ISSN 2616-5775 Vol. 5, Issue 10: 70-77, DOI: 10.25236/AJCIS.2022.051012

Bilateral Filtering Based Image Cartoon Stylization Design

Yuankai Peng

School of Information and Communication Engineering, Communication University of China, Beijing, China, 100024

Abstract: *With the rapid development of multimedia technology and rising demand for entertainment, comic-styled images are increasingly becoming a necessity of life. Cartoonists can produce comic stylized works by making comic-style modifications to real images, but this is laborious and time-consuming. By extracting the features of comic style paintings, we can analyze the differences of comics relative to real images in terms of contrast, presentation, and color, and thus write computer programs to perform comic stylized processing of images. In this paper, we extract the features of comics and use MATLAB tool to write the corresponding feature transformation program to get the comic stylized images step by step, so as to achieve the effect of fast comic stylization of images. Then the images are selected to validate the program, and the analysis concludes that the program has advantages such as fast and good effect, and also has shortcomings such as elimination of details and single style.*

Keywords: *Digital image processing; Cartoon stylization; matlab*

1. Introduction

Cartoons are a common form of artistic expression. The emergence of cartoons has made it easier for people to create without having to stick to realistic details, and their unique style and exaggerated expressions are loved by many people. Comics are everywhere in real life, on food bags, street advertisements, bags and clothes, and comic elements are used as decorative embellishments to enrich our lives. Many artists are engaged in the work of adapting real-life photos into comic book style, i.e. reproducing a picture with the same content as the original one but in comic book style. However, such work is time-consuming and labor-intensive. Since there is a corresponding spatial mapping between the physical image (hereinafter referred to as the physical image) and the manga-style image (hereinafter referred to as the manga image), it is natural to think whether it is possible to use the physical image as an input and obtain a manga image as an output through computer processing. If such a transformation from a physical image to a cartoon image is left to a computer, it can save a lot of time for cartoonists, and if it is developed into an APP or an executable file, ordinary people can realize the transformation from a physical image to a cartoon image on their own cell phones and computers [1], and even set different cartoon styles, adjust the contrast, hue, etc., which will bring great convenience to people's lives. This will bring great convenience to people's life.

2. The characteristics of the cartoon style and the way to achieve it

The main characteristics of the comic style can be summarized as follows.

①The omission of elaborate details, i.e., smoother color jumps and limited use of colors.

②There are often obvious lines outlining the edges of the image, separating different parts of the picture.

③In terms of color the comic pictures will have a higher degree of saturation than the physical pictures.

Around these three problems, analyze what methods can be used to achieve each of these three problems in image processing, then perform the actual derivation and code writing, then integrate them into a program, process the image step by step, debug the program, and get the desired results.

2.1 Programming Language Selection

Because the color image in the computer is a three-dimensional array, and matlab is currently known to be a very powerful tool for processing arrays, and its built-in functions can easily achieve the functions we need, and its powerful processing matrix kernel can reduce the computer running time, so this experiment I choose matlab as the language to write the program [2].

2.2 Step-by-step problem solving

①Omitting complicated details to make the image blurred - bilateral filtering

Blurring is also known as smoothing, which is the concept of filtering in signals. We can pass a digital signal through a filter that removes the high frequencies from the signal, theoretically smoothing the image. However, it is worth noting that in order to distinguish between the various areas of the image, we want to keep more detail on the edges of the image, so we cannot use the Gaussian algorithm of overall blurring. I finally choose to use bilateral filtering: Bilateral filtering is a spatially non-linear filtering method commonly used for image edge preservation, which mainly uses the spatial proximity of pixels in the neighborhood and the similarity of pixel values to construct a Gaussian weight filter. This algorithm was proposed by Tomasi et al. and has a wide range of applications in image processing, such as noise removal, mosaic removal, stream estimation, etc[3], which can also be used as the core algorithm for blurring in this project. Bilateral filtering is able to smooth and preserve the boundary because it combines the features of Gaussian filter and α-truncated mean filter, considering both spatial and value domains, i.e., its kernel is obtained by multiplying the spatial and value domain kernels. According to the bilateral filtering equation.

$$\mathbf{g}(\mathbf{i},\mathbf{j})=\frac{\sum_{(k,l)\in S(i,j)} f(k,l)w(i,j,k,l)}{\sum_{(k,l)\in S(i,j)} w(i,j,k,l)} \tag{1}$$

Among them.

$$\begin{gathered}\mathbf{w}=\mathbf{wswr}\\ \mathbf{ws}=e^{\left(\frac{(i-k)^2+(j-l)^2}{2\sigma_s^2}\right)}\\ \mathbf{wr}=e^{\left(\frac{\|f(i,j)-f(k,l)\|^2}{2\sigma_r^2}\right)}\end{gathered} \tag{2}$$

Using this formula for each pixel value yields an image that is flat and maintains edge detail. Bilateral filtering and the corresponding spectra (original, once, twice), each filtering makes the image smoother and with lower high frequency components. Moreover, the two filters have a better effect on the blurring of the image, and the two filters will be used in the future. So far, the image blurring problem has been solved by the bilateral filtering method.

②Edge line outline - edge detection + saturation adjustment

In order to imitate the stroke strokes of the artist in the comic, it is necessary to find the edges of the picture and outline them. Naturally, one can think of using edge detection to do the detection. The Sobel algorithm is a combination of Gaussian smoothing and differential operations, so it is very resistant to noise and has many uses. In particular, it is most used when high efficiency is required and fine textures are less relevant. In matlab, it is easy to do this by first converting an RGB image or color map into a grayscale image by eliminating the image hue and saturation information while preserving the luminance using the rgb2gray() function, and then calling the edge() function to output the edge matrix of our image (same size and width as the image, 0 pixels for the non-edge part, 1 pixel for the edge part). (size and width are the same as the image, non-edge pixels are 0, edge pixels are 1). With this matrix we can know where the edges are and outline them.

Method 1, direct subtraction of edge regions in all 3 dimensions.

New image matrix = Original image matrix - Original image matrix . * Edge matrix

The elements of the edge matrix that are 0 are left unchanged and the elements that are 1 are removed.

This method sets all the edges of the image to 0 and the color will be pure black. Although it is simple to implement, the result is not very satisfactory.

The processed black line is stark and unnatural. To have a more natural edge transition, the edges need to be related to the color they are in (not black). That's why we propose Method 2.

Method 2, also using the edge matrix, instead of subtracting it directly, we first create a colored edge outline, with all black values as 0 and color as the color of the corresponding position of the outline, and then do a hue processing on the edge so that it can outline better than the black line without departing from the original color system. and then replace the values of the new edge matrix that are not 0 with the corresponding values of the original image to obtain the image after the edge outline.

New image = Iterate through the new edge matrix, and when it reaches a non-zero element, replace the pixel point at the corresponding position of the original image with it.

Although it is more troublesome, this method is more natural and closer to the cartoon style than the direct edge removal method, so I use this method as the edge outlining method of this algorithm.

③Improve the overall saturation of the picture - interpolation and extrapolation

Comic style corresponds to more vivid colors, and color saturation refers to the degree of vividness of colors. We can approximate the effect of the comic style by turning up the saturation. Changing saturation means that the pixel components must be shifted toward or away from the luminance value of the pixel.

For saturation changes, interpolation and extrapolation are usually done using the grayscale image of the original image as a template. The hue change can be adjusted by adjusting the ratio of luminance to color. Here we have chosen -0.95 and 2 as parameters, and the image results are basically what we had expected.

The method summary: The image cartoon styling is done in matlab with bilateral filtering, edge detection, and saturation adjustment, which is basically as expected. However, it is worth noting that this method always uses a 799*531 image, and the effect of edge detection on images with too high or too low resolution needs to be tested. The next step is to use a few classic test images and a few downloaded images from the Internet as benchmarks.

3. Download image selection

3.1 Lena

The classic Reina image, as shown in Figure 1, was used for this color figure test, and the texture and shading of the image, as well as the color, minute detail, and percentage of smooth parts of the image make it a fairly good test material for digital image processing studies.

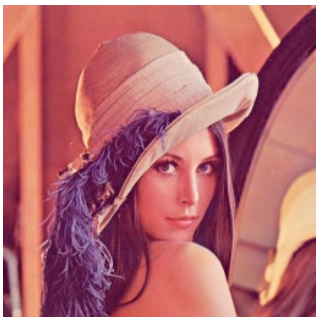

*Figure 1: Lena 256*25*

3.2 Façade

The Figure 2 of the building has rich details and strong light and dark contrasts. Sharp variations such as the texture of the windows as edge edges, are very high frequency signals. was also selected as the test image for this.

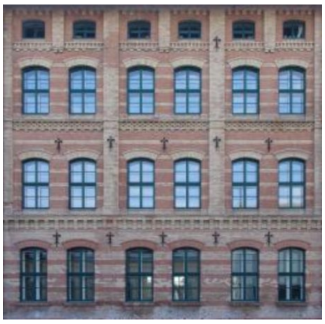

*Figure 2: Facade 256*25*

3.3 Jimi

This picture is shown in Figure 3, with a relatively low brightness and high contrast, while the picture is larger, by having rich details, the task of the hair texture is also exaggerated, this photo can be used as a representative of the multi-detail picture, the test of our algorithm.

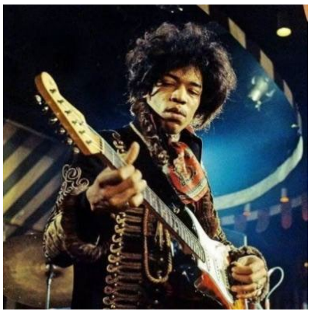

*Figure 3: Jimi 600*600*

3.4 Sailboat

Figure 4 is a representative of a landscape picture with both flat filled spaces and complex foliage, as well as water reflections, which can be a good test of the algorithm's blurriness and edge detection.

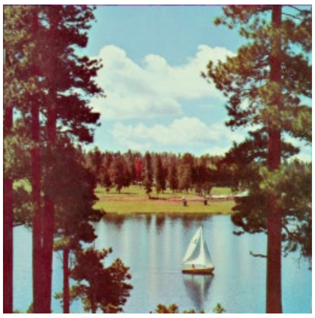

*Figure 4: Sailboat 256*25*

3.5 House

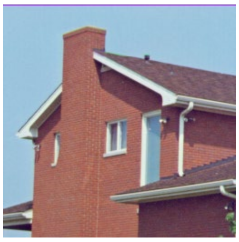

*Figure 5: House 256*25*

Figure 5 has a flat sky with complex wall tiles. Focus on the wall tiles during the test to see if these details can be omitted after filtering, rather than still maintaining the details and identifying the edges.

3.6 Airplane

The plane in Figure 6 has bright red and dark blue colors, the plane is relatively smooth, while the snowy mountain has quite a lot of details. Let's see if the algorithm can eliminate the details for the snowy mountains and have good edge detection for the airplane.

Figure 6: Airplane

4. Experimental results and analysis

4.1 Experimental results

Figure 7 shows the process and results of cartoon stylization for all test images.

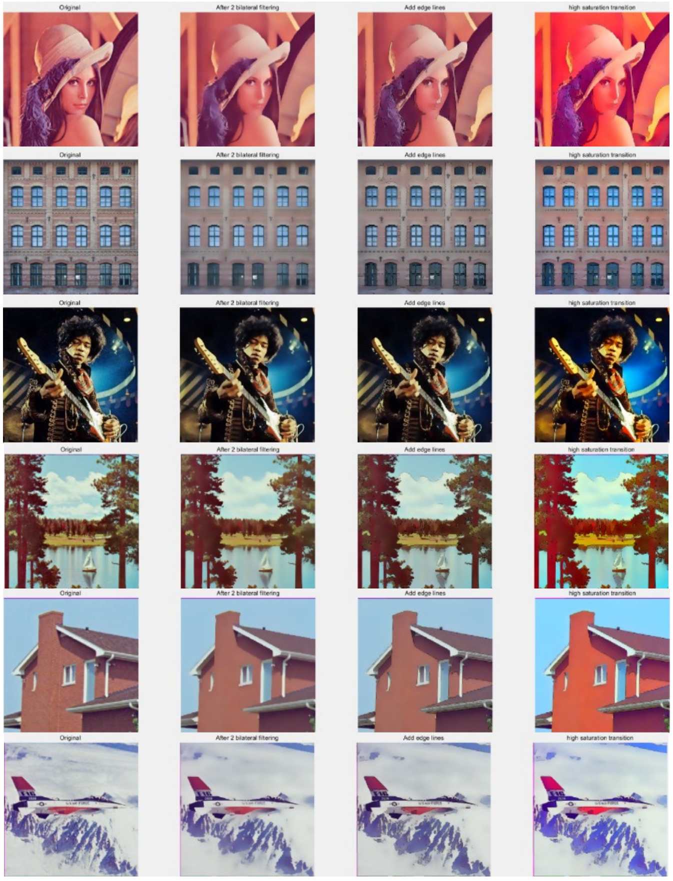
Original
After 2 bilateral filtering
Add edge lines
high saturation transition
Original
After 2 bilateral filtering
Add edge lines
high saturation transition
Original
After 2 bilateral filtering
Add edge lines
high saturation transition
Original
After 2 bilateral filtering
Add edge lines
high saturation transition
Original
After 2 bilateral filtering
Add edge lines
high saturation transition
Original
After 2 bilateral filtering
Add edge lines
high saturation transition

Figure 7: From left to right, the original image, the image after secondary bilateral filtering, the image after adding edge lines, and the image after high saturation transformation. image after adding edge lines, image after high saturation transformation

4.2 Experimental analysis

To facilitate comparison, the original image of the two is compared with the final result combination.

4.2.1 Lena

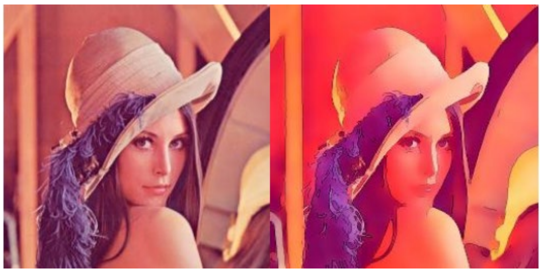

Figure 8: Lena before and after comparison

From Figure 8, you can clearly see that the details of the face and hair have been omitted. The edges that are visible to the human eye are also outlined, and the lines The lines are soft and accurate, very much in the style of comics [4].

4.2.2 Façade

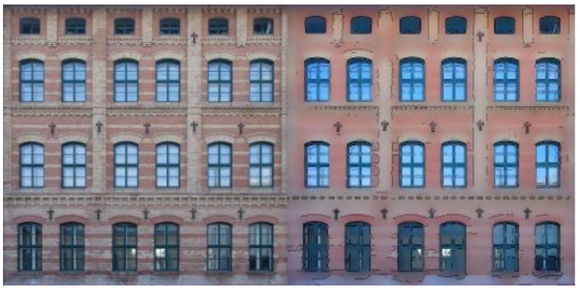

Figure 9: Facade before and after comparison

The brickwork details on the front of the building are largely omitted after the caricature shown in Figure 9, leaving the windows as well as the black cross. In the smooth place, the details are basically blurred, but the edges, such as the windows and the bottom translucent light source are still clear, which all fully illustrates that our algorithm is very good at keeping the edges when dealing with blurred objects. The final color is like a watercolor, just like the moving castle in Hayao Miyazaki's work, which is exciting and desirable.

4.2.3 Jimi

Figure 10 shows a good effect. jimi's face has been divided into pieces, and the edges are outlined with lines similar to the skin tone. The same goes for the hands. The guitar part also has some details omitted from the headstock, which is a watercolor style, keeping the original shape and omitting The details of the strings were omitted. In the background, the complex and noisy (blue) parts have been flattened and brightened, as has the yellow part in the lower left corner. The same is true for the yellow part, where the details have been removed. The color is also just right, like a cartoon on a movie poster.

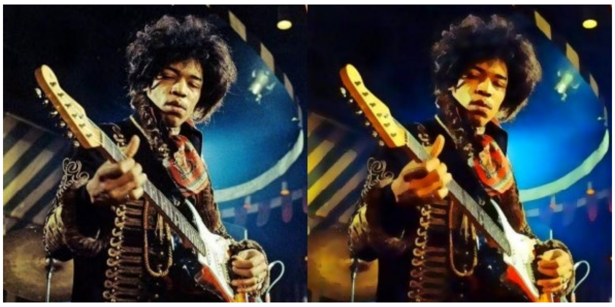

Figure 10: Jimi before and after comparison

4.2.4 Sailboat

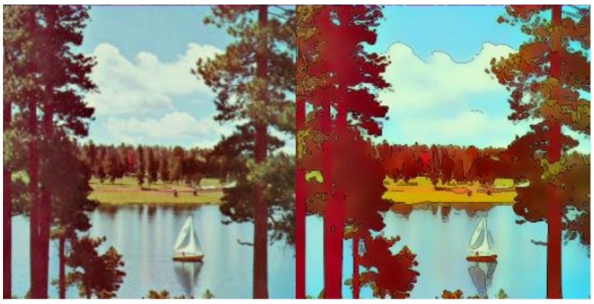

Figure 11: Sailboat before and after comparison chart

The output images, such as the clouds and the lake in Figure 11, are smooth, with a smooth transition, and the tops of the clouds have clear junction lines with the blue sky, as if the artist had used a pencil to outline them. The trees in the distance and near have also been removed from the details of color mutation, and replaced by single color and obvious edges. Look at the boat in the middle of the picture, white alum brown background, outlined by lines, like the sketches of children when they first learn to draw, cartoon style is obvious. Finally, the whole picture feels like an illustration in a fairy tale book, simple and beautiful, very different from the style of the original picture, which shows that the algorithm is very successful.

4.2.5 House

The processed Figure 12 omits the details between the bricks, and the edges are well outlined due to the strong contrast in the rest Clear, the overall tone is also very consistent with the style of the cartoon. However, there is a shortcoming: the chimney in the middle is rectangular, but the processed The image after processing is difficult to see the three-dimensional feeling, which is caused by the loss of a lot of details, originally in these three positions plus the line would Better (manually added). More desirable picture effects is shown in Figure 13. However, this relies on more sophisticated AI algorithms, which are not discussed in this experiment for the time being.

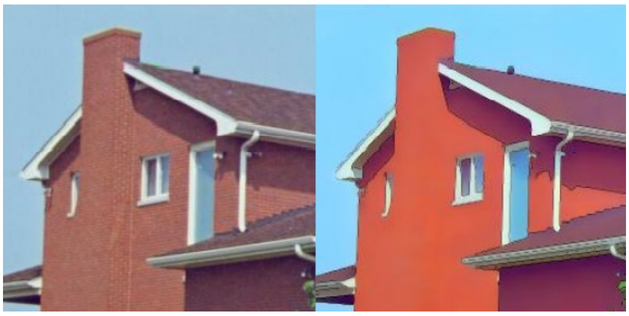

Figure 12: House before and after comparison

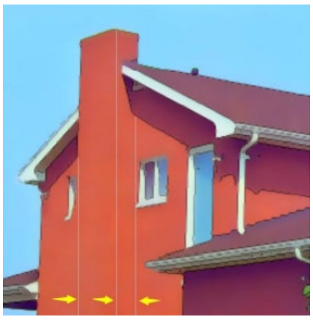

Figure 13: More desirable picture effects

4.2.6 Airplane

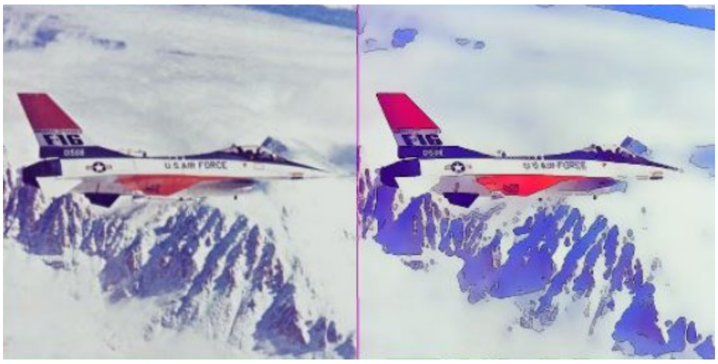

Figure 14: Airplane before and after comparison

As in Figure 14 the algorithm retains the details of the edge of the aircraft and discards the details of the snow color gradient, making the outline of the aircraft clear and the snowy mountains basically only blue and white, with good results.

5. Conclusion

The algorithm can effectively discard the excess details of the image without losing the edge information, achieve the blurring of the image, accurately find the edges and outline them naturally, and finally improve the saturation of the overall image to achieve the purpose of cartoon-style image. The above results and analysis of the output images fully illustrate the superiority of our algorithm, but also reveal some shortcomings of the algorithm: for example, in the house experiment, the loss of details will lead to a lack of three-dimensional images, which is different from the cartoonist's drawing; careful observation of each result shows that each edge line has a certain jaggedness, which seems less realistic; there are various cartoon styles, but the algorithm intelligently implements a cartoon style. However, the algorithm is smart enough to realize the conversion of one cartoon style, and there is not much plasticity except for adjusting the parameters of hue. The algorithm can transfer the style of one painting to another painting [5], which can solve this problem perfectly.

References

[1] Zhang, Jinglei, Hou, Ya-Wei. Image style migration based on improved recurrent generative adversarial network [J]. Journal of Electronics and Information, 2020, 42(5):7.

[2] https://github.com/Sun-Yize-SDUWH/Photo-Cartoonization

[3] Fan Ke-Yue, Liu S-G. Stylization method for portrait photo cartoons [J]. Journal of Computer-Aided Design and Graphics, 2022, 34(1): 8.

[4] Wang Dechu. Research and application of automatic face caricature generation algorithm [D]. South China University of Technology, 2019.

[5] Chen, Shuhuan, Wei, Yuke, Xu, Le, et al. A review of image style migration research based on deep learning [J]. Computer Application Research, 2019, 36(8):6.