolve this error as mentioned in the piazza post. $6^{th}$ attempt on bonnie submission was made to test that bonnie issue is resolved without any major changes to agent.py file.

### 2.6 Journal Entry 7:

#### 2.6.1 *Date of Submission:*

2020-06-05 18:03:16 UTC

#### 2.6.2 *Changes in this version:*

In this version, I am targeting to solve the problem where an additional shape is superimposed(Added) on the figure or a shape is removed from the figure. I tried to implement the XOR function given the options in "Pillow" library. After experimenting with various functions in ImageChops class,it turns out to be a series of steps to find the additional shape which is superimposed on the figure and construct the resulting image by adding that shape. The steps I followed are:
a).Take a pixel wise difference between 2 images using ImageChops.difference (In a row/column)
b.)Invert the resulting image and c). Multiply it with the image in parallel row/column using ImageChops.multiply
. This predicted image will then be compared against available answer choices.
This method only worked for scenarios where the additional shape has been superimposed on the image , while failed for the scenario where a shape is removed .

#### 2.6.3 *Agent Vs. Human Approach*

The agent is still no match with humans ability to spot the difference and establish the pattern . The agent is working well in one way(shape addition) but failing the other way i.e shape removal. The human approach is a bit different when it comes to solving such problems. Humans just spot the difference and further add/subtract the difference(some shape) onto the figure. While the

agent takes a little technical approach of finding the predicted image by first producing the difference, inverting it and then multiplying it with the image .

#### 2.6.4 *Outcome ,Efficiency and Performance*

*Table 5*—Autograder Results

| Basic | Raven's | Test | Challenge |
|---|---|---|---|
| 9/12 | 7/12 | 6/12 | 6/12 |

**Execution Time** : 6.46 Secs

Looks like this approach has degraded the performance , instead of improving it. It failed one of the basic test cases which was passed earlier . The pattern where a shape is being removed was earlier addressed by imperfect verbal implementation which was establishing relationship based on just no. of objects and giving the correct results since the answer choices have only one option with specific no. of objects. 1 test case in basic problem set failed due to this incorrect implementation of XOR ,that replaced the verbal approach.

### 2.7 Journal Entries 8 and 9:

#### 2.7.1 *Date of Submission:*

2020-06-05 22:09:08 UTC and 2020-06-05 23:43:24 UTC

#### 2.7.2 *Changes in this version:*

In my quest for a more generalized approach , I came across this paper by (Joyner, 2015) and decided to implement the Agent 3 based on "visual heuristics".
1).First , convert the images to binary images with a threshold value for dark pixels.
2). Inverted it - that assigned 1 for black and 0 for white pixels.This makes it easy to count black pixels.
3). Calculate the difference in the number of dark pixels between adjacent images in row and column.Ratio is not needed since all the images are of the same size (184x184). (i.e. DPR_AC and DPR_AB).
4). Calculate the DPR between each of the answer choices and images B and C as mentioned in the paper *"Compare each of the test-answer pairs, the combinations of any cell adjacent to the empty slot and each answer choice."*

5). Compare the respective ratios obtained in the steps 3 and 4 .
6). The answer choice which is closest to either row wise or column wise cell ratio will be chosen.

However above mentioned approach did not work in 8th attempt of bonnie submission due to some issue in the order of many if /else statements in the code.After fixing the issue , it gave a better performance in 9th attempt of bonnie submission

#### 2.7.3 *Agent Vs. Human Approach*

I would say that this approach is less likely the way humans approach the problem. It is a very quantitative way of finding the relationship or pattern by calculating the ratio of dark pixels. On the other hand ,humans can spot the difference easily without doing too much calculation , because of the spatial and structural understanding of the shapes in the image .

#### 2.7.4 *Outcome ,Efficiency and Performance*

***Table 6***—Autograder Results

| Basic | Raven's | Test | Challenge |
|---|---|---|---|
| 10/12 | 10/12 | 8/12 | 4/12 |

**Execution Time** : 6.18 Secs

This approach worked wonders and gave me the desired results I needed for the test problem set.The better performance in unseen raven's test also ensures that the solution is generic one and not overfitting to the basic problem set. But somehow it reduced the test cases passed for challenge problem set.This might be due to changes dome in existing flow of if/then statements in the code.

### 2.8 Journal Entry 10:

#### 2.8.1 *Date of Submission:*

2020-06-08 00:46:14 UTC

#### 2.8.2 *Changes in this version:*

Just fixed the broken code for challenge problems and it achieved the improved score on challenge set which was affected in the last iteration.But 1 of the Raven's

test which was passing before started failing.

### 2.8.3 *Outcome ,Efficiency and Performance*

*Table 7*—Autograder Results

| Basic | Raven's | Test | Challenge |
|---|---|---|---|
| 10/12 | 8/12 | 7/12 | 6/12 |

**Execution Time** : 6.29 Secs

I decided to stop improving my agent at this point. However , I still feel there is lot that can be fixed here.

## 3 CONCLUSION:

**a).Implementation Approach :** I started with a very basic implementation of "Affine method" using few generalized transformations.That gave me a head start on the project and 50% of the problems in basic and test sets were solved with these. In a quest of achieving just a few more successes , I went ahead with a verbal approach to solve the simple problems which were failing before.Hoping that similar problems would be existing in the "test" problem set. After several attempts of tackling one problem at a time and trial /error method , I realized that there is a need for a more generic approach since the "Test" set score was not improving. I read more papers and found the DPR method mentioned in Dr. Joyner's paper is worth giving a try. That gave me the passing results I needed. I did few quick fixes in the code after that but did not try to improve the results further due to time constraints and instead decided to complete the journal.

**b).Agent Vs Human Approach :**
I experimented with 3 different types of approaches - In Visual- Affine and Heuristic based and Verbal. I experienced that visual affine methods and verbal methods approach the problem more like humans do.
Verbal Method is based on standard data which describes the characteristic of a problem in a structured manner and it is very similar to what humans perceive the image.
Affine method used in visual approach is also simulating human intelligence upto some extent , but fails in differentiating spatial distribution of different

kind of shapes in the image.
Visual heuristic based approaches are more quantitative and depends on the pixel ratio calculation which humans can not do in a very precise manner. On the other hand, it can also be considered similar to how humans perceive and differentiate easily between dark areas , light areas and different shapes in the image without any mental calculations.
The final outcome of the Agent is also displaying similarities with what is defined as human intelligence.
The agent is following a complex structure of “if /then/else" rules, that simulates the metacognitive ability of humans. I believe the agent is trying to mimic human intelligence to an extent and can be improved further to reach a higher level of accuracy. As stated by carpenter et.al -

> “What one intelligence test measures according to the current theory, is the common ability to decompose problems into manageable segments and iterate through the differential ability to manage the hierarchy of goals and subgoals generated by the problem decomposition and the differential ability to form higher-level abstractions.” (Carpenter, 1990)

**c).Further Improvements:**
My agent is not perfect , but performing above average in all 4 problem sets. There are many different strategies which could not be tested due to time constraints.
In Affine method , I could test very few(only 6) transformations. Apart from identity , different angle rotation and horizontal /vertical reflection, I could have implemented the AND,NOR and XOR transformations as well. I tried some but failed due to limited experience in python and image processing libraries.It has a learning curve and would have definitely performed better given more time .
Efficiency and time complexity of operations performed in image processing can be improved further if implemented on images as numpy arrays.
There are different visual heuristics which can also be implemented such as IPR mentioned in (Joyner, 2015)
I am looking forward to implement these strategies in upcoming assignments.

## 4 REFERENCES


[1] Carpenter, Patricia A. (1990). "What One Intelligence Test Measures: A Account of the Processing in the Raven Matrices Test." In: Carnegie-Mellon Univ., Pittsburgh, Pa. Psychology.

[2] Joyner, D.A. (2015). "Using Human Computation to Acquire Novel Methods for Addressing Visual Analogy Problems on Intelligence Tests". In: *In Proceedings of the Sixth International Conference on Computational Creativity.* Provo, Utah.

[3] Maithilee Kunda, McGreggor and Goel, Ashok (2011). "Two visual strategies for solving Raven's Progressive Matrices Test". In: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

[4] Maithilee Kunda, McGreggor and Goel, Ashok (2012). "Reasoning on the Raven's Advanced Progressive Matrices Test with Iconic Visual Representations". In: *n 34th Annual Conference of the Cognitive Science Society.* Sapporo, Japan.