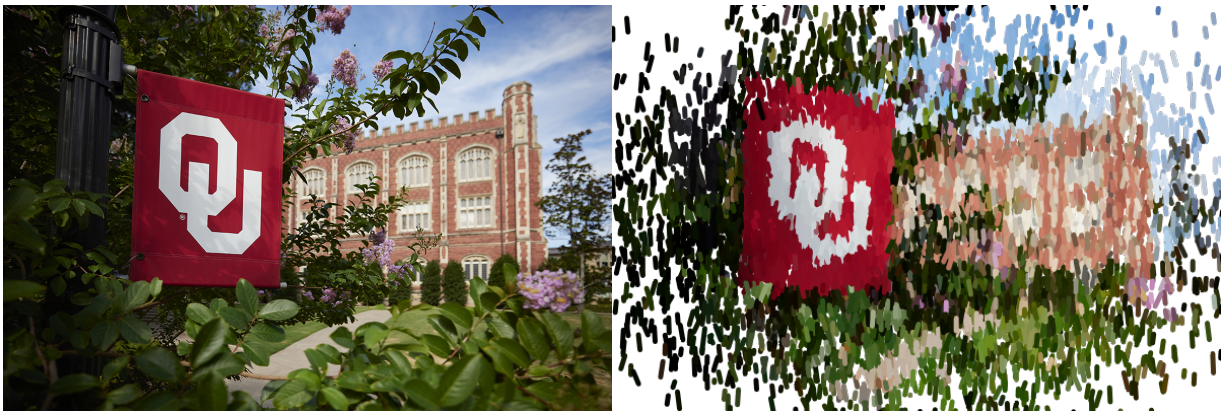
Reimplementation: Paint By Numbers

Matthew Carroll
University of Oklahoma



(a) Source image

(b) Paint-like rendering (thick strokes)

Figure 1: An example of the reimplemented method

ABSTRACT

Producing photo-realistic images has limited degrees of freedom. Producing artistic interpretations of digital images, especially those similar to paintings, was presented in 1990 by Haeberli [1]. Using JavaScript and the Canvas2D API, these effects were in-part reproduced in a portable and simple environment. Most core features of Haeberli's solutions were re-implemented using modern, flexible tools.

1 REVIEWED TECHNIQUES

1.1 Paint By Numbers: Abstract Image Representations

A method for abstract image representations using color sampling and simulated brush strokes was presented.[1] The core novelty of this approach is automatically selecting the color of the paint stroke based on a source image, where the location of the stroke matches the sampled pixel from the source. This allows users to quickly create an interesting abstract representation of the image while maintaining some true-to-source likeness through color.

This sampled color was then painted along an angle at a certen length/size to recreate the effect of a long paint stroke. These strokes are z-buffered such that one stroke always appears above/below another, in a similar way to realistic painting (one cannot paint under another paint-stroke, it is always additive).

Brush strokes were varied in 5 main categories:

- Location - Position of the brush stroke.
- Color - The RGB and Alpha color of the stroke.
- Size - How big the stroke is.
- Direction - Angle of the stroke in the painting.
- Shape - The look of the brush stroke.

Several example usages of the methods allowed these parameters to vary even within an image, such that some details were especially emphasized, while others were more-or-less covered up by broad, fat strokes. The ability to vary the shape of each stroke allowed for especially varied results, from stippling, to oil painting, to a somewhat "cellular characteristic".

1.2 Processing Images and Video for An Impressionist Effect

Litwinowicz builds upon many of the methods posed by Haeberli in order to process videos to produce Impressionist-like video reproductions.[2] Many of the same methods are used, including sampling the source frame in the generation of the reproduction. Linear strokes are taken from each source pixel. These strokes are then cut off or directed according to a few different methods.

- Edge cutoff
- Flow field estimation

Edges of the image are extracted using Sobel filtering. These edges are used to cut off strokes in order to avoid color bleed. Flow fields are estimated throughout the image by comparing colors in consecutive video frames. This allows for estimated strokes to be placed along these flow fields, such that the video's movement can be transitioned using paint strokes. They also use the density of strokes to purge certain strokes, such that certain regions become less dense according to their visual density in the source image.

2 IMPLEMENTATION

The core implementation presented follows the design structure of Haeberli.[1] An interactive, web-based tool to generate painterly images was presented. A web Canvas is present, allowing an interactive painting stroke-by-stroke, or automatically in batches of 100 or 1000 short strokes. Upon each stroke, the source image is sampled for the RGBA values, and the stroke color is set accordingly. Then, a base circular brush texture is painted along the path of the stroke (a straight line in the case of the automatic strokes).

These strokes can vary in all of the core aspects presented by Haeberli as follows:

Figure 2: Thin strokes with consistent angle

Figure 3: Uniformly distributed strokes with random angle

- Location - Manual, uniform, or Gaussian
- Color - Sampled from the source image
- Size - Stroke length: uniform or Gaussian.
- Direction - Custom (manual), uniform, or Gaussian
- Shape - Stroke radius: uniform or Gaussian

An arbitrary image can be uploaded.

3 Analysis

Varied stroke qualities allow for varying aesthetic representations. Figure 2 shows the effect of shrinking the stroke size. The image shows a higher apparent resolution, though the decreased density means more strokes should be added. Conversely, Figure 1b shows a more Impressionist representation of the source image, blurring some features in an artistically appealing way.

Figure 3 shows the effect of uniform stroke distribution. These images were generated using automatic stroke generation, and the level of detail of the uniformly distributed strokes detracts from some realistic artistic representations in the presence of a subtle gradient; Haeberli mainly showed reproductions with strokes centered towards the relevant parts of the images, most likely due to manual painting. Figure 3 also shows the randomized direction of strokes, a feature not implemented by Haeberli. Though this betrays much of the typical painting methods, it reflects an interesting texture, similar to grass or carpet.

The core limitation of this implementation is the lacking of custom textures for brushes. One major source of Haeberli's unique aesthetics are more painting-like stroke textures, which better reproduce a realistic abstract representation. Some work was done to import custom paint stroke textures, including realistic/high-resolution textures.

Further techniques including dynamic angle of paint strokes according to features such as a preset map or the gradient of the source image were left unimplemented. Such features allow for some more realistic reproductions of paintings, but don't feature as the core novelty of this paper. Additions may be made to allow for these further features.

4 Utility and Applicability

Modern day stylistic representations of images typically take more advanced approaches. Modern methods utilize Artificial Neural Networks to represent images in a paint-like texturing. These methods allow for the specific tuning of the model to emulate a specific artist or style, allowing for a more recognizable reproduction. This specific, recognizable representation is a drawback of Haeberli's methods, aside from the very most obvious and broad styles like Pointillism and Impressionism. Modern methods are also capable of recomposing images, moving segments around the relative canvas, allowing for the source image to be more strongly recomposed.

These methods are however limited in their manual methods. Haeberli and Carroll's methods allow for manual, stylistic painting of these reproductions, allowing for deviance from predefined or historical styles and allowing for stylistic omissions. Given that most of artistic success is attributed to the artist, keeping this manual intervention has allowed for more longstanding usage in artistic compositions and these methods remain in the form of many graphics programs today.

Haeberli's methods are also computationally cheap and require no pre-trained machine learning models. They can be done on minimal hardware and dynamically using conventionaly programming methods.

5 Conclusion

Throughout the process of re-implementing these (nowadays) ancient stylistic graphics techniques, I have gained an interest in computer-generated art. Approaches presented to reproduce stylistic, especially Impressionist, versions of images have a longstanding part in many video and photo editing software products. These initial approaches at stylistic rendering are relatively easy to reproduct using modern technologies, but it reveals the roots of much of artificial/stylistic rendering techniques used today in movies and videogames.

Supplemental Materials

The repository for this implementation can be found at:
`https://github.com/matthewCmatt/paint-by-numbers/`

References

[1] P. Haeberli. Paint by numbers: Abstract image representations. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pp. 207–214, 1990.

[2] P. Litwinowicz. Processing images and video for an impressionist effect. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 407–414, 1997.