An AI System for Visual Creativity Task

Chuilian Kong (chuilian.kong@vanderbilt.edu)
Department of Electrical Engineering and Computer Science, 2201 West End Ave
Nashville, TN 37211 USA

Abstract

In this paper, I built a AI system to solve the visual creativity task. The system emulate the way how human solve this kind of problem and try to think visually by using internal representation of shapes and images. The paper will first describe the overall design of the system and then give a detailed example to help reader get a better knowledge of how the system works, then give some results and analysis of the system.

Keywords: visual thinking, visual creativity,artificial intelligence

Introduction

Creativity Task

The goal of this project is to build an AI system that could complete creativity task(Kim, 2006): generate different drawings using some simple shapes as the starting point. For example, if I give a circle and a triangle to the system as input, the system should return some creative drawings that only using circles and triangles as basic elements. However, the size and number of those circles and triangles could be vary in order to form a meaningful image.

The Way a Human may Solve the Task

I think people usually solve this kind of visual creativity task using their imagination, and we are not doing imagination by wandering without limitation, instead I think imagination is based on our experience or memory. When I were given a set of starting shapes and were told to create meaningful drawings using those starting shapes, the first thing I would think about is to find something in my memory that looks like a starting shape and use those things to construct a meaning image and use several starting shapes to replace them. In this way, I created an image that both has a meaningful sense and contains only starting shapes.

Therefore, I think my creative thinking AI system should perform in a similar way: it needs an imagery space which contains many simple meaning images just like I have in my memory. It also needs a complete and precise mechanism to judge whether a starting shape(or some combination of them) looks like a part of an image, and a method that use those shapes to replicate that image in order to generate a new drawing.

Why interesting

I think this kind of AI system is interesting, because it is interesting to see a computer to automatically generate creative drawings as if it really is creating something and you will have so much fun to see the clumsy drawings sometime it returns just like a baby draws.

System Design

The inputs of model should be a list of starting shapes. The starting shapes are supposed to be extracted from an input image.

System Inputs

The inputs of model should be a list of starting shapes. The starting shapes are supposed to be extracted from an input image.

System Outputs

The outputs of model should be an image that describes a simple concept or conveys a meaningful information such as a cat is sitting on a tree or simply a figure of a robot.

General Idea

First, my system has a imagery space, which is a bunch of simple pictures that describe simple concept such as a cat, a face and an apple. Then the system will break those pictures into pieces according to their colors. For each of those small pieces, the system will use one or more starting shapes to fit them, pick the best fit pattern as the representation of the piece, and record fitting area percentage at the same time. Then the system will combine those representations into one picture according to their spatial relationship in original picture and compute overall fitting area percentage. After generate all of the pictures, the system will output first three pictures that have higher overall fitting area percentage.

Image Representation

This AI system is a visual AI system, which means the thinking process of the system is driven by images. Basically the smallest unit of manipulation is a shape, which is a piece of an image. The shape is an object maintained by the system, it contains not only a bit map of the image but also a list of helpful information about characteristics of the shape including center point, orientation, bounding box, shape category and area. There are also a list of method that can be used to manipulate a shape including move, rotate, resize.

Detailed Explanation

My AI system can be divided into three part. The first part is called learning, in the learning process the system will load sample pictures into imagery space and break them down into different pieces according to their colors. These pieces then are used as the minimum unit of fitting.

The second part is called fitting, in the fitting process the system will analyze each piece of every image in the imagery space and use the most appropriate starting shape to fit the piece. Then generate a best fitting drawing for each image.

The last part is called picking, in the picking process the system will select the image that have better overall fitting performance as the output of the system.

Learning Process Learning is the process that the system emulate human to get knowledge about some simple concept of real world objects so that the system could generate meaningful rather than random images. First, the system will load every image in the imagery space (which is located on the disk) into working memory and then break them into pieces according to the color. The learning process is not supposed to run every time the system is start up for solving visual creativity task, it is supposed to run only if there is a new image imported into its imagery space, and when it happens, the system will learn the new coming image by breaking it into pieces and analyze and store the information of every piece separately on the disk. When next time the system starts, it only need to reload the stored information to the memory rather than reloading original images to analyze again.

Fitting Process Fitting the core process of the system since it controls how to interpret the image using starting shapes. The detailed fitting algorithm will be discussed further in the next section.

Picking Process Picking is the process that the system determine which one to be output. After fitting process, we get many fitting solutions for one original shape using different starting shapes. How do my system decide which one to choose? We need a measurement that can tell which one is fitting better. The basic idea of a better fitting is, the fitting shape is supposed to have the similar size as the original shape and both shapes are supposed to cover a similar region. The measurement my picking algorithm using is the percentage of pixels that has true value in both images to pixels that has true value in either images. This is a good measurement because it considers both sides of coverage: it ensures a large coverage area of fitting shape to original shape which means the fitting shape should not be too small and also ensures a large coverage area of original shape to fitting shape which means the fitting shape should not be too large.

Fitting Algorithm

As mentioned above, the fitting process is the core of the AI system. The main idea of fitting is to find a combination of same or different size of starting shapes that looks like the shape that needs to be fit. In my AI system, for simplicity, we only use one kind of starting shape to fit one fitting shape, that is, for every starting shape, the system may use many different size of it to fit a original shape, and then use the similar selecting technique as mentioned above to pick one that has better fitting performance.

Since my system is capable of recognizing circles and rectangles, there should be special cases to process those special shapes to achieve a better overall performance. I will first discuss some of those special situation when the algorithm uses special procedures to fit, then I will give a general fitting procedure.

Use circle to fit circle This is the simplest situation, the fitting algorithm will simply read the shape information including center and radius from the circle that needs to be fit, then draw a circle that has the same center and radius as the fitting shape. Why don't duplicate the original circle to be the fitting shape? Duplicating the original circle is the way I first use to process circle fit circle situation, the problem is, sometimes the original shape is not a real circle, it may be a shape that is very similar to a circle and being classified as a circle in the leaning process. In that case, duplicating the original shape may lead to a output image that contains a shape that is not included in the set of starting shapes.

Use rectangle to fit rectangle This is the similar situation as using circle to fit circle. The fitting algorithm is supposed to read the shape information of the rectangle that needs to be fit, including center, length, height and orientation. And then use those exact information to draw a new rectangle as the fitting shape.

Use rectangle to fit unknown category shape Use rectangle to fit any shape should be easy if we introduce a concept called bounding box. The bounding box is the minimum square that contains the whole part of a shape. In this case, what we need to do is simply draw a rectangle that has the same center as the original shape and has the same size as the bounding box of the original shape.

Use circle to fit rectangle This is a little bit complicated situation than above two. The general idea is to use circles of the same size but located on different center to form an approximation of the rectangle. The diameter of circles should be the height of the rectangle and the circles are supposed to evenly distributed on the body of rectangle. In order to maintain the central point of the original shape, the fitting should be processed from center to sides. So firstly, the fitting algorithm will read the shape information of the rectangle that needs to be fit, including center, length, height and orientation. Then the fitting algorithm will draw the first circle using the height of the rectangle as its radius and the center of the rectangle as its center. After drawing the first circle, the algorithm will check if there is enough room to draw another two circles, if there is, draw two circles with the same radius and center as the first one does and move them one diameter length towards the orientation and the counter orientation of the rectangle, respectively. Then the algorithm will recursively check if there is enough room for next two circles and draw them and move them if there is. Finally, the algorithm will generate a series of circles that cover most region of the rectangle.

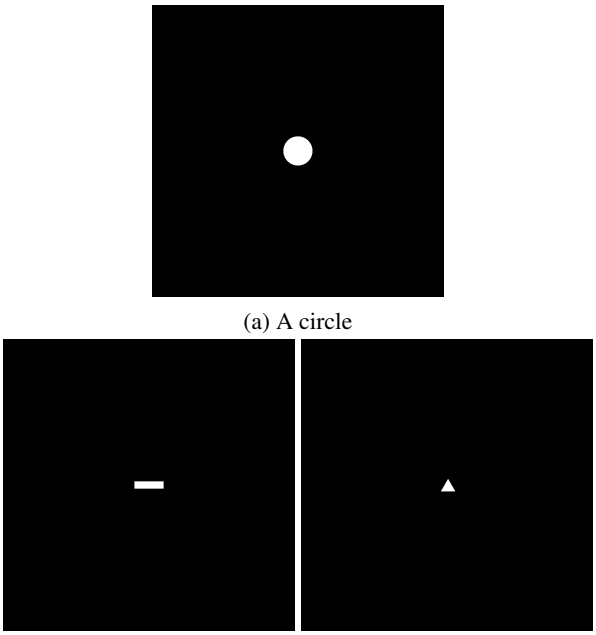
Use unknown-category shape to fit rectangle This is the situation that similar to the last one we discussed before – use circle to fit rectangle. The general idea is the same, to use same size of unknown shapes but located on different center to form an approximation of the rectangle. To make a un-

known shape like a circle, we need to refer to a concept called bounding box. the bounding box is the minimum square that contains the whole part of a shape. When we look at the bounding box of the unknown shape, it is much easier to fit them with the rectangle. So firstly, the fitting algorithm will read the shape information of the rectangle that needs to be fit, including center, length, height and orientation.Then the fitting algorithm will draw the first shape by resizing the shape to a size which the height of bounding box is the same height of the rectangle, and move the shape to be centered as the center of the rectangle. After drawing the first shape, the algorithm will check if there is enough room to draw another two shapes, if there is, draw two shapes with the same size and center as the first one does and move them one length of bounding box towards the orientation and the counter orientation of the rectangle, respectively.Then the algorithm will recursively check if there is enough room for next two shapes and draw them and move them if there is. Finally, the algorithm will generate a series of unknown shapes whose bounding box cover the rectangle completely.

General fitting algorithm The general idea of using unknown category shape to fit unknown-category shape is to find a combination of same or different size of starting shapes that looks like the shape that needs to be fit. We also want to maintain the center of the original shape, so the first starting shape we draw should be at the center of the original shape. At what size? This is the core question for our general fitting algorithm. When we know the center we are going to draw the shape, how can we decide the size of it? Since both fitting shape and original shape are in unknown category, we cannot just set the size to a fixed value like what we did in last two situation, we should find a size in which the fitting shape fit the original shape the best. Although we cannot determine exactly what the size to fit best, we do have upper and lower bound of the best size, which are maximum and minimum value of length and height of the original shape's bounding box. Then for each size in between, we can try them all and pick one the fit the best. After determine the size of the fitting shape, the remaining steps are similar as we did in the last two situation except we need to draw both sides of shapes separately because the original shape is no longer symmetric. To be specific, after drawing the first shape, the algorithm will recursively check if there is enough room to draw another shape on the orientation side of the center, if there is, draw one shape and move it on length of bounding box towards the orientation and find the best fitting size to be the size of it. Then the algorithm will do the same procedure for the other side of the center. Finally, the algorithm will generate a series of starting shapes with different size and center but they as a whole have the center of the original shape and cover most region of the original shape.

A Concrete Example

Now lets go through a concrete example for a better understanding of how the system works.



(a) A circle

(b) A rectangle (c) A triangle

Figure 1: Input set of the example.

Sample Input

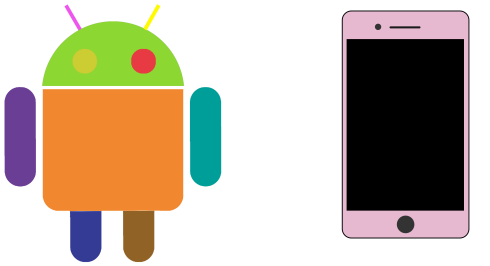
In our example, we give the system three starting shapes: a circle, a rectangle and a triangle. Since my system is not capable of recognizing triangle by now and there are no special fitting algorithm for triangle either, the triangle is supposed to be treated as a known-category shape. The specification of these three input elements are shown in the Figure 1.

Learning Process

The imagery space of my current system is relatively small, but functioning well. The imagery space contains four images as shown in Figure 2 and each picture has a name associated with it. In the learning process, the system will load all images into the working memory and break them into colors according to the colors and analyze the properties of each piece, then record them in the memory. Note the imagery space is considered as a part of the AI system just as the memory we have in our brain, thus these four images are not considered as the input of the system.

Fitting Process

After learning, the system get four concept along with a list of shapes that consist of each concept. The system then try to use starting shapes to fit them. Lets take the robot as the example. The robot has been divided into 10 parts according to its color, we show 3 of them here in Figure 3. Then for each of these 10 parts, we use the circle, the rectangle and the triangle in the starting point set to fit them, respectively, where we use different fitting algorithm as we discussed before. The fitting results for the part A as an example are shown in Figure 4. The algorithm then picks the best fitting solution, the rect-



(a) A robot (b) A iphone

(c) A tree (d) A cat

Figure 2: Picture of animals

angle fitting solution in this example, to be the fitting result for part A, where we use the picking method as we discussed above. Then the algorithm will find the best fitting solution for every part of the robot. Then combine them into one shape as the best fitting solution for the image robot. After the algorithm find all best fitting solutions for every image in the imagery space, it enters into the final picking process.

Picking Process

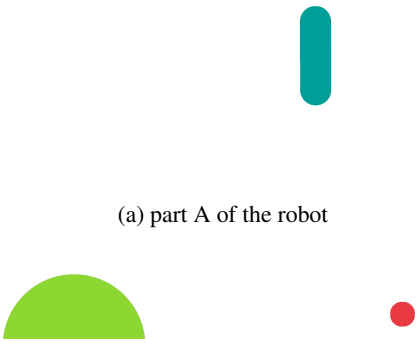
Before entering the final picking process, the system has already find the best fitting solution for every images in the imagery space. So in the picking process, the system will simply calculate the measurement for fitting performance we discussed before, then pick top three best fitting shapes to be the output of the system. The four best fitting shapes for the example input are shown in Figure 5. And the system finally picked the robot, the iphone and the tree as the output.

Discussion

Performance on problem sets

I tested my AI system on seven set of input shapes and the result is promising. Since the system will only display top-three best fitting shapes, which means it will omit one bad fitting shape, all of the output result is acceptable. Here I will show you 4 good fitting result and 2 bad fitting result and explain why it fails.

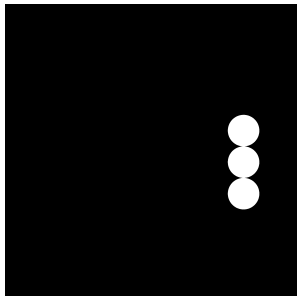
Figure 6 shows the four good fitting results that can be recognized the concept of the shape by human. Figure 6a shows a fitting shape of using only circle to fit all the part of robot, it's not easy but we can recognize the robot from the shape. Figure 6b shows a fitting shape of using only rectangles to fit



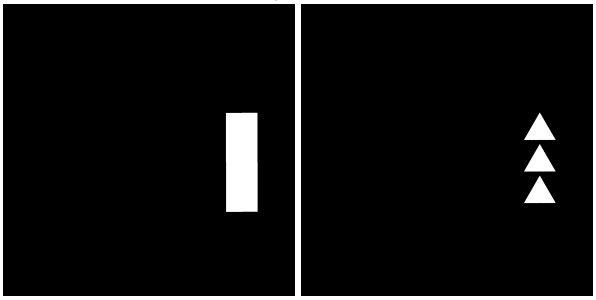
(a) part A of the robot

(b) part B of the robot (c) part C of the robot

Figure 3: Parts of the robot.

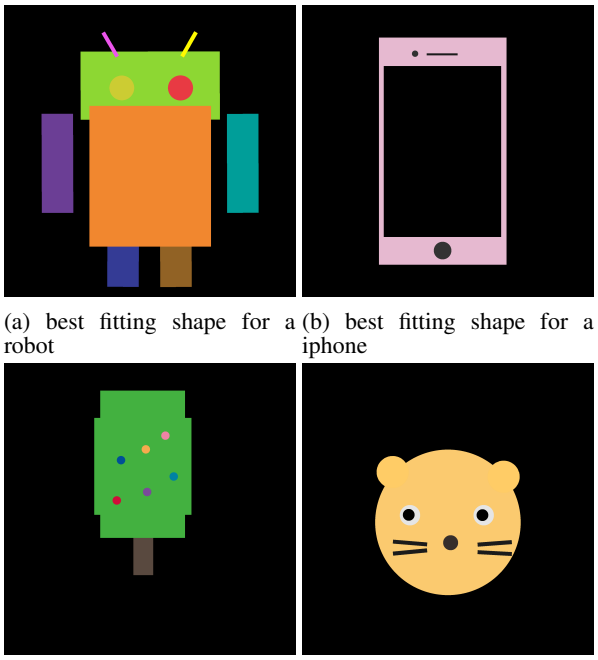


(a) Using circles to fit



(b) Using rectangles to fit (c) Using triangles to fit

Figure 4: Fitting result for the part A of the robot.



(a) best fitting shape for a robot (b) best fitting shape for a iphone

(c) best fitting shape for a tree (d) best fitting shape for a cat

Figure 5: Best fitting shapes for every image in imagery space

an iphone, we can easily get the concept of an iphone by looking at the shape. Figure 6c shows a fitting shape of using only triangles to fit a tree, although there might be some imperfection on the fitting of the trunk, that won't effect our understanding of the concept of the shape. And the last Figure6d shows a fitting shape of using circles and rectangles to fit a cat, the shape is recognizable and cute.

Figure 7 shows the two bad fitting results. The first bad fitting example is try to use only circles to fit a iphone, which is hard obviously since the base shape of an iphone is a rectangle and the fitting algorithm for using circle to fit a rectangle is to draw a series of same size circle to cover the rectangle. The diameter of the circles should be the smaller side of rectangle. And apparently there is no room for more circles after drawing one.

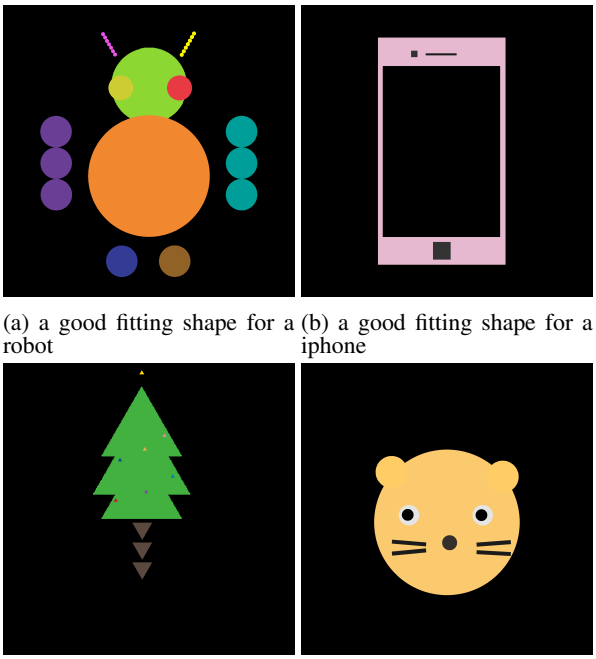
The second bad fitting example is try to use only triangles to fit a cat. It's actually better than the first example because at least we can recognize the cat, but the looking of the cat is not that good and the reason is similar to the first example.

In conclusion, it is hard to create a meaningful and creative images using only one starting shape.

Problems and Improvements

My current AI system is still a beta version, it can complete some of visual critivity tasks and have a good performance of it, but there still be much space for improvements.

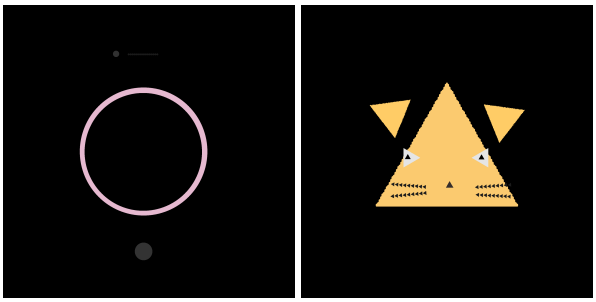
Imagery Space It is quite obvious that based on the current design of my system, the larger imagery space the system maintained, the more creative the system could be. However, unfortunately, the imagery space of my system has only four



(a) a good fitting shape for a robot (b) a good fitting shape for a iphone

(c) a good fitting shape for a tree (d) a good fitting shape for a cat

Figure 6: Good fitting shapes examples



(a) a bad fitting shape for a robot (b) a bad fitting shape for a iphone

Figure 7: Bad fitting shapes examples

images by now, although it has a good overall performance just based on these four images, but it could be a lot better if we add more images into that space. And the system thus has an unlimited potential of creativity.

Preprocessing As you might noticed, the images in the imagery space are preprocessed by dividing into different parts and labeled by different colors, the purpose of the preprocessing is to make it easier for later breaking down and image analysis in the learning process. This preprocessing stage can also be completed by the system by using some image processing algorithms and a proper logic. The system then can automatically preprocess the raw image in the database and learn from real-world image.

Shape Recognition The system is capable of recognizing circles and rectangles and thus can define some shape specific fitting algorithms to get a better fitting performance. If our system can recognize more shapes such as triangles, then we can define more shape specific fitting algorithms to improve overall fitting performance.

Acknowledgments

I really appreciate the help from Professor.Kunda for the guidance of this project, it really helps me clear my mind and focus on the essential problem of this project.

References

Kim, K. H. (2006). Can we trust creativity tests? a review of the torrance tests of creative thinking (ttct). *Creativity Research Journal*, *18*(1), 3-14. doi: 10.1207/s15326934crj1801_2