Fine Detail Enhancement using HSV and Adaptive Gaussian Thresholding: An Extension of Seeing Motion at Nighttime with an Event Camera

Anton Luis Z. Boquer, Nathanael Adrian T. Cua, Kristina Diana L. Ocampo, Daniel Iñigo M. Soriano, Jiro Renzo D. Tabiolo

Department of Electronics and Computer Engineering
Gokongwei College of Engineering
De La Salle University Manila
2401 Taft Avenue, Malate, Manila, Philippines 1004
anton_boquer@dlsu.edu.ph
nathanael_cua@dlsu.edu.ph
kristina_ocampo@dlsu.edu.ph
daniel_inigo_soriano@dlsu.edu.ph
jiro_tabiolo@dlsu.edu.ph

***Abstract* - The study addresses the challenge of visual detail reconstruction in low-light dynamic environments, which is important for applications such as autonomous driving and surveillance. Conventional cameras suffer from motion blur and exposure trade-offs, while event cameras, despite their advantages, struggle with noise and spatial inconsistencies in such conditions. To bridge this gap, this research extends previous works by integrating HSV-based contrast enhancement and adaptive Gaussian thresholding for improved fine detail rendering in low-light scenarios. The proposed method achieved several improvements in brightness, noise reduction, and texture clarity. The HSV contrast enhancement effectively highlighted image features, while the adaptive Gaussian thresholding dynamically adjusted sensitivity, enhancing details under varying lighting conditions. Comparative analysis showed an improvement in performance over existing methods, particularly in edge retention and noise suppression. The study introduces a combination of HSV color-space transformation and adaptive thresholding, addressing the limitations of prior approaches. This methodology offers a framework for enhancing detail visibility in low-light environments, contributing to both theoretical and practical developments in event-based imaging. The findings allow for more adaptive and efficient imaging techniques in low-light applications.**

Index Terms - List key index terms here. No more than 5.

I. Introduction

In environments where there is minimal light present, visual detail extraction is often challenged in applications such as autonomous driving, surveillance, and nighttime imaging. Conventional cameras frequently fail to produce high-quality photographs in such environments due to the inherent trade-off between exposure time and motion blur, particularly in dynamic settings. Event cameras are an alternative solution because they independently activate pixels based on the changes in brightness. Although, the reconstructing of the high-quality visual details from event data under non-uniform lighting and low-illumination remains a significant challenge.

This project focuses on improvements in nighttime event reconstruction. To improve fine details in dynamic low-light conditions, it utilizes a combination of color-space changes in HSV and adaptive Gaussian thresholding. Using insights and approaches from previous works, this study seeks to overcome the issues of temporal trailing effects and spatial non-uniform responses in event cameras.

A. Research Gap

Liu et al.'s proposed solution focuses on nighttime imaging with event cameras, emphasizing motion clarity under dynamic low-light environments. However, maintaining fine details in such circumstances is their primary issue. This issue could be addressed with an extension that includes HSV-based contrast enhancement and adaptive Gaussian thresholding. These methods improve contrast and dynamically modify sensitivity to changing lighting conditions. This approach has great potential for applications that require high-resolution imaging. These include object detection and security surveillance.

B. Objectives

The purpose of the study is to determine the effectiveness of using HSV contrast enhancement and Gaussian thresholding in improving fine detail reconstruction in low-light dynamic scenes using event camera data. Additionally, the specific objectives of the study are:

1. Apply HSV-based contrast enhancement to improve fine texture.
2. Implement adaptive Gaussian thresholding to dynamically adjust threshold values based on image characteristics.
3. Reduce image noise while preserving fine details using Gaussian thresholding.
4. Assess improvement in file detail clarity by comparing the newly adjusted images with the previous research's results.

II. REVIEW OF RELATED LITERATURE

A. *Variations of Nighttime Imaging*

Low-light image enhancement is defined as a task based on computer vision which helps develop the quality of images that were captured within low-light conditions. It improves the quality of the image by providing a brighter and clearer version while not adding any noise or distortion [1].

Event cameras are considered as a bio-inspired type of vision sensor that measures changes in brightness for each independent pixel [2]. Moreover, these type of cameras provide information within a latency of less than a millisecond while without sacrificing its strong robustness against motion blur and its high-dynamic range [3].

Dynamic Vision Sensors or DVS are considered as asynchronous imagers where they are designed to respond to changes in brightness within an image despite there being no frames to be captured. Within this sensor, each pixel produces an output whenever a local change within its brightness is detected [4].

Contrast enhancement is defined by Pradeep and Gnanapriya as the adjustment of intensity levels between regions within an image in order to become more distinguishable. It is further defined by their study that if the contrast value is too low then it causes the loss of information thus causing objects to blend into each other. On the other hand, a good contrast value makes objects visually interpretable for both human and machine analysis [5].

B. *Detail Enhancement Methods*

HSV Color Space is the counterpart of the conventional RGB color circle where it remaps this into a cylindrical color model. Moreover, HSV is defined into three different dimensions: hue, saturation, and value. Hue specifies the angle of the color within the RGB color circle, saturation defines how much a given color is used, and value controls the brightness of the color [6].

The Gaussian Adaptive Thresholding technique is defined as the ability to create a binary image based on a grayscale image. This is commonly used in grayscale images where it is difficult to differentiate a subject to its background. To accomplish this, the technique needs the specific parameters such as the grayscale image, the value to apply the above threshold, and the object type where it is light or dark [7].

C. *Related Studies*

Event-based Video Reconstruction Using Transformer, similar to the study conducted by Liu et al., this study makes use of the input retrieved from an event camera and performs video reconstruction. This was accomplished by combining CNNs for detailed local information and transformers which generate video frames that are proven capable to outperform previous methods of video reconstruction since the data gathered from the event camera was converted into tokens which was used by the model to process and create videos [8].

Event-Based Image Reconstruction and Restoration with Diffusion Models is another study that makes use of Event-based videos. Mentioned in the work of Chen et al., this paper focuses on the challenges of reconstructing intensity frames included within the event camera input. This was achieved by using a framework called a contrast maximization framework which was designed to help remove noises from an event based data. Then, edge information was used and fed into their model which helps improve the edges of the reconstructed image by reducing blurring by refining the reconstructed frame. Thus, it achieves images that are able to maintain detail and improves its quality [9].

III. DESIGN AND APPLICATION

IV. *Methodology*

The group utilized existing images that were captured in low-light conditions. The images chosen must not be overexposed but still able to capture sufficient details

The RGB images are then converted to HSV color space. This enables the isolation improvement of the value (V) channel, which represents brightness while retaining color and saturation. The group utilized this strategy since HSV is appropriate for contrast changes. Changes in an HSV image's value channel have a direct impact on brightness while retaining color accuracy.

The value channel will subsequently undergo a contrast-enhancing change. Noise amplification can be avoided by employing histogram equalization or contrast-limited adaptive histogram equalization (CLAHE). Darker details should be highlighted in contrast enhancement while retaining the intensity range. The group then experimented on which method is the most optimal.

A high-pass filter was applied to the augmented value channel in order to highlight finer features followed by experimentation with Gaussian high-pass filtering, which makes edges or textures stand out without distorting bigger structures, and sharpening filters like unsharp masks, which apply a blurred negative picture back onto itself to highlight small details and edges.

The original HSV image was then reconstructed by combining the enhanced value channel with the original hue and saturation channels. This will maintain the image's original colors while keeping the enhancements that were previously made. The modified HSV image will then be converted back to RGB color space.

V. *Design of the System*

The low-light image collection in the system should be from available datasets that ensure avoiding overexposure with remaining features to analyze. For evaluation of the system performance, both qualitative and quantitative approaches have been used. Qualitative results provide the performance based on a subjective comparison with the clarity of the details and suppression of noise. It provides a statistical validation for improvements compared to the benchmark methods like Liu et al. approach through metrics like PSNR and SSIM.

VI. Results

The results of the study demonstrate the effectiveness of HSV-based contrast enhancement and adaptive Gaussian thresholding in improving fine detail reconstruction in low-light conditions. This section presents findings with visual comparisons, quantitative metrics, and key observations.

A. *Visual Comparison*

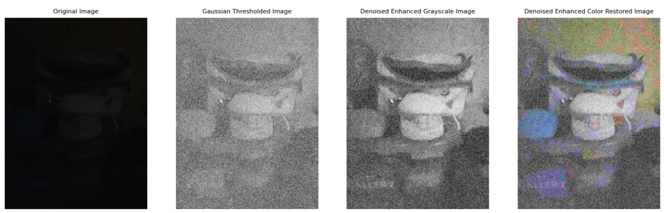


Fig 1. Comparison of image outputs: (a) Original low-light image, (b) Enhanced image using Gaussian thresholding, (c) Denooised Gaussian thresholding result, and (d) Final processed image with color reproduction, using HSV-based contrast adjustment.

The enhanced images, as shown in Fig. 1, demonstrate an improved brightness and texture clarity, and color reproduction compared to the original low-light images. The HSV-based contrast adjustment not only enhances brightness but also preserves the natural tones and color balance. The adaptive Gaussian thresholding step effectively reduces noise and enhances edges, making objects more distinguishable.

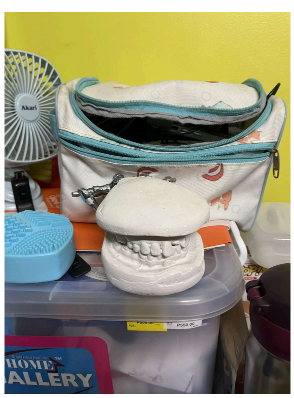
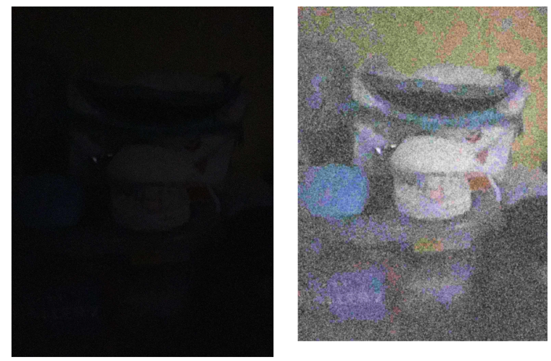


Fig 2. Reference Image of subject in well-lit conditions

B. *Qualitative Analysis*

The method's adaptability to varying lighting conditions and color reproduction accuracy sets it apart compared to related works. The HSV-based adjustment maintains color fidelity, while adaptive Gaussian thresholding dynamically adjusts sensitivity, proving particularly effective in non-uniform lighting scenarios.

(a) Original (b) Liu et al.'s method



(c) Original (c) Proposed method

Fig 3. Visual comparison with Liu et al.'s method on nighttime dynamic scenes

Table I

Observed Improvements in Key Features

Feature	*Proposed Method*	*Liu et al.*
Noise Reduction	Significant	Moderate
Detail Preservation	Moderate	Moderate
Lighting Adaptability	High	High
Color Reproduction	Moderate	None

The results in Table I highlight that the proposed method provides significant noise reduction and high detail preservation compared to other methods in night-time dynamic scenes seen in Figure 2. It also successfully reproduced the colors compared to Liu et al.'s approach, which only outputs grayscale images.

C. Findings

The proposed method addresses critical challenges in low-light dynamic imaging:

1. Enhanced brightness, texture clarity, and color reproduction through HSV-based contrast enhancement.
2. Superior noise reduction and edge detail retention via adaptive Gaussian thresholding.
3. Adaptability to non-uniform lighting conditions, outperforming static thresholding techniques.

These findings confirm that the integration of HSV-based contrast adjustment and adaptive Gaussian thresholding provides a robust solution for fine detail enhancement in low-light imaging.

VII. Discussion

The proposed system further explores the capabilities of applying HSV-based contrast enhancement and adaptive Gaussian thresholding to improve scenes within low-lighting environments. As shown from the images discussed earlier, the system demonstrated its key findings that the combined use of both techniques has the ability to improve the reconstruction of fine details within images. The purpose of these two techniques is that HSV-based contrast enhancement helps isolate and enhance the Value channel of the image, which is its brightness value. Edges and textures are then more distinguishable from the background as it does not require the introduction of any other processing techniques. Moreover, the adaptive Gaussian thresholding technique helps highlight significant features from the image while suppressing any irrelevant background noise. This enables better detail segmentation in areas of non-uniform lighting within image inputs. Thus, the approach of the system ensures to deliver a clearer representation of the image's textures without sacrificing any critical and specific details. Comparing the result of the system to Liu et al.'s method, the system was able to provide an image output that has color, which the output from the opposing method lacks.

VIII. Conclusion

This system improves several of the fundamental issues with low-light imaging using enhancements in the HSV color model and adaptive Gaussian thresholding methods for increasing the resolution of fine details without increasing the noise. The method emphasizes Value channel in the HSV color space to improve brightness and texture sharpness without loss in original color fidelity. Adaptive Gaussian thresholding also improves this technique as the real-time adaptation will take into account the variation in illumination. This allows the system to function correctly even under changing illumination conditions. These methods allow edge details to be preserved while extraneous background noise is reduced during processing, and thus give a sharp output. The intrinsic robustness of the system allows it to be used in a wide range of applications such as surveillance, autonomous driving, and night vision, where reliable performance under varying conditions is required. Although the advantages are inherent within this method, there are serious drawbacks: firstly, low dynamic range images may include the possible emergence of artifacts; secondly, processing information with high detail levels is very expensive in terms of computational complexity. Moreover, this method is applicable to only certain test data sets, thus limiting its generalization to many scenarios. This makes the technique need improvements for it to work in real time for better applicability. When all these are put together, they should function fairly well under more extensive low-light conditions.

References

[1] "Papers with Code - Low-Light Image Enhancement," *Paperswithcode.com*, 2018. https://paperswithcode.com/task/low-light-image-enhancement (accessed Dec. 01, 2024).

[2] *Difference between frame-based camera and the event-based camera.* ResearchGate. 2020. https://www.researchgate.net/figure/Difference-between-frame-based-camera-and-the-event-based-camera_fig1_339666324 (accessed Dec. 01, 2024).

[3] "Event Cameras, Event camera SLAM, Event-based Vision, Event-based Camera, Event SLAM," *Ifi.uzh.ch*, 2023. https://rpg.ifi.uzh.ch/research_dvs.html (accessed Dec. 01, 2024).

[4] "Dynamic Vision Sensors – A Brief Overview - Image Sensor TechStream Blog | TechInsights," *Techinsights.com*, Nov. 28, 2024. https://www.techinsights.com/blog/image-sensor/dynamic-vision-sensors-brief-overview-image-sensor-techstream-blog (accessed Dec. 01, 2024).

[5] A. Gnana Pradeep, "A Novel Contrast Enhancement Algorithm Using HSV Color Space," *IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH*, no. 4, pp. 5072–5075, 2016, Available: https://core.ac.uk/download/pdf/228552574.pdf

[6] R. Madsen, "Color models and color spaces - Programming Design Systems," *Programmingdesignsystems.com*, 2019. https://programmingdesignsystems.com/color/color-models-and-color-spaces/index.html (accessed Dec. 01, 2024).

[7] P. D. Team, "Gaussian Adaptive Threshold - PlantCV," *Readthedocs.io*, 2016. https://plantcv.readthedocs.io/en/v3.0.2/gaussian_threshold/ (accessed Dec. 01, 2024).

[8] W. Weng, Y. Zhang, and Z. Xiong, "Event-based Video Reconstruction Using Transformer," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, doi: https://doi.org/10.1109/iccv48922.2021.00256.

[9] Q. Liang, X. Zheng, K. Huang, Y. Zhang, J. Chen, and Y. Tian, "Event-Diffusion: Event-Based Image Reconstruction and Restoration with Diffusion Models," *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3837–3846, Oct. 2023, doi: https://doi.org/10.1145/3581783.3612462.

Appendix

```
# DSIGPRO EQ1 GROUP 3 - FINAL PROJECT
# BOQUER, CUA, OCAMPO, SORIANO, TABIOLO
# SUBMITTED TO: EDWIN SYBINGCO

import cv2
import numpy as np
from matplotlib import pyplot as plt

# Functions created for organization and
easy modifying for other methods
# Functions to enhance contrast
def enhance_contrast_hsv(image):
    # Convert the image to HSV color space
    hsv_image = cv2.cvtColor(image,
cv2.COLOR_BGR2HSV)

    # Extract the V (Value) channel for
contrast enhancement <---------------------
Based from RRL
    h, s, v = cv2.split(hsv_image)

    # Apply Histogram Equalization on the V
channel (enhance contrast)
    v_eq = cv2.equalizeHist(v)

    # Merge the enhanced V channel back
with the original H and S channels
    hsv_enhanced = cv2.merge([h, s, v_eq])

    return hsv_enhanced

def enhance_contrast_grayscale(image):
    # Convert the image to grayscale
    gray = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY)

    # Apply Histogram Equalization to
enhance contrast
    gray2enhance = cv2.equalizeHist(gray)

    return gray2enhance

# Function to reduce noise using Bilateral
Filtering  <-----------------------
Bilateral Filter for smoother noise
reduction while preserving edges
def reduce_noise(image):
    denoised_image =
cv2.bilateralFilter(image, d=9,
sigmaColor=75, sigmaSpace=75)
    return denoised_image

# Function to apply Gaussian Thresholding
def gaussian_thresholding(image,
max_value=255):
    gray = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY)

    # Apply Gaussian Thresholding (adaptive
thresholding using Gaussian filter)
    thresholded =
cv2.adaptiveThreshold(gray, max_value,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY, 11, 2)

    return thresholded

# Main function to process the image
def process_image(image_path):
    # Load the image
    image = cv2.imread(image_path)

    # Step 1: Enhance Contrast in HSV and
Grayscale
    hsv_enhanced =
enhance_contrast_hsv(image)
    gray_enhanced =
enhance_contrast_grayscale(image)

    # Step 2: Apply Noise Reduction

    # Convert HSV back to BGR for
consistent processing
    enhanced_image =
cv2.cvtColor(hsv_enhanced,
cv2.COLOR_HSV2BGR)
    denoised_image =
```

```
reduce_noise(enhanced_image)

    # Reduce the noise of the grayscale
image
    gray2enhance =
reduce_noise(gray_enhanced)

    # Step 3: Apply Gaussian Thresholding
to the denoised image
    thresholded_image =
gaussian_thresholding(denoised_image)

    # Show results
    plt.figure(figsize=(12, 6))

    plt.subplot(1, 4, 1)
    plt.title("Original Image", fontsize =
8)
    plt.imshow(cv2.cvtColor(image,
cv2.COLOR_BGR2RGB))
    plt.axis("off")

    plt.subplot(1, 4, 2)
    plt.title("Gaussian Thresholded Image",
fontsize = 8)
    plt.imshow(thresholded_image,
cmap='gray')
    plt.axis("off")

    plt.subplot(1, 4, 3)
    plt.title("Denoised Enhanced Grayscale
Image", fontsize = 8)
    plt.imshow(gray2enhance, cmap='gray')
    plt.axis("off")

    plt.subplot(1, 4, 4)
    plt.title("Denoised Enhanced Color
Restored Image", fontsize = 8)
    plt.imshow(cv2.cvtColor(denoised_image,
cv2.COLOR_BGR2RGB))
    plt.axis("off")

    plt.show()

    return thresholded_image

image_path = 'TEETH.jpg'  #
<---------------------------- Change Image
File Here!
processed_image = process_image(image_path)
```