



ISSN: 0067-2904

Painterly Rendering by Using K-mean Segmentation

Raad Falih Hassan Foadi

Astronomy and Space Department, College of Sciences, University of Baghdad, Baghdad, Iraq

Received: 20/9/2022

Accepted: 4/1/2023

Published: 30/4/2023

Abstract

The problem statement discussed in this paper is a new technique for the presentation of painterly rendering that uses a K-mean segmentation to divide the input image into a set of regions (depending on the grayscale of the regions). Segmenting the input image helps users use different brush strokes and easily change the strokes' shape, size, or orientation for different regions. Every region is painted using different brush kinds. The properties of the brush strokes are chosen depending on the region's details. The brush stroke properties, such as size, color, shape, location, and orientation, are extracted from the source image using statistical tools. The number of regions is set up manually and depends on the input image. This method allows the user to apply different painting styles to different regions and create a painterly rendering. MATLAB is used to render the images into paintings.

Keywords: Painterly rendering, K-means segmentation, Image segmentation.

تصيير الصورة لرسم بواسطة استخدام تجزئة الصورة

رعد فالح حسن فوادي

قسم علوم الفلك والفضاء، كلية العلوم، جامعة بغداد، بغداد، العراق

الخلاصة

بيان المشكلة الذي تمت مناقشتها في هذه الورقة البحثية هو تقنية جديدة لتصيير الصور الفوتوغرافية الى رسم باستخدام طريقة خوارزمية التجزئة لتقسيم الصورة المدخلة الى مجموعة من المناطق اعتماداً على التدرج اللوني. هذا التقسيم سيساعد المستخدم على استخدام ضربات فرشاة مختلفة ويتحكم باتجاه الضربة لمنطقة محددة دون المناطق الاخرى. كل منطقة من الممكن ان ترسم باستخدام نوع مختلف من فرشاة الرسم. مواصفات ضربة الفرشاة اختيرت اعتماداً على تفاصيل المنطقة المرسومة. مواصفات ضربة الفرشاة مثل حجم ولون وشكل ومكان واتجاه الضربة والتي استخلصت من الصورة الاصلية المدخلة باستخدام حسابات احصائية. عدد المناطق التي تقسم لها الصورة يتم ادخاله بواسطة المستخدم واعتماداً على الصورة المدخلة. تمنح الطريقة المقترحة في هذه الورقة الحرية باستخدام انماط مختلفة من الرسم لمناطق مختلفة من الصورة المدخلة لتوليد رسم من الصورة الاصلية. في هذا البحث تم استخدام الماتلاب لتصيير الصور الى لوحات فنية.

1. Introduction

Non-photorealistic rendering has become one of the major research subjects in image processing. Non-photorealistic rendering deals with essential topics, including oil paintings, watercolor paintings, cartoons, pen-and-ink illustrations, and other subjects. One of the main purposes of painterly rendering is to convert the image to a painting by placing brush strokes on a canvas to make the final work look hand-crafted.

This research's contribution is to use K-mean segmentation to divide the input image into a set of regions depending on the grayscale of the input image. The next step is using a method to determine the location, size, orientation, and shape of brush strokes. The properties of the brush strokes are taken from the local source image. The steps of this method can be summarized as follows:

1. The image is divided into k-regions by using the k-segmentation method. Each region has pixels that have the same grayscale or are close to it. The user can choose the number to divide the image.
2. Converting the color from RGB to Lab and dithering to place the strokes.
3. The stroke size is chosen carefully over the image canvas. The highly detailed regions are painted with small strokes, while flat regions with low details are painted with large strokes' sizes. This variety of strokes' sizes helps to get a painted image close enough to the actual painting. To control the stroke size, the dithering method is used and will be explained in section 4.3.2.
3. The information of each brush stroke is extracted from the source image, such as location, orientation, and color. These properties are essential to represent the details of the source image to get a perfect painting.
4. Each region is painted separately. The painted strokes do not exceed the boundary of the region. Therefore, the painted regions do not overlap with each other.

In section two, the related literature review is discussed. In section three, the K-mean segmentation method and dithering method will be explained. In section three, the painterly rendering algorithm, in general, is described. Section four will show the results, and section five will highlight the findings of this research.

2. Literature review

Haeberle introduced a new painterly rendering technique that changed the static image to painting using a collection of strokes whose location was determined by the dithering method. Haeberle used four elements: size, shape, color, and orientation, extracted from local image properties, to describe the brush strokes [1]. Litwinowicz introduced an automated algorithm to produce video paintings using strokes aligned to normal image gradients. He depended on Haeberle's work using strokes' location and colors [2]. Hertzman introduced a novel method to paint an image with a series of b-spline strokes. In his method, he imitated artists by drawing large, broad strokes representing the underpainting and then adding more minor strokes to create details [3]. Gooch et al. proposed a new method to segment the image into features, find the average center axes of these features, and then use these axes to paint the brush stroke. Their method helped reduce the number of brush strokes [4]. Chen et al. suggested a particle system that was fully accessed with Compute Unified Device Architecture (CUDA) and accurately and realistically replicated wet-wet paint mixtures [5]. Gatys et al. proposed a machine learning method that used convolutional neural networks to mimic the artistic style of a painting to photography while keeping the original details [6] [7]. Tan et al. and Aharoni et al. presented methods for decomposing images into pigments and

thickness channels for later recoloring and image enhancement. The color of the painted image was extracted from an input image, and each color presented in the image consisted of a weighted mixture of base pigments [8] [9].

3. Theory

3.1 K-mean segmentation method:

The K-mean algorithm is an algorithm that is based on distance. The similarity is estimated by distance. In other words, when two objects have a smaller distance, the similarity is higher and becomes lower when the distance between the objects becomes greater. In this way, the algorithm obtains an independent cluster. The algorithm process consists of the following steps: First, random points are chosen from different regions in the image as cluster centers to form K clusters.

The second step is to calculate the distance to each cluster center for each other region and classify the region nearest cluster center. The third step, calculate the cluster centers of each region again after separating them. Repeat the second and third steps in the fourth step until the procedure is completed and the new center is equal to or less than the chosen threshold. The Euclidean distance formula is used to determine the sum of the squares of the distances from each point in the region to the cluster center, which separates objects into K regions ($K = c_1, c_2, \dots, c_k$), each of which has a cluster center K [10] [11]

$$Mean(c_k) = \sum_{x_i \in c_k} |x_i - \mu_k|^2 \quad (1) [10]$$

The clustering method aims to reduce the square sum mean of distance ($k_c = \sum_{k=1}^k mean(c_k)$). The clustering used in this research involves replacing the original values of the same sort of pixels with their prescribed values (colors), using each color component of the RGB as an input parameter. The resulting regions were presented on an image without being displayed one by one [10] [11].

3.2 Dithering method

Dithering is a technique used to imitate the display of colors not available in the present image's color palette. By combining multiple neighboring hues into a single visible one, dithering tricks the human eye into seeing more colors than are presented in an image. Many dithering functions (such as thresholding, patterning, random dither "Robert's algorithm," ordered dither, and error diffusion dither) can be used to generate a dithered image. The error diffusion dither techniques distribute the error (produced by quantizing a pixel) over surrounding pixels, and the value of a pixel is compared to a single 50% gray threshold (that is, when black is 0 and white is 255, each pixel is compared to 128). Based on that comparison, the pixel is either white or black. Following the acquisition of the dithered pixel, the algorithm computes the difference between the source and output pixels (by simple subtraction) and then distributes this difference (the "error") over neighboring pixels. In other words, the algorithm modifies the input image as it filters it. The error is divided between pixels, as shown in the following Figure (1) [12] [13].

It begins with an input grayscale image in floating point format with pixel values ranging from 0 (black) to 1 (white). The error, e , from one pixel is added to its neighbors at each step using the following algorithm, see Figure (2). [12] [13]:

$$I_{acc}(i, j + 1) = I_{acc}(i, j + 1) + \frac{7}{16} e$$

$$I_{acc}(i + 1, j - 1) = I_{acc}(i + 1, j - 1) + \frac{3}{16} e \quad (2) [12]$$

$$I_{acc}(i + 1, j) = I_{acc} (i + 1, j) + \frac{5}{16} e$$

$$I_{acc}(i + 1, j + 1) = I_{acc} (i + 1, j + 1) + \frac{1}{16} e$$

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & \text{●} & \frac{7}{16} \\ \frac{3}{16} & \frac{5}{16} & \frac{1}{16} \end{pmatrix}$$

Figure 1: Spread error over neighboring pixel [12]

4. The painterly rendering algorithm

The process of painterly rendering consists of three steps: the first is dividing the input image into several regions using the K-mean segmentation algorithm. The second step is calculating the dithering for the input image. The white dots of dithering represent the center of the brush strokes. The third step is extracting the stroke properties from an input image, such as orientation, size, color, and shape, which are chosen depending on the region details.

4.1 Number of regions (k)

The user can choose the number of regions manually using K-mean segmentation. The main factor that divides the image is the grayscale (color) of input image pixels. Figure (3) shows different values of k that are used to divide the image into k regions.

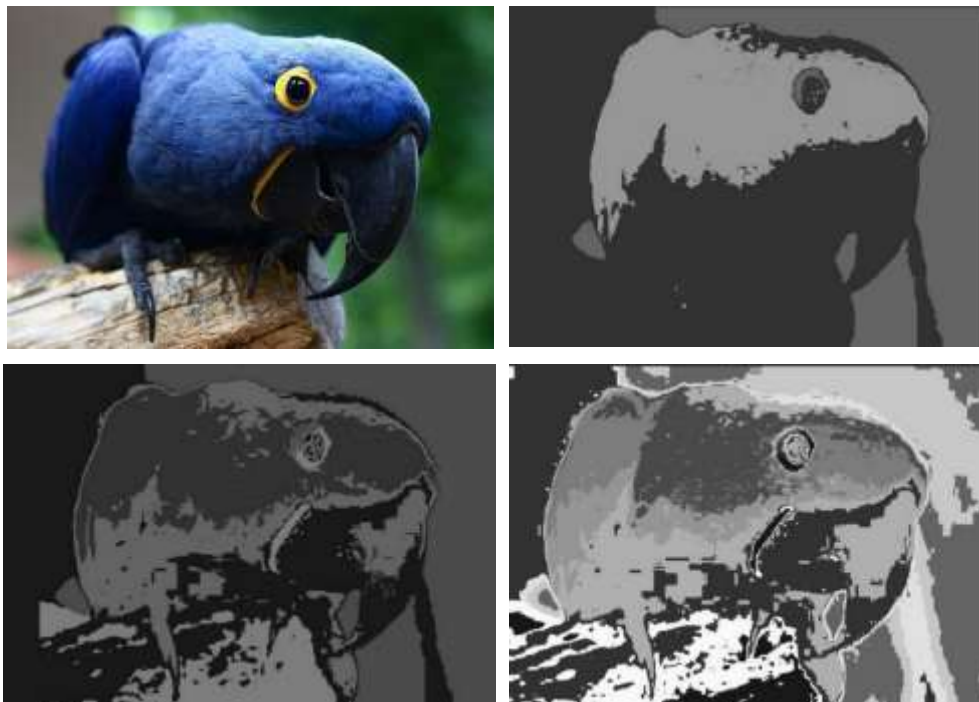


Figure 3: a- The upper left is the original image [14] b- The upper right divides the image to three regions c- The lower left divides the image to five regions d- The lower right divides the image to ten regions.



Figure 2: On the left is the original image [14] and on the right is the dithered image

4.2 Dithered image

The dithering algorithm was applied to the input image; the result is a dotted image (white and black). The process of dithering was explained above in section (3.2). The white dots represent the location of the strokes. The highly detailed region has more white dots than the low detailed regions, as shown in Figure 2.

4.3 Strokes' properties:

4.3.1 Strokes' orientation

To measure the stroke orientation, the gradient was calculated at the center of the stroke (white dot) from the dithering of the input image. Small random values were added to the amount of gradient to make the stroke more realistic. The amount of random addition to the gradient differs from region to region and by user choice. The gradient magnitude and direction were calculated using the following equations:

$$G_x = I(x + 1, y) - I(x - 1, y) \quad (3)$$

$$G_y = I(x, y + 1) - I(x, y - 1) \quad (4)$$

G_x and G_y are the gradients in the x-axis and y-axis directions, respectively.

Applying the Pythagorean theorem to get the gradient magnitude:

$$G = \sqrt{G_x^2 + G_y^2} \quad (5)$$

The gradient orientation can be given as the ratio of G_y to G_x .

$$\tan \theta = \frac{G_y}{G_x} \quad (6)$$

4.3.2 Brush strokes' size

High Pass Filter (HPF) was applied to the input image to determine the size of brush strokes. HPF detects the high details area in the regions. To expand the detected area, the morphological operator was used. The dilation operator works to enlarge the boundaries of the detected area gradually. Thus, the detected area grows in size while holes within the same region become smaller. The centers of the strokes were located in the detected area (high

details area), which was pointed out by the previous process and was painted in small size. At the same time, other areas were painted with large brush strokes size. This process allows the user to keep accurate details of the image without distortion, see Figure (4).



Figure 4: a- Input image [15] b- Image after using HPF c- Image after applies Dilation Operator

4.3.3 Shape of the strokes

Different templates of brush strokes were used; each region was painted using a different stroke shape assigned by the user. Figure (5) shows some of the templates of brush strokes.



Figure 5: Shows different templates of brush strokes [16]

4.3.4 Color of the strokes

The color of the stroke was extracted from the input image. The pixel's color in the input image representing the stroke's center was chosen as the brush stroke's color.

5. Results

Figure 6 shows the result of two images used as examples of applying a painterly rendering algorithm. The input images are shown in Figures 6- a and 6- c. Figures 6-b and 6-d are generated with ten different regions, each with a different brush stroke template. The size of strokes was determined depending on the details of the region to be painted. The color was extracted from the input images using the center pixel of the stroke in input images. The orientation of the strokes was calculated using the gradient plus a small amount of random to give the painted image a hand-crafted touch. Figure 7 shows the result of using different brush strokes and orientation sizes. Figure 7-a shows the input image with size 1009*1458 RGB colored image. Figure 7-b shows the painting image with large brush strokes, and the orientation was calculated by the gradient and adding a small amount of random. The painting

was generated with 20 different regions, and each region has different templates of brush strokes. Figure 7-c shows the painting of the image with small brush strokes, and the orientation was calculated by the gradient and adding a small amount of random. The painting was generated with ten different regions, and each region has different templates of brush strokes. Figure 7-d shows the painting of the image with large brush strokes, and the orientation is random. The painting was generated with ten different regions, each with different strokes. Small strokes give better results but are more like an image than a painted one.



Figure 6: a- The upper left is the original image [14] b- The upper right is a painted image
c- The lower left is the original image [17] d- The lower right is a painted image.

Figure 8 shows the effect of the number of regions to which the image was divided. Figure 8-a shows the input image. Figures 8- b, c, d, e, and f showed the painted image with 5,10, 15, 20, and 25 segmented regions, respectively. From this figure, one can realize that the painted image has become increasingly better with increasing the number of regions. More details are nicely painted with increasing the number of regions such as lips, nose, and eyes.

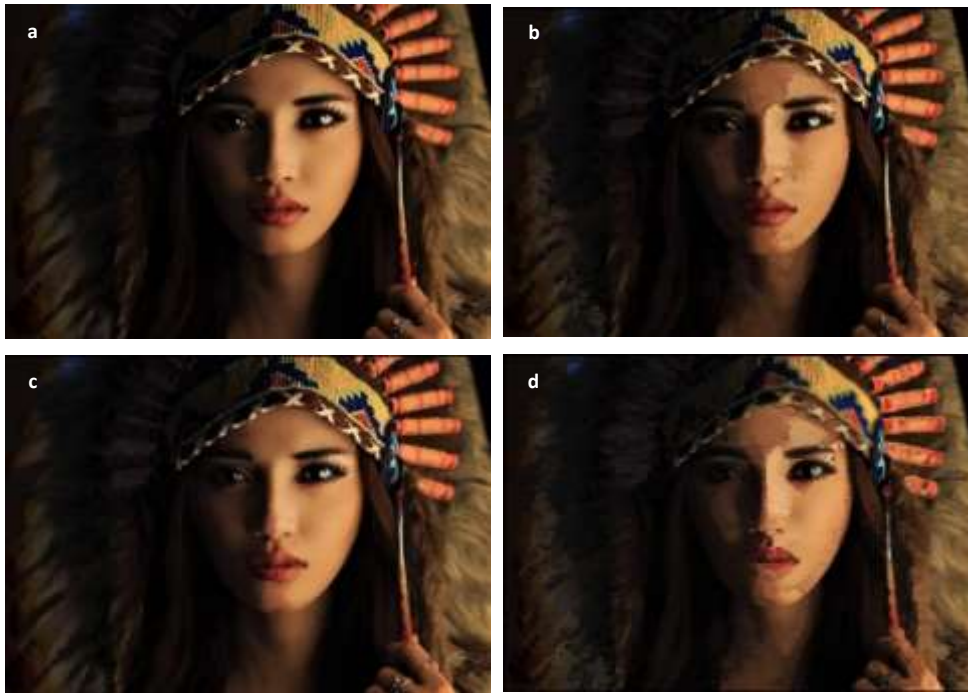


Figure 7: Shows the effect of the size of strokes and the orientation



Figure 8: Shows the effect of segmentation on the painted image. a- Input image [15] b- The painted image with 5 segmented regions. c- The painted image with 10 segmented regions. d- The painted image with 15 segmented regions. e- The painted image with 20 segmented regions. f- The painted image with 25 segmented regions.

6. Conclusion

Using the K-mean segment helps to manipulate how the user can render the image to paint. Therefore, one can choose different templates of brush strokes for different regions. Also, the user can change the strokes' orientation for each region separately. The number of regions affects the final result of the painted image. That is to say that the more the regions are, the better the result is, as shown in Figure 8. This research has also shown that different sizes of brush strokes gave different results (Figure 7). Using HPF with a morphological operator during painting the image helped save fine details (such as nose, lips, eyes, etc.), a matter that made the painted image more refined and realistic. One shortcoming of this research is that this method is time-consuming and requires CPU and memory-intensive processing. Also, the brush types were manually chosen.

7. Future work:

As a future work, this research might consider automatic brush selection for different regions of the image. Also, the time consumption might be reduced considerably

References

- [1] P. Haeberli, "Paint by numbers: Abstract image representations," in *Proceedings of the 17th annual conference on Computer graphics and interactive techniques.*, 1990.
- [2] P. Litwinowicz, "Processing images and video for an impressionist effect," in *In Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997.
- [3] A. Hertzmann, "A survey of stroke-based rendering," *Institute of Electrical and Electronics Engineers*, vol. 23, no. 4, pp. 70-81, 2003.
- [4] B. Gooch, G. Coombe and P. Shirley, "Artistic vision: painterly rendering using computer vision techniques," in *In Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, 2002.
- [5] Z. Chen, B. Kim, D. Ito and H. Wang, "Wetbrush: GPU-based 3D painting simulation at the bristle level," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 1-11, 2015.
- [6] L. A. Gatys, A. S. Ecker and M. Bethge, "A neural algorithm of artistic style," arXiv preprint arXiv:1508.06576, 2015.
- [7] L. A. Gatys, A. S. Ecker and M. Bethge, "Image style transfer using convolutional neural networks," in *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [8] E. Aharoni-Mack, Y. Shambik and D. Lischinski, "Pigment-based recoloring of watercolor paintings," in *In Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, 2017.
- [9] J. Tan, S. DiVerdi, J. Lu and Y. Gingold, "Tan, Jianchao, Stephen DiVerdi, Jingwan Lu, and Yotam Gingold. "Pigmento: Pigment-based image analysis and editing."," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 9, pp. 2791-2803, 2018.
- [10] P. Shan, "Image segmentation method based on K-mean algorithm," *EURASIP Journal on Image and Video Processing*, no. 1, pp. 1-9., 2018.
- [11] S. Jardim, J. António and C. Mora, "Graphical Image Region Extraction with K-means Clustering and Watershed," *Journal of Imaging*, vol. 8, no. 6, p. 163, 2022.
- [12] J. Hughes, . A. van Dam, M. McGuire, D. Sklar, . J. Foley, S. Feiner and K. Akeley, *Computer graphics: principles and practice*. Pearson Education, Pearson Education, 2014.
- [13] Z. Yao and Y. Wan, "A high-performance dithering method for gray and color image quantization," in *In 2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, 2010.
- [14] "<https://www.pinterest.com/pin/561542647262590201/>," [Online].

- [15] " <https://www.wallpaperbetter.com/en/hd-wallpaper-eanxn>," [Online].
- [16] "https://www.hdwallpapers.in/beautiful_eyes_girl_model_is_wearing_black_dress_hd_girls-wallpapers.html," [Online].
- [17] "<https://stock.adobe.com/de/search?k=splatter%20watercolor%20brush>," [Online].