Modern Art Rendering

Aneesh Akella
CS 194-26: Computational Photography
University of California, Berkeley
December 13, 2017

Abstract

Using edge detection, gradient alignment, and random cluster generation, digital photos can be rendered as impressionist and pointillist painting-like images. For both algorithms, a photo is taken as an input. In the impressionist algorithm, random rectangles are generated across the image to simulate brush strokes. Edges are detected and rectangles are clipped if said rectangle falls onto an edge. These shapes are filled with the average color within their region and oriented in the gradient normal direction. To generate a pointillist effect, the original picture is subsampled to reduce computations and to reduce high frequencies. K-means analysis is applied to the photo to gather a basis of commonly used colors. The compliments of these colors are generated as well. For each pixel in the subsampled image, three colors are chosen: two with the closest distance to the corresponding pixel and one random color. A larger cluster of colored rectangles will be generated for each subsampled pixel. The color probability distribution is created with the colors closely matching the pixels having a higher probability distribution than the random color. The final products closely match impressionist and pointillist replicas of the input image.

Introduction

Modern art is characterized by a variety of genres including cubism, divisionism, fauvism, surrealism, and minimalism. Here are analyzed algorithms to convert digital images into pseudo-modern-art painting replicas. Particularly, impressionist and pointillist art rendering is analyzed. Impressionism is a 19th century art movement described as an art movement known for small thin, visible brush strokes and its inclusion of movement. Famous impressionist painters include Monet, van Gough, and Picasso. Pointillism is another 19th century art movement characterized by small, distinct blobs of color applied in a pattern to form an image. Famous pointillist painters include van Gough, Seurat, and Signac.

Impressionist Art Rendering

Brush Stroke Generation

To create a painting on a canvas, brush strokes have to be made. These brush strokes are simulated as quadrilaterals created at random points across the image. The number of random points,, is determined by dividing the area of each brush stroke by the area of the image. This alone, however, can leave holes in the image since not every point in the generated image may be filled. The number of true brush strokes should be multiplied by a random constant, c , that is greater than 1 that will ensure all pixels–or a strong majority–are filled. The total number of brush strokes is $n_{rand} \times c$.

Figure 1: Original Image

Figure 2: $\theta = 30$ and No Clipping

Edge Detection and Brush Stroke Clipping

The saying ”Draw within the lines” is conserved in this algorithm. To ensure the algorithm follows this axiom, the canny operator is applied to the input image. If the rectangle falls within an edge in the image, the rectangle is cut at the edge of the line. For each rectangle defined by points

A,B,C,D that stretches across a line, two points—E and F–corresponding to the two intersections of that line with the rectangle are created. If the center used to generate the original quadrilateral falls within the quadrilateral ABEF, a brush stroke will be generated at ABEF instead of ABCD. Otherwise, a brush stroke will be generated at quadrilateral CDEF.

Figure 3: $\theta = 30$ and Clipping

Gradient Alignment

In actual paintings, artists vary the direction of their brush strokes to simulate movement and texture. The impressionist painting algorithm takes this into account. For each pixel in the image, the y-gradient and the x-gradient is calculated to determine the angle orientation. The brush stroke is then drawn in the direction normal to this orientation.

Figure 4: Gradient Normal Direction and Clipping

Brush Stroke Size

This algorithm can be parameterized through brush stroke size. The smaller the brush stroke size the better the details of the image. However, the problem lies in computation time: the smaller the brush stroke size, the more points, and herego a longer runtime.

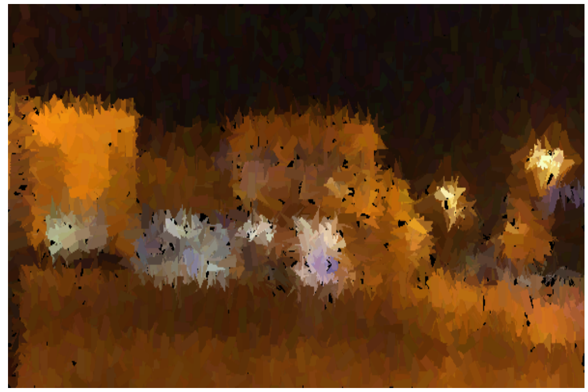

Figure 5: Brushstroke Area = 40 pixels^2

Figure 6: Brushstroke Area = 10 pixels^2

Pointillism Art Rendering

Subsampling Image

To reduce computation time and high-frequencies, the image is subsampled. Computation time for time consuming computations such as K-means color determination and cluster generation will be decreased.

Figure 7: Original Image

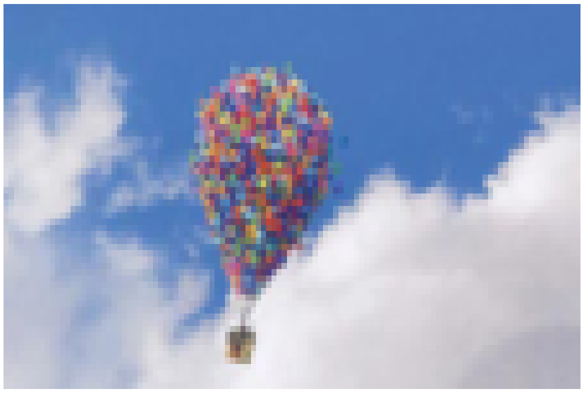

Figure 8: Subsampled Image: 4 times smaller

K-Means Color Determination

Seurat and other pointillist painters were known for their limited use of color. Some of Seurats paintings only include 8-11 different colors. So, a K-means analysis is performed on the pixel values of each image to minimize the colors utilized. When the 8 primary colors are determined, their brightness should be enhanced to give more of an artistic flare. These colors are converted from RGB to HSV. H determines the hue or the color represented by an angle. S determines the saturation and V determines the overall brightness of the pixel. The S and V should be incremented by about 1.2 of its current value or not at all if these channels are already at their maximum value. The colors in the HSV are now transformed back into the RGB space.

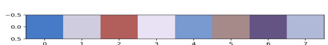

Figure 9: K-Means Cluster Colors

Complimentary Colors

Furthermore, Seurat and other impressionist painters were known their use of complimentary colors. They would mix in these colors to enhance contrast. As described before, H represents the color wheel in terms of angle. A compliment color is therefore positioned 180 degrees away from the original color. So, the complimentary colors for the original colors can be quickly determined.

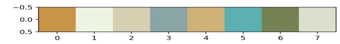

Figure 10: K-Means Compliment Cluster Colors

Cluster Color Determination

For each pixel in the subsampled image, a cluster is generated. To determine the colors of each cluster, the Sum of Squared Differences (SSD) of each color and the current pixel value are taken. The two colors with the smallest SSD and a random color are chosen as the basis of this cluster. Seurat added random colors into this painting, necessating its inclusion in the algorithm.

Cluster Size and Location

For each pixel, a square area located within the new canvas is chosen to place the brush strokes. Random points in the square are chosen with a Gaussian distribution. A square area of $4\sigma^2$ will contain 68 % of the points because it is within a standard deviation based radius from the mean in either direction. So, if the height and width of each corresponding area is chosen to be 2σ and 2σ, then each canvas region will be densely packed. If the clusters need to be less dense, the height and width should be increased while keeping the standard deviation constant, herego increasing the distance between each cluster, d_{clust}. Or the standard deviation can be decreased as well, reducing the spread within the cluster while keeping d_{clust} constant. This areas location is chosen respective to the pixel location. For instance, the pixel with position (a, b) in the subsampled image will have a region in the new canvas corresponding to points ($a \times h$, $b \times w$), ($(a+1) \times h$, $b \times w$), ($(a+1) \times h$, $(b+1) \times w$), and ($a \times h$, $(b+1) \times w$).

Figure 11: $d_{clust} = 15$ pixels

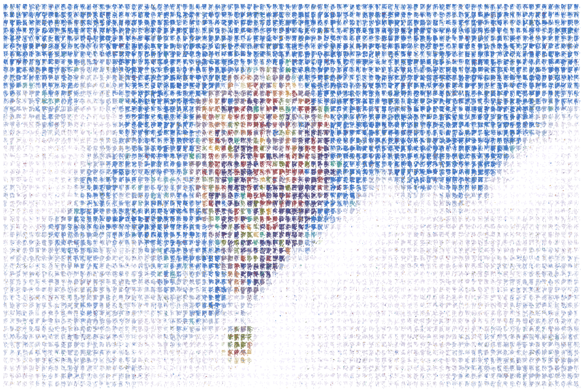

Figure 12: $d_{clust} = 20$ pixels

Brush Stroke Generation

Random points within this decided canvas region are chosen to generate the brush strokes in a matter similar to the impressionist algorithm. Orientation is also decided in a similar way. Each brush stroke within this given area will have the same orientation normal to the basis pixel.

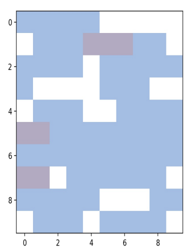

Figure 13: Canvas Portion Displaying Aligned Brush Strokes in Above Image

Number and Color of Brush Strokes

The probability distribution is determined in a matter such that the color of a strong majority of the brush strokes will be from the color with the lowest SSD, a few brush strokes will be from the color with the second lowest SSD, and potentially zero or a minimal number will be from a random color. An chosen value, A, is used to determine the maximum number of brush strokes with a given cluster. To determine how many brush strokes are actually utilized, the intensity of each pixel is calculated. An intensity value of 0 will correspond to the maximum number of brush strokes while an intensity value of 1 will correspond to no brush strokes. This is because an intensity value of 0, all black, necessitates strong aggregates.

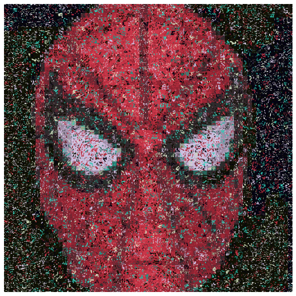

Figure 14: Constant Intensity, I = 1

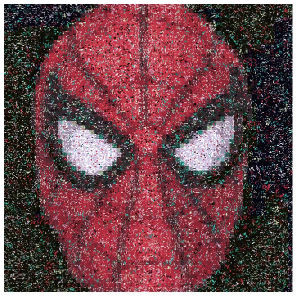

Figure 15: Intensity-Mapped (Note Difference in Eyes)

Transparency

An alpha channel is added to the brush strokes to provide transparency as a feature.

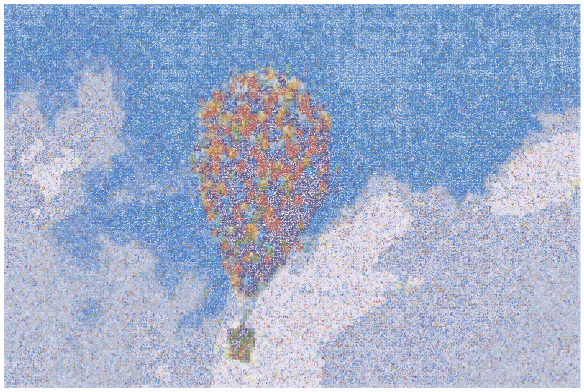

Figure 16: $\alpha = 1$, d_{clust} = 10 pixels, Intensity-Mapped

Figure 17: $\alpha = 0.5$, d_{clust} = 10 pixels, Intensity-Mapped

References

- Hong. Yanshu. Liu. Tiffany. ”Create Pointillism Art from Digital Images”. Web. 9 December 2017.
- Litwinowicz, Peter. Processing images and video for an impressionist effect. SIGGRAPH (1997).

Acknowledgements

I appreciate the guidance of Professor Efros and would like to extend additional graditude towards the CS-194-26 for instructing an enjoyable course.

Appendix

I have posted more images online. Please visit `https://inst.eecs.berkeley.edu/~cs194-26/fa17/upload/files/projFinalProposed/cs194-26-aeh/` for more photos.