Project 3: Raven's Progressive Matrices

Marc Micatka
mmicatka3@gatech.edu

1 INTRODUCTION

The agent designed for Project 3 follows the same visual approach as the agent in Project 1 and 2 First, the agent creates affine transformations for seven image pairs (see 5.2). The agent calculates similarity values between the images for seven transformations (see 5.1). Using these image pairs, the agent determines which pairing and which transformation results in the best match. Once the pairing and transformation is determined, the agent applies that transformation to the corresponding image pair and all answers, choosing the one that most closely matches.

Basic Problem E-08 (see 5.11 and **Figure 1** below.**)**, will be used to illustrate the solution process. The agent returns Horizontal Addition (A+B = C) is the best pairing match and outputs the similarity values [0.98 0.92 0.91 0.92 0.92 0.85 0.95] for all transformations from this pairing. Transformation 1 (identity) has the highest similarity value (0.98) so the agent creates a new image, G+H, and compares it against the answer choices for the highest match for identity transformation. That similarity check returns: [0.96 0.81 0.86 0.83 0.85 0.83 0.84 0.85] for the eight options. Image 1 has the highest similarity value (0.96) so it is chosen as the answer.

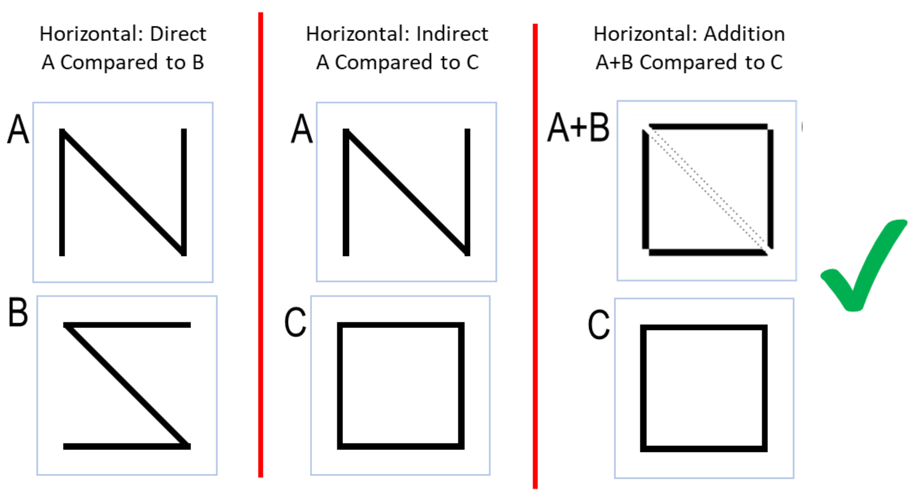

Figure 1— Horizontals comparisons used for Basic Problem E-08. This process is repeated for vertical and diagonal comparisons.

2 JOURNAL ENTRIES

2.1 Submission 1: 2019-11-20 06:43:35 UTC

	Problem Set D			Problem Set E		
Basic	3 of 12	25%	-	4 of 12	33%	-
Test	5 of 12	42%	-	3 of 12	25%	-
Challenge	1 of 12	8%	-	3 of 12	25%	-
Raven's	3 of 12	25%	-	3 of 12	25%	-
Runtime	12.322 s					

What did you change for this version? Why?

For the first submission, I reused my Project 2 code and ran the Problem Set D and E images. See 5.4 for a discussion of how Project 2 functions. I used this method for Project 2 as well (using my code from Project 1) and it sets a good baseline for submissions going forward.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it "think" similarly to how you think, or differently?

My agent approaches problems like I would from a very high level. It has a bank of simple transforms and it looks at each problem and determines which of those transforms best describe what's happening. Beyond that, humans would be able to look a bit deeper and see more complicated transforms. My most non-human-like transformation is the pixel count metric which is responsible for most of my correct answers. It seems a bit counter-intuitive, but pixel count is a good proxy for image for shape transformations or images addition or subtraction without the agent performing more complex image recognition functions.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

For the first attempt using an agent that passes Project 2, the agent struggled. There are many diagonal transformations in Project 3 (see 5.6) that the agent does not catch because it is not looking for them. The agent is also not reviewing images for addition or subtraction between 3 elements. This problem type is featured heavily in Basic Set E (see 5.7). In general, the agent performs better on the more complex problems that it can solve with dark pixel ratio comparisons and transformations that use simple affine transformations.

The efficiency of this algorithm is bounded by its transformation checks between adjacent neighbors which yields an efficiency of $T(n) = O(n^2)$. Because my agent is using very simple numpy operations and vectorized scoring algorithms, my agent can analyze the images very quickly – in less than 15 seconds.

2.2 Submission 2: 2019-11-26 16:32:00 UTC

	Problem Set D			Problem Set E		
Basic	4 of 12	33%	▲+1	5 of 12	42%	▲+1
Test	5 of 12	42%	▶	3 of 12	25%	▶
Challenge	1 of 12	8%	▶	3 of 12	25%	▶
Raven's	3 of 12	25%	▶	4 of 12	33%	▲+1
Runtime	9.598 s					

What did you change for this version? Why?

For this submission, I was reviewing my code from Project 2 and I found a pretty simple coding error where I was comparing the wrong images for some horizontal indirect comparisons (see 5.2). I fixed the error and the agent performed a bit better as expected.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it "think" similarly to how you think, or differently?

This revision makes the agent ever so slightly human-like because I was comparing two images that were not in the same row (A and F) and a human would not be making that comparison..

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

The agent improved by 3 solutions overall (all deltas in the results table are compared to the baseline submission). These were all RPMs that involved an affine or DPR transformation between elements that I was mis-comparing in the prior solution. The same struggles with diagonal transformations and addition/subtraction transformations is present in this version. The efficiency remains unchanged and the overall computation time decreased. I think this variation is from the online submission tool.

2.3 Submission 3: 2019-11-26 16:33:53 UTC

	Problem Set D			Problem Set E		
Basic	7 of 12	58%	▲ +4	1 of 12	8%	▼-3
Test	4 of 12	33%	▼ -1	2 of 12	17%	▼-1
Challenge	1 of 12	8%	▶	2 of 12	17%	▼-1
Raven's	3 of 12	25%	▶	3 of 12	33%	▶
Runtime	9.629 s					

What did you change for this version? Why?

For submission 3, I implemented the intersection pixel ratio mentioned in (Joyner, 2015) and recommended by John Pham in his review of my Project 2 report. I calculated the intersection ratio and added the metric to my affine transformation matrix. I also found that keeping the DPR decreased my performance on the Basic D problem set so for this version, I removed it from my transformation matrix. Overall, the performance on this submission was not great (-5 from submission 2) but I passed the 7/12 threshold on the Basic Set D so I submitted it anyway. I'm unsure why the IPR improves the performance on Basic Set D and decreases it on every other set.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it "think" similarly to how you think, or differently?

Again, the approach is similar to human cognition, but the metrics used to achieve the solutions are non-human. The introduction of the intersection pixel ratio (IPR) is another good proxy for human cognition that is much more simplistic to implement. IPR will allow for simple shape recognition and will result in a higher similarity score for images containing similar shapes.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

This version improves on Problem Set D but the performance decreased on Problem Set E. Looking at the two problem sets, E has a heavy addition/subtraction focus and D has almost no addition/subtraction transforms. This agent has no subtraction/addition capabilities yet (see 2.4 for the introduction of addition transformations). I also saw a huge performance decrease when I introduced diagonal transformations, so I removed that portion and will try and implement it later. The agent still gets Basic Problem D-02 incorrect (see 5.6) but can correctly solve problems like D-04 (see 5.8) with the introduction of the IPR. The efficiency is unchanged and the runtime hasn't changed significantly.

2.4 Submission 4: 2019-11-29 16:00:35 UTC

	Problem Set D			Problem Set E		
Basic	3 of 12	58%	▶	7 of 12	8%	▲ +3
Test	3 of 12	33%	▼ -2	5 of 12	17%	▲ +2
Challenge	1 of 12	8%	▶	3 of 12	17%	▶
Raven's	3 of 12	25%	▶	7 of 12	33%	▲ +4
Runtime	12.348 s					

What did you change for this version? Why?

For submission 4 I added diagonal transformation checking along the main diagonal (A-E) only. I also checked for addition transformations (A+B=C) horizontally and verticallyI made this change after assessing Problem Set E and seeing how many transformations in E were addition based.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it "think" similarly to how you think, or differently?

I'm still using IPR and DPR as computationally lightweight shape detection and addition proxies. Although these metrics aren't replicated in humans directly, they help the agent reach decisions in a similar fashion. The introduction of diagonal transforms and addition has also helped the agent achieve more human-like performance.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

My agent is checking for diagonal transformations but it's still missing Basic Problem D-02 (see 5.6) which is a simple identity transform from A-E. The reason for this is clear when you overlay A and E (see 5.9). What looks like a simple identity transformation is a small translation between A and E. This means that the identity checker returns a result less than 1.0 because of the offset. Overall, this version performs quite a bit better (+9 over submission 4 and +8 over the baseline) but worse on problem set D and I'm still missing quite a few challenge and test problems. The efficiency is unchanged, and the runtime has increased because I've added some additional image operations.

2.5 Submission 5: 2019-11-30 19:07:20 UTC

	Problem Set D			**Problem Set E**		
Basic	5 of 12	42%	▲ +2	8 of 12	67%	▲ +4
Test	2 of 12	17%	▼ -3	8 of 12	67%	▲ +5
Challenge	3 of 12	25%	▲+2	2 of 12	17%	▼ -1
Raven's	3 of 12	25%	▶	5 of 12	42%	▲ +2
Runtime	87.247 s					

What did you change for this version? Why?

Submission 5 separates the affine transformation checks from the IPR and DPR checks. The rationale is that the agent should prioritize pure affine transformations – identity, rotations, mirroring – over more abstract metrics like the IPR and DPR. To achieve this, the agent loads all the images and runs them through an affine first. If any transformation exceeds a threshold value, that answer is chosen. If not, the images are loaded into a transformation class that compares the IPR and DPR values for the images.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it "think" similarly to how you think, or differently?

I'm happy that this version scores the best overall and on problem set E because I think it's the best reflection of human cognition on any of my submissions so far. Humans will prioritize the simpler transformation over a more complex transformation and this agent reflects that logic.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

This agent performs the best of any submission yet. I've now hit my 7/12 threshold on 3 of the 4 problem sets which is wonderful. This agent does well on Problem Set E because there are a lot of simple affine transformations between diagonal members that were not being properly addressed by the previous submissions. It still struggles on some transformations that don't line up perfectly (see 5.9) and it has started missing some problems from set D that it used to solve using DPR or IPR that it is now solving incorrectly using affine transformations.

The efficiency was decreased to $T(n) = 2 * O(n^2)$ because every image is checked twice – once for affine and once for non-affine. I tried to be more efficient in my classes and logic to solve everything with one loop but, seeing as this class doesn't reward efficiency or punish inefficiency, I just wrote an algorithm with two loops through the images. This inefficiency is reflected in the runtime increase.

2.6 Submission 6: 2019-11-30 20:52:59 UTC

	Problem Set D			Problem Set E		
Basic	6 of 12	50%	▲ +3	7 of 12	58%	▲ +3
Test	5 of 12	42%	▶	8 of 12	67%	▲ +5
Challenge	6 of 12	50%	▲+5	2 of 12	17%	▼ -1
Raven's	3 of 12	25%	▶	7 of 12	58%	▲ +4
Runtime	115.825 s					

What did you change for this version? Why?

In an ongoing effort to get the final test cases for Problem Set D, I set about tweaking my algorithm to see how if it would change the performance. I first removed the additional IPR/DPR transformation check and added only the IPR to the original affine matrix. This along with some weighting adjustment improved the performance overall by a large margin (19 over the baseline, 8 over submission 5) but did not improve on Problem Set D.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it "think" similarly to how you think, or differently?

This version does not fundamentally alter the operation of the agent. For more complete answers, see 2.4 and 2.5.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

The agent continues to struggle on nearly all of the challenge problems in Problem Set E. These are complicated transformations that don't have a strict pattern to them, so I have not attempted to introduce metrics to address them. This version does the best overall on the Problem Set D Challenge Set after removing the DPR.

Because I removed the second pass of the algorithm, the operational efficiency is back to $T(n) = O(n^2)$ however the runtime increased significantly. Seeing as this version is equally as efficient as the first four submissions, I believe the runtime increase is an issue with Bonnie and not with my algorithm.

3 CONCLUSION

How would you characterize the overall process of designing your agent? Trial-and-error? Deliberate improvement? Targeting one type of problem at a time?

I began with my framework from Project 2. I knew it had some shortcomings – namely that it was not looking at diagonal transformations or addition or subtraction of neighboring images. The initial changes were very deliberate. As I saw the type of problems it struggled on (diagonal transformations, addition and subtraction of images) I began to more clearly target specific challenge areas. Once I achieved the baseline for the Basic Sets, I worked on tweaking my existing algorithms to improve my performance in the hopes that it would translate to better performance on the Test Sets.

How similar do you feel your final agent is to how you, a human, would approach the test? Why or why not?

This is answered exhaustively in every single submission but in general my agent approaches problems like I would from a very high level. It has a bank of simple transforms and it looks at each problem and determines which of those transforms best describe what's happening. Beyond that, humans would be able to look a bit deeper and see more complicated transforms. My most non-humanlike transformation is the pixel count metric and the intersection ratio. Although the actual metric is non-human, they serve as simple proxies for what humans do intuitively.

What improvements would you make if you had more time and/or more computational resources?

With more time, I'd like to add a few layers of understanding to my agent. For Project 3, I think it would have been useful to implement a transformation checker for constant addition/subtraction. Some images (see 5.10) don't exhibit a proportional pixel increase that would be caught by addition or the DPR. Instead, it's a constant addition or subtraction of elements. Adding detection to allow my agent to count connected components and compute the delta between frames would help with those problems. Adding additional metrics like corner detection or simple shape detection would be useful in answering the challenge problems as well.

4 REFERENCES

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

5 APPENDIX

5.1 Image Transformations used in Project 3

Table 1 — Affine Transformations used in Project 3 with weights.

Transformation	Description	Weight
Identity	Compare directly	1.00
Reflection: X	Reflect over X-Axis	0.95
Reflection: Y	Reflect over Y-Axis	0.95
Rotation: 90°	Rotate 90° clockwise	0.95
Rotation 180°	Rotate 180° clockwise	0.95
Intersection Pixel Ratio	$\frac{A \cap B}{number\ of\ pixels\ in\ A}$	1.00
Dark Pixel Ratio	$\frac{\lvert A_{dark\ pixel\ count} - B_{dark\ pixel\ count}\rvert}{number\ of\ pixels\ in\ A}$	0.95

5.2 Image Pairs used in Project 3

Table 2 — Image Pairs used in Project 3.

Transformation Description	Original	Test
Horizontal: Direct	A compared to B	H compared to X
Horizontal: Indirect	A compared to C	G compared to X
Horizontal: Addition	(A+B) compared to C	(G+H) compared to X
Vertical: Direct	A compared to D	F compared to X
Vertical: Indirect	A compared to G	C compared to X
Vertical: Addition	(A+D) compared to D	(C+F) compared to X
Diagonal: Direct	A compared to E	E compared to X

5.3 Affine Transformations and Weights from Project 1

Table 3 — Affine transformations used in Project 1 to evaluate answer.

Transformations	Description	Weight
Identity	Compare A and B directly	1.0
Reflection: X	Reflect B across X-axis	0.50
Reflection: Y	Reflect B across Y-Axis	0.50
Rotation: 90°	Rotate B 90° clockwise	0.25
Rotation: 180°	Rotate B 180° clockwise	0.25
Histogram Comparison	Calculate the ratio of white/black pixels	0.25

5.4 Project 2 Project Description

The final agent designed for Project 2 attempts to follow the same approach as the agent designed in Project 1. First, different affine transformations are calculated for horizontal and vertical figures. This includes direct transformations like A to B but also indirect, like A to C. The direct and indirect transformations are computed for D to F, E to F, E to H, and B to H. The transformation (the different transformations are shown in Appendix 5.1) that returns the highest similarity score is chosen along with the direction that resulted in that score (direct versus indirect, horizontal or vertical).

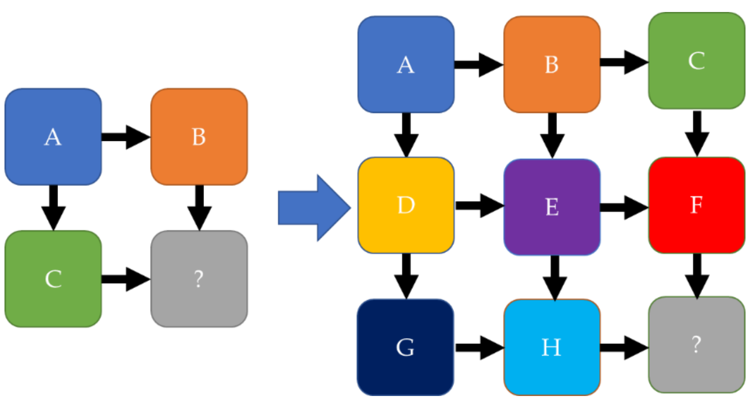

Figure 2 — Different transformations between frames and the frame letters corresponding to each image.

The agent then calculates the similarity between the H/F or C/G and choices 1- 6 using only the transformation chosen in earlier steps. The option that returns the highest similarity match over a threshold value is chosen as the answer. If no answer scores over the threshold, the agent will revert to simpler weighted voting method applied in Project 1.

Basic Problem C-07 (shown in Appendix 5.5) will be used as an example. Initially, A will be compared to B and will return the highest similarity value and the transformation that returns this score. This is repeated for the other comparison images. For C-07, "Horizontal, Indirect", returns the highest score for "Y-Axis Mirror" across all transformations and all images. The agent would then *only* assess image G and the answers 1-6 for this transformation.

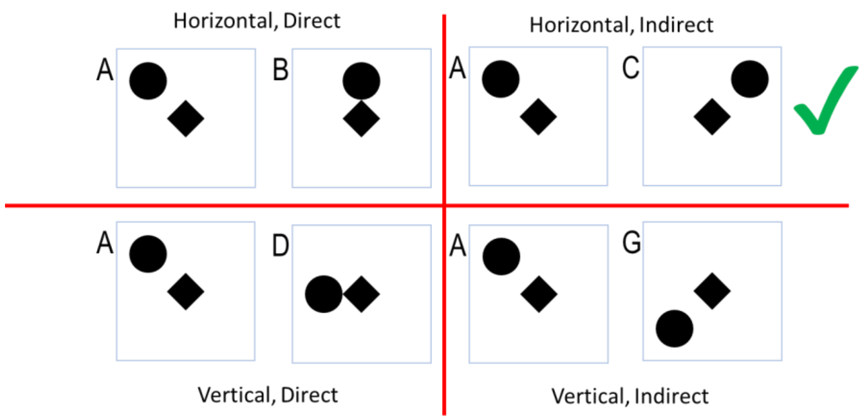

Figure 3 — Problem solving approach to Basic Problem C-07.

5.5 Basic Problem C-07

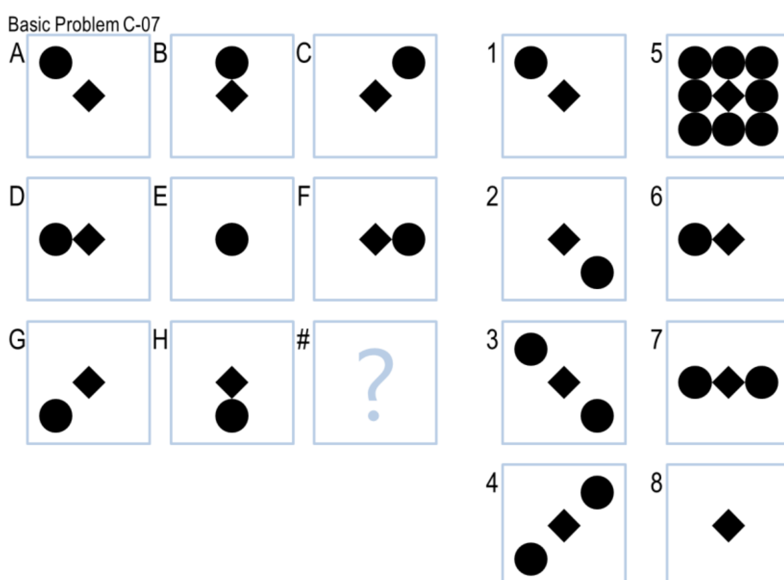

Figure 4 — Basic Problem C-07 has a straightforward A-C transformation but a difficult A-B transformation.

5.6 Basic Problem D-02

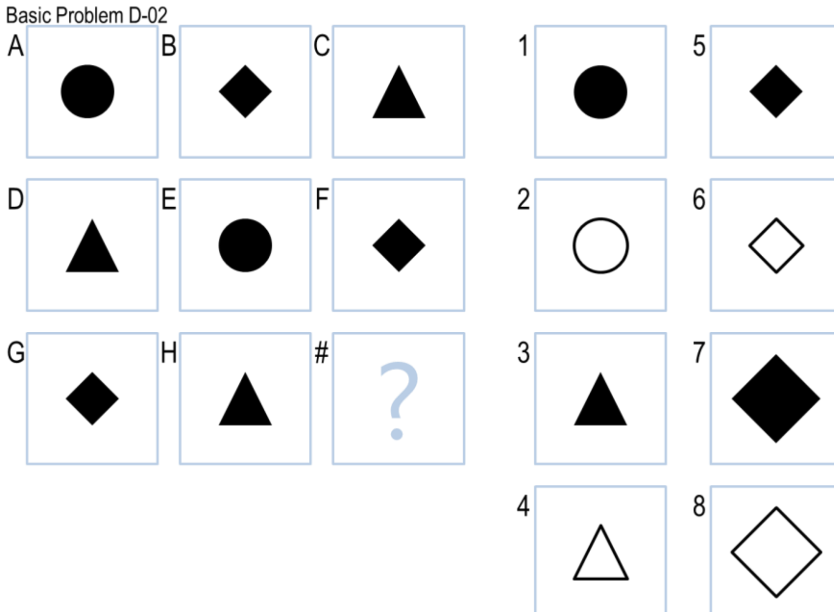

Figure 5 — Basic Problem D-02 has a straightforward diagonal transformation (A-E) but a complex horizontal or vertical transformation.

5.7 Basic Problem E-01

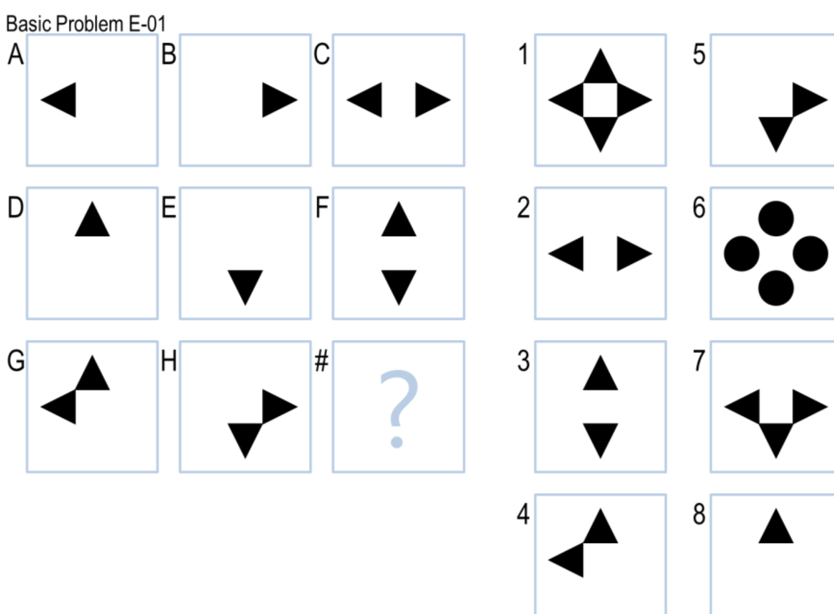

Figure 6 — Basic Problem E-01 exhibits addition horizontally and vertically.

5.8 Basic Problem D-04

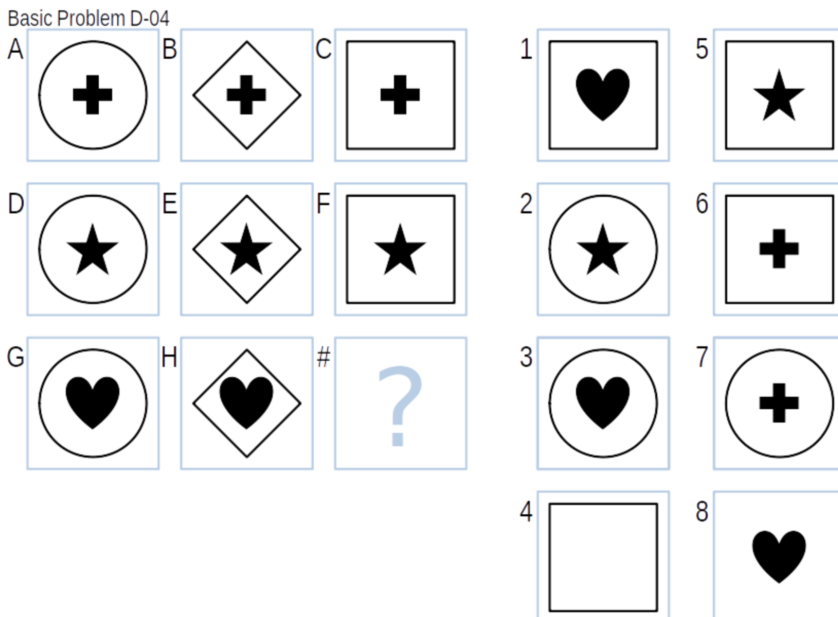

Figure 7— Basic Problem D-04 is solved using IPR.

5.9 Basic Problem D-02 Overlap

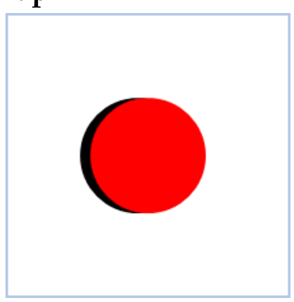

Figure 8— Basic Problem D-02 is a bit tricky because what looks like a simple identity transform is actually a slight translation.

5.10 Basic Problem D-12

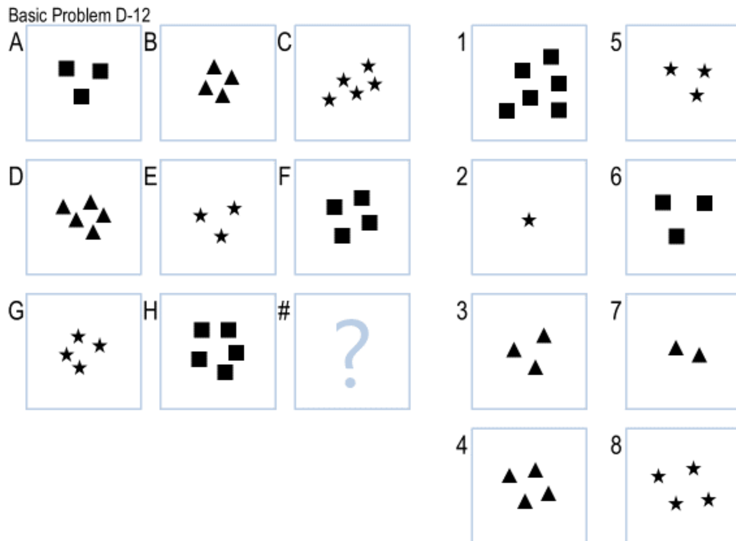

Figure 9— Basic Problem D-12 doesn't exhibit a consistent pixel addition from frame to frame.

5.11 Basic Problem E-08

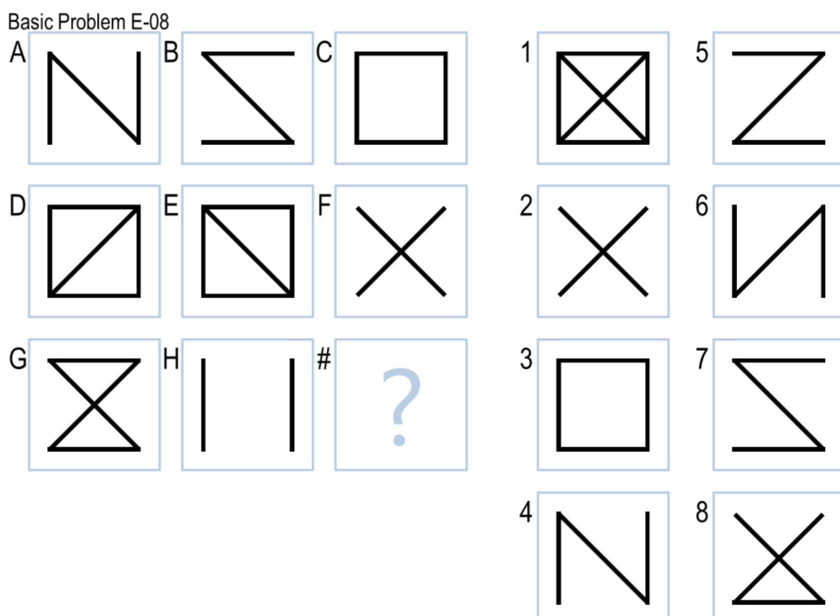

Figure 10— Basic Problem E-08 is a fun addition operation.