CANNY EDGE DETECTOR BASED IMAGE/VIDEO ABSTRACTION (CARTOONIZATION)

[1]Deepa C Kugor, [2]Hameem. Shanavas I,
[1]Post Graduate Scholar, [2]Associate Professor,
[1]Department of Electronics & Communication, MVJ College of Engineering,Bangalore

Abstract - Abstraction, is a short summary of the content of a longer document. Specifically, a Video Abstract is a sequence of still or moving images representing the content of a video such that the target image (or images) is rapidly provided with concise information about the content while the essential features of the original are well preserved. It's getting more important to develop fully automated video abstraction algorithms. This work presents a non-photorealistic rendering technique to create stylized abstractions from color images and videos. It aims at modifying the contrast of visually important features. We reduce the contrast in low-contrast regions using a normal recursive filter and artificially increase the contrast in higher contrast regions with a Canny edge detector. Finally, a cartoonized version of an image is obtained by combining the outputs of Canny edge detector and the recursive filter.

Keywords – Abstraction (or Cartoonization); Canny Edge Detector; Line Drawing,; Recursive Filter; Smoothing.

1 INTRODUCTION

A cartoon is a form of two-dimensional illustrated visual art. The concept originated in the Middle Ages and first described a preparatory drawing for a piece of art, such as a painting, fresco etc. That is the reason the dictionary meaning of a cartoon is prescribed to drawings as "a humorous drawing or series of drawings." In the 19th century, it came to refer to humorous illustrations in magazines and newspapers, and in the early 20th century and onward it referred to comic strips and animated films. So, cartoonizing is a field that is developed, still emerging and full of scope.

Cartoons are given much importance because they help to evoke mood or ideas and even elicit emotional response from the viewer. Cartoons provide a vivid sense of reality that is impossible in a realistic video shot through a camera. But creating such animations is very laborious work and needs significant artistic mettle as well as time. So, cartoonization and other forms of stylized rendering are topics of considerable interest in the field of non-photorealistic rendering (NPR). Cartoonizing helps to simplify the visual cues and convey certain aspects of the scene more effectively. Ultimately, it reduces the amount of data to be processed on an image or video.

When artists design imagery to portray a scene, they do not just render visual cues veridically. Instead, they select which visual cues to portray and adapt the information each cue carries. Similarly, the Nonphotorealistic rendering (NPR) in general involves abstraction and stylization of the target scene, which helps simplify the visual cues and convey certain aspects of the scene more effectively. In recent years, automatic abstraction for efficient visual communication is becoming more and more popular to make images and/or videos easier or faster to understand. Depicting information about shapes by line drawing (or Edge Detection) is clearly effective and natural. Lines can be a simple yet effective tool for describing shapes, as demonstrated in many technical or artistic illustrations. Also, use of filters to remove the insignificant details is a basic process in any image processing task.

In this paper, present an automatic technique that generates a stylistic visual abstraction from a photograph or a video. This method is designed to abstract both shapes and colors of the images but in a feature preserving manner. First, the edge detector is imposed on the image or video to extract the boundaries or edges. Then the image smoothing filter is applied on the same set of input images to filter out the insignificant color variations and to enhance the significant parts. Here, the RGB smoothed image ids converted into CIELab colorspace. In the case of video, first the video is converted into continuous frames and then the above mentioned edge detector and the filter are applied on each of the frames. Finally, the edge extracted output of image (or images) is combined with smoothed image (or images) output to yield the cartoonized version of the image or video. Finally, each of these sub-outputs and the cartoonized output will be displayed on the screen.

Real time image abstraction is gaining a huge importance in recent times. Also it is finding its application in the fields of communication, entertainment, broadcasting, video games, novel writing and many more areas. So, an automatic and real time image or video abstraction technique is quite important in the present era.

The organization of the paper is as below. Section II depicts the Literature Review. Section III tells about the exact problem definition and the tasks. Section IV is gives the Proposed Method tat is implemented. Section V gives a detailed depiction about the results of our method. Finally Section VI gives the conclusion and future work.

II LITERATURE REVIEW

Image abstraction, that focuses more on facilitating visual communication and data reduction. This line of work concerns capturing and conveying important image features while minimizing possible distractions from unimportant details. As shape and color are two of the most important features to convey, the existing approaches have focused on solving the corresponding two problems of line drawing and region smoothing.

DeCarlo and Santella employed the Canny edge detector and the mean-shift filter [9] to obtain a cartoon - style image abstraction. They use the edge detector to produce line drawing, while the mean-shift filter performs region smoothing and segmentation. Wang et al. developed an anisotropic mean-shift filter and applied it to create a sequence of image abstractions from a video. Collomosse et al. [10] similarly used the mean shift filter to solve an offline video abstraction problem, focusing on achieving good spatiotemporal coherence. Jan Eric Kyprianidis et al. [7] presented a system that produces a rough sketch of the scene, based adaptive bilateral filtering.

One limitation of the mean-shift segmentation is that it typically produces rough region boundaries as a result of the density estimation in a high-dimensional space. The resulting region boundaries thus require additional smoothing or post editing to obtain stylistic image abstraction. Region segmentation based on the mean-shift filtering is useful for flattening regions but less ideal for producing a sophisticated line drawing, because each segmented region inevitably forms a closed boundary.

Fischer et al. presented a system for producing a stylized augmented reality that incorporates 3D models into a video sequence in a nonphotorealistic fashion. They applied the Canny edge detector and the bilateral filter. Orzan et al. developed a multiscale image abstraction system based on the Canny edge detector and the gradient reconstruction method. Kang et al. showed that it is also possible to obtain image abstraction via stroke-based rendering, constrained by the lines generated from a modified Canny edge detector.

Though Canny edge detector is used for line drawing, there are other line extraction methods as well. Gooch et al. presented a facial illustration system based on a difference-of-Gaussians (DoG) filter, originated from the Marr-Hildreth edge detector. They used this filter in conjunction with binary luminance thresholding

to produce a black-and-white facial illustration. Winnemoller et al. recently extended this technique to

abstract general color images and video, employing the DoG filter for line drawing and the bilateral filter for region smoothing.

III PROBLEM STATEMENT

Given an image that we view as a height field of pixel intensities, the task of image or video abstraction involves the following subtasks:

A.Edge Detection: Captures and displays "significant" height discontinuities.
B. Region smoothing: Removes all "insignificant"height discontinuities.
Solving the first task results in a "line drawing" (see Fig. 3.6), while the second results in a "smoothed" or "flattened" height field (see Fig. 3.7). The combination of these two solutions often results in a cartoonlike image (see Fig. 3.8 and 3.9).

IV PROPOSED METHOD

A . Overview of the System

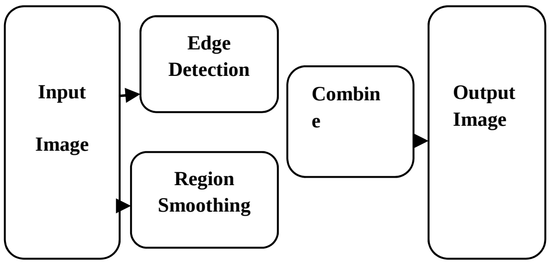

Fig 1. The flow of Abstraction process.

The basic and man block diagram of the abstraction process is shown in the above figure 1. The initial step of abstraction is to feed the input image or a video to the Edge Detection algorithm as well as Region Smoothing filter individually. In case of a video abstraction, first the video has to be fragmented into a number of frames before applying for any of the below algorithms. In this project, we provide the input video frames to the Canny Edge Detector and a basic Recursive IIR Filter. Each of these subtasks are applied o each of the frames for a video input. The line extracted output of the Canny detector is then combined with the Smoothed output of Recursive filter to result in the cartoonized outputAll these algorithms are implemented using MATLAB. Matlab version of 2009 and above can be used for the simulation. This work was carried with Matlab R2009.

B. Canny Ede Detection Algorithm

The purpose of edge detection in general is to significantly reduce the amount of data in an image, while preserving the structural properties to be used for further image processing. Canny edge detector is the optimal and most widely used algorithm for edge detection. Compared to other edge detection methods like Sobel, Prewitt etc Canny edge detector provides robust edge detection, localization and linking. It is a multi-stage algorithm and the stages involved are illustrated in Figure 2. Thus, instead of providing the whole algorithm as a single API, kernels are provided for each stage. This way, the user can have more flexibility and better buffer management.

C. Smoothing

Blurring of the image to remove noise. It is inevitable that all images taken from a camera will contain some amount of noise. To prevent that noise is mistaken for edges, noise must be reduced. Therefore the Gaussian is applied to convolve with the image. This step will slightly smooth the image to reduce the effects of obvious noise on the edge detector. The equation for a Gaussian filter kernel with the size of 2k+1*2k+1 is shown as follows:

$$g(x, y) = \frac{1}{2\pi\sigma^2} \exp\{\frac{-(x^2+y^2)}{2\sigma^2}\} \quad (1)$$

Where $g(x, y)$ is the Gaussian operator, σ is the standard deviation, and finally x and y are the coordinates.

It is important to understand that the selection of the size of the Gaussian kernel will affect the performance of the detector. The larger the size is, the lower the detector's sensitivity to noise. Additionally, the localization error to detect the edge will slightly increase with the increase of the Gaussian filter kernel size. A 5*5 is a good size for most cases, but this will also vary depending on specific situations.

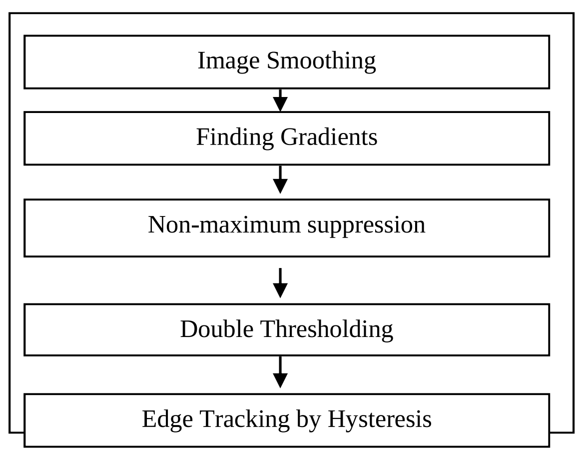

Fig 2. Flow Diagram of Canny Edge Detector

D.Finding Gradients

The Canny algorithm basically finds edges where the grayscale intensity of the image changes the most. An edge in an image may point in a variety of directions, so the Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image. The edge detection operator (Roberts, Prewitt, Sobel for example) returns a value for the first derivative in the horizontal direction and the vertical direction .

From this the edge gradient magnitudes (also known as the edge strengths) and direction can be determined as in equation 2 and 3. The computed edge strengths are compared to the smoothed image.

$$|G| = \sqrt{(Gx^2 + Gy^2)} \quad (2)$$

$$\Theta = \arctan\left(\frac{|Gy|}{|Gx|}\right) \quad (3)$$

where Gx and Gy are the gradients in the horizontal and vertical directions respectively.

The edge direction angle is rounded to one of four angles representing vertical, horizontal and the two diagonals (0, 45, 90 and 135 degrees for example). An edge direction falling in each color region will be set to a specific angle values, for example alpha lying in yellow region (0 to 22.5 degrees and 157.5 degrees to 180 degrees) will be set to 0 degree.

E. Non-Maximum Suppression

Non-Maximum suppression is applied to "thin" the edge. After applying gradient calculation, the edge extracted from the gradient value is still quite blurred. there should only be one accurate response to the edge. Thus non-maximum suppression can help to suppress all the gradient values to 0 except the local maximal, which indicates location with the sharpest change of intensity value.

The algorithm for each pixel in the gradient image is:

1) Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient directions.
2) If the edge strength of the current pixel is the largest compared to the other pixels in the mask with the same direction(i.e, the pixel that is pointing in the y direction, it will be compared to the pixel above and below it in the vertical axis), the value will be preserved. Otherwise, the value will be suppressed.

At every pixel, it suppresses the edge strength of the center pixel (by setting its value to 0) if its magnitude is not greater than the magnitude of the two neighbors in the gradient direction. For example,if the rounded gradient angle is zero degrees (i.e. the edge is in the north–south direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the east and west directions. Same way, for all the other directions.

If the gradient angle is between 45 degrees and 90 degrees, interpolation between gradients at the north and north east pixels will give one interpolated value, and interpolation between the south and south west pixels will give the other (using the conventions of last paragraph). The gradient magnitude at the central pixel must be greater than both of these for it to be marked as an edge.

F. Double Thresholding

After application of non-maximum suppression, the edge pixels are quite accurate to present the real edge. However, there are still some edge pixels at this point caused by noise and color variation. In order to get rid of the spurious responses from these bothering factors, it is essential to filter out the edge pixel with the weak gradient value and preserve the edge with the high gradient value. Thus two threshold values are set to clarify the different types of edge pixels, one is called high threshold value and the other is called the low threshold value. If the edge pixel's gradient value is higher than the high threshold value, they are marked as strong edge pixels. If the edge pixel's gradient value is smaller than the high threshold value and larger than the low threshold value, they are marked as weak edge pixels. If the pixel value is smaller than the low threshold value, they will be suppressed. The two threshold values are empirically determined values, which will need to be defined when applying to different images.

G.Edge tracking by hysteresis

Strong edges are interpreted as “certain edges”, and can immediately be included in the final edge image. Weak edges are included if and only if they are connected to strong edges. The logic is of course that noise and other small variations are unlikely to result in a strong edge (with proper adjustment of the threshold levels). Thus strong edges will (almost) only be due to true edges in the original image. The weak edges can either be due to true edges or noise/color variations. The latter type will probably be distributed independently of edges on the entire image, and thus only a small amount will be located adjacent to strong edges. Weak edges due to true edges are much more likely to be connected directly to strong edges. Edge tracking can be implemented by BLOB-analysis (Binary Large OBject). The edge pixels are divided into connected BLOB's using 8-connected neighborhood. As long as there is one strong edge pixel is involved in the BLOB, that weak edge pixel can be identified as one that should be preserved.

H.Recursive Filter for Image Smoothing

The goal of region smoothing is to remove unimportantdetails from the region interiors while preserving important shapes. Filters are a basic component of all image processing systems to achieve smoothing. The primary function of a filter can be many. But, here the purpose of using filter is to smoothen the image or removal of noise from the images. This can be called as Image Filtering.

Image Filtering

It computes a function of a local neighborhood at each pixel position. Maps each pixel's value to a new value. It's called as "local operator" or "neighborhood operator". In local Operator the value of pixel in output image is a function of the corresponding pixel in the input image plus other nearby pixels (usually defined by a square *or* rectangular window centered on the given pixel. It does the following.

- ❖ Enhance images - Noise reduction, smooth, resize, increase contrast, recolor, artistic effects, etc.
- ❖ Extract features from images - Texture, edges, distinctive points, etc.
- ❖ Patterns - Template matching, e.g., eye template

Recursive filters are an efficient way of achieving a long impulse response, without having to perform a long convolution. They execute very rapidly. Recursive filters are also called Infinite Impulse Response (IIR) filters, since their impulse responses are composed of decaying exponentials. This distinguishes them from digital filters carried out by convolution, called Finite Impulse Response (FIR) filters.

The main difference between IIR filters and FIR filters is that an IIR filter is more compact in that it can usually achieve a prescribed frequency response with a smaller number of coefficients than an FIR filter. A smaller number of filter coefficients imply less storage requirements and faster calculation and a higher throughput. Therefore, generally IIR filters are more efficient in memory and computational requirements than FIR filters.

Consider x[n], x[n-1], x[n-2]…..x[0] be the input signal values and y[n], y[n-1], y[n-2]…..y[0] be the output sequences to be computed. The definitions of non-recursive and the recursive filter are as follows.

Non-Recursive filter: The current value of output value (y[n]) is calculated solely from the current and previous input values (x[n], x[n-1], x[n-2]…..).

Recursive Filter: Here, in addition to the input values, the previous output values are also necessary to compute the current output y[n].

All these previous values are stored in the processor's memory. The algorithm would be in the form:

$$y[n] = a0x[n] + a1x[n-1] + a2x[n-2] + \cdots b1y[n-1] + b2y[n-2] + \ldots \quad (4)$$

In words, each point in the output signal is found by multiplying the values from the input signal by the "a" coefficients, multiplying the previously calculated values from the output signal by the "b" coefficients, and adding the products together. Notice that there isn't a value for b0 because this corresponds to the sample being calculated.

The order of a recursive filter is the largest number of the previous input or output values required to compute the current output. We can see from the above definitions that recursive filter is same as the non-recursive filter with additional terms y[n-1], y[n-2]…etc. The alternative symmetrical form of the equation can be written as

$$b0y[n] + b1y[n-1] + b2y[n-2] = a0x[n] + a1x[n-1] + a2x[n-2] \quad (5)$$

The coefficients of inputs are denoted by a's and coefficients of outputs are denoted by b's. Recursive filters are useful because they bypass a longer convolution.

V. RESULTS AND DISCUSSION

Fig 3.1: Input Image

Fig 3.5: After Thresholding

Fig 3.7: Filtered image (using Recursive IIR Filter)

C. Cartoonized Output Images

Fig 3.8: Cartoonized image due to fusion

Fig 3.9: Cartoonized image due to repmat function

In the experimentation using Matlab, the final step is to combine the two images from Canny detector and the filtered output. This can be done by using the Matlab function to fuse the images, WFUSIMG as shown in Fig 3.8. The more cartoon effect is given by using the matlab function for fusion, REPMAT as shown in the Fig 3.9. The same process was carried out for each frame in the case of video.

VI. CONCLUSION AND FUTURE SCOPE

An automatic technique for image and video cartoonization (or abstraction) was presented in the paper. For the video abstraction, first the video has to be converted into frames. And then Canny edge detector and Recursive filter are applied individually for each frame in both the edge tracking and smoothing process. Later, these frames portraying the edges from Canny are to be superimposed with the frames that were smoothed by the filter. The output will be a cartoonized version of the inputted video. Here we have applied the algorithms on the extracted frames of a video that were saved in a directory. The future works can focus on cartoonizing the video directly without the need of saving into a directory. Also, the delay between the cartoonization of each frame can be minimized using some alternatives. This abstraction worked well for short span videos. So, works can also be taken to abstract longer videos.

VII. REFERENCES

[1] Henry Kang, Seungyong Lee and Charles K. Chui, “Flow-Based Image Abstraction”, IEEE Transactions On Visualization And Computer Graphics, Vol. 15, No. 1, January/February 2009.

[2] Jan Eric Kyprianidis and Henry Kang, “Image and Video Abstraction by Coherence-Enhancing Filtering”, EUROGRAPHICS 2011 / M. Chen and O. Deussen, Volume 30 ,Number 2,2011.

[3] Hanli Zhao, Xiaogang Jin, Jianbing Shen, XiaoyangMao, and Jieqing Feng, “Real-time feature-aware video abstraction”, Original Article,Visual Comput DOI Volume 10, 2008.

[4] Y Senthil Kumar, “Canny Edge Detection Implementation on TMS320C64x/64x+ Using

VLIB", Application Report SPRAB78– November 2009, Texas Instruments.

[5] S.Jagadeesh Babu, P.Karunakaran, S.Venkatraman, I.Hameem Shanavas and T.Kapilachander, "Edge Detection System using Pulse Mode Neural Network for Image Enhancement", I.J. Image, Graphics and Signal Processing, Volume 3, Page No: 42-48, 2012.

[6] Mitra Basu, "Gaussian-Based Edge-Detection Methods—A Survey", IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 32, No. 3, August 2002.

[7] Jan Eric Kyprianidis and Jürgen Döllner, "Image Abstraction by Structure Adaptive Filtering", EG UK Theory and Practice of Computer Graphics ,2008.

[8] Hameem Shanavas.I , Gnanamurthy R K, "Wirelength Minimization in Partitioning and Floorplanning using Evolutionary Algorithm" VLSI Design" Vol.2011.

[9] Doug DeCarlo Anthony Santella, "Stylization and Abstraction of Photographs", In SIGGRAPH 2002.

[10] J. P. Collomosse and P. M. Hall, "Cubist Style Rendering from Photographs", IEEE Transactions On Visualization And Computer Graphics, Vol. Xx, No. Y, Month 2002.

[11] Parimal Sikchi, Neha Beknalkar, and Swapnil Rane, "Real-Time Cartoonization Using Raspberry Pi", IJCAT, International Journal of Computing and Technology, Volume 1, Issue 6, July 2014.