Image Cartoonization Methods

Using LBG, KPE, KMCG quantization

Dr. Archana B. Patankar
Associate Professor: Computer Dep.
Thaodomal Shahni College of engineering ,Bandra (W), Mumbai.
Email : patankararchu@gmail.com

Ms. Purnima A. Kubde
Assistant Professor: Information Technology Dep.
St. Francis Institute of Technology, Borivali (W), Mumbai.
Email : purnima.tawde10@gmail.com

Ms. Ankita Karia
Assistant Professor: Computer Dep.
St. Francis Institute of Technology, Borivali (W), Mumbai.
Email : ankita.karia@gmail.com

Abstract— **Different methods to generate cartoonized painterly effect on grayscale and colored images are discussed in this paper. The Concept of vector quantization is used to get the painterly effect on images. Algorithms LBG, KPE and KMCG are used to generate cartoonized painterly results. The results of applying each algorithm on images is compared based on time taken and effect generated. The results obtained by discussed methods can be incorporated in many application which are used for movie to comic conversions and digital art software.**

Keywords— Vector quantization, LBG, KPE, KMCG, Image paint effect and Image Cartoonization.

I. Introduction

There is a huge growth in people using photo fun , digital art applications on mobiles to enhance images , creating posters, image stylization, image shading , converting to movie to comic book effects for the fun of reading. The application, where images are converted to painterly cartoonized images falls under the category of Non-photorealistic rendering (NPR). NPR is a growing field of Image processing, Computer graphics and Computer vision [1]. NPR is simple stylization of images. Cartoon effect stylized images can be used to generate paintings, drawing, technical illustration, and animated cartoons. Stylized images not only prove useful in the field of art and creativity but also can be beneficial in the field of visualization, architectural illustration and animated explanation of experiments [2] [3]. The concept of vector quantization can be used to generate a result which can prove helpful in applications of NPR. The methods which can generate cartoonized painterly effect on the images is discussed in the paper. The input required by the system is a true colored image which on quantization will be converted to painterly effect.

A. Problem Statement

Image is given as input to the system and the output generated is a cartoon like effect on the image as seen in comic books and painterly rendering. The cartoonization process involves Line extraction and Color Reduction, Combining and Image enhancement.

1) *Line Extraction:* The task of line extraction is to locate discontinuities in the image intensities. The pixel where brightness of the image changes sharply represents the edge that is to be extracted. Different methods like Canny's Edge detection, Difference of Gaussian Edge detection or Gradient edge detection are used for Line Extraction [1][5][6].
2) *Color Reduction:* It is the process of reducing the number of colors in the image. Smoothening techniques like Bilateral filtering and Anisotropic filtering can be applied for color reduction. Another approach for color reduction in images is quantization .Vector Quantization (VQ) consists of three steps - Codebook design, Encoding and Decoding [7].
3) *Image Enhancement:* The effect on image produced by applying line extraction and color reduction is combined and image sharpness and contrast is improved to produce a good cartoonization effect.

II. Methods of Cartoonization

A. Cartoonization using Smoothening Filters

a) Bilateral Filtering (BLF) Cartoonization

Process of image cartoonization using BLF method is given in the steps bellow:

1) Feature space conversion is performed to extracts the contrasts in the given image. Feature space, such as CIE Lab so that image contrast is adjusted depending on just noticeable differences. We follow this advice and our parameter values assume that L ∈ [0,100] and (a,b) ∈[−127,127].
2) Second step in the process is image smoothing which is done by using a filter called bilateral filter to smoothen the input image.

$$\boldsymbol{I'(x) = \frac{1}{W_p}\sum_{x_i \in \alpha} I(x_i).m} \qquad (1)$$

$$\boldsymbol{m = f_r(||I(x_i) - I(x)||).g_d(||(x_i) - (x)||)} \qquad (2)$$

$I(x_i)$ is the original image, $I'(x)$ is the filtered image, f_r is the Gaussian kernel for intensity difference, g_d is Gaussian range kernel for coordinate difference. x_i is neighboring pixels of current pixel x_i. Neighboring kernel value is of current pixel x[8].

At the third step the result of second step(2) is used to perform luminance quantization . Q is Quantized image $\Delta q \in [8 ,10]$ is bin width $q_{nearest}$ is the bin boundary.

$$\boldsymbol{Q = q_{nearest} + \frac{\Delta q}{2}\tanh(s.(x - q_{nearest}))}$$

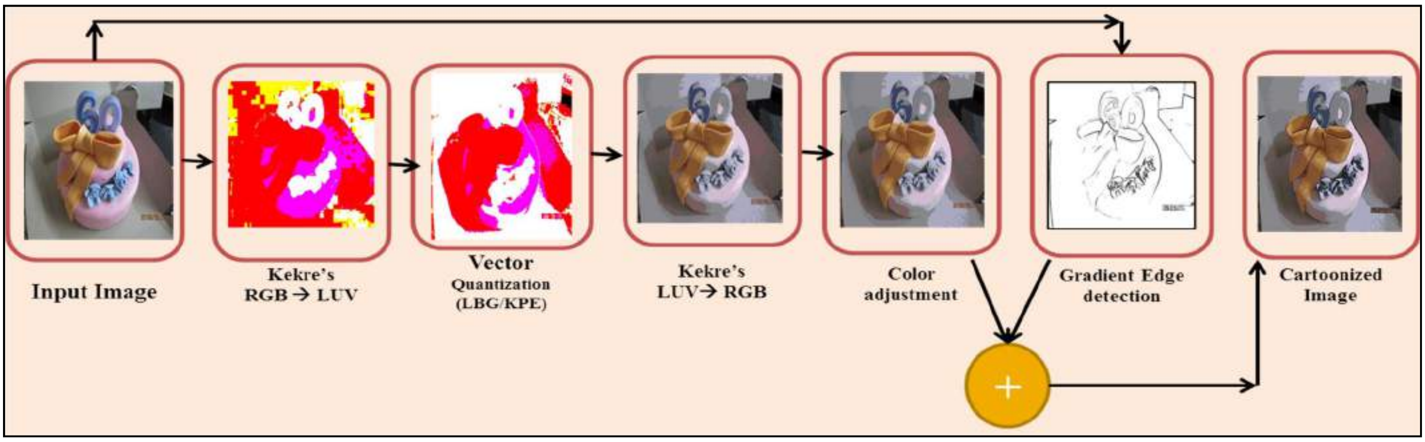


Figure 1: Cartoonization Process using LBG and KPE VQ

In the fourth step the result of second step is used to detect edges of the image. The result of the third and fourth step is merged to obtain stylized image to enhance the stylized image result image warping is performed [7].

$$\boldsymbol{D1 = I * \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}} \tag{3}$$

$$\boldsymbol{D2 = I * \frac{1}{2\pi K\sigma^2} e^{-(x^2+y^2)/2K\sigma^2}} \tag{4}$$

$$\boldsymbol{D = D1 - D2} \tag{5}$$

$$\boldsymbol{E = \begin{cases} 1 & if(\,D > 0) \\ 1 + \tanh(s.D) & otherwise \end{cases}} \tag{6}$$

E is the edge detected by using difference of Gaussian method. The sharpness parameter is $s \in [0.75,5.0]$ and D1, D2 are Gaussian function and K is a constant value $K = \sqrt{1.6}$ [8].

b) Anisotropic diffusion Filtering

1) Feature space conversion is performed to extracts the contrasts in the given image. Feature space, such as CIE Lab so that image contrast is adjusted depending on just noticeable differences.
2) Smoothening and color reduction using Perona and Malik's smoothing filter also called as Anisotropic diffusion filter. ∇I denotes gradient of image I. Diffusion coefficient is $c(x, y, t)$ where x, y are image coordinates and t is iteration, div() is the divergence operator[10][11].

$$\boldsymbol{\frac{\partial I}{\partial t} = div(c(x, y, t)\nabla\, I)} \tag{7}$$

3) Canny's edge detection Canny edge detection is a four step process.A Gaussian blur is applied to clear any speckles and free the image of noise. A gradient operator is applied for obtaining the gradients' intensity and direction. Non-maximum suppression determines if the pixel is a better candidate for an edge than its neighbors. Hysteresis thresholding finds where edges begin and end.

4) Result of step 2) and step 3) are combined to obtain the cartoonization effect on the given image.

III. New Cartoonization Method

A. Cartoonization using LBG and KPE VQ

The process of new cartoonization technique that uses LBG, KPE vector quantization is discussed in this section is shown in Figure 1.

(1) First step is to use Kekre's RGB to LUV color space conversion to the input image [10]. The result of this step helps in better quantization effect.

(2) Second step is to use Vector Quantization algorithm. LBG (Linde Buzo Gray) or KPE (Kekre's Proportionate Error) is applied to obtain a quantized output.

(3) Once image is converted from LUV color space to RGB color space.

(4) Next step is to smoothen the image and enhance its features by applying smoothening filters.

(5) The edges from the original image are extracted to define prominent border color in the cartoonized image.

(6) The final cartoonized output obtained is achieved by combining the results of edge detection step e) and quantized and smoothened image at step d) [1].

B. LBG Vector Quantization

The LBG VQ design algorithm is an iterative algorithm; the algorithm iteratively minimizes the total distortion by representing the training vectors by their corresponding code vectors. The algorithm requires an initial codebook C obtained by taking the average of the training vector X. The frame work for cartoonization using LBG and KPE vector quantization are shown is shown in the Fig 1.

Steps of LBG algorithm[1] [11] [12][16].

(1) Take image as input and divided into the windows of size 2x2 pixels (each pixel consisting of red, green and blue components).
(2) Put in a array to get 12 values per vector. Collection of these values is a training set.
(3) Initial codebook is Mean (M) training vector . i.e. Mean of rows of X.
(4) Split the initial Codebook C1=M +1 and C2=M -1.

(5) Compare between C1 and C2 using training vector Euclidean distance form Codebook CB1 and CB2. Compute D1 Euclidean distance between X and C1 and X and D2 Euclidean distance between X and C2. if D1 > D2 then X goes to new cluster CB1 else X goes to another cluster CB2.
(6) (Calculate mean of new codebooks CB1 and CB2 and continue the process till codebook size is N (desired value) N=2^n where n=1,2,3,….

C. *KPE Vector Quantization*

Instead of adding constant value, a proportionate error is added to the centroid in positive and negative direction in order to get initial two code vectors in codebook. The error ratio is decided by magnitude of coordinates of the centroid. Hereafter the procedure is same as that of LBG.

Steps of KPE Quantization [1][13][14][15][16].

(1) Take image as input and divided into the windows of size 2x2 pixels (each pixel consisting of red, green and blue components).
(2) Put in an array to get 12 values per vector. Collection of these values is a training set.
(3) Initial codebook is Mean (M) training vector. i.e. Mean of rows of X.
(4) Compute proportionate error vector $E = c_i/c_j$ where c_i the initial codebook is and c_j is the minimum value in the initial codebook.
(5) Split the codebook CB1=M+E and CB2=M-E.
(6) Compare between CB1 and CB2 using training vector Euclidean distance and form Codebook V1 and V2. Compute Euclidean distance D1 between X and CB1, X and D2 Euclidean distance between X and CB2. If D1>D2 then X goes to new cluster V1 else X goes to another cluster V2.
(7) Calculate mean of new codebooks CB1 and CB2 and continue the process till codebook size is N (desired value) N=2^n where n=1,2,3,…..

D. *Cartoonization using KMCG*

The Cartoonization process using KMCG requires an RGB or Gray scale image as input which is shown in Figure 2. The steps of Cartoonization using KMCG are as follows[18][19]:

(1) The input image is smoothened using image enhancement algorithms.
(2) The next step is to apply KMCG (Kekre's Median Codebook Generation vector quantization algorithm which reduces the colors in the image giving it a complete painted image look.
(3) For providing dark border effect edge extraction algorithm like Difference of Gaussian is applied.
(4) The output of edge extraction and KMCG are combined to produce a proper painterly effect on images. This effect makes the output image look cartoonized.

The above two algorithms LBG and KPE discussed in section *B* and *D* requires Euclidean distance computations in every iteration they take large time generate codebook. In order to reduce the computational complexity of LBG and KPE, every Euclidean computation is replaced by simple comparison. Here quick sort algorithm is used. This algorithm takes least time to generate codebook, since Euclidean distance computation is not required.

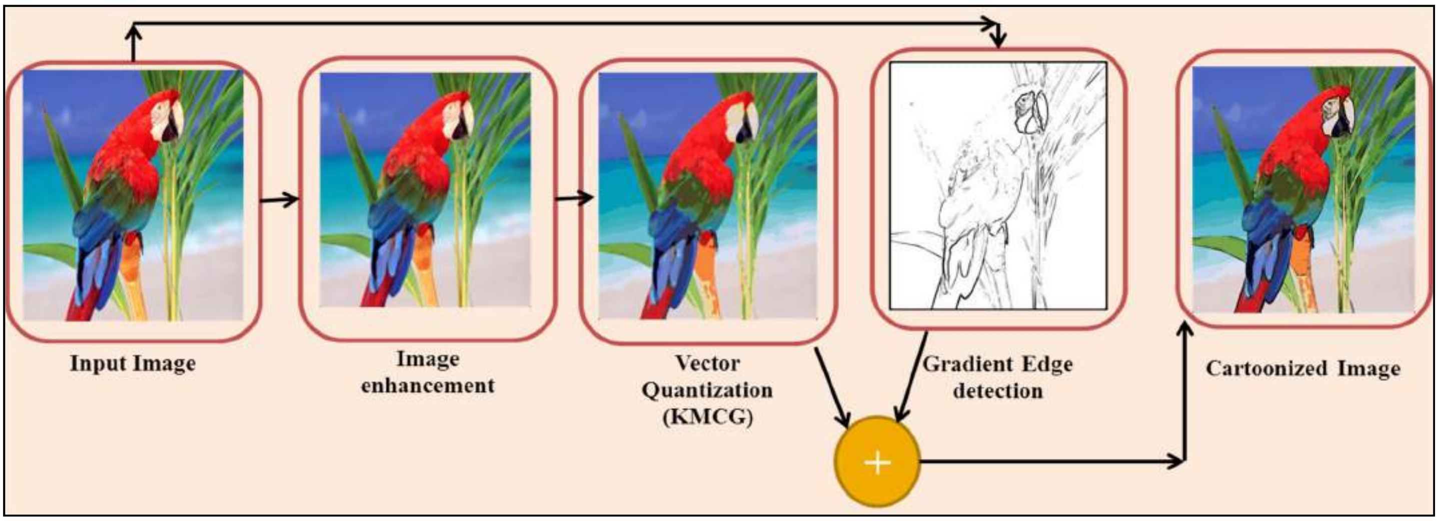


Figure 2: Cartoonization Process using KMCG VQ

Steps for KMCG Algorithm [18] [19]:
Image is divided into the windows of size 2x2 pixels (each pixel consisting of red, green and blue components).

(1) These are put in a row to get 12 values per vector. Collection of these vectors is a training set. The training set is sorted with respect to first column. The Median of the first column is used to divide the training set in two parts and the median vector is put in the codebook. Set the codebook size equal to 1.
(2) Further each part is then separately sorted with respect to second column to get two median values and these

two median vectors are put into the codebook. Set the codebook size equal to 2.

(3) The process of sorting is repeated till codebook of desire size is obtained.

IV. RESULTS

All Cartoonization methods Bilateral Filtering method, Linde Buzo Gray(LBG) Vector Quantization method, Kekre's Proportionate error (KPE) method and Kekre's Median Codebook Generation (KMCG) method produce a cartoonized image output. The process of cartoonization and implementation is different. The result produced by each of the method is compared in this section based on the time (milliseconds) taken to produce cartoonized image. Also the cartoonized output generated by each of the method is compared based on the appearance of painterly effect in the output. Which is evaluated based on a survey done by showing the output images to common people and also to commercial artists. Time take for cartooning a given image by all the methods are shown in Table 1.

Table 1: TIME COMPARISON OF LBG , KPE AND KMCG

Image	Size	LBG (ms)	KPE (ms)	KMCG (ms)	BLF (ms)
PM	512*512	20	20	18	33
Plane	512*512	22	21	17	36
Julia	512*512	20	20	18	33
Pancake	256*256	6	6	4	10
Coconut	512*512	6	6	5	9
LH	512*512	20	20	18	30
Mary.K	512*512	20	20	18	29
Food	256*256	5	5	5	8
Gere	512*512	20	20	18	30
Me	256*256	6	6	5	10

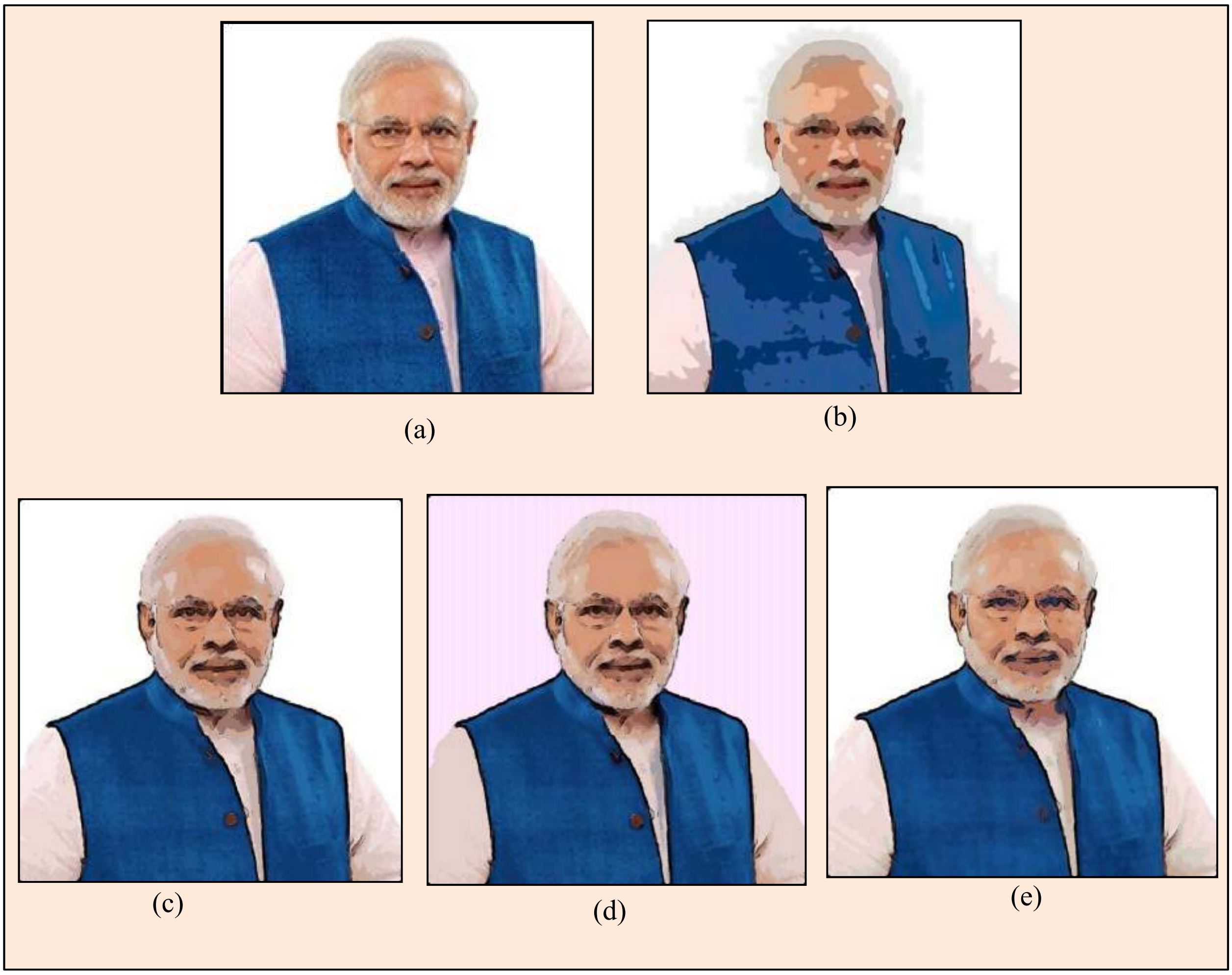
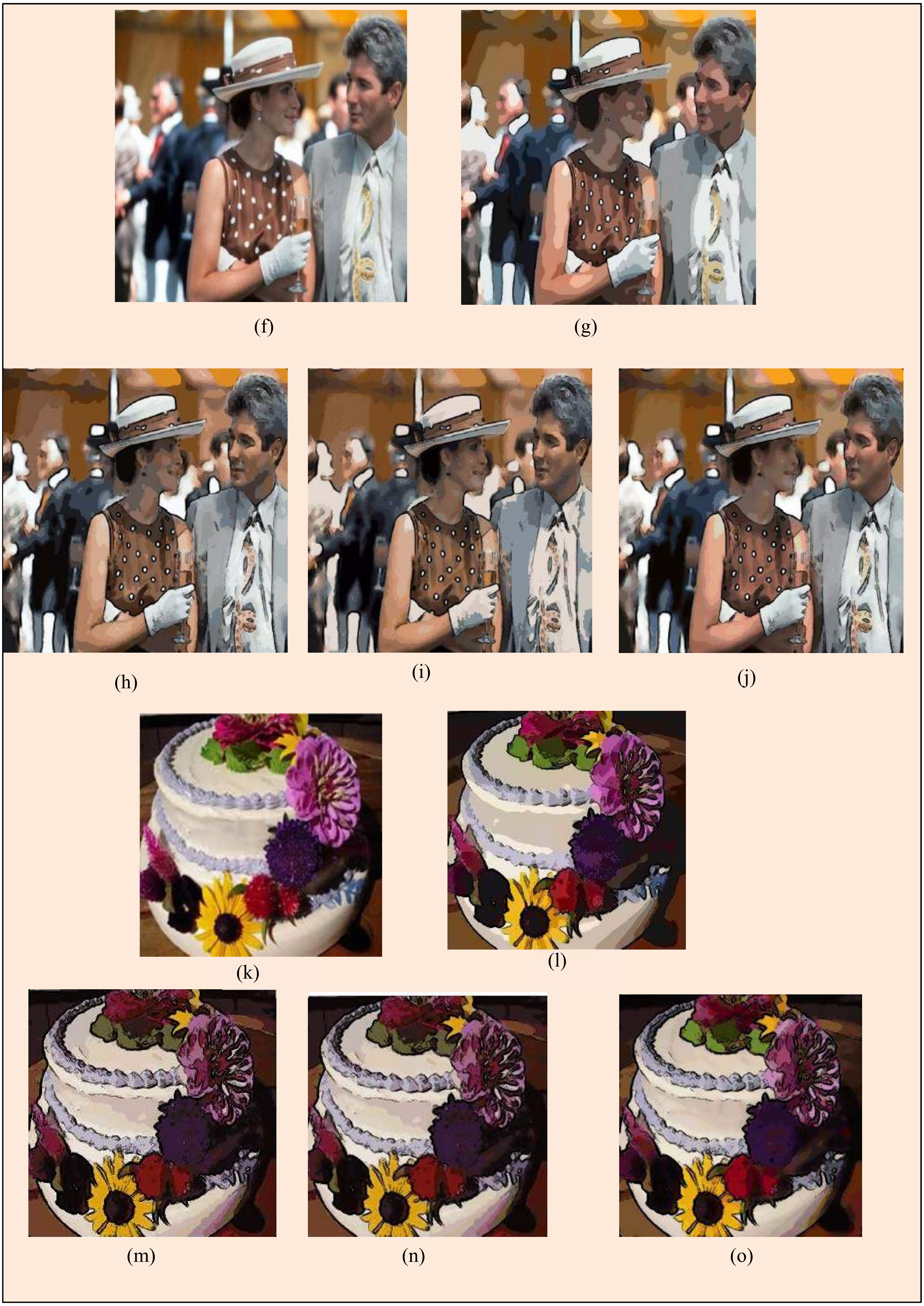


Figure 3: (a) Original Image (b) BLF cartoon (c) LBG cartoon (d) KPE cartoon (e) KMCG cartoon



(f) (g)

(h) (i) (j)

(k) (l)

(m) (n) (o)

Figure 4: (k)(f) Original Image (l)(g) BLF cartoon (m)(h) LBG cartoon (n)(i) KPE cartoon (o)(j) KMCG cartoon

Bilateral filter is a smoothening filter it produces good quality cartoonization but time taken for computation is more than cartoonization using LBG ,KPE , KMCG when codebook size is up to 64. A 64 codebook size is apt for producing good cartoonization. BLF requires more time for computation since it requires calculating range kernel and intensity difference.

LBG is a very old and effective method of vector quantization when applied in accordance with line extraction method and feature space conversion method. It produces good cartoonization effect or paint effect on the image. LBG algorithm uses constant error deviation for clustering it results in cluster elongation horizontal axis in two dimensional. This results in inefficient clustering. Also LBG requires Euclidean distance calculation which increases the time taken for processing the image[13][1].

KPE overcomes the drawback of LBG since it doesn't use fixed error value. It uses a proportionate error. The drawback of KPE is that it uses Euclidean distance which increases the time taken for processing the image just as in LBG [1].

KMCG vector quantization when applied produces good quality cartoonization. It doesn't require color space conversion and Euclidean distance calculation as seen in LBG and KPE cartoonization. It is fastest to cartoonize an image as compared to LBG and KPE cartoonization. The effect of all cartoonization methods discussed in the paper on few images is shown in Figure 3 and Figure 4[19].

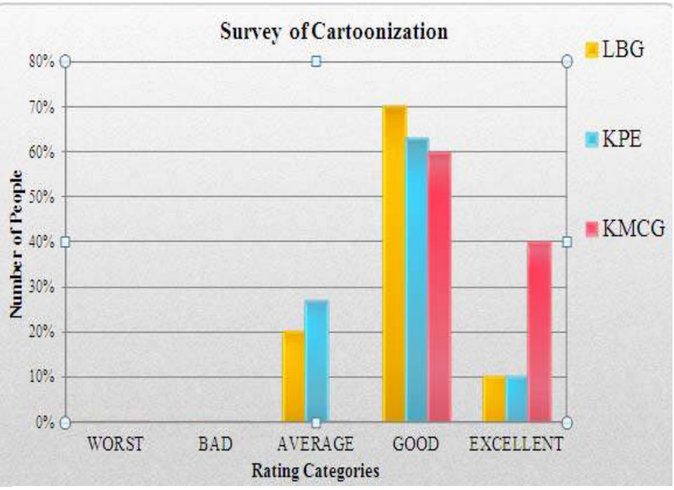


Figure 5: Cartoonized image output review by 30 viewers.

A survey was conducted on vector quantization cartoonization methods to check if the images cartoonized by the method LBG,KPE and KMCG are giving results closer to paint effect cartoonization. 30 participants who use image stylization applications and are in the age group of 15 years to 40 years were shown 10 original images and cartoonized images generated by LBG, KPE and KMCG cartoonization method. The participants were asked to rate the cartoonized effect on the scale of 1 to 5. Where 1 denoted Worst, 2 denoted Bad, 3 denoted Average, 4 denoted Good and 5 denoted Excellent. A bar graph of the result of survey is shown in Figure 5.From the survey it is concluded that maximum participants rated LBG and KPE cartoonization with good cartoonization and KMCG with excellent.

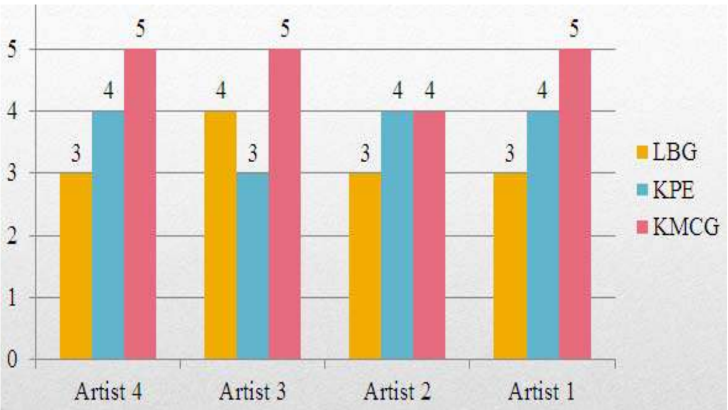


Figure 6: Cartoonized image output review by 4 commercial artists.

IV. CONCLUSION

From the results seen it is concluded that KMCG Cartoonization method works faster than KPE and LBG cartoonization. KPE and LBG produce good cartoonization and KMCG produces excellent painterly effect. The Methods described in the paper can be used for application like movie to comic conversion and other image stylization related applications.

ACKNOWLEDGMENT

We would like to convey our thanks to commercial artists Ms. Snehal Ambre, Mr. Sagar Arun Pitale, Ms. Mansi Dalvi, and Ms. Lubdha Joshi for rating our work from artist point of view. Their suggestions helped us evaluate our work . We would also like to show our gratitude to all the other people who observed and gave their review about the output.

REFERENCES

[1] Dr.Archana B. Patankar, Purnima Tawde," Cartoonization using LBG and KPE Vector Quantization", *International Journal of Modelling and Simulation, 18(2),* 1998, 112-116.

[2] Santella.A and DeCarlo.D, "Visual Interest and NPR: an evaluation manifesto", *Proc. ACM SIGGRAPH'06,* pp. 1221-1226, 2006.

[3] Thomas Strothotte and Stefan Schlechtweg, "Non-photorealistic computer graphics: Modeling, Rendering and Animation (First Edition"), *Publisher: Morgan Kaufman,* 2002.

[4] Richard Hong,Xiao-Tong Yuan , Shuicheng Yan and Tat-Seng Chua,"Movie2comic:Towards a lively video content presentation", *IEEE Tranc. On Multimedia,* Vol(14). No.3,June 2012.

[5] Marr.D and Hildeth.E.C, "Theory of Edge detection", *Proc. Royal Soc.London,Bio.SCi* 1980.

[6] Canny.J.F , "A Computational approach to edge detection". *IEEE Tranc. On Pattern Recognition and machine Intelligence,* Vol(8). No.6,1986.

[7] Tomasi .C and Manduchi .R, "Bilateral filtering for gray and colored images", *Proc. ICCV'98,* pp. 839-842, 1998.

[8] Holger Winnemoller, Sven C. Olsen, Bruce Gooch, "Real time video abstraction", *Proc. ACM DIGGRAPH'06*, pp. 1221-1226, 2006.

[9] Zoya Shahcheraghi, John See, "On the Effects of Pre-and Post – Processing in video cartoonization with Bilateral Filters", *Signal and Image Processing Applications (ICSIPA), 2013 IEEE International Conference* , October 2013.

[10] PietroPerona and JitendraMalik,"Scale-space and edge detection uses anisotropic diffusion". Proceedings of IEEE Computer Society Workshop on Computer Vision,. pp. 16–22, (November 1987).

[11] Kyprianidis J E, Kang H, D¨ollner J, "Anisotropic Kuwahara filtering on the GPU", GPU Pro—advanced rendering techniques, 2010.

[12] Dr. H.B Kekre and Sudeep D. Thepade, "Creating the Color Panoramic view using medley of grayscale and color partial images". *Word Academy Journal of Science and Engineering Technology*, Vol (2). 2008.

[13] Yoseph Linde, Aandres Buzo, “An algorithm for vector quantizer design”, *IEEE Trans Communication, 28(1), 1980.*

[14] M. Gray, “Vector quantization”, *IEEE ASSP Magazine, 28(1), 4-29, 1980.*

[15] Dr. H. B. Kekre, Archana Patankar, Hema Ramesh Galiyal, “Segmentation of blast using vector quantization technique”, *International Journal of Computer Application Security*, 72(15), 2013.

[16] Dr. H.B Kekre, Tanuja K. Sarode, Bhakti Raul, “Color Image Segmentation using KPE for vector quantization”, *International Journal of Computer Science. 3(4),2008, 287-292.*

[17] Dr. H.B Kekre, Tanuja K. Sarode,” New clustering algorithm for vector quantization using rotation of error vector”, *International Journal of computer science and information security,7(3),2010,* 159-165.

[18] H.B.Kekre, TanujaSarode, Sudeep D. Thepade, “Color-Texture Feature based Image Retrieval using DCT applied on Kekre’s Median Codebook”, International Journal on Imaging (IJI), Vol 2, No. A09, Autumn 2009,pp. 55-65.