IJCAT International Journal of Computing and Technology, Volume 1, Issue 6, July 2014
ISSN : 2348 - 6090
www.IJCAT.org

Real-Time Cartoonization Using Raspberry Pi

[1] **Parimal Sikchi,** [2] **Neha Beknalkar,** [3] **Swapnil Rane**

[1, 2, 3] School of Electronics Engineering (SENSE)
VIT University, Vellore-632014 India

Abstract - Cartoonization is a method of image stylization in which input moreover looks like a sketch or cartoon like image. It includes making edges bolder, making colors brighter and lively, accompanied with smoothening operation so as to filter out the high frequency details. Furthermore, reduction in the luminance contents is carried out, which helps to quantize the image for obtaining cartoon like effect. Different image processing techniques such as contrast stretching, bilateral filtering, luminance quantization, and edge detection are being implemented to achieve the desired outcome. We have proposed and implemented a device using Raspberry pi that could create cartoonized images on the fly from a camera feed. OpenCV programming is used for implementing image processing techniques.

Keywords - **Cartoonization, Bilateral Filter, Luminance quantization, Camera feed, Raspberry pi, OpenCV.**

1. Introduction

Real time image abstraction is gaining importance in recent times, and finding its application in the fields of communication, entertainment, broadcasting, video games, and many more areas. Image abstraction is used to simplify color and shape which makes it easier to recognize a scene. Our paper presents an idea to generate cartoon of a captured image in real time.

Lines play a vital role in expressing any scene swiftly. They are capable of describing the figures efficiently and naturally. Portraying information of shapes by lines has been effectively used by humans for ages. Line drawing approach can be seen in various contexts, such as medical diagrams, architectural design, technical illustration, storytelling, and most importantly in cartoon drawing.

Our design of real-time cartoonization can be briefed in eight steps: 1) Pre-process an image with contrast stretching 2) Apply bilateral filter for smoothening purpose. 3) Convert filtered image to gray scale for edge enhancement. 4) Employ edge detection. 5) Convert RGB smoothed out image to CIELab color space. 6)Set various threshold levels to the intensity values to achieve luminance quantization 7) Once again perform RGB color space from CIE-Lab color space. 8) Combine edge enhanced image and output received from the step of luminance quantization.

We have built an algorithm to implement above steps using MATLAB which indeed runs faster on hardware system. The paper is organized as follows: Section II describes different related work being carried out for cartoonization. Section III gives details about the algorithm to achieve a cartoon from a captured image. Section IV tells about design and realization of the required system. Section V illustrates our results. Finally, the conclusion of our work is presented in section VI.

2. Literature Survey

Generating a cartoon requires numerous image processing techniques. Few papers have briefed the methodology to form a cartoon. It is observed that there is a huge importance of edge detection along with use of selective colors [1]. Its features have been explicitly described in the paper "On the Effects of Pre- and Post-Processing in Video Cartoonization with Bilateral Filters" [2]. But to decide which among the many edge detection techniques is suitable for our purpose, we referred "Study and Comparison of Various Image Edge Detection Techniques" [3]. "A Computational Approach to Edge Detection" [4] helps us to decide why Canny edge detection method is appropriate for this application. It is an optimal edge detector as it would not create a false edge of some noise in the image. This property is vital because edges are significant part of a cartoon.

Apart from detecting the edges, it is also essential to do smoothening of the image to give an experience of cartoon. Bilateral filter helps to achieve this purpose [5, 6]. Subsequent essential step in forming a cartoon is to reduce color levels [7]. Based on human perception color levels can be reduced by the means of luminance quantization and color quantization [8].

3. Framework for Cartoonization

Our system is based on the technique suggested by Winnemöller et al. [5], which has since been established as the generalized methodology for cartoonization. We tried to extend Winnemöller's framework with adding pre-processing and post-processing techniques to enhance the effects. First step of the framework is to enhance the image contrast using contrast stretching at the same time maintaining balance in range of intensity. Bilateral filtering is applied on this contrast stretched image.

3.1 Algorithm for Cartoonization

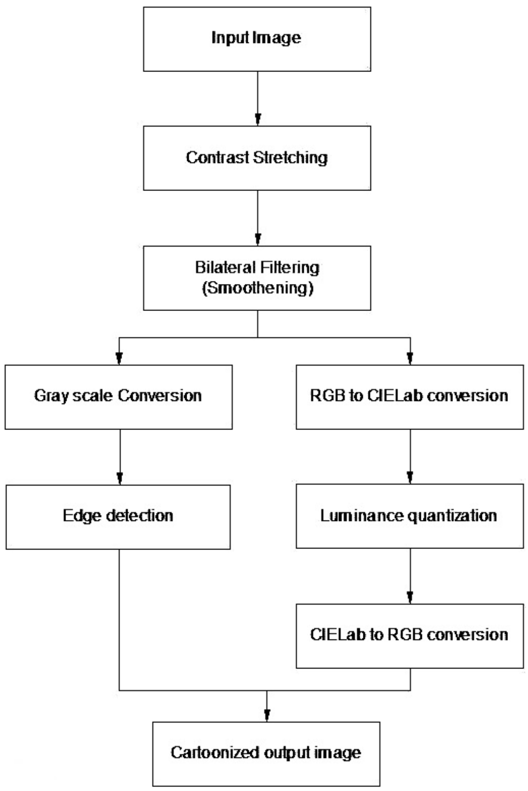

Fig.1 Control flow of algorithm

3.1.1 Bilateral Filter

Bilateral filter is a key step in cartoonization. It smoothens images while preserving edges with a nonlinear mixture of neighboring image intensity values. The main idea of bilateral filter is to apply Gaussian based weights to neighbor pixels to obtain smoothing effect. Bilateral filtering is achieved by combination of domain and range filtering. The domain operates as a low pass filter as it averages adjacent pixel values. While the range filter plays key part in conserving edges.

Only perceptually alike colors are averaged and only perceptually evident edges are conserved. Geometric spread σ_d is selected on the basis of the required amount of low pass filtering. Large σ_d blurs more with the intention of combining values from more distant image progresses. Mathematically, the output of a bilateral filter at a pixel location x is calculated as follows，

$$\tilde{I}(x) = \frac{1}{c}\sum_{y\in N(x)} e^{\frac{-||y-x||^2}{2\sigma_d^2}} e^{\frac{-|I(y)-I(x)|^2}{2\sigma_r^2}} I(y) \qquad (1)$$

There are two parameters that control the behavior of the bilateral filter. As the range parameter $\sigma_r$increases, the bilateral filter becomes closer to Gaussian blur.Hence, increasing the spatial parameter $\sigma_d$which helps to smooth image.N(x) can be considered as a spatial neighborhood of pixel I (x), and C represents the normalization factor which is given as $1/W_p$.

$$c = \sum_{y\in N(x)} e^{\frac{-||y-x||^2}{2\sigma_d^2}} e^{\frac{-|I(y)-I(x)|^2}{2\sigma_r^2}} \qquad (2)$$

There are different techniques available for noise suppression but we have used bilateral filter technique which reduces noise and preserves the edges at the same time.

3.1.2 Edge Detection

For cartoonization edges play crucial role. Many basic drawings, sketches, cartoons in line drawing format are only based on the edges of the object. So it is necessary to accentuate this feature. Edge detection is another feature abstraction method. Edges are region of high local contrast. The distinctiveness of these locations are emphasized. Edges which are more noticeable to human perception are enhanced.

In our work, canny edge detection algorithm operator is used to identify pixels which have sharp brightness changes. It involves several steps such as: Smoothing, Finding gradients, Non-maximum suppression, Double thresholding, Edge tracking by hysteresis.

3.1.3 RGB to CIE-Lab Color Space Conversion

It is an essential step before moving forward. Image sequence is converted into CIELa*b* color space from original RGB color space. In this step, the color channels and the luminance channel get separated. By varying the value of L we can adjust the brightness of individual pixel while preserving the color effects of the pixel. Colorspace conversion
is carried out as follows :

$$L = 0.299 * 0.587 * G + 0.114 * B \qquad (3)$$
$$a = 0713 * (R - L) \qquad (4)$$
$$b = 0.564 * (B - L) \qquad (5)$$

Where, R, G, B are the individual color values in RGB colorspace. L is the value of the luminance, while a and b represents two color channel values in CIELa*b* colorspace.

Perceptually more linear than other colorspaces is obtained by this conversion. It simply indicates that a variation of the particular amount in a color intensity value will results in a change of approximately equivalent visual importance.

3.1.4 Luminance Quantization

Luminance quantization is extensively used to get cartoon like shading effect on the smoothened filtered image. It maps the luminance values of an image to discrete levels.

CIELab allows image contrast to be adjusted based unnoticeable differences. By natural design, CIELab approximates the human vision by describing all colors that can be seen by the human eye. Its luminance component 'L' is closely equivalent to the human perception of lightness. Quantization is achieved by dividing 256 intensity levels into 16 quantized levels.

3.1.5 CIE-Lab to RGB Color Space Conversion

Finally, the image should be converted back to RGB color space from the CIE-Lab color space, for final representation. Conversion is carried out using following equations:

$$R = L + 1.403 * a \quad (6)$$
$$G = L - 0.714 * a - 0.334 * b \quad (7)$$
$$B = L + 1.773 * a \quad (8)$$

3.1.6 Contrast Stretching

Contrast refers to the relative differences in the brightness of pixel values. Contrast stretching is a method for improving the contrast in an image so as to highlight important details of image. It stretches the range of intensity values to the extent of desired range of values. It is obtained by remapping or the gray level values so that the histogram extents to the full range.

4. Hardware Design

We have implemented raspberry pi based system which creates cartoonized images according to the status of push button which is connected to raspberry pi. Open CV programming on raspberry pi platform enables us to combine various image processing techniques on hardware.

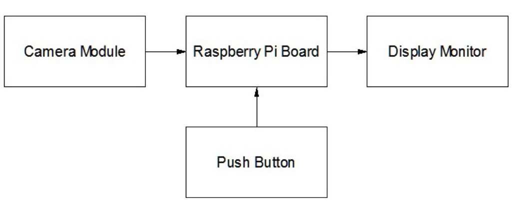

Fig. 2 System Block Diagram

5. Results

Following are some of the results we obtained after executing different filtering and transform operations. Results of real time image captured from serial camera are being shown.

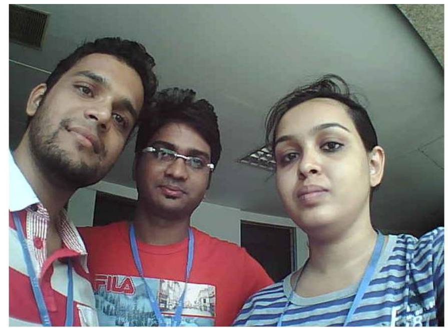

Fig.5. Original Image

Fig.6. Contrast Stretching Output

Fig.7. Bilateral Filtered Output

Fig.8. Edge Detection Output

Fig.9. Edge Enhancement Output

Fig.10. Quantization Output

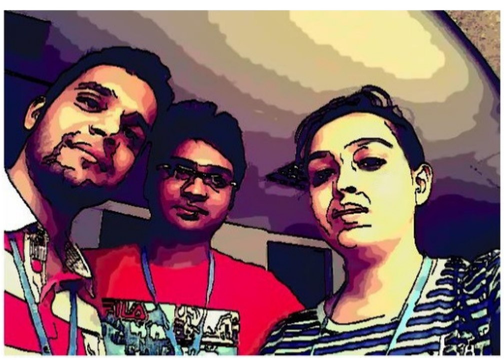

Fig.11. Cartoonized Output

6. Conclusions

We have implemented several filtering and image enhancement techniques to obtain desired effects of cartoonization. We have used the theory of bilateral filtering, to filter out the small discontinuities and strengthening the salient edges, along with unsharp masking for feature detailing. In addition to this, image frames are processed by image enhancement techniques like contrast enhancement, edge enhancement. Luminance quantization helped for getting color reduction and quantization effects. It is seen that accuracy and performance of this algorithm also depends on the resolution of the image. Some important details might be lost while cartoonifying due to poor resolution of the input image, but it is also seen that bigger the image more time it takes to form cartoonized image. Hence, there was an inevitable trade-off between these two factors. To get better performance and speed up the system, we have developed hardware system which improves efficiency.

References

[1] Chaoqun Hong, Zhi Yang, Jiajun Bu, Yi Liu, Chun Chen, Cartoon Like Stylization of video real time applications, Zhejiang University, IEEE International Conference on Multimedia and expo 978-1-4244-2571-6/08/©2008 J.ClerkMaxwell,ATreatiseonElectricityandMagnetism,3rded.,vol.2.Oxford:Clarendon,1892,pp.68–73.

[2] ZoyaShahcheraghi #1, John See, On the Effects of Pre- and Post-Processing in Video Cartoonization with Bilateral Filters, IEEE international conference on Signal and Image processing applications(ICSIPA),978-1-4799-0267-5,

[3] Raman Maini, Dr. Himanshu Aggarwal "Study and Comparison of Various Image Edge Detection Techniques", International Journal of Image Processing (IJIP), Volume (3) : Issue (1). J. Canny. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis & Machine Intelligence,8(6):679–698,November 1986.M.Young,TheTechnicalWriter'sHandbook.Mill Valley,CA:UniversityScience,1989.

[4] H. Winnemoller, S. C. Olsen, and B. Gooch, "Real-Time Video Abstraction," ACM Transactions on Graphics, pp.1221-1226,2006J.ClerkMaxwell,ATreatiseonElectricityand Magnetism,3rded.,vol.2.Oxford:Clarendon,1892,pp.68–73. C. Tomasi and R. Manduchi. Bilateral Filtering for Gray and Color Images. In International Conference on Computer Vision (ICCV), pages 839–846, 1998.

[5] N. Chaaklha, W.C. Tan, Teresa H. Y. Meng: Color Quantization of Images Based on Human Vision Perception. IEEE International Conference, 1994.R.Nicole,"Titleofpaperwithonlyfirstwordcapitalized," J.NameStand. Abbrev.,inpress.

[6] Fast color quantization using weighted sort-means clustering. JOSA A, Vol. 26, No.11, pp.2434-244, 2009.