JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

REAL TIME IMAGE CARTOONIFICATION USING COMPUTER VISION EDGE AND CONTOUR FILTER METHODS

1Ms.P.Sasitha, [2]K.Sireesha, [3]M.Poojitha,

[4]R.Venkata Sai, [5]M.Dinesh,[6]D.Dinkar

[1]Assistant Professor, Department of ECE, PBR VITS, India,
[2]UG Student Department of ECE, PBR VITS, India,
[3] UG Student Department of ECE, PBR VITS, India,
[4] UG Student Department of ECE, PBR VITS, India,
[5] UG Student Department of ECE, PBR VITS, India,
[6] UG Student Department of ECE, PBR VITS, India,

Abstract : This paper presents a novel approach for real-time image cartoonification leveraging advanced computer vision techniques, specifically edge and contour filter methods. The proposed method aims to transform input images into cartoon-like representations while preserving important visual features and details. The process involves multiple stages, including edge detection to extract key outlines and contours, followed by a series of filtering operations to enhance and stylize the image. Additionally, real-time performance is achieved through efficient algorithm design and implementation, enabling seamless integration into various applications such as image editing software, augmented reality, and digital entertainment platforms. Experimental results demonstrate the effectiveness and versatility of the proposed approach in producing high-quality cartoon-like images with minimal computational overhead, paving the way for enhanced user experiences in visual content creation and consumption.

Index Terms **– Bilateral Filtering, Adaptive Thresholding, Virtual Reality, Augmented Reality**

I. INTRODUCTION

Real-time image cartoonification, a fascinating application of computer vision, employs edge and contour filter methods to transform ordinary images into captivating cartoons instantly. Leveraging edge detection algorithms like Canny or Sobel, it identifies and enhances prominent edges in the image, mimicking the bold outlines characteristic of cartoons. Additionally, contour filter techniques refine these edges, smoothing out imperfections and amplifying visual appeal. By combining these methods, real-time cartoonification breathes new life into images, offering users a playful and artistic way to engage with their digital content.

II. LITERATURE SURVEY

A comprehensive literature survey on real-time image cartoonification involves a thorough exploration of existing research aimed at transforming digital images into cartoon-style representations in real-time. Researchers typically begin by scouring academic databases for recent publications, focusing on papers that discuss techniques and methodologies relevant to this specific task. Key areas of interest include edge detection, contour filtering, color quantization, and stylization methods, all of which contribute to achieving the desired cartoon effect. The survey entails a detailed examination of the approaches adopted in various studies, including their computational complexity, robustness, and ability to produce visually appealing results in real-time applications. Researchers analyze the effectiveness of different algorithms and evaluate their performance based on metrics such as edge preservation, color fidelity, and overall cartoon quality. Moreover, they identify common challenges encountered in real-time cartoonification, such as balancing computational speed with image fidelity, handling diverse image content, and ensuring consistency in cartoon style. By synthesizing findings from multiple papers, researchers gain insights into emerging trends, promising techniques, and areas requiring further investigation, ultimately contributing to advancements in the field of real-time image cartoonification

III. EXISTING METHOLOGY

One existing method for real-time image cartoonification using computer vision edge and contour filter methods involves a multi-stage process.

3.1. Edge Detection: Begin by detecting edges in the image using techniques like Canny edge detection or Sobel operators. This step highlights the boundaries between different regions in the image.

3.2. Contour Detection: Next, identify and extract the contours from the detected edges. This step helps in delineating the shapes and forms within the image.

3.3. Simplification: Simplify the detected contours to reduce the complexity and achieve a more stylized cartoon-like appearance. This can involve techniques such as line thinning or simplification algorithms.

3.4. Color Reduction: Reduce the number of colors in the image to give it a more graphical, cartoon-like appearance. This can be achieved through methods like color quantization or palette reduction.

3.5. Enhancement: Optionally, apply further enhancements such as adding outlines or adjusting contrast and brightness to enhance the cartoon effect.

These steps can be optimized for real-time performance by employing efficient algorithms and parallel processing techniques, allowing for the cartoonification process to be performed on video streams or live camera input in real-time.

IV. DISADVANTAGES OF EXISTING METHOLOGY

Computational demands for real-time processing can strain systems, particularly with high-resolution images or video streams. Accuracy poses another challenge, as edge and contour filters may not faithfully capture intricate details, leading to information loss or distortions in the resulting cartoonified images. Additionally, the process may introduce artifacts or undesired effects, especially around edges and fine features. There's a degree of subjectivity involved, as optimal settings for desired cartoon effects can vary and require manual adjustment. Furthermore, the approach's effectiveness might not generalize well across different image content and complexities. Resource-intensive demands limit its practicality on devices with constrained computational capabilities. Over-simplification of certain features or textures can compromise visual fidelity or semantic information. Dynamic scenes with moving objects or changing backgrounds pose additional challenges, necessitating continuous tracking and updating of cartoonified outputs. Meeting user expectations for high-quality cartoonification comparable to hand-drawn illustrations remains an ongoing challenge, necessitating further research and refinement of existing methods.

V. PROPOSED METHOLOGY

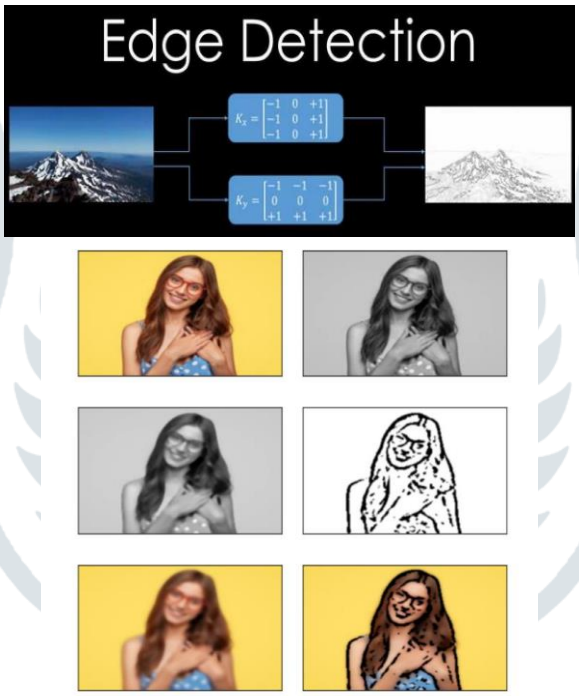
To achieve real-time image cartoonization. It starts by converting the captured frame to grayscale of median filtering and then applies bilateral filtering to reduce noise while preserving edges. Edge detection is performed using adaptive thresholding, and the detected edges are combined with the color image using bitwise AND operation.

(a)

We propose to create a website, which consists of image upload functionality using which the user can upload his image, the uploaded image is then processed by server using Neural style transfer algorithm and the resulting image is presented to the user on the website. Which then user can download & share. Neural fast style transfer is used by Apps such as https://deepart.io, Prisma, Artisto etc. We decided to choose this approach over traditional image filters (e.g. using image filters such as median & bilateral filters to posterize an Image) as Neural fast style transfer is quite new and challenging technique which uses machine learning & image processing to produce various styled images based on variety of input & style images. The algorithm can be implemented in Python/JavaScript/Lua to perform neural style transfer. We will use Python to implement the backend and the front end of the website will be in JS. Basically, in Neural Style Transfer we have two images- style and content. We need to copy the style from the style image and apply it to the content image. By, style we basically mean, the patterns, the brushstrokes, etc. we will provide a set of style images which a user can use to apply different kinds of Cartoon like effects to his image.

User will be provided with a set of pretrained style images to choose from. Based on the chosen style and the content image provided by the user, the Resulting image with cartoon like effect is generated by the program. The implementation is based on of the combination of Gatys' A Neural Algorithm of Artistic Style, Johnson's Perceptual Losses for Real-Time Style Transfer and Super-Resolution, and Ulyanov's Instance Normalization (all 3 papers mentioned above).

The algorithm iterates over each captured frame in real-time, continuously applying these steps to achieve the desired cartoonization effect. The combination of noise reduction, edge preservation, and color integration results in visually appealing cartoon-style representations of live video feeds or captured images. This algorithmic approach enables dynamic and interactive cartoonization of images, making it suitable for applications requiring real-time processing and visualization.



VI. RESULTS AND DISCUSSION

The real-time cartoonization, leveraging a series of meticulously chosen image processing techniques. Initially, the conversion to grayscale simplifies images by stripping away color information while preserving essential luminance values, laying a foundation for subsequent processing stages. Subsequently, median filtering with blur value of (5) in to mitigate noise, a common nuisance in webcam or video feed images, ensuring that the input for further processing remains pristine and artifact-free. The introduction of adaptive thresholding with thresholding values of (255,9,9) facilitates the detection of edges, dynamically adapting to local image characteristics and delineating object boundaries from the background. This is particularly crucial in realtime scenarios where lighting conditions may vary, ensuring robust edge detection across different environments. Bilateral filtering with thefiltering values of (9,300,300) emerges as a key player in the pipeline, orchestrating the delicate balance between noise reduction and edge preservation. By considering both spatial distance and intensity difference, this filter smooths the color image while respecting significant edges, laying the groundwork for subsequent stylization. The Bitwise AND operation then takes center stage, artfully combining detected edges with the color image or mask image to yield a visually striking cartoon-like effect which was clearly shown in fig

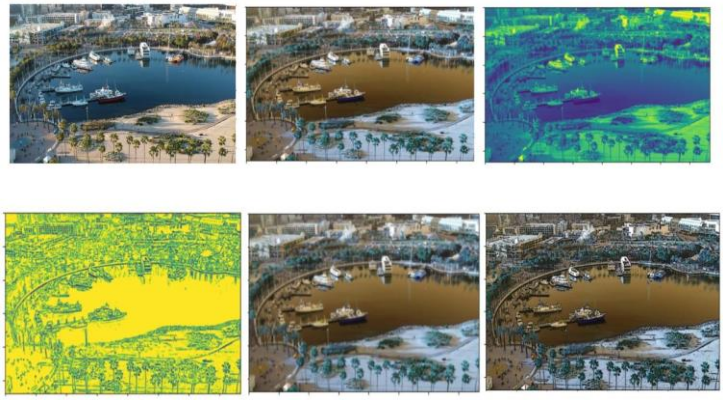


Fig: Output Of Real Time Image Cartoonification

VII. ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression, “Oneofus(R.B.G.)thanks...”
Instead,try“R.B.G.thanks”.Putapplicablesponsoracknowledgmentshere;DONOTplacethemonthefirstpageofyourpaperorasafootnote.

VIII. REFERENCES

[1]. Abdou, E.; and Pratt W. K. (1979), ―Quantitative Design and Evaluation Enhancement/Thresholding Edge Detectors,‖ Proceedings of IEEE, 67(5): 753–763.

[2]. Abreu, E.; Lightstone, M.; Mitra, S. K.; Arakawa, K. (1996), ―A New Efficient Approach for Removal of Impulse Noise from Highly Corrupted Images‖, IEEE Transactions on Image processing, 5(6):1012-1025.

[3].Archibald, R.; Gelb, A.; and Yoon, J. (2005),‖Polynomial Fitting for Edge Detection in Irregularly Sampled Signals and Images,‖ SIAM Journal on Numerical Analysis, 43(1):259-279, 2005.

[4]. Archibald, R.; Gelb, A.; and Yoon, J. (2008), “Determining the Locations and Discontinuities in the Derivatives of Functions,‖ Applied Numerical Mathematics, 58(5):577-592.

[5]. Argenti, F.; and Alparone, L. (2002), ―Speckle Removal from SAR Images in Undecimated Wavelet Domain,‖ IEEE Transactions on Geoscience and Remote Sensing, 40(11): 2363-2374.

[6]. Bennamoun, M.; Boashash, B.; and Koo, J. (1995), ―Optimal Parameters for Edge Detection,‖ in Proceedings of IEEE International Conference on SMC, 2:1482–1488.

[7]. Bergholm, E. (1987), ―Edge Focusing,‖ IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(6):726-741.

[8]. Blake, A.; and Yuille, A. (1992), Active Contour, MIT Press, Cambridge,Massachusetts.

[9].Davis, L. S. (1975), ―A Survey on Edge Detection Techniques,‖ Computer Graphics and Image Processing,‖ 4: 248-270.

[10]. Demystifying Neural Style Transfer, 2017 - Yanghao Li, Naiyan Wang, Jiaying Liu, Xiaodi Ho

[11]. Huang, J.S.; and Tseng, D.H. (1988), ―Statistical Theory of Edge Detection,‖ Computer Vision Graphics Image Processing, 43(3):337–346.

[12]. Hueckel, M. (1971), ―An Operator which Locates Edges in Digitized Pictures,‖ Journal of the ACM, 18(1):113–125.