OIL PAINTING EFFECT OF REALISTIC DIGITAL IMAGE

Roshni Sahoo, Joel Mascarenhas, Riya Manu Jacob

St. Xavier's College, Mumbai, India

Abstract

Simulating artistic abstractions of real-life images has become widely popular in the world of photoshop and digital image processing. This is known as non-photorealistic rendering (NPR) which is concerned with giving artistic effects to realistic photos. Our work is based on generating oil paintings of images using a Kuwahara filtering technique. A kuwahara filter is an edge- preserving smoothing non-linear filter that generates color blocks that resemble the blocky texture of oil paintings. These color blocks are generated by calculating the variance of the neighbouring pixel values of any given pixel and replacing it with the mean of the region with the minimum variance. This not only generates the sought after blocky effect but does an excellent job of preserving the edges of the shapes in an image giving it the resemblance of an oil painting.

Keywords: Non- Photorealistic Rendering (NPR), artistic image abstraction, kuwahara filter, oil painting effect, digital image, color image processing, color model

Introduction

The main aim of this project is to convert an image into a painting-like picture using python and by a technique known as Non-photorealistic rendering (NPR) that focuses on allowing the user to access a wide variety of expressive styles for digital art, in contrast to traditional computer graphics, which focuses on photorealism. NPR deals with the representation of painting styles (for example: generating strokes that are similar to hand-drawn strokes), and it often adopts filtering methods to keep the edge features and smooth the unimportant image regions.

These different filtering methods are often used in creating various kinds of applications such as photoshops and in doing so gives their clients' options to convert any image into a painting like picture of anything they want by giving cheap artistic effects in their own hands. An example of such type of filter is the Kuwahara filter. The Kuwahara filter is a non-linear smoothing filter that maintains the overall sharpness of the image or the positions of the edges. The advantage of this filter is that, while most of the filters used for image smoothing are low-pass filters that effectively reduce noise, they tend to result in the blurring of the edges. This problem is entirely eliminated by the Kuwahara filter.

Background

This filter is named after *Michiyoshi Kuwahara, Ph.D* and was proposed in 1976. Initially, it was developed with the purpose of assisting in the processing of RI-angiocardiography images of the cardiovascular system. It is popular in the biomedical field, in the identification of anomalies in noisy images (such as detection of brain tumors), and hence is an important contribution to medicine. Currently, it is used in artistic imaging and photography due to its ability to smoothen the image, sharpen the edges and create a painting effect to a desired level of abstraction.

However, the filter does pose a few and minor disadvantages. For example, it does not consider the case where two regions have equal standard deviations. But this is a rare case as it is difficult to find two regions with exactly the same standard deviation in a realistic image as some amount of noise is always present. The filter is also known to create slight distortions in the image, called block artifacts, in regions of the image that are highly textured. This reduces the smoothness of the image and becomes counterproductive to our aim of creating a painting-like effect.

Theory

The process of Kuwahara filtering involves the division of a pixel grid into four overlapping sub-grids and then calculating the mean and variance for each (as shown in Image). The output value is defined as the mean of the sub-grid that presents the least variation and then assigning the output value to the central pixel of each region analyzed by the algorithm.

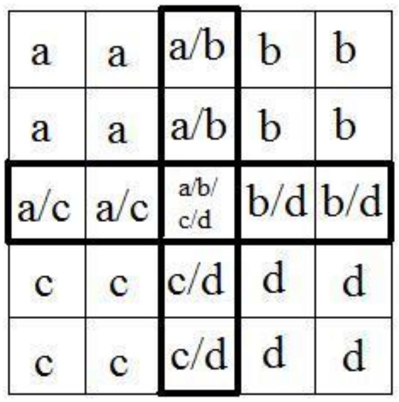


Figure 1: Window used in Kuwahara Filtering (Source: https://en.wikipedia.org/wiki/Kuwahara_filter)

-

In the figure above, we consider a square window of size (2a+1) around a point (x,y) in the image. Then we divide it into four smaller regions: a,b,c and d, with the pixels located on the central row and column belonging to more than one region, causing an overlap between the sub regions. The arithmetic mean and standard deviation of each of the four regions are calculated and that with the least variance determines the value of the central pixel. So the central pixel takes the average value of the area that has least variation.The pixel position with respect to an edge is a main determinant of which region will have the greater standard deviation. For example, if a pixel is located on the dark side of an edge, it tends to take the average value of the dark region. A similar working follows if it is located on the lighter side of an edge. If it is located on the edge, it takes the value of the most homogenous region. Edge-preservation is hence maintained as the Kuwahara filter considers the homogeneity of the regions.

The window size also has a part to play in the resulting effect we wish to produce. Smaller the window, greater the extent of detail preservation. For a more abstract image, we use a larger window size. For the application of the Kuwahara filter to Color images, as the one done in this project, the filter is applied to each of the RGB channels separately, and then recombined to give the resultant painting-like image.

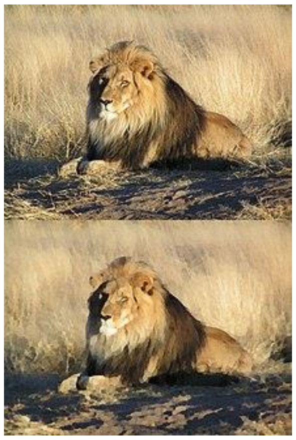
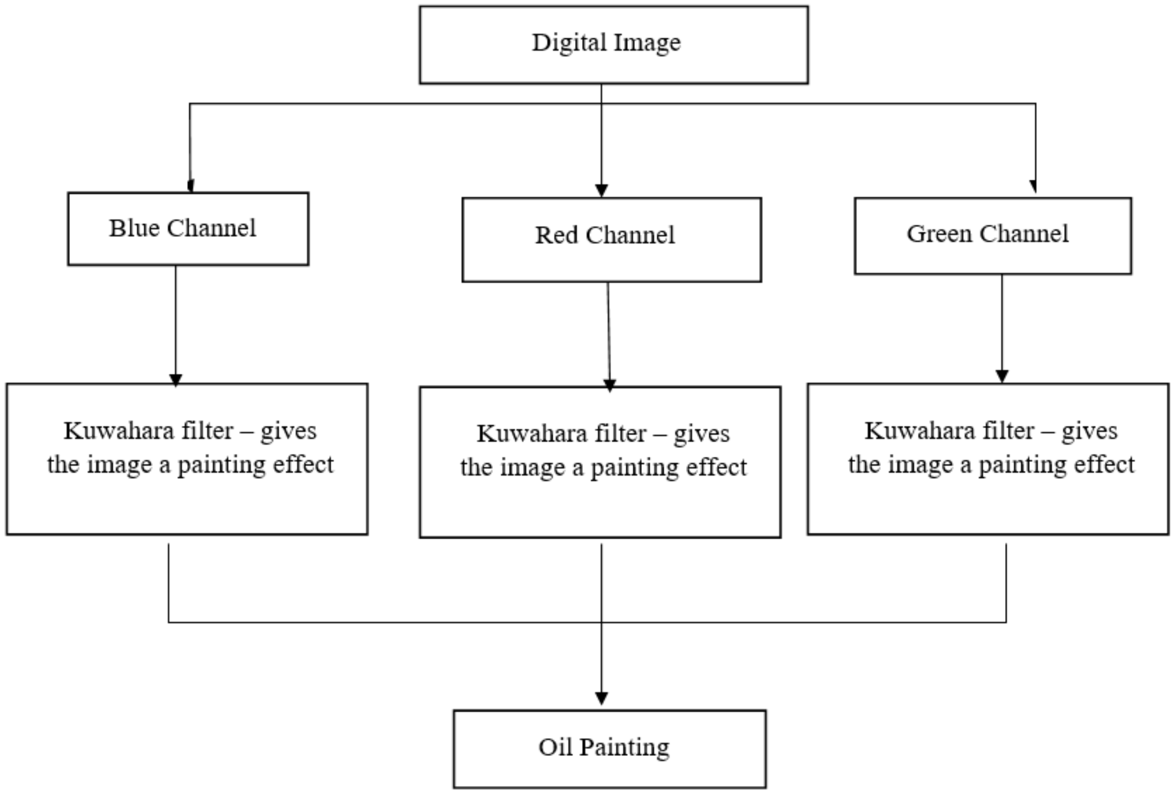


Figure 2: Example of the application of the Kuwahara Filter to a color image for NPR (Source: https://en.wikipedia.org/wiki/Kuwahara_filter)

Computational

Flowchart


Digital Image
Blue Channel
Red Channel
Green Channel
Kuwahara filter – gives the image a painting effect
Kuwahara filter – gives the image a painting effect
Kuwahara filter – gives the image a painting effect
Oil Painting

Algorithm

1. Take the digital image and split into its Red, Green and Blue components
2. Write the code for implementing the Kuwahara Filter
    i. For any given pixel "p" in the R/G/B component of the image, search the 4 quadrants of any kernel size of your choice around pixel "p" and store the intensity values of the pixels in 4 different lists.
    ii. Calculate the mean and variance of the values in each of the 4 lists.
    iii. Choose the list which has the least variance and replace the pixel "p" with the mean of that list which is essentially the mean of the values of the pixels that have the least variance.
3. Apply this filter to the Red, Green and Blue components of the image.

4. Merge the filtered red, green and blue components to get the final painting effect using np.dstack

Code

```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
import cv2
from google.colab.patches import cv2_imshow
import random
from google.colab import drive
drive.mount('/content/drive')
img = cv2.imread('/content/drive/MyDrive/DIP/project/painting.png')
cv2_imshow(img)
print(np.shape(img))

#Kuwahara filter implementation
def kuwahara(x,a):
       # x = image on which filter is used, (2a + 1) = size of window
  a = int(a)
  n,m = x.shape
  x_kuw = np.zeros((n,m))
  for i in range(0,n):
    for j in range(0,m):
      quad_1 = []
      quad_2 = []
      quad_3 = []
      quad_4 = []
      for dx in np.arange(-a,1):
        for dy in np.arange(-a,1):
          if 0<=i+dx<n and 0<=j+dy<m:
            quad_1.append(x[i+dx,j+dy])
      quad_1 = np.array(quad_1)
      m1 = np.mean(quad_1)
      v1 = np.var(quad_1)

      for dx in np.arange(0,a+1):
        for dy in np.arange(-a,1):
          if 0<=i+dx<n and 0<=j+dy<m:
            quad_2.append(x[i+dx,j+dy])
      quad_2 = np.array(quad_2)
      m2 = np.mean(quad_2)
      v2 = np.var(quad_2)

      for dx in np.arange(-a,1):
        for dy in np.arange(0,a+1):
          if 0<=i+dx<n and 0<=j+dy<m:
            quad_3.append(x[i+dx,j+dy])
      quad_3 = np.array(quad_3)
      m3 = np.mean(quad_3)
      v3 = np.var(quad_3)

      for dx in np.arange(0,a+1):
        for dy in np.arange(0,a+1):
          if 0<=i+dx<n and 0<=j+dy<m:
            quad_4.append(x[i+dx,j+dy])
      quad_4 = np.array(quad_4)
      m4 = np.mean(quad_4)
      v4 = np.var(quad_4)

      variance_list = [v1,v2,v3,v4]
      mean_list = [m1,m2,m3,m4]
      minimum = min(variance_list)
      pos = variance_list.index(minimum)

      x_kuw[i,j] = mean_list[pos]
  return x_kuw

#applying kuwahara filter to the B, G and R components individually
b_kuw = kuwahara(img[:,:,0],6)
g_kuw = kuwahara(img[:,:,1],6)
r_kuw = kuwahara(img[:,:,2],6)

#merging R, G and B components
painting = np.dstack((b_kuw,g_kuw,r_kuw))
cv2_imshow(painting)                                #displaying final painting
```

Results

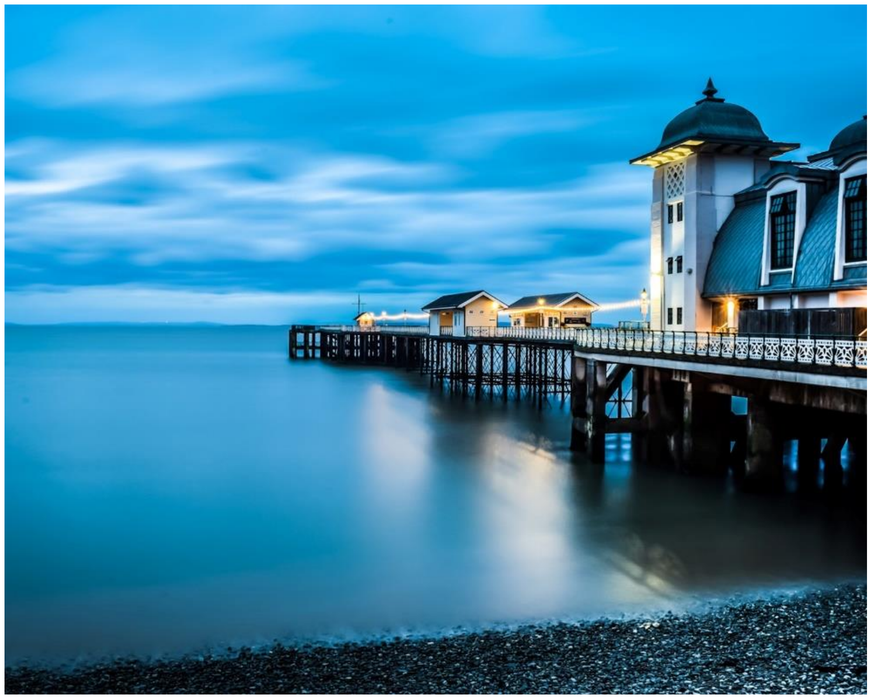


Figure 3. Realistic Image

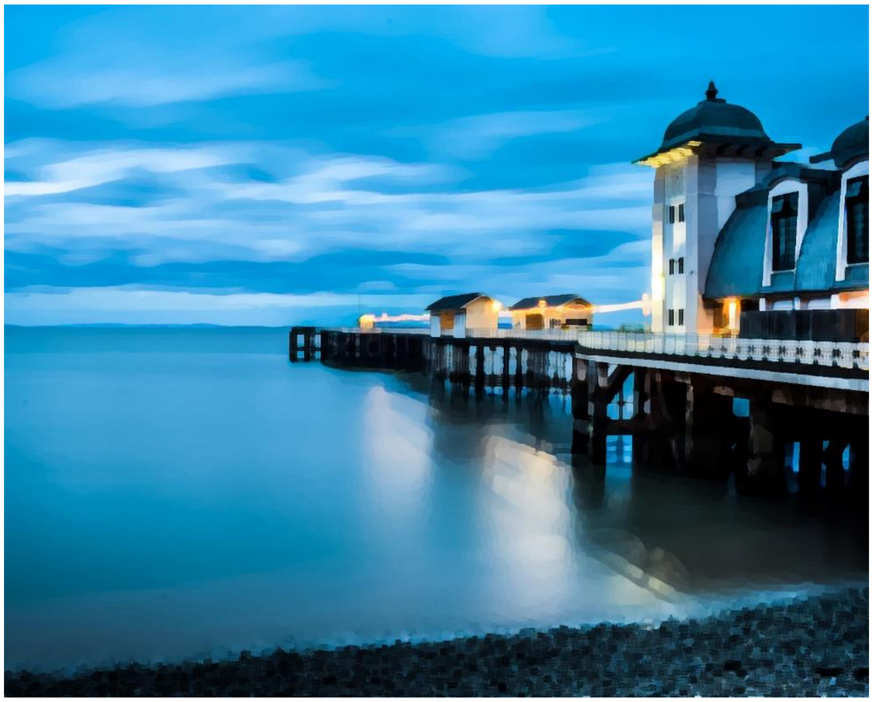


*Figure 4. Painting Effect. Filter size = 6*6*

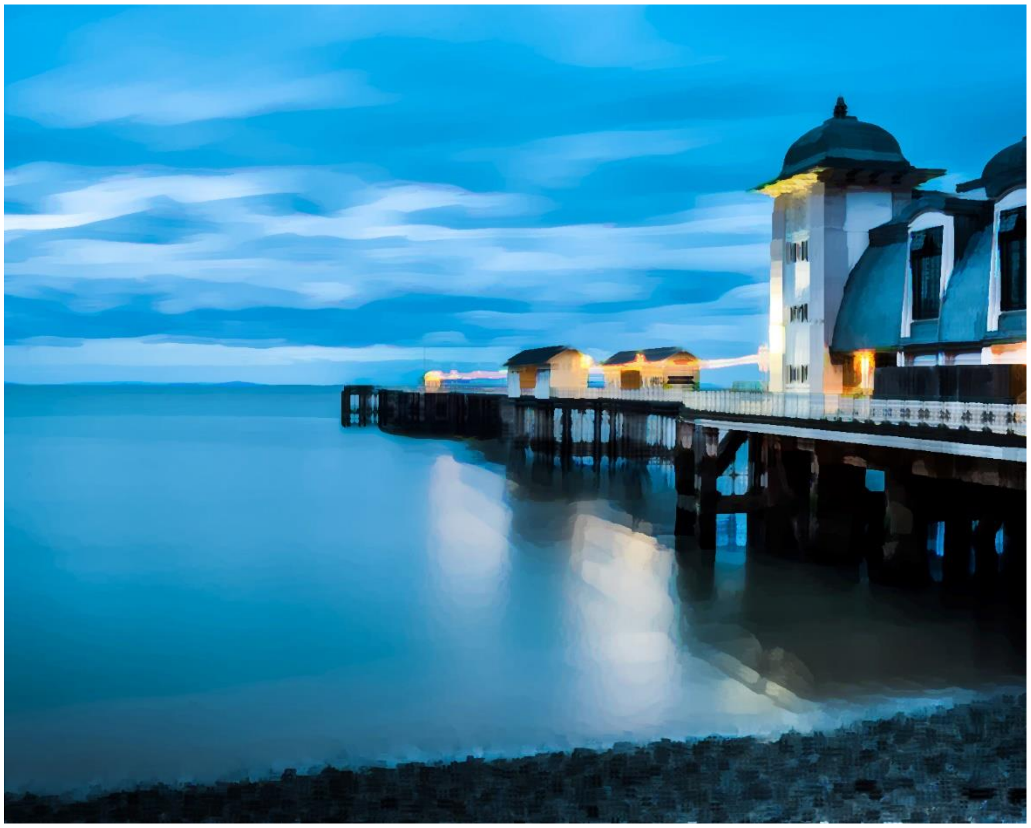


*Figure 5. . Painting Effect. Filter size = 8 * 8*

Discussion

Our method uses the Kuwahara filtering technique to blur out the details of the image while preserving the high frequency regions (or edges). This gives the colors a blocky effect while retaining the overall shape of the object. There are several different methods in literature to achieve this artistic effect. One method that has been implemented before[1] is the technique of searching the neighboring pixels of any given pixel 'p' in either one of the RGB components of an image, and replacing 'p' with the most frequently occurring value in its neighborhood. This technique however, fails to retain the edges of objects thereby resulting in the loss of their overall shape. The implementation of this technique is shown in Figure 6.

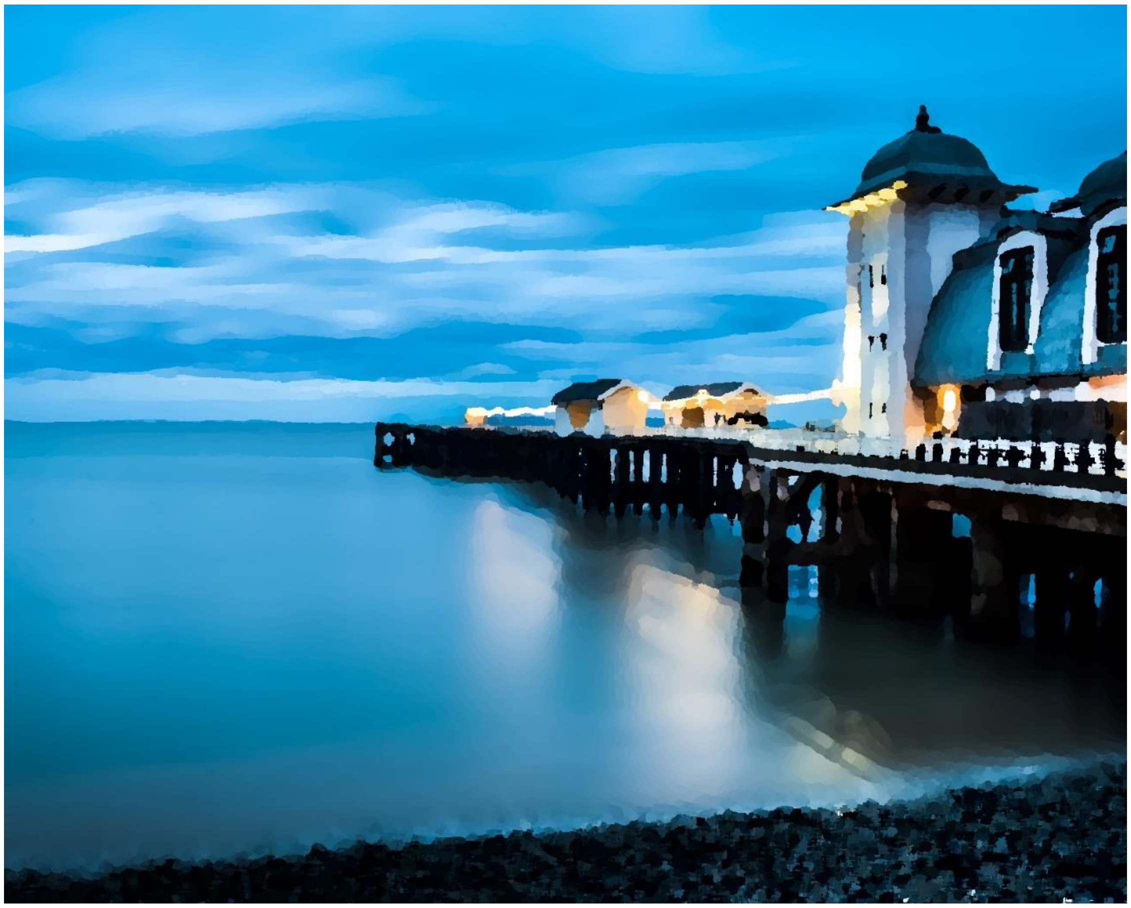


Figure 6. Painting effect using technique in reference[1]. This method fails to retain the overall shape of the objects, however, it does successfully give the image a blocky effect.

In this rendering, the obvious shapes of the objects in the image are distorted, moreover important details are completely lost. Whereas, in our technique (Figure 4)), much of the details like the railings and narrow bars along with the overall shapes of the objects is retained while still giving a blocky color effect to the overall image.

Conclusion

In this project, we have peformed non-photorealistic rendering (NPR) of an image to produce an artistic effect using the Kuwahara filtering technique. This technique works by giving the colors a 'blocky' effect while still preserving the edges of the objects in the image and thus successfully transforming it into an oil painting.

Applications

1. Movies and Video Games: NPR (Non-photorealistic rendering) is a technique inspired by other artistic forms such as painting, drawing, and animations. It has appeared in numerous movies and video games in the form of cel-shaded animation (also known as "toon" shading). This is a type of NPR designed to make 3-D computer graphics appear to be flat by using less shading color instead of a shade gradient or tints and shades, in order to mimic the style of a comic book or a cartoon.
2. Medical Diagnosis: Medical image processing is expected to be one of the most useful applications of image processing. The Kuwahara filter was originally proposed for use in processing RI-angiocardiographic images of the cardiovascular system. The edge-preservation characteristic of this filter is especially useful for feature extraction and segmentation and is the reason why it is extensively used in medical imaging. X-ray cineangiocardiography has been a very helpful technique in examining cardiac functions. This involves the use of a movie camera to film the passage of a contrast medium through chambers of the heart and great vessels for diagnostic purposes.
3. Photo Editing Applications: The use of NPR can be commonly seen in mobile photo editing apps, such as *Prisma*, enabling users to access a wide variety of artistic stylization, with the freedom to choose the level of abstraction, in order to mimic different types of mediums (such as watercolor effect, oil painting effect etc.). Fine art photography is one such field, where the desired outcome is to express a certain emotion or message. Kuwahara filtering enables this by giving an abstract feeling to the photograph, and still preserving the edges.

4. Architecture and Engineering: Three dimensional NPR is used in simplifying technical illustrations and for creating computer simulations. Many 3D-modelling softwares use this technique to create building plans, blueprints etc. A similar approach is used for Scientific Visualization, where scientific phenomena are displayed with the help of computer graphics. Non-photorealistic rendering is an important tool used in many of the STEM fields.

References

1 Zhu, Z., Tang, J., Liu, C., & He, W. (2013). Filters Design for Image based Oil, Watercolor and Cartoon Special Visual Effects Rendering. *Sensors & Transducers*, *161*(12), 27.

2 Gao, J., Li, D., & Gao, W. (2019). Oil painting style rendering based on kuwahara filter. *IEEE Access*, *7*, 104168-104178.

3 Papari, G., Petkov, N., &Campisi, P. (2007). Artistic edge and corner enhancing smoothing. *IEEE Transactions on Image Processing*, *16*(10), 2449-2462.

4 Kyprianidis, J. E., Kang, H., &Döllner, J. (2009, October). Image and video abstraction by anisotropic Kuwahara filtering. In *Computer Graphics Forum* (Vol. 28, No. 7, pp. 1955-1963). Oxford, UK: Blackwell Publishing Ltd.

5 Gooch, Bruce & Gooch, Amy. (2001). Non-Photorealistic Rendering. 10.1201/9781439864173.