

Abstract:

This report details my attempts at creating a computationally creative system that will produce artwork made of shapes using an image as a template. It is implemented with JavaScript utilizing the p5.js library. The sketch that is created using shapes allows for manipulation from the user to further boost creativity. I will discuss the idea I came up with to create art from images as well as background of similar projects. I will analyse the design, methods and choices made throughout development. I will provide examples of art produced from different input parameters and images. Finally I will provide an evaluation of the art generated and the system behind it, with final notes on future improvements.

Introduction:

This program takes a users image as well as all of their parameters such as shape type or size used to allow for own artistic design and sketch manipulation. It does this by going through the image in rows and columns of these shapes in the most standard form e.g if using 20 pixel squares it will place a square with 20 by 20 and place the next square at the start of the shapes end. When using the fill option it will generate the image and then go through pixel by pixel to check for empty space and place shapes in these areas. These shapes will check all the pixels within the shape and find the most common pixel colour and set the shapes colour to that colour. It will then use mostly random functions to act upon the parameters which is useful in this scenario as it can produce varying results but results in a less intelligent project.

Background:

Similar projects in this area of generative art have explored the use of algorithms and computational methods to create visually compelling artworks. An example that I researched before making mine, was Primitive by Michael Fogleman which is a more sophisticated version of what I aimed to generate. It has all the features that mine has but with a lot more such as the option to animate generated images, more shapes and more sophisticated placement of shapes. I also did comprehensive research into existing techniques used in geometric art generation.

Methodology and design:

Original design how it worked at first

This will take the input image and will analyse the blocks in a set x by x with x being the shape size to place blocks along the x coordinates until it reaches the end then go down a y coordinate and repeat. These shapes will take the colour of the most used colour pixel in its shape. The easiest shape was the square so it was used as a starting point before implementing other shapes.

Changes from original design

From the original design I added extra parameters to affect the generated sketch based on users input which was the main source of creativity in this system. Implementing functions such as fill to fill in the blank space which goes through the image pixel by pixel checking if theres a shape over it if not then it will generate the shape in this location. As well as extra functions such as colour variation, rotation and outlines.

Parameters that can be adjusted:

The input image will be inputted by the user and will fit the image to the canvas and create a sketch atop when the start button is pressed. Selected shape which provides the options for Square, Triangle, Circle or all of them. Size of the shape is the size of the shapes generated or if the random size is checked then it will use this as the max shape size. Theres checkboxes for rotation which will randomly rotate the shapes generated and outlines either on or off. The checkbox Fill will go through the image and find free space then place a shape in that space. Colour variation is the amount that the shape colours can vary from the original colour. The background colour can be selected from its rgb value or it can use the starting image and keep that in the background.

Results:

The following results in the table have been generated using the following parameters with the varying shapes specified the figure title. These will be used as the base parameters for art generation. Other examples that may have varying parameters will be specified. I have provided a range of different images and input to understand the extent of the program.

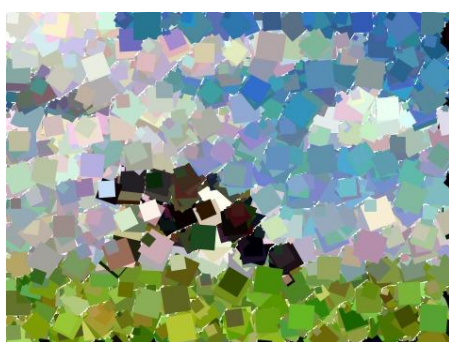
Figure 1. Cow generated using squares



Starting Cow



Cow1



cow2



cow3

Cow2 has outlines off and size 30 as well as colour variation.

Cow3 has random size off, outlines off, fill off, size increased to 30 as well as colour variation

Figure 2. Pizza generated through Triangles



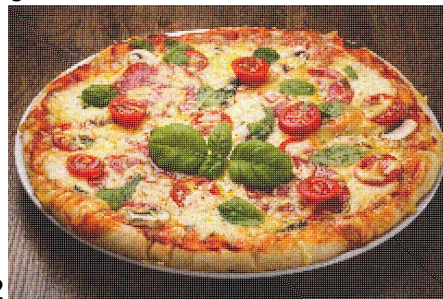
Starting Pizza



Pizza1



Pizza2

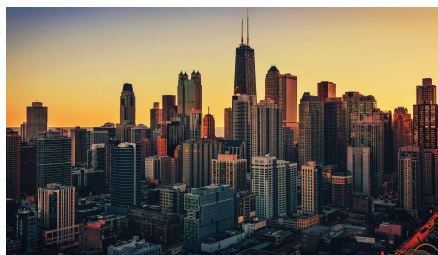


Pizza3

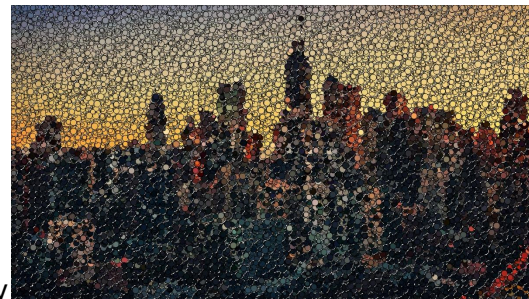
Pizza2 has outlines off and size 100.

Pizza3 has outlines off, random sizes off, rotation off and a size of 5.

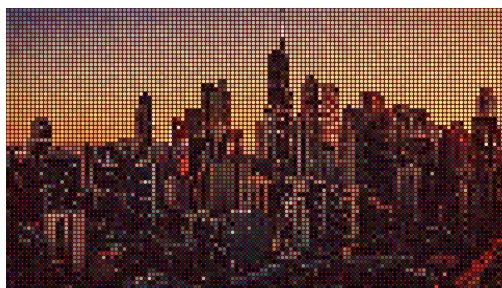
Figure 3. City generated through circles



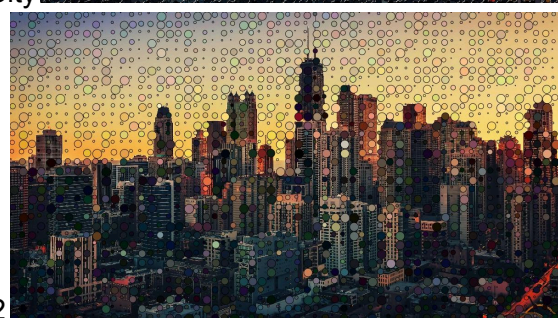
Starting City



City1



City2



City3

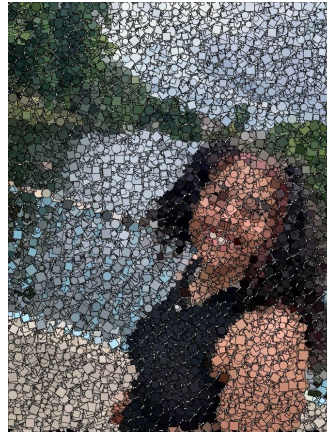
City2 has random sizes off, fill off, colour variation of 20, size of 10 and background colour is red.

City3 has fill off, colour varion of 20 and background set to the background image.

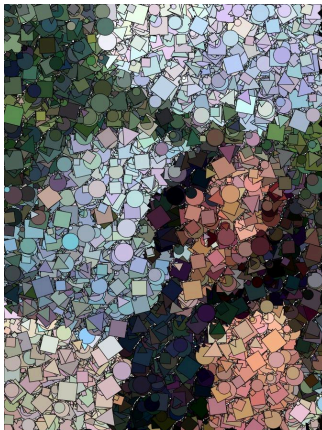
Figure 4. Generating people with all the shapes



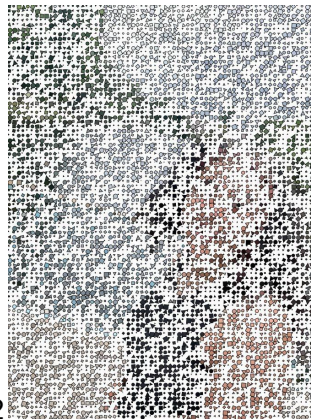
Starting Person



Person1



Person2



Person3

Person2 has a size of 40 and a colour variation.

Person3 has fill off a size of 10, fill off and a background colour of white.

In figure 1 this produces a variety of cows in a field with squares using different parameters to change how the image appears. It shows that when the sizes are smaller it creates more accurate sketches and with bigger shapes it creates a more blocky image which is more of an abstraction of the starting image. Using squares that are size 1 without any other parameters it will recreate the image exactly the same as the squares are the same size and colour as the pixels. Figure 2 is creating a pizza out of triangles and the same rule applies when the shapes are larger its less accurate however, it still resembles a pizza. When using triangles without fill there tends to be a lot of blank space as its placing them in the same columns as it would a square. Figure 3 is creating a detailed city using circles which due to the colours and details it may be somewhat unclear. Although, it still resembles the image there can be ways to make it more accurate. Random sizes affects the image a lot if fill isnt on as it leaves a lot of empty space, but the user can choose a background colour or to have the image as the background to improve how it looks. In Figure 4 it uses all the shapes to generate a picture of a person which is clear that its a person but will lack some of the facial details. If the size of the shapes is large it may not show the teeth from the smile but if the size is smaller it can show more of the details.

Evaluation:

This program does do as intended converting images into sketches of shapes, which can look good work that resembles the starting image. However, the does this mean the generations are actually good, or are they creative. Below I will try to analyse this.

Skill

The system has limited skill as it uses a lot of random inputs and relies on input from the user to create a more interesting image. However, it does do a decent job in creating varying images with the same inputs and the users creativity is used to boost projects creativity. Although, the sketches all have one rough style due to all the shapes being generated the same way.

Appreciation

This is the most lack of section due to the systems level of thinking being limited. Analysing the image pixel by pixel, to check if its been drawn over or its colour is. It doesn't critically assess the image and place the shapes accordingly using advanced methods such as edge detection or colour grouping which would help take this project to the next level and if I had more time in the future I'd plan to implement this.

Imagination

Due to the range of randomness it can create a different sketch every time but still represent the same image. Therefore it can show the most basic form of imagination but where it uses more is based on the users imagination and their goal for the sketch. Changing the image over and over until they are happy with the result, altering the range of parameters to do so.

Conclusion:

Overall, this project was successful in completing its end task goal. However, it could be done in better ways using more computational methods to create more complex images as well as to increase the systems creativity. It allows for a lot of input from users to edit the sketches and hope they are generated to customise images as intended. It generates sketches that can resemble the starting image or completely different to where its not recognizable.

Improvements:

The main improvement I wanted to add was to use advance computational creative methods such as edge detection and colour grouping to create more advanced sketches that will resemble the images more. This will also help turn the project into a creative project. I also wished to add a textbox where uses could type for example a cow and it would use OpenCV.js to generate the image and then sketch that image. Another future improvement I want is so that it will always fill the screen and fix the blank space issues with random sizes, then change fill to overlapping to enable or disable this. Then I can use the overlapping to make the colour combine where it overlaps to create new colours and patterns in the sketch. Another future improvement I wish to add if i had more time is to generate images in a pattern format such as a spiral to make it more creative and less resemble the starting image. My final improvement is to take input from the users mouse to physical draw shapes onto the image or delete shapes that may not be placed where the user wants. These will give the user more access to creative skills which will develop this system further.