Create Pointillism Art from Digital Images

Raghav Sharma

Indian Institute of Technology, Ropar
Electrical Engineering Department
2015eeb1071@iitrpr.ac.in

Abstract

Pointillism is a technique of painting in which small, distinct dots of color are applied in patterns to form an image. The technique relies on the ability of the eye and mind of the viewer to blend the color spots into a fuller range of tones. Combining this artistic inspiration with the techniques of digital image processing, an image-processing algorithm is implemented that creates pointillism art from ordinary digital images.

Introduction

Pointillism artists take advantage of how the human visual system blends small discrete dots of color into other colors. Therefore, if a viewer looks at a pointillism painting up close, the painting looks like a bunch of randomly colored dots, but if he or she steps back, the dots blend together to form colors, which make up an image.

Algorithm

The proposed pointillism algorithm can be broken down into multiple steps. The general work-flow is to first create a color palette, then convert every pixel in the original image into a cluster of dots with these specific colors and finally paint these dots onto the canvas.

1. $Preprocessing$: The main aim of this process is to reduce the amount of computations in the pointillism algorithm. The image is filtered with a low pass normalised gaussian filter to reduce aliasing that may result from the subsequent subsampling. The downsampled factor is determined by the size of the dot cluster upon image generation.

2. $ColorSelection$: After downsampling the image, the colors used to create the pointillism image are identified in this step. The total number of colors used in this algorithm is 16. Seurat, the father of pointillism art is known to use atmost 11 colors in his paintings. Initially, 8 primary colors are determined using K-means clustering algorithm (used in unsupervised machine learning). This algorithm is used so as to make sure that our final image looks similar to the original image.

   (a) K-Means Clustering Algorithm: It begins by picking 8 random locations in the image as the initial centroids. Then for each pixel in the image, the algorithm assigns it to the centroid with the shortest Euclidean distance to the RGB values of that pixel. New clusters are then formed by classifying each pixel with its assigned centroid. A cost function is defined that sums the distances between each pixel color values and its centroid color values. The algorithm repeats until the cost function converges.

   (b) RGB to HSV color space transformation: In the RGB color space, brightness and saturation of a color are difficult to manipulate because these characteristics are embedded in all three color channels. The HSV color space separates color type (hue), saturation and brightness (value) into three separate components. Therefore, the RGB values of the 8 primary colors are converted to values in the HSV color space.

   (c) Color Saturation and Brightness Boosting: In the HSV color space, after hit and trial method, the parameters are tweaked a bit so as to provide brighter colors for the resultant pointillism image.

      i. Kcolor = (R, G, B) to (H, S, V)
      ii. H' = H
      iii. S' = pow(S, 0.75) + 0.05
      iv. V' = pow(V, 0.75) + 0.05
      v. (H', S', V') to (R', G', B')

   (d) Complimentary Colors: In modern color theory, the use of complement colors is known as color juxtaposition. Color juxtaposition is described to be when two colored areas are observed to be quite close to each other, no matter in space or in time, each of the colors will shift its hue and lightness. While Seurat used a limited color palette, analysis of his painting shows that he used complimentary colors to enhance features. This algorithm encorporates the use of complimentary colors by enhancing the 8-colored primary color palette to that of 16 colors. The color complements are calculated in the HSV color space by shifting the hue of a primary color by a random degree uniformly distributed from 0 to 180 degrees.

3. $ColorTransformation$: In the final result image, clusters of dots are used to represent a single pixel of original image. Each individual cluster consists of three different colors from the 16-colored palette. The third color is a randomly chosen color from the remaining 14 colors so as to mimic the real case scenario of an artist using an unclean brush for individual strokes.

   The intensity map is used to ensure that lighter regions of an image have significantly fewer dots than the darker regions. The use of dot concentration to represent intensity in an image is known as stippling.

   The opacity of each individual dot is set to be less than 100 percent so as to make sure that they appear to blend with each other.

4. $ImageGeneration$:

   (a) Dot Distribution: To create the final image, the algorithm ”paints” the dots onto the blank canvas. Each dot on the final image is part of a dot cluster, which represents a single pixel from the original image. The center of each dot cluster is ”d” pixels away from each other dot cluster center in the resulting image. The dots within a single cluster are distributed via normalised Gaussian Distribution.

   (b) Brushstroke Size and Orientation: Each dot is made up of a kernel that maps out the shape of a brushstroke. The orientation of each block is derived from local gradients. For each pixel, 4 Kirsch operators are applied to find the direction of strongest gradient, to which the brushstroke is then aligned.

5. Future Considerations:

   (a) In the pre-processing step, additional Gaussian filters might be applied to create aesthetically better artwork.

   (b) The lowpass filter could also be applied locally in the original image so as to produce depth of focus in the result.

   (c) Morphological image processing with different shapes of structuring elements could be used to dilate features in the image.

   (d) When the edges are found in the image, complement colors can be strategically placed to outline edges in an image. This may make edges appear sharper because contrasting colors are easier to perceive than brushstroke orientation.

   (e) The selection of the third color can be more systematic to ensure more predictable outcomes.

Acknowledgment

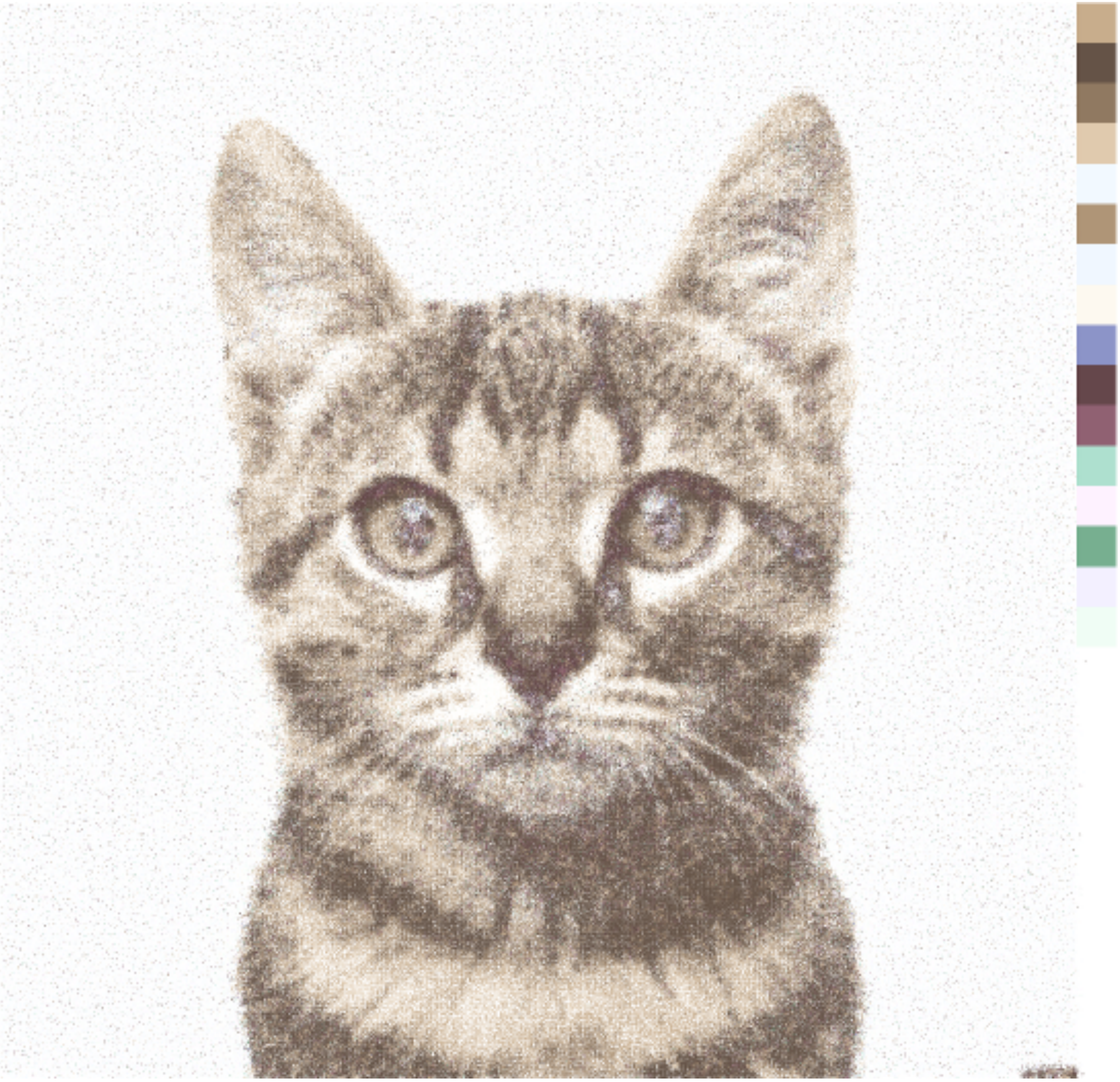
I would like to thank Dr. Puneet Goyal for his unconditional support during the duration of the course CSL 461 - Digital Image Analysis.

References

1. Yanshu Hong, Tiffany Liu. ”Create Pointillism Art from Digital Images”

2. https://mathematica.stackexchange.com/questions/92295/coloured-image-transformation-in-pointillism

3. https://en.wikipedia.org/wiki/Pointillism



Modified Image

Original Image

Modified Image

Original Image

Modified Image

Original Image

Modified Image

Original Image