International Journal of Science and Engineering Applications
Volume 14-Issue 07, 26 – 29, 2025, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1407.1005

Design and Implementation of FPGA-based Low-Light Image Enhancement Algorithm

Liu AHao
School of Electronic Information and Electrical Engineering
Yangtze University
Jingzhou, China

Abstract: In order to solve the problems of low brightness and loss of details inimages under low illumination environment, an improved low illumination image enhancement algorithm based on FPGA is designed and implemented. Based on the LIME algorithm, this study improves the original algorithm by introducing bilateral filter and YCrCb color space optimization method, which enhances the brightness and naturalness of the image while effectively retaining the edge information of the image. The RTL design of the improved algorithm is completed by Verilog language and hardware testing is carried out on FPGA development board. The results show that the improved algorithm performs well in brightness enhancement, contrast enhancement and image naturalness preservation, and is suitable for scenes with high real-time processing requirements.

Keywords: low-light image enhancement; FPGA; LIME algorithm; bilateral filtering;YCrCb color space

1. Introduction

With the development of informationtechnology, image processing in low-light environment has importantapp lication value in the fields of security,medical treatment and unmanned driving.However, low illumination images often fail to meet the practical needs due to the problems of low brightness and poor contrast [1]. The traditional histogram equalization and Retinex theory related algorithms improve the brightness of the image, but there are problems such as color distortion or over-enhancement [2]. To address this problem, Guo et al.proposed the LIME algorithm, which strikes a good balance between computational efficiency and enhancement [3].As the complexity of low illumination image enhancement algorithms increases, FPGA-based hardware implementations have become an important option due totheir efficient parallel computingcapabilities [4]. In this paper, we propose an improved LIME algorithm that improves the image quality through filter optimization and color space adjustment,and complete the hardware implementation on FPGA.

2. Improved algorithmdesign

2.1 Bilateral filter optimization

Traditional Gaussian filtering is prone to blurring image edges while reducing noise. In this paper, a bilateral filter is used to achieve noise reduction while retaining the edge information by considering both spatial proximity and pixel gray scale similarity. Its mathematical expression is as follows:

$$D(x,y) = \frac{\sum I(i,j) \times \omega_s(i,j) \times \omega_r(i,j)}{\sum \omega_s(i,j) \times \omega_r(i,j)}, \quad (i,j) \in \Omega(x,y)$$

$$\omega_s(i,j) = \frac{1}{\sqrt{2\pi}\sigma_s} e^{-\frac{(i-x)^2+(j-y)^2}{2\sigma_s^2}}, \quad (i,j) \in \Omega(x,y)$$

$$\omega_r(i,j) = \frac{1}{\sqrt{2\pi}\sigma_r} e^{-\frac{[I(i,j)-I(x,y)]^2}{2\sigma_r^2}}, \quad (i,j) \in \Omega(x,y)$$

Where,D (x,y) is the filtered image,I(x,y) is the input image, Ω (x,y) is a neighborhood space of pixels (x,y),S (x,y) is the spatial proximity factor,and ωγ(x,y) is the gray scale similarity factor.

2.2 Effectiveness testing

First of all, generate an image containing noise, as shown in Figure 1 (a) of , and compare the effect of the two filters; set the parameters of the two filters the same, and the filtering effect is shown in Figure 1 (b) and Figure 1 (c), respectively. It can be seen that both filters can filter the noise relatively well, but the edges of the Gaussian filtered image are obviously blurred, and the loss of edge information is very serious.

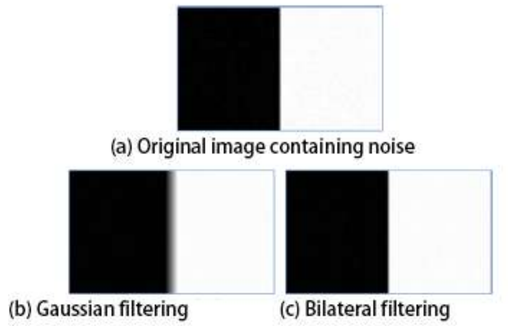


Figure. 1 Noise image

2.3 LIME algorithm for bilateral filter opti mization

A comparison between the optimized algorithm and the original algorithm is shown in Figure 2, where the rows of images are the original low-light image, the LIME algorithm, and the LIME algorithm optimized by bilateral filtering. Intuitively, there is no obvious difference between the optimized algorithm and the original algorithm, but according to the analysis in subsection 3.1.3, the use of bilateral filters will make the edge information of the image better preserved. Take img1 as an example, zoom in its local details, as shown in Figure 3. From the figure, it can be seen that the optimized LIME algorithm does not generate noise at the edge of the image, which indicates that the optimized LIME algorithm can retain the edge information of the image well.

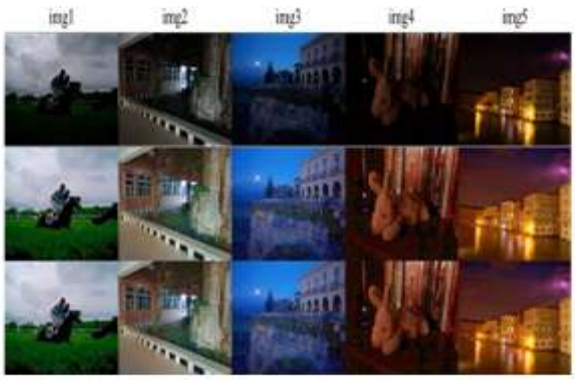


Figure.2 Comparison of Algorithm Effects

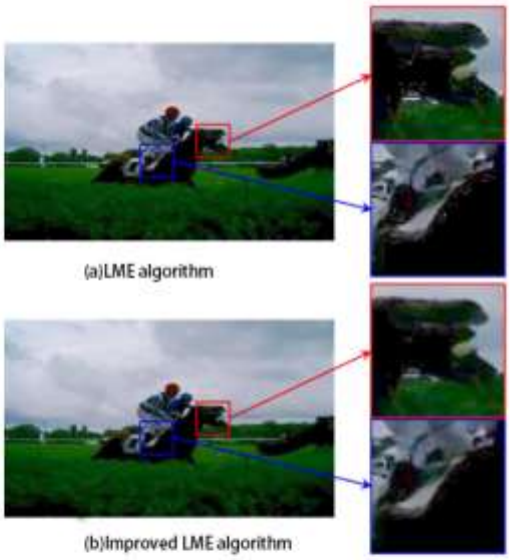


Figure.3 Local zoom in detail comparison

2.4 Color space optimization

2.4.1 YCrCb Color Space

In the traditional RGB color space, the three colors have the same importance, but according to Eq. , the three colors do not have the same degree of contribution to the luminance. The YCrCb color space separates the luminance information from the color information, with Y representing the luminance of the pixel and the color information stored in Cr and Cb. The conversion equations between the two color spaces are shown in equations.

$$\begin{cases} Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \\ Cr = (R - Y) \times 0.713 + 128 \\ Cb = (B - Y) \times 0.564 + 128 \end{cases}$$

$$\begin{cases} R = Y + 1.403 \cdot (Cr - 128) \\ G = Y - 0.714 \cdot (Cr - 128) - 0.344 \cdot (Cb - 128) \\ B = Y + 1.773 \cdot (Cb - 128) \end{cases}$$

2.4.2 LIME algorithm for YCrCb optimization

In the enhancement effect of the LIME algorithm shown in Figure 4, although there is no color distortion, when the overall color tone of the image is biased towards a certain color, the hue of the enhanced image will be shifted to a certain extent. For example, the color of the cliff in img3 is bluish, and the color of the doll in img4 is orangeish, which are both affected by the color tone of the image environment. The YCrCb color space separates the luminance information from the color information, which can effectively prevent the output image from the problem of color deviation when processing images. Design and implementation of FPGA-based low-light image enhancement algorithm The comparison of the optimization algorithm and the original algorithm is shown in Figs. 3-4, in which each row of images is the original low-light image, the optimized LIME algorithm with double-sided filtering, and the optimized LIME algorithm with YCrCb. It can be seen from the figure that the optimized algorithm slightly improves the brightness, and more importantly, the color deviation of the image is obviously reduced, indicating that the YCrCb optimized LIME algorithm can effectively prevent the output image from color deviation.

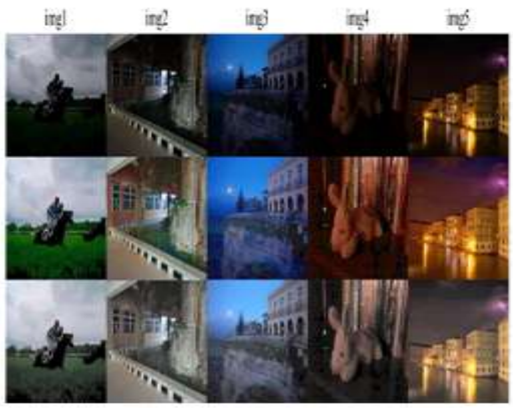


Figure.4 Color Separation Diagram

2.5 Improved LIME algorithm

The flowchart of the improved LIME algorithm is shown in Figure 5. Firstly, the input image is converted to YCrCb color space, the Y channel is used as the initial illuminance map, and after Gamma correction and bilateral filtering, the illuminance estimation value L is obtained; according to the Retinex theory, the value of the Y channel of the enhanced image is obtained by equation (2-20), and then the image is finally restored to the RGB color space for output. The effect of the algorithm is shown in the third row of Figure 5.

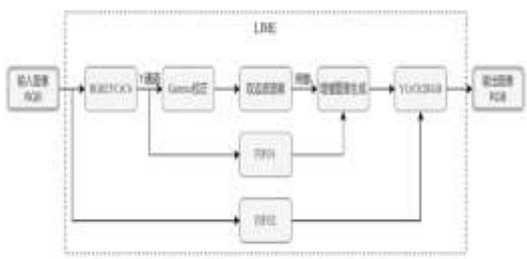


Figure.5 LME algorithm

The improved LIME algorithm is also evaluated objectively using the five evaluation indexes in Section 2.2, as shown in Tables 1 through 4. As can be seen from the table, the standard deviation of luminance, entropy of image information, and sharpness index of LIME algorithm before and after the

International Journal of Science and Engineering Applications
Volume 14-Issue 07, 26 – 29, 2025, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1407.1005

improvement do not have much change, while the mean value of luminance of the improved LIME algorithm has been increased, which indicates that the enhancement ability of the improved algorithm for low illuminance has been improved. At the same time, the luminance order error of the improved LIME algorithm is significantly reduced, which indicates that the improved LIME algorithm is more capable of maintaining the naturalness of the original image, this is because the algorithm uses a bilateral filter, the edge information of the enhanced image4 is not damaged, and it is relatively consistent with the edge information of the original low-light image.RGB color space

	AVERAGE BRIGHTNESS	BRIGHTNESS STANDARD DEVIATION	IMAGE INFORMATIO N ENTROPY	Articulation Index	Brightness sequence error
LIME	112.33	78.902	4.9039	0.5830	132.22
Improve LIME	120.72	79.646	4.7830	0.6651	98.728

Table 1 Objective evaluation indicators under img1 of LIME algorithm before and after improvement

	AVERAGE BRIGHTNESS	BRIGHTNESS STANDARD DEVIATION	IMAGE INFORMATIO N ENTROPY	Articulation Index	Brightness sequence error
LIME	101.42	47.801	5.2156	0.5946	267.17
Improve LIME	107.50	45.582	5.1180	0.5595	126.17

Table 2 Objective evaluation indicators under img2 of LIME algorithm before and after improvement

	AVERAGE BRIGHTNESS	BRIGHTNESS STANDARD DEVIATION	IMAGE INFORMATIO N ENTROPY	Articulation Index	Brightness sequence error
LIME	71.988	34.612	4.8616	0.7665	243.10
Improve LIME	97.855	44.077	5.0014	0.5904	237.16

Table 3 Objective evaluation indicators under img3 of LIME algorithm before and after improvement

	AVERAGE BRIGHTNESS	BRIGHTNESS STANDARD DEVIATION	IMAGE INFORMATIO N ENTROPY	Articulation Index	Brightness sequence error
LIME	42.476	37.145	4.4430	0.8767	243.35
Improve LIME	55.608	42.907	3.9958	0.8114	240.71

Table 4 Objective evaluation indicators under img4 of LIME algorithm before and after improvement

3. Hardware Implementation

3.1 Test platform

The hardware test is based on the Zynq-7020 FPGA development board, and the system peripherals include the OV5640 camera module, DDR3 memory module and TFT-LCD display module. The real-time captured low-light video is processed by the algorithm and displayed on the LCD screen. Field Programmable Gate Array (FPGA) is a kind of programmable logic device, which is programmed by using hardware description language such as Verilog to change its internal structure in order to realize the required functions.FPGA has basic logic resources such as LUT, FF, RAM, IO, etc., and there will also be some embedded special hardcores such as DSP, Block RAM, etc. The designers will use compiler software to change the internal structure of the FPGA to realize the required functions. Designers use compiler software to synthesize the hardware code into a bitstream file, which is burned into the FPGA. FPGA configures the contents of the lookup table according to this file, so as to realize different logic functions. The advantages of FPGA's parallel computation and pipelined operation make it very suitable for real-time image processing. In this paper, we use Xilinx's Artix-7 series FPGA, model XC7A35TFGG484-2L, and the on-chip logic resources are shown in Table 5. The development board is equipped with a 256MB DDR3 SDRAM chip and peripheral interfaces such as camera, TFT-LCD, HDMI, etc., which is very suitable for realizing and verifying image processing algorithms.

LUT	FF	BRAM	DSP	IO
20800	41600	1.8Mbit	90	250

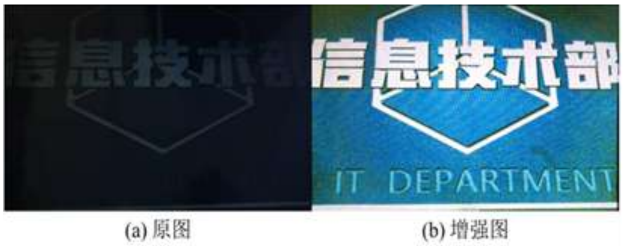
Table 5 The FPGA chip on-chip logic resources used in this article

3.2 Hardware System Board Level Testing

The optimised code is programmed into the FPGA development board, with the OV5640 camera module serving as the video image input and the TFT-LCD displaying the enhanced image. The physical test system is shown in Figure 6. The on-board buttons can be used to switch between displaying the original image and the enhanced image.

Figures .6 Physical image of the testing system

The before-and-after image comparisons are. In these figures, the original images in Figures 7 and 8 were captured under low-light conditions, and the enhanced images effectively restore the details in the images; the original image in Figure 9 was captured under normal lighting conditions, and while the enhanced image shows a slight increase in brightness, there is no overall over-enhancement. Based on these three sets of comparison images, it can be concluded that the improved LIME algorithm proposed in this paper can effectively enhance low-light images without causing issues such as over-enhancement or colour bias.



Figures . 7 Comparison diagram of board level testing

Figures . 8 Comparison diagram of board level testing

Figures . 9 Comparison diagram of board level testing

4. REFERENCES

[1] Pizer S M, et al. Adaptive histogram equalization and its variations[J]. Computer Vision, Graphics, and Image rocessing, 1987, 39(3): 355-368.

[2] Land E H. The retinex theory of color vision[J]. Scientific American, 1977, 237(6): 108-129.

[3] Guo X, et al. LIME: Low-light image enhancement via illumination map estimation[J]. IEEE Transactions on Image Processing, 2017, 26(2): 982-993.

[4] Shiau Y C, et al. A real-time image enhancement technique for fog-penetration using FPGA[J]. IEEE Transactions on Consumer Electronics, 2013, 59(3): 556-562.