

A Low-Cost Hardware Architecture for Illumination Adjustment in Real-Time Applications

Yeu-Horng Shiao, Pei-Yin Chen, *Member, IEEE*, Hung-Yu Yang, and Shang-Yuan Li

Abstract—For real-time surveillance and safety applications in intelligent transportation systems, high-speed processing for image enhancement is necessary and must be considered. In this paper, we propose a fast and efficient illumination adjustment algorithm that is suitable for low-cost very large scale integration implementation. Experimental results show that the proposed method requires the least number of operations and achieves comparable visual quality as compared with previous techniques. To further meet the requirement of real-time image/video applications, the 16-stage pipelined hardware architecture of our method is implemented as an intellectual property core. Our design yields a processing rate of about 200 MHz by using TSMC 0.13- μ m technology. Since it can process one pixel per clock cycle, for an image with a resolution of QSXGA (2560×2048), it requires about 27 ms to process one frame that is suitable for real-time applications. In some low-cost intelligent imaging systems, the processing rate can be slowed down, and our hardware core can run at very low power consumption.

Index Terms—Hardware architecture, illumination adjustment, real-time applications.

I. INTRODUCTION

NOWADAYS, there is an increasing demand for cameras and intelligent surveillance systems aiming at monitoring private and public areas. For example, the camera-based advanced driver assistance system can inform drivers of the appropriate speed or help keep the vehicle between lane markers to improve vehicle/road safety. In intelligent transportation systems (ITSs), the cameras keep track of the road and driving situations to detect the traffic flow or to record information related to vehicle crashes or accidents automatically. However, insufficient visibility due to the influence of weather or poor atmospheric light will cause the intelligent system to be inoperative. Many image preprocessing algorithms are integrated

to increase the visibility of the system, such as denoising, scaling, and illumination adjustment. In such a real-time intelligent system, high-speed processing for those enhancement algorithms is necessary and must be considered. A hardware implementation for the algorithms [1]–[4] is a better solution, which can be included in end-user camera equipment to meet the requirement. In this paper, we focus on the development of a fast and efficient illumination adjustment algorithm that is suitable for low-cost high-speed hardware implementation.

Dynamic range is the illumination ratio between the darkest and the brightest region in a scene. A captured image in surveillances shows an irrevocable loss of visual information in some places under strong background illumination or in dark environments. This is mainly because the dynamic range of natural scene is far larger than the dynamic range of image captured by the common digital devices. Such overflow will cause a blurred image particularly in the low-luminance regions, and the viewer cannot obtain enough information of the image. To overcome this annoying problem, the existing enhancement methods [5]–[18] for a single image are proposed and can be classified into two categories: spatial-domain processing [5]–[14] and compression-domain processing [15]–[18]. The spatial-domain processing can further be divided into two subgroups: the histogram-based techniques [11]–[14] and the Retinex-based techniques [5]–[10]. To consider the hardware implementation to meet the real-time requirement, the above image enhancement algorithms suffered from some problems. The histogram-based techniques are effective contrast enhancement algorithms due to their straightforward and intuitive implementation qualities. However, the histogram-based techniques need to scan the whole image twice to complete the enhancement process, i.e., one for calculating the probability density function and one for obtaining the enhanced image by using the mapping function. They need to store the whole image inside the hardware core that requires a great amount of hardware resource. Longer processing time is needed since it requires reading the whole image data twice from the memory module. The Retinex-based techniques and compression-domain processing techniques usually keep the details of the original image and generate high-quality enhanced images with low noise. However, the algorithms with high time complexity are those that will make the hardware cost high and increase the processing time.

In this paper, we propose a low-complexity illumination adjustment algorithm based on the Retinex theory. The fast illumination estimation by using the concept of bidimensional empirical mode decomposition (FIEEMD) is presented. Then, a modified gamma correction is employed to adjust the illumination component to obtain a more pleasing image. The

Manuscript received July 1, 2013; revised November 8, 2013, March 15, 2014, and July 16, 2014; accepted July 28, 2014. Date of publication September 16, 2014; date of current version March 27, 2015. This work was supported in part by the National Science Council of Taiwan under Grant NSC-101-2221-E-006-151-MY3, by the Ministry of Economic Affairs (MOEA) of Taiwan under Grant MOEA 102-EC-17-A-05-S1-192, and by the Headquarters of University Advancement at National Cheng Kung University, which is sponsored by the Ministry of Education, Taiwan. The Associate Editor for this paper was P. Grisleri.

Y.-H. Shiao is with the Department of Electrical Engineering, National Yunlin University of Science and Technology, Yunlin 64002, Taiwan (e-mail: shiauyh@yuntech.edu.tw).

P.-Y. Chen, H.-Y. Yang, and S.-Y. Li are with the Digital IC Design Laboratory, Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 70101, Taiwan (e-mail: pychen@mail.ncku.edu.tw).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TITS.2014.2347701

experimental results demonstrate that our method achieves far shorter execution time and comparable visual quality as compared with other enhancement methods [10], [18]. Moreover, we also present a 16-stage pipelined hardware architecture implemented as an intellectual property (IP) core that can be integrated with other circuits in the system-on-chip (SoC). By using hardware sharing and parallel processing techniques, the proposed IP core can efficiently operate to meet the requirement of real-time applications with lower cost. In some low-cost imaging systems, the processing rate can be slowed down, and our hardware core can run at very low power consumption.

The rest of this paper is organized as follows. In Section II, the background information and the main idea are briefly introduced. Then, the proposed illumination adjustment method is presented in Section III. Section IV describes the hardware architecture of the proposed method in detail. Section V illustrates the simulation results and hardware implementation. Finally, the conclusions are provided in Section VI.

II. BACKGROUND AND MAIN IDEA

In 1971, Land and Mccann [19] developed the Retinex theory as a simple and effective model of visual perception. The Retinex theory assumed that the intensity of visible light reaching a camera and the observer's eyes depends on the product of two components: illumination, which represents the ray of light source from the sun or sky in the scene, and reflectance, which represents the illumination variation of the object surfaces. According to the assumption, an observed image can be expressed as

$$I(x, y) = L(x, y) \times R(x, y) \quad (1)$$

where $I(x, y)$ is the pixel of an observed image located in the (x, y) coordinate, $L(x, y)$ is the illumination of the scene, and $R(x, y)$ is the reflectance of the objects.

Based on the Retinex theory, the image enhancement algorithm first estimates the illumination denoted as $\hat{L}(x, y)$ by some signal processing technique, and the estimated reflectance denoted as $\hat{R}(x, y)$ can be determined by (1). Then, the estimated illumination $\hat{L}(x, y)$ can be adjusted by a nonlinear operator $G(\cdot)$ to enhance the visual quality of the image. Finally, the adjusted luminance data can be obtained as

$$\hat{I}(x, y) = \hat{R}(x, y) \times G\left(\hat{L}(x, y)\right). \quad (2)$$

The image \hat{I} is an enhanced version of the input observed image.

The main challenge of the Retinex method is how to estimate the illumination component $\hat{L}(x, y)$ properly. Recently, a powerful signal analyzing technique called empirical mode decomposition (EMD) is proposed [20]. By extracting oscillation information of the data, EMD decomposes a signal into several frequency components, which are called characteristic intrinsic mode functions (IMFs), and a last function that cannot be decomposed into any IMFs, which is called residue. The process of EMD required several iterations to obtain each IMF. To analyze 2-D data such as a 2-D image and to overcome

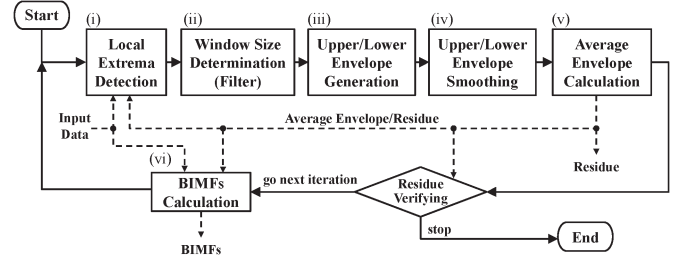


Fig. 1. Flow diagram of the FABEMD.

the exhaustive computation, Bhuiyan *et al.* [21] proposed the fast and adaptive bidimensional empirical mode decomposition (FABEMD). The process of FABEMD for extracting bidimensional IMFs (BIMFs) is shown as in Fig. 1, where the solid arrow means the step flow of the algorithm, and the dashed arrow represents the important data flow for the step block. For more detailed description, the reader can refer to [21].

In the image application, some existing research [22]–[27] found that each BIMF and the residue contain different information of an image such as color variety, edge, noise, and illumination. The first BIMF contains the majority of color variation, and the residue contains the illumination trend. With this assumption, we treat the residue as the illumination plane $\hat{L}(x, y)$ in the Retinex theory for adjusting the uneven illumination. According to the FABEMD process, the residue is included in the average envelope of each i th decomposition. To reduce the computational load, we only calculate the first average envelope as the illumination plane $\hat{L}(x, y)$. Then, an adaptive gamma correction is developed to adjust $\hat{L}(x, y)$ to enhance the illumination contrast without oversaturation.

III. PROPOSED ILLUMINATION ADJUSTMENT ALGORITHM

The flow diagram of the proposed illumination adjustment algorithm is shown as in Fig. 2. First, the color image is mapped from the RGB domain into the HSV color space. The HSV domain is an ideal way for color descriptions that are natural and intuitive to humans. Second, the first average envelope is extracted as the illumination plane L by using FIEEMD only from the intensity layer V of the HSV domain. Third, a novel gamma correction is employed to adjust the illumination component to improve the visual quality of the details in low- and high-luminance regions. Finally, the enhanced image represented in the HSV color space is converted into the RGB domain. The details of FIEEMD and modified gamma correction are described as follows.

A. Fast Illumination Estimation By Using the Concept of Bidimensional Empirical Mode Decomposition (FIEEMD)

The three processing steps of the FIEEMD are described as follows.

- (i)' Find the local maximum and local median points of the intensity layer V of the input image I . Finding the local maximum and minimum points from the original image is an important step in the process of FABEMD. In this illumination adjustment application, these extrema points

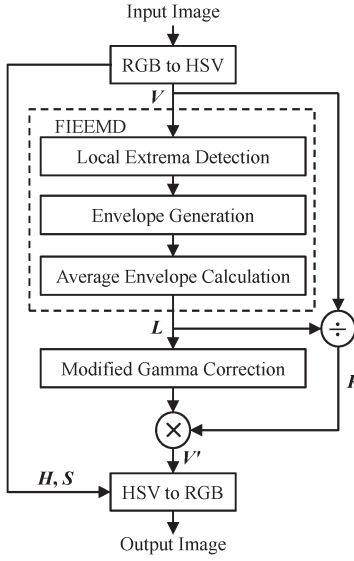


Fig. 2. Flow diagram of the proposed illumination adjustment algorithm.

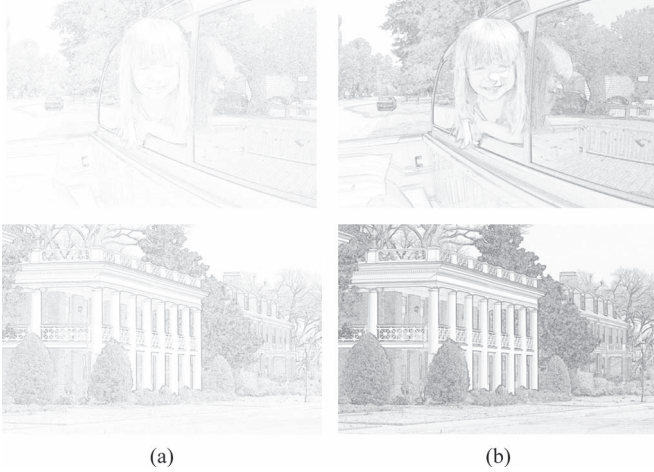


Fig. 3. Reflectance obtained from different illumination. (a) FABEMD in the first decomposition. (b) Our FIEEMD.

not only determine the trend of illumination L but also affect the reflectance R . According to the Retinex theory, the reflectance can be calculated by dividing the intensity layer V by the illumination plane L . Generally, the illumination should be quite uniform on large portions of the image. Thus, the image contrast is kept in the reflectance. The more flat plane the illumination is estimated, the more contrast information can be kept in the reflectance, and the higher contrast image can be reconstructed. When the first average envelope is extracted from the FABEMD as the illumination L , the estimated reflectance loses the contrast of image. In this paper, we change Step (i) of FABEMD into a new local extrema detection [Step (i)'], which uses the local maximum and local median filter to determine the trend of illumination. The local median filter instead of the local minimum used here is to get a smoothed illumination and leads to a more contrasting reflectance, as shown in Fig. 3.

- (ii)' Estimate the upper and lower envelopes by using local maximum and local median points, respectively. After the

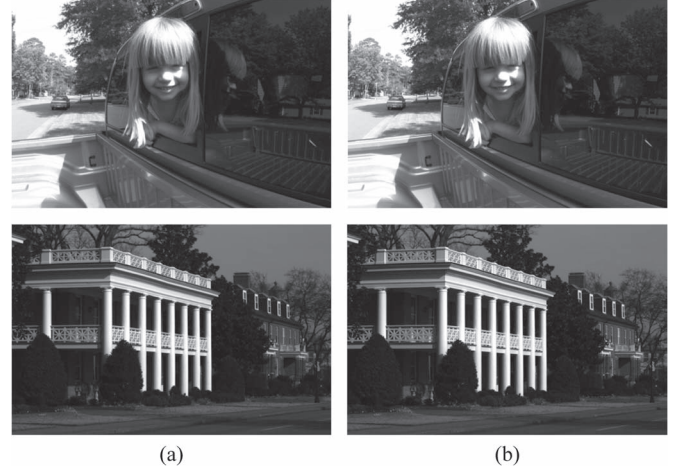


Fig. 4. Illumination plane generated by two methods. (a) FABEMD in the first decomposition. (b) Our FIEEMD.

local maximum and local median points are obtained, the next step is to generate the upper and lower envelopes. In the FABEMD, the upper and lower envelopes are created by three major procedures: window size determination [Step (ii)], envelope generation [Step (iii)], and smoothing operation [Step (iv)]. These three steps consume most execution time of the average envelope decomposition. In the simulation, we found that the larger window size we choose, the more computational time we need to perform the maximum and medium filters. Furthermore, the larger window size means that we need more line buffers (higher hardware cost) to store those previous image rows for hardware implementation. Hence, to minimize the time complexity and save computational resources, we fix the window size of maximum (median)-order statistics filters to a square region Z_{xy} of size 3×3 for envelope generation. Once the window size is fixed, the determination of window size in Step (ii) is ignored in our FIEEMD. Then, the upper and lower envelopes in our Step (ii)', i.e., E'_U and E'_L , are obtained as

$$E'_U(x, y) = l_{\max}(x, y), \quad E'_L(x, y) = l_{\text{med}}(x, y) \quad (3)$$

where $l_{\max}(x, y)$ and $l_{\text{med}}(x, y)$ are the local maximum and local median point in the region defined by Z_{xy} , respectively. Obviously, in (3), $l_{\max}(x, y)$ and $l_{\text{med}}(x, y)$ have been found in Step (i)'. We skip Step (iv) of the FABEMD to save computational resources. Thus, the upper and lower envelopes can be obtained without any calculation in FIEEMD.

- (iii)' Calculate the first average envelope E'_A by averaging the upper and lower envelopes obtained from Step (ii)'. The obtained E'_A is just the estimated illumination plane L and will be further processed for illumination adjustment.

Fig. 4 shows the estimated illumination L , generated by the FABEMD in the first decomposition and our proposed method. Although the processing is simplified, it can be seen that the illumination L in Fig. 4(b) is very close to that in Fig. 4(a). Assume that the image is of size $M \times N$. Table I shows the comparison of operation for FABEMD executed in the first

TABLE I
COMPARISON OF OPERATION COMPLEXITY FOR FABEMD EXECUTED IN
THE FIRST DECOMPOSITION AND OUR FIEEMD

FABEMD	FIEEMD
Step (i) : $(3 \times 3 \text{ window}) \times M \times N$ compare operations	Step (i)' : $(3 \times 3 \text{ window}) \times M \times N$ compare operations
Step (ii) : $(\text{number of local extrema})^2$ Euclidean distance calculations + $(\text{number of local extrema})$ compare operations	Step (ii)' : no operation
Step (iii) : $(Z_{xy} \text{ window}) \times M \times N$ compare operations	
Step (iv) : $(Z_{xy} \text{ window}) \times M \times N \times 2$ smooth operations	
Step (v) : $M \times N$ average operations	Step (iii)' : $M \times N$ average operations

decomposition and our FIEEMD, respectively. Because the window size used in Step (i)' and Step (ii)' of our method is identical and the smooth operation is skipped in Step (ii)', the computational complexity of the envelope generation for our FIEEMD can be greatly reduced from $O(M \times N)$ to $O(1)$.

B. Modified Gamma Correction

Once the illumination L is estimated, the corresponding reflectance plane R can be calculated by dividing the intensity layer V by the illumination plane L according to the Retinex theory. To adjust the illumination, the modified gamma correction is proposed and represented as

$$V'(x, y) = R(x, y) \times \left(\frac{L(x, y)}{C} \right)^r \times K_{(x, y)} \quad (4)$$

where $V'(x, y)$ is the enhanced intensity value, r is the parameter for the gamma function, and $K_{(x, y)}$ is the normalization value for each $V(x, y)$. The parameter C is the normalization constant, which is set as 255 because each pixel is an 8-bit integer data in the illumination plane L in our design. The parameter r is chosen as 0.4 to increase the variation of the gamma function in the low-luminance regions. To enhance the illumination contrast and avoid causing oversaturation simultaneously, the normalization value $K_{(x, y)}$ for each point (x, y) of L whose size is $M \times N$ is regulated as

when $y = 1$: (Let $K_{(1,1)} = 200$)

$$K_{(x,1)} = \begin{cases} K_{(x-1,1)} - 1, & \text{if } V(x, 1) \geq 128 \\ K_{(x-1,1)} + 1, & \text{if } V(x, 1) < 128 \end{cases}$$

when $y > 1$:

$$K_{(x,y)} = \begin{cases} K_{(x,y-1)} - 1, & \text{if } V(x, y) \geq 128 \\ K_{(x,y-1)} + 1, & \text{if } V(x, y) < 128. \end{cases} \quad (5)$$

If $V(x, y)$ is greater than 128, we classify point (x, y) as a bright point and decrease $K_{(x, y)}$ to make the bright region not too bright. Otherwise, point (x, y) is determined as a dark point, and $K_{(x, y)}$ is increased to make the dark region not too dark. The flowchart for the modified gamma correction is shown in Fig. 5.

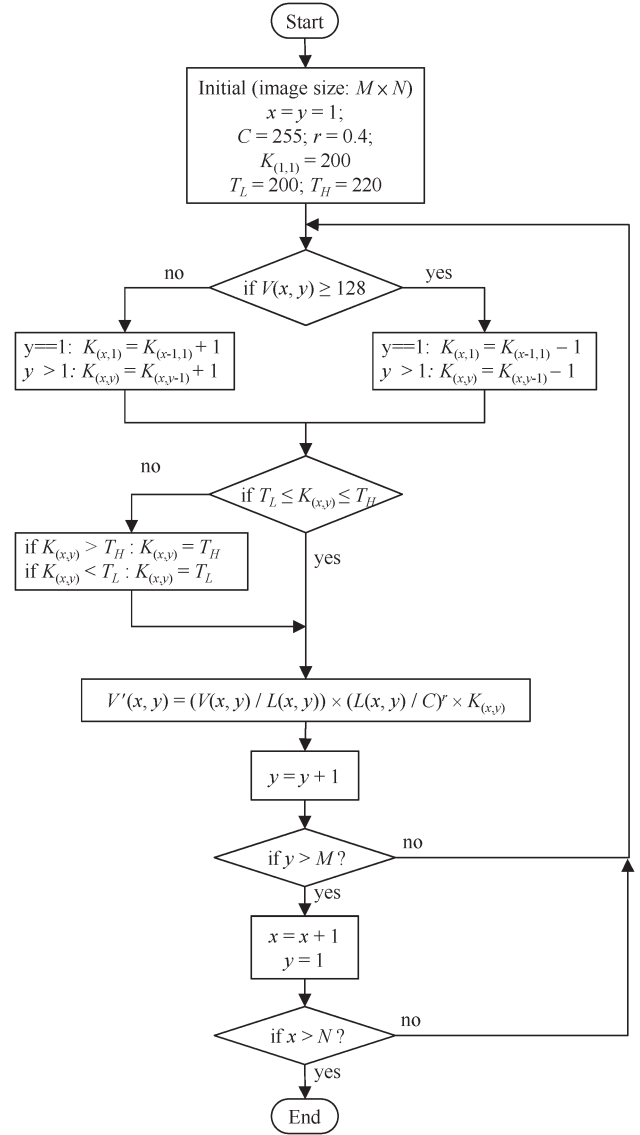


Fig. 5. Flowchart of the modified gamma correction.

To avoid luminance oversaturation in the brighter pixels, the adjustment range of $K_{(x, y)}$ is limited between T_L and T_H , which are determined by the curve of the gamma correction function in our experiments. Fig. 6 shows the experiment results by using different r and K values in (4). Note that $K_{(x, y)}$ is substituted with a fixed K value in (4) for the experiments because we want to see how different K values affect the luminance. The original image, which is on the top row in Fig. 6(a)–(d), was created by combining several blocks with different gray levels. The leftmost block was the darkest block with gray level 0. Then, the gray level increased 20° in the next block and so on. Finally, the rightmost block reached gray level 255. The other rows below the first row (original image) were created by (4) with different K values. It can be seen that when K is set as 200 or 220, we can avoid oversaturation in high-luminance regions efficiently. Thus, the range of $K_{(x, y)}$ is limited from 200 to 220. For different r values, in Fig. 6(a) and (b), the contrast is lost particularly in the brightness region. Fig. 6(d) shows the best contrast, but it looks too dark. Thus, we chose $r = 0.4$ for our implementation.

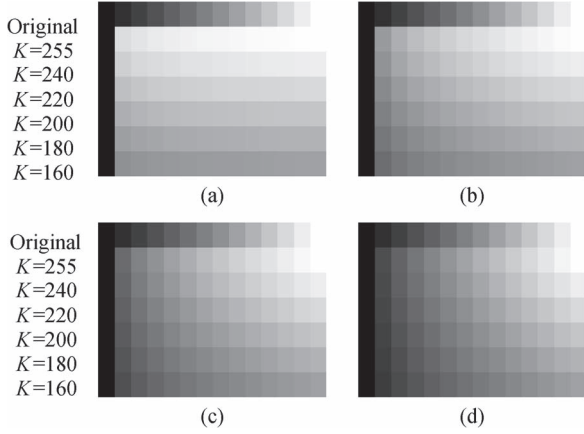


Fig. 6. Experiment results by using different r and K values in the gamma function. (a) $r = 0.05$. (b) $r = 0.2$. (c) $r = 0.4$. (d) $r = 0.6$.

TABLE II
INPUT VALUES AND OUTPUT EQUATIONS OF OUR
ILLUMINATION ADJUSTMENT PROCEDURE

	Inputs	Outputs
Step 1: RGB to HSV	$R(x, y)$ $G(x, y)$ $B(x, y)$	$H(x, y) = \begin{cases} (6 + \frac{G(x, y) - B(x, y)}{\text{MAX} - \text{MIN}}) \times 60, & \text{if } R(x, y) = \text{MAX} \\ (2 + \frac{B(x, y) - R(x, y)}{\text{MAX} - \text{MIN}}) \times 60, & \text{if } G(x, y) = \text{MAX} \\ (4 + \frac{R(x, y) - G(x, y)}{\text{MAX} - \text{MIN}}) \times 60, & \text{if } B(x, y) = \text{MAX} \end{cases}$ $S(x, y) = \frac{\text{MAX} - \text{MIN}}{\text{MAX}}, \quad V(x, y) = \text{MAX}$
Step 2: FIEEMD	$V(x, y)$	$L(x, y) = \text{FIEEMD}(V(x, y))$
Step 3: Modified Gamma Correction	$V(x, y)$ $L(x, y)$	$V'(x, y) = \frac{V(x, y)}{L(x, y)} \times \left(\frac{L(x, y)}{C} \right)^r \times K_{(x, y)}$
Step 4: HSV to RGB	$V'(x, y)$ $S(x, y)$ $H(x, y)$	$(R'(x, y), G'(x, y), B'(x, y)) =$ $\begin{cases} (V'(x, y), m(x, y), p(x, y)) & \text{if } hi = 0 \\ (n(x, y), V'(x, y), p(x, y)) & \text{if } hi = 1 \\ (p(x, y), V'(x, y), m(x, y)) & \text{if } hi = 2 \\ (p(x, y), n(x, y), V'(x, y)) & \text{if } hi = 3 \\ (m(x, y), p(x, y), V'(x, y)) & \text{if } hi = 4 \\ (V'(x, y), p(x, y), n(x, y)) & \text{if } hi = 5 \end{cases}$ <p>where</p> $hi = \left\lfloor \frac{H(x, y)}{60} \right\rfloor \bmod 6, \quad f(x, y) = \frac{H(x, y)}{60} - hi$ $p(x, y) = V'(x, y) \times (1 - S(x, y))$ $n(x, y) = V'(x, y) \times (1 - f(x, y) \times S(x, y))$ $m(x, y) = V'(x, y) \times (1 - (1 - f(x, y)) \times S(x, y))$

IV. PROPOSED HARDWARE ARCHITECTURE

The input values and output equations for our illumination adjustment procedure are summarized in Table II, where $R(x, y)$, $G(x, y)$, $B(x, y)$, and normalization coefficient $K_{(x, y)}$ are given inputs, and the final output results are $R'(x, y)$, $G'(x, y)$, and $B'(x, y)$. The MAX and MIN in Step 1 mean the maximum and minimum pixel values among the R , G , and B plane in the (x, y) coordinate, respectively. Fig. 7 shows the state flowchart of our four-step illumination adjustment procedure. We assume that no serial execution of two or more operations in each state is allowed. The MAX, MED, and MIN

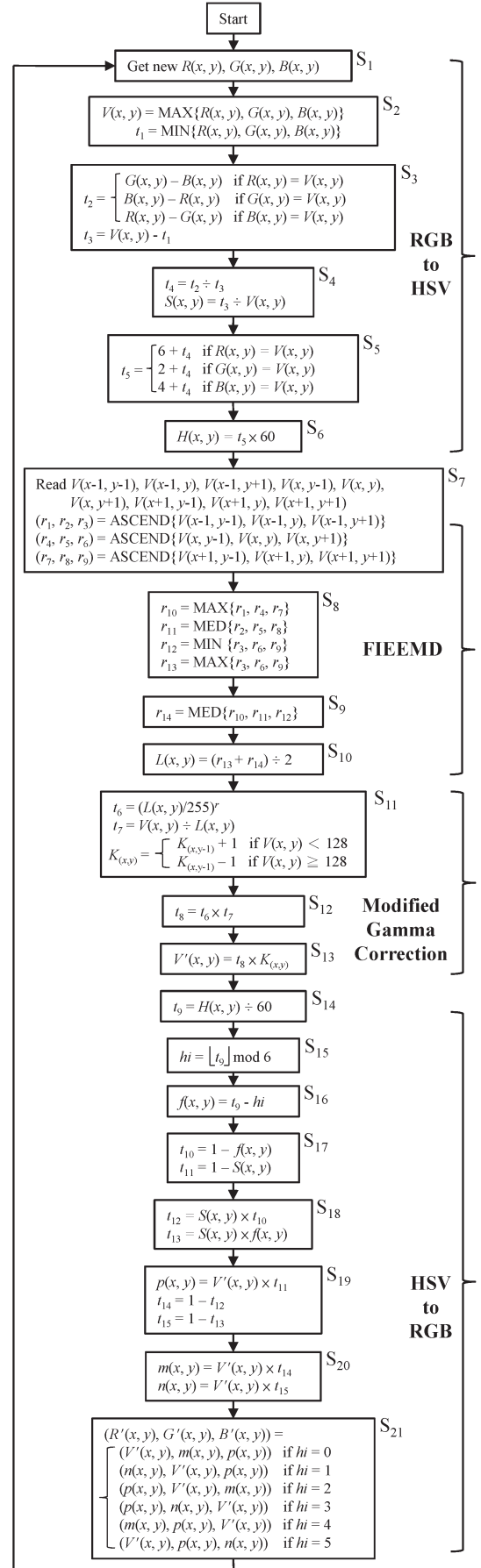


Fig. 7. State flowchart of our four-step illumination adjustment procedure.

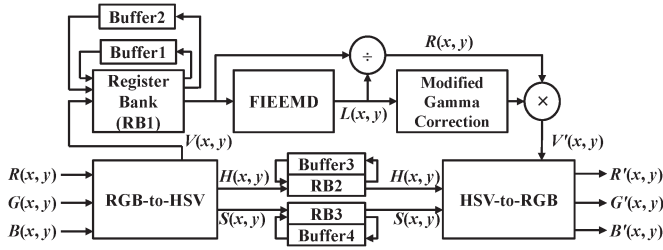


Fig. 8. Overall, block diagram of the proposed hardware architecture.

are the operations to find the maximum, median, and minimum values for three input elements, respectively. The ASCEND is the operation to sort three input elements in ascending order. The first step, i.e., *RGB-to-HSV*, is processed from S_1 to S_6 , and $V(x, y)$, $S(x, y)$, and $H(x, y)$ can be obtained in S_2 , S_4 , and S_6 , respectively. The second step, i.e., FIEEMD, is processed from S_7 to S_{10} , where the implementation of the median- and maximum-order statistics filters ($S_7 \sim S_9$) is based on the efficient sorting method presented in [28]. The median and maximum values are determined in states $S_9(r_{14})$ and $S_8(r_{13})$, respectively, and $L(x, y)$ is generated in S_{10} . The third step, i.e., modified gamma correction, is processed from S_{11} to S_{13} , and $V'(x, y)$ is obtained in S_{13} . The final step, i.e., *HSV-to-RGB*, is processed from S_{14} to S_{21} , and $R'(x, y)$, $G'(x, y)$, and $B'(x, y)$ can be determined in S_{21} . To achieve the goal of low-cost and high-performance hardware implementation, hardware sharing and parallel processing are considered in our design. Moreover, the pipelined architecture is proposed to get better performance of our hardware core. The overall block diagram of our proposed architecture is shown in Fig. 8. The architecture is divided into five units: the *RGB-to-HSV* unit, the buffer and register bank unit, the FIEEMD processing unit, the modified gamma correction unit, and the *HSV-to-RGB* unit. The details of design are described in the following sections.

A. Operation Reduction and Hardware Sharing

As shown in Fig. 7, since $H(x, y)$ produced in S_6 is kept without changing during the FIEEMD step ($S_7 \sim S_{10}$) and modified gamma correction step ($S_{11} \sim S_{13}$), the two variables t_5 and t_9 must be the same. Thus, the equations in S_6 and S_{14} are unnecessary and can be removed. S_{15} and S_{16} are changed to S'_{15} and S'_{16} , as shown in Fig. 9. Furthermore, we consider the possible reduction of the *HSV-to-RGB* step to reduce hardware resource by using a sharing technique. Since the equation of $m(x, y)$ is similar to $n(x, y)$ from state S_{16} to state S_{20} , and $m(x, y)$ and $n(x, y)$ are exclusively used according to the value of hi in state S_{21} , states S_{18} , S_{19} , and S_{20} are modified as S'_{18} , S'_{19} , and S'_{20} , as shown in Fig. 10. Obviously, the two multiplications in S_{18} and S_{20} can be reduced to one, and one subtraction can be removed from S_{19} .

B. Parallel Processing Technique

Observing the state flowchart in Fig. 7, $V(x, y)$ is already produced in state S_2 , but the FIEEMD and modified gamma correction steps are operated from S_7 to S_{13} . Thus, we rearrange the four-step illumination adjustment procedure by a parallel

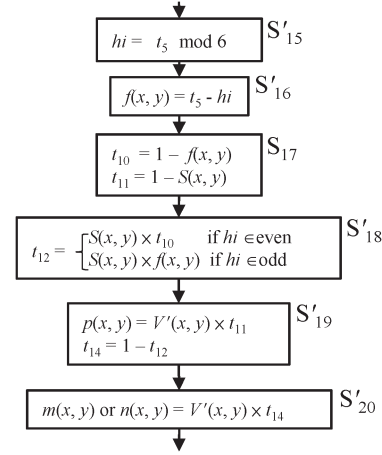


Fig. 9. State flowchart of the modified procedure in our design.

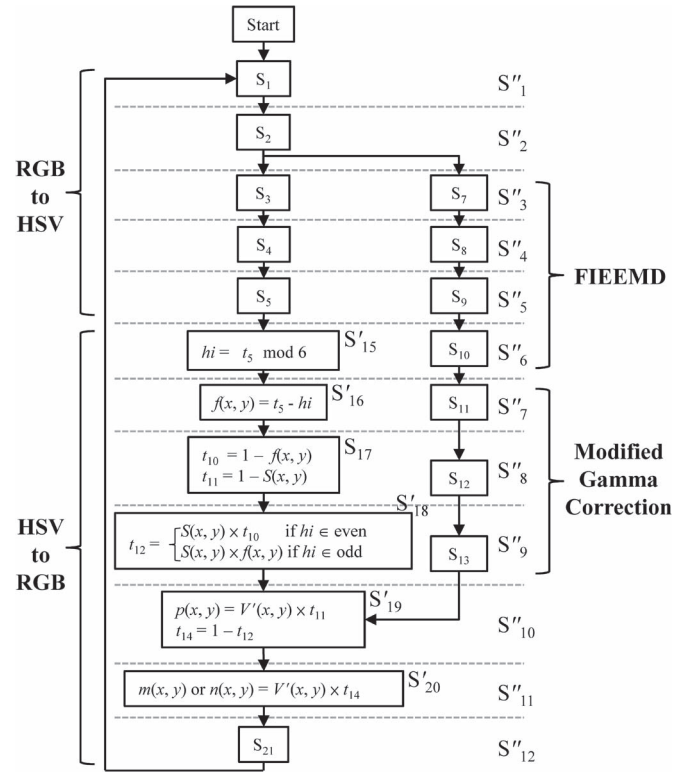


Fig. 10. State flowchart of the parallel four-step illumination adjustment procedure.

processing technique as drawn in Fig. 10. In this arrangement, the FIEEMD and modified gamma correction steps can be simultaneously executed with the *RGB-to-HSV* and *HSV-to-RGB* steps to increase the performance of our design. Obviously, the total number of execution states is reduced from 21 to 12. The reduction is almost half the original execution time.

C. Pipeline Hardware Architecture

By employing pipeline scheduling to the optimized state flowchart (see Fig. 10), we develop a 16-stage pipelined hardware architecture as shown in Fig. 11. In Stage 2, we realize a module with two line buffers and a register bank to produce nine values of a 3×3 window concurrently, as shown in Fig. 12(a).

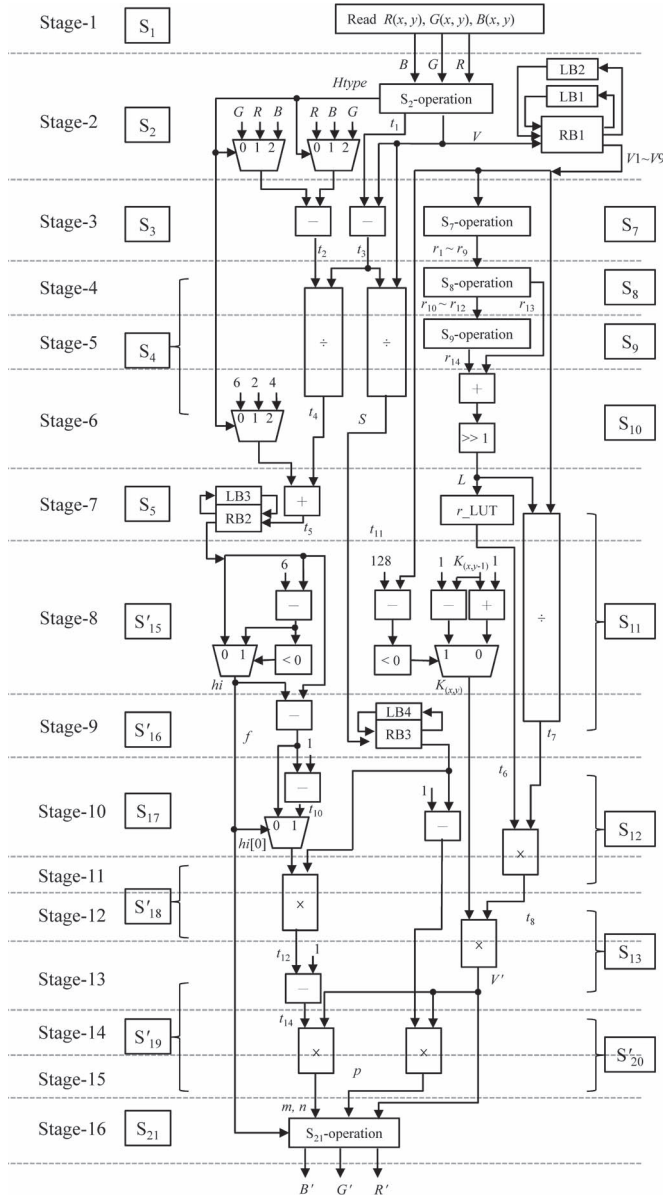


Fig. 11. 16-stage pipelined architecture of our design.

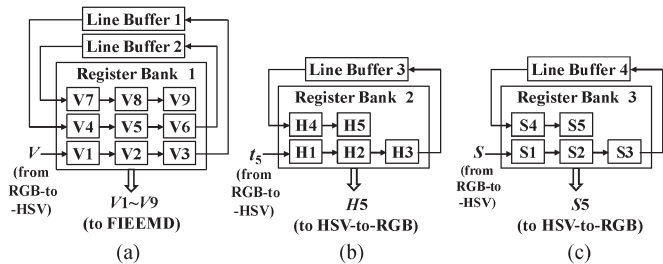


Fig. 12. Architecture of line buffers and the register banks: (a) for V, (b) for H, and (c) for S storage.

Two line buffers store two previous rows of V component. Each of them is of size 2000×8 bits, which can be applied to process an image with 1312×2000 pixels. The line buffer outputs one pixel to the register bank and replaces the output pixel with a new pixel from the register bank at every cycle. The register bank is designed with a shift register style that saves

the nine pixels of a 3×3 window and can output these pixels concurrently at every cycle. Then, the nine pixels are used in Stage 3 for the FIEEMD. Similarly, in Stage 7 and Stage 9, the two other modules are applied to store previous H and S components, as shown in Fig. 12(b) and (c). The Htype is the control signal of the multiplexer in Stage 2 and Stage 6 to complete the transformation of $H(x, y)$. The function of Htype is described as follows:

$$Htype = \begin{cases} 0, & \text{if } R \text{ is the maximum among } R, G, \text{ and } B \\ 1, & \text{if } G \text{ is the maximum among } R, G, \text{ and } B \\ 2, & \text{if } B \text{ is the maximum among } R, G, \text{ and } B. \end{cases} \quad (6)$$

The design of S7-operation, S8-operation, and S9-operation is based on a three-stage pipelined architecture [24]. The r_LUT is a lookup table component with 8-bit input ($L(x, y)$) that can generate all possible results of $(L(x, y)/255)^r$ in $S_{11}(t_6)$. For the modulo operation in S'_{15} , since the result of $S_5(t_5)$ is between 2 and 8, the equation of state S'_{15} can be changed as follows:

$$hi = \begin{cases} t_5, & \text{if } (t_5 - 6) < 0 \\ t_5 - 6, & \text{if } (t_5 - 6) \geq 0. \end{cases} \quad (7)$$

Thus, the modulo operation is replaced by a subtract operator and multiplexer to cost down the hardware resource and speed up the execution time in Stage 8. The symbol $hi[0]$ is denoted as the least significant bit of variable hi to show whether variable hi is even or odd. Note that the three-stage pipelined division and two-stage pipelined multiplication are automatically synthesized by using the SYNOPSIS Design Vision.

For an image of size $M \times N$ pixels, when the start signal is enabled, the hardware core will output the first pixel after $(2M + 18)$ clock cycles. Then, it can process one pixel per clock cycle. Totally, it would take $\{M \times (N + 2) + 18\}$ cycles to process an image.

V. IMPLEMENTATION RESULTS AND COMPARISONS

To evaluate the performance of the illumination adjustment algorithms, we compared our method with three previous methods, namely, Retinex With Robust envelope (RWRE) [10], Color Enhancement by Scaling (CES) [18], and NASA technology [29]. The results of the four methods are compared in terms of computational load, objective testing (metric evaluation), and subjective testing (visual quality), respectively.

To show the actual computational load, we compared the computation complexity of different methods in terms of actual number of comparisons, additions (or subtractions), multiplications (or divisions), and exponentiation operations. The compared result of three methods is summarized in Table III, where *COMP* denotes the number of comparison, *ADD* denotes the number of additions (or subtractions), *MUL* denotes the number of multiplications (or divisions), and *EXP* denotes the number of exponentiation operations for each pixel in the original image. Note that because the detailed algorithm of NASA technology cannot be found in any literature, we cannot evaluate the computation complexity of NASA technology in Table III. The result indicates that our method required the

TABLE III
COMPARISON IN TERMS OF ACTUAL NUMBER OF OPERATIONS
PER PIXEL FOR DIFFERENT METHODS

Methods	COMP	ADD	MUL	EXP
CES	0	60.55	38.52	0.02
RWRE	3	100	57	1
Proposed	24	9	15	1

least computational load. The evaluations for the computation complexity of different methods are described in the following sections.

In CES, computation of RGB to $YCbCr$ required eight additions and nine multiplications per pixel. The same estimation can be done for $YCbCr$ to RGB , which also required eight additions and nine multiplications per pixel. To compute DCT coefficients, we applied the fast 1-D DCT algorithm proposed in [30] for estimating the number of operations. According to [30], the 1-D eight-point fast DCT algorithm required 29 additions and 11 multiplications. Thus, the 8×8 2-D DCT required $29 \times 8 \times 2$ additions and $11 \times 8 \times 2$ multiplications. Because there are three components ($YCbCr$) in one pixel, computing three DCT coefficients of one pixel, on average, required 21.75 additions and 8.25 multiplications. The same estimation can be applied to the inverse DCT coefficients, and the pixel also required 21.75 additions and 8.25 multiplications. Finally, the computation for scaling DCT coefficients required 1.05 additions, 4.02 multiplications (or divisions), and 0.02 exponentiation operations per pixel. The more detailed descriptions can be found in [18].

In RWRE, the computation of RGB to HSV according to Table II (step 1) required three comparisons, three additions (or subtractions), and three multiplications (or divisions) per pixel. For the iteration of estimating the illumination, the first step is coarse estimation by applying a gradient-descent algorithm to minimize the cost function, which required eight additions and three multiplications per pixel. The second step is to refine the illumination, which requires at least 42×2 additions and 20×2 multiplications per pixel for edge detection with the filter size of 7×7 in vertical and horizontal directions. Then, the computation of the gamma correction required four multiplications (or divisions) and one exponentiation operation. Finally, to transform HSV to RGB according to Table II (step 4), five additions (or subtractions) and seven multiplications (or divisions) were required per pixel. Note that the number of operations for RWRE listed in Table III is only one iteration for the illumination estimation.

In the proposed method, the computation of RGB to HSV required three comparisons, three additions (or subtractions), and three multiplications (or divisions) per pixel. The computation of FIEEMD required 20 comparisons, one addition, and one multiplication per pixel. Then, the number of operations per pixel in the modified gamma correction is one comparison, one addition, four multiplications (or divisions), and one exponentiation. Finally, the computation of HSV to RGB required five additions (or subtractions) and seven multiplications (or divisions). According to Table III, our method required the least computational load.

For objective assessment, we applied discrete entropy (DE) [31], edge-based contrast measure (EBCM) [32], and colorfulness metric (CM) [33] as quantitative measures. These metrics were also used for objective assessment in [11], [13], and [18], respectively.

The DE of image I measures the content of an image, where a higher value indicates that an image has richer details. The DE is defined as

$$DE(I) = - \sum_{\forall i} p(x_i) \log(p(x_i)) \quad (8)$$

where $p(x_i)$ is the probability of the pixel intensity, which is estimated from the normalized histogram.

The EBCM is based on the observation that human perception mechanisms are very sensitive to edges [32]. Thus, it is suitable for evaluating the contrast of images. The gray level corresponding to object contours is obtained by computing the average value of the pixel gray levels weighted by their edge values. The definition of contrast $c(i, j)$ for a pixel of image I located at (i, j) is defined as

$$c(i, j) = \frac{|I(i, j) - e(i, j)|}{|I(i, j) + e(i, j)|} \quad (9)$$

where the mean edge gray level $e(i, j)$ is defined as

$$e(i, j) = \frac{\sum_{(k, l) \in N(i, j)} g(k, l) I(k, l)}{\sum_{(k, l) \in N(i, j)} g(k, l)} \quad (10)$$

where $N(i, j)$ is the set of all neighbors of pixel (i, j) , and $g(k, l)$ is the edge value at pixel (k, l) . Here, the Sobel operator [34] is applied to calculate $g(k, l)$ with 3×3 neighborhood. Then, the EBCM for image I is thus computed as the average contrast value, which is defined as

$$EBCM(I) = \frac{\sum_{i=1}^M \sum_{j=1}^N c(i, j)}{MN} \quad (11)$$

where M and N are the height and the width of image I , respectively. A larger EBCM value indicates that an image has higher contrast.

The CM that measures colorfulness is based on the mean and standard deviations of two axes in a simple opponent color representation with $\alpha = R - G$ and $\beta = 0.5 \times (R + G) - B$, where R , G , and B denoted the red, green, and blue components of an image in the RGB color space. Then, the metric is defined as

$$CM = \sqrt{\sigma_\alpha^2 + \sigma_\beta^2} + 0.3 \sqrt{\mu_\alpha^2 + \mu_\beta^2} \quad (12)$$

where σ_α and σ_β are standard deviations of α and β , respectively, and μ_α and μ_β are the mean of the pixel cloud along directions α and β , respectively. In our comparison, we employed the ratio of CMs between the enhanced image and its original for observing the color enhancement factor (CEF) [18].



Fig. 13. Twenty-two reference images for testing. (a) Image1. (b) Image3. (c) Image4. (d) Image5. (e) Image6. (f) Image9. (g) Image10. (h) Image11. (i) Image14. (j) Image15. (k) Image17. (l) Image24. (m) Image2. (n) Image7. (o) Image8. (p) Image16. (q) Image18. (r) Image19. (s) Image20. (t) Image22. (u) Image23. (v) Image27.

TABLE IV
COMPARISON OF DE FOR IMAGE ENHANCEMENT WITH FOUR METHODS

Image No.	Original	CES	RWRE	NASA	Proposed	Image No.	Original	CES	RWRE	NASA	Proposed
1	4.22	4.39	4.29	4.68	4.28	14	5.23	5.17	5.03	5.32	5.09
2	5.14	5.33	5.18	5.25	5.20	15	4.73	5.10	4.96	5.31	4.93
3	5.26	5.13	5.33	5.45	5.34	16	5.09	4.99	5.21	5.40	5.19
4	5.18	5.24	5.23	5.41	5.21	17	4.99	5.03	5.21	5.36	5.17
5	5.14	5.20	4.94	5.24	4.99	18	4.79	5.22	4.93	5.29	4.93
6	5.23	5.22	5.07	5.36	5.12	19	4.82	3.01	4.38	4.88	4.50
7	5.02	5.28	5.08	5.38	5.06	20	5.22	5.35	5.20	5.43	5.23
8	5.01	5.30	5.00	5.21	5.01	22	4.74	5.14	4.87	5.16	4.87
9	4.40	4.70	5.11	5.30	4.83	23	4.94	5.13	5.30	5.43	5.22
10	5.05	4.87	4.96	5.21	4.98	24	4.60	5.01	4.99	5.27	4.88
11	5.15	5.13	4.91	5.29	4.99	27	5.09	5.31	5.03	4.91	5.12
						Avg.	4.96	5.01	5.01	5.25	5.01

We use 22 testing images downloaded from the NASA website [29], as shown in Fig. 13, to explore the performance of objective testing for the four-image enhancement method. The size of the testing image is 283×432 [see Fig. 11(f)], 640×501 [see Fig. 13(v)], or 1312×2000 [see Fig. 13(a)–(e) and Fig. 13(g)–(u)]. The compared results of DE, EBCM, and CEF for the four methods are shown in Tables IV–VI, respectively. Note that the enhanced images of CES and RWRE are produced by the source codes, which were provided by the original authors, and the results of NASA technology are downloaded from the website [29]. In Table IV, the NASA method enhances the image with the richest detail. Our method is a little worse than the NASA method but similar to CES and RWRE. In Table V, the result of the NASA method has the best image contrast. Our method is a little worse than the NASA method but better than CES and RWRE. This is because the FIEEMD estimates a smoothed illumination and preserves more contrast

TABLE V
COMPARISON OF EBCM FOR IMAGE ENHANCEMENT WITH FOUR METHODS

Image No.	Original	CES	RWRE	NASA	Proposed	Image No.	Original	CES	RWRE	NASA	Proposed
1	0.017	0.017	0.015	0.032	0.019	14	0.016	0.017	0.015	0.026	0.018
2	0.029	0.029	0.025	0.049	0.034	15	0.025	0.025	0.022	0.052	0.029
3	0.027	0.027	0.024	0.052	0.032	16	0.028	0.028	0.024	0.054	0.032
4	0.022	0.023	0.020	0.042	0.027	17	0.025	0.025	0.024	0.045	0.032
5	0.017	0.017	0.015	0.024	0.019	18	0.030	0.030	0.025	0.051	0.033
6	0.019	0.020	0.017	0.033	0.022	19	0.013	0.011	0.011	0.016	0.014
7	0.020	0.020	0.017	0.029	0.021	20	0.028	0.028	0.022	0.047	0.030
8	0.017	0.017	0.016	0.030	0.020	22	0.025	0.025	0.022	0.046	0.028
9	0.079	0.080	0.068	0.084	0.086	23	0.049	0.048	0.038	0.096	0.053
10	0.022	0.021	0.017	0.034	0.023	24	0.026	0.027	0.023	0.044	0.031
11	0.022	0.022	0.017	0.033	0.022	27	0.062	0.059	0.048	0.080	0.077
						Avg.	0.028	0.028	0.024	0.045	0.032

TABLE VI
COMPARISON OF CEF FOR COLOR IMAGE ENHANCEMENT WITH FOUR METHODS

Image No.	CES	RWRE	NASA	Proposed	Image No.	CES	RWRE	NASA	Proposed
1	1.47	1.70	1.66	1.55	14	1.53	1.54	1.39	1.44
2	1.48	1.67	1.49	1.54	15	1.66	2.16	2.33	1.90
3	1.40	1.48	1.24	1.39	16	1.42	1.59	1.49	1.47
4	1.50	1.75	1.7	1.58	17	1.67	1.87	1.83	1.71
5	1.56	1.69	1.62	1.56	18	1.78	2.40	2.73	2.14
6	1.62	1.77	1.66	1.62	19	1.25	1.20	1.11	1.17
7	1.55	1.70	1.72	1.55	20	1.50	1.68	1.57	1.54
8	1.62	1.8	1.85	1.64	22	1.73	2.07	1.98	1.86
9	1.60	2.14	2.30	1.88	23	1.70	2.17	2.02	2.01
10	1.12	1.23	1.08	1.18	24	1.76	2.46	2.81	2.23
11	1.57	1.84	1.72	1.68	27	1.88	2.25	2.04	2.01
					Avg.	1.56	1.83	1.79	1.67

in the reflectance. In Table VI, RWRE has the most colorful result. Our method is similar to the NASA method and better than CES. This is because the Retinex theory can keep and enhance the color information in the reflectance. RWRE, NASA, and our method are all based on the Retinex theory. In average, our method obtains comparable image quality as compared with other methods.

Figs. 14–16 show the enhanced results to illustrate the performance of visual quality with the four methods. The enhanced results of the NASA method have better visual quality and color contrast than ours and the other two methods. However, our method can efficiently enhance the details of the image to obtain the comparable results as compared with other methods, particularly in the low- and high-luminance regions. More experimental results, which are not reported here due to space limitations, are available as supplementary materials in the website <http://dic.csie.ncku.edu.tw/retinex-mfabemd/>.

To illustrate the applications for the ITS, Figs. 17 and 18 show the comparison results of the vehicle image and road image at night, respectively. It can be seen that our method reveals the clearer license plate of the vehicle and the more detailed background on the road than CES. Our results are similar to RWRE, but RWRE generates more noise than ours. Furthermore, the computation load of our method is far less than RWRE. Our method is very suitable for the applications in the real-time ITS.

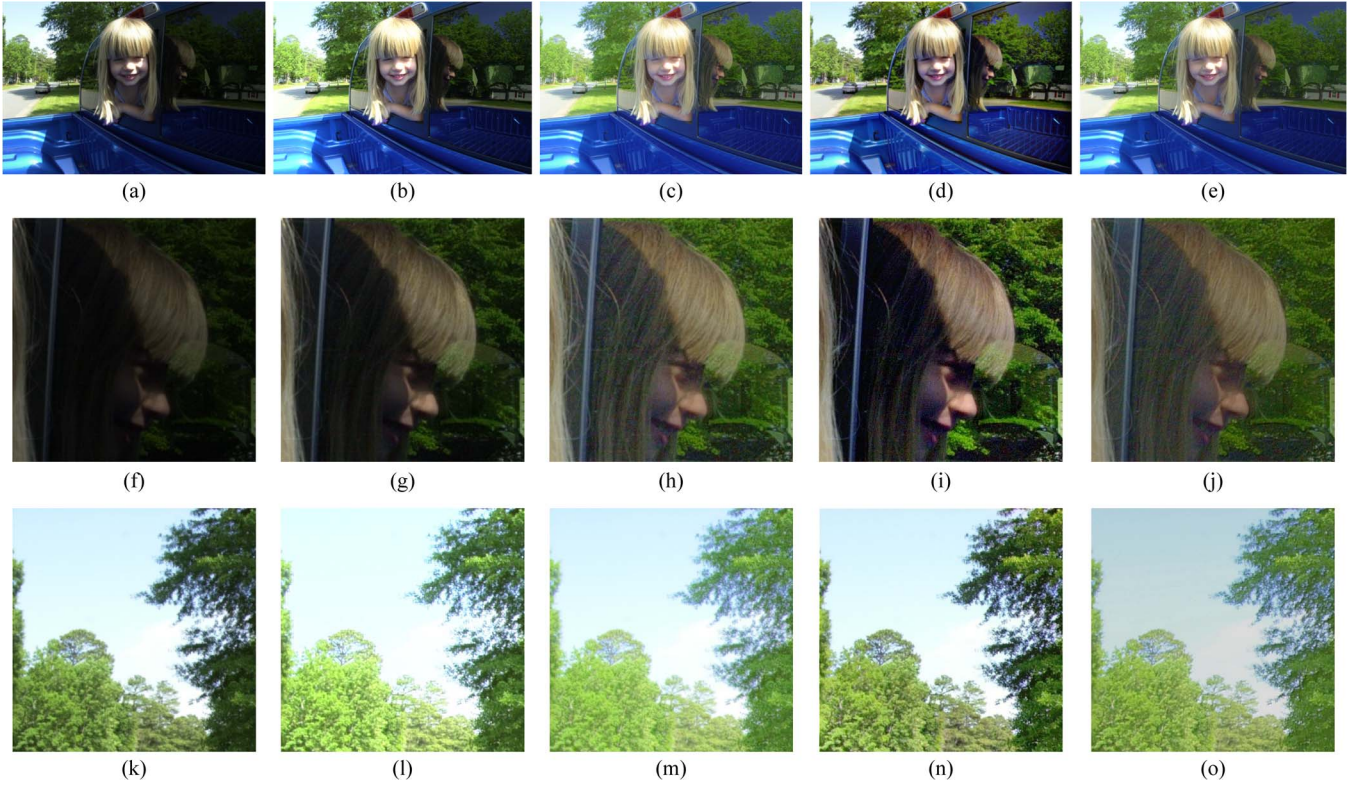


Fig. 14. Enhanced results and the magnified region for Image16 with four methods. (a) Original image. (b) CES. (c) RWRE. (d) NASA. (e) Proposed method. (f)–(j) Close-up of the shade window for images (a)–(e), respectively. (k)–(o) Close-up of the leaves and sky for images (a)–(e), respectively.

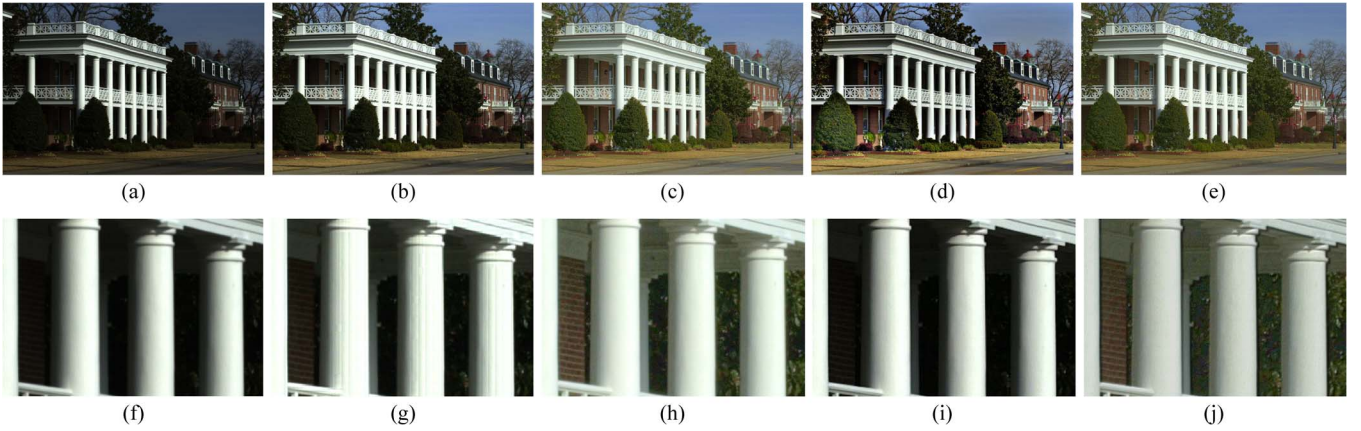


Fig. 15. Enhanced results and the magnified region for Image23 with four methods. (a) Original image. (b) CES. (c) RWRE. (d) NASA. (e) Proposed method. (f)–(j) Close-up of the pillar for images (a)–(e), respectively.

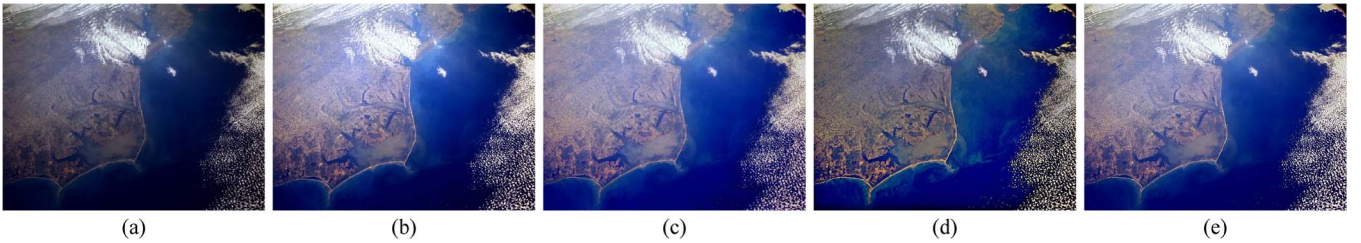


Fig. 16. Enhanced results for Image27 with four methods. (a) Original image. (b) CES. (c) RWRE. (d) NASA. (e) Proposed method.

We also use Macbeth ColorChecker [12] to measure color distortion as shown in Fig. 19. In our experiment, images of ColorChecker were captured by the camera in low luminance.

Fig. 20 shows the comparison of the enhancement results with other two methods, i.e., CES and RWRE. Three test ColorChecker images captured with different exposures (1/5, 1/6, and



Fig. 17. Enhanced results of the vehicle image with three methods. (a) Original image. (b) CES. (c) RWRE. (d) Proposed method.



Fig. 18. Enhanced results of the road image with three methods. (a) Original image. (b) CES. (c) RWRE. (d) Proposed method.



Fig. 19. Macbeth ColorChecker

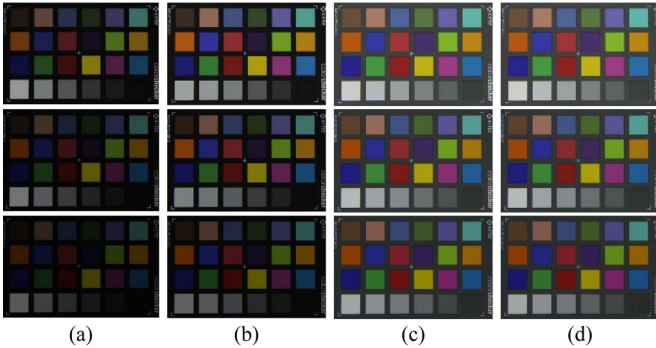


Fig. 20. ColorChecker images. (a) Original images. The enhancement results generated by (b) CES, (c) RWRE, and (d) the proposed method.

1/8 s) are displayed from top to bottom. It can be seen that our method has low color distortion and can achieve the comparable results as compared with other methods.

In summary, from the comparison in terms of computational load, objective testing, and subjective testing, while considering the tradeoff between image quality and computational resources, the proposed method has lower complexity and can output comparable image quality as compared with those previous techniques and is very suitable for low-cost very large scale integration (VLSI) implementation.

The hardware architecture of our illumination adjustment algorithm is implemented as an IP core that can be integrated with other circuits in the SOC system. To evaluate our VLSI implementation, our design was implemented by using Verilog HDL.

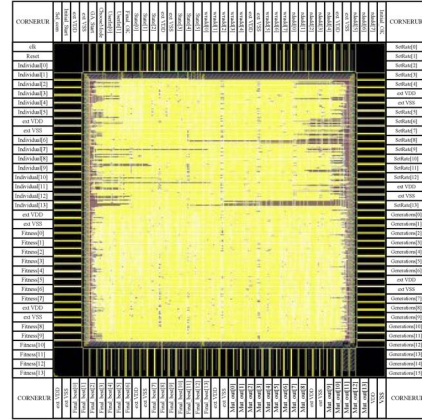


Fig. 21. Layout of our design.

We used SYNOPSIS Design Vision to synthesize the design with TSMC 0.13- μ m cell library. Finally, the layout of our design, as shown in Fig. 21, was generated with SYNOPSIS Astro (for auto placement and routing) and verified by MENTOR GRAPHIC Calibre (for DRC and LVS checks). SYNOPSIS PrimeTime was employed to measure the total power consumption after auto placement and routing. Its core size and gate count are about 1.835 mm \times 1.835 mm and 11.6 K, respectively. In the simulation, it works with a clock period of 5 ns and can achieve a processing rate of 200 Mpixels/s. For an image with a resolution of QXGA (2560 \times 2048), it requires about 27 ms to process one frame, which is suitable for real-time applications. For further verification, the circuit was also implemented on an Altera field-programmable gate array (FPGA) board. The operating clock frequency of FPGA implementation is 55.55 MHz with 1.784-K logic elements. The summary of the implementation results is listed in Table VII. To meet the different requirements of the image processing system, we provide the power consumption of our core at different clock periods, as listed in Table VIII. In low-cost imaging systems, the processing rate can be slowed down, and our circuit can run at low power consumption.

TABLE VII
SIMULATION RESULTS OF OUR DESIGN

FPGA Device/ ASIC Library	Line Buffer	Total Logic Elements	Flip Flops	Gate Counts	Frequency (MHz)
Altera Cyclone IV EP4CE30	4	1,784	510	N/A	55
TSMC's 0.13- μ m	4	N/A	N/A	15,742	200

TABLE VIII
POWER CONSUMPTION OF OUR DESIGN
WITH DIFFERENT CLOCK PERIODS

Clock Period (ns)	5	10	20	40	80
Power Consumption (mW)	19.8	8.92	3.968	2.49	1.25

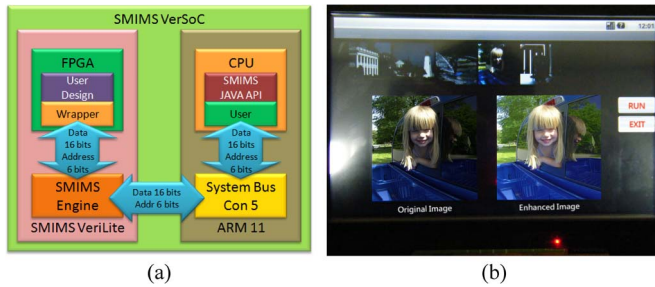


Fig. 22. Implementation on the hardware-software codesign platform. (a) Block diagram of the hardware-software codesign platform. (b) Result of hardware-software coverage verification.

To integrate our IP core in the system, we implemented it on the hardware-software codesign platform (SMIMS VeriSoC-6410), which included an ARM-based SoC board and an FPGA board. In this verification system, a graphical user interface APP was designed in JAVA language for an Android system on the 533-MHz Samsung S3C6410A ARM11 processor with 256-MB memory. The proposed illumination adjustment circuit and a system bus wrapper were implemented in Verilog HDL for FPGA on the Altera Cyclone IV EP4CE30. This platform uses memory mapping to connect the ARM CPU and the FPGA device. The APP in the board was used to send the original image to Altera FPGA and to receive the result generated from Altera FPGA. The block diagram of the hardware-software codesign platform and the result of our design implemented on this platform are shown in Fig. 22.

VI. CONCLUSION

In ITSs, the cameras keep track of the road and driving situations to detect the traffic flow or to record information related to vehicle crashes or accidents automatically. Insufficient visibility due to the influence of weather or poor atmospheric light will cause the intelligent system to be inoperative. Thus, many image preprocessing algorithms are integrated into ITSs for enhancing the visibility of the system, such as denoising, contrast enhancement, and illumination adjustment. The enhanced image will increase the successful rate of the following processing steps of the ITS, such as car license plate detection and recognition, human detection and tracking, and vehicle crash or accident detection. Since the ITS is a real-time system, its high-speed processing for enhancement algorithms is necessary

and must be considered. Many methods are proposed to reach higher visual image quality without considering the computational complexity. However, when it comes to implementing those algorithms on hardware core employed in the end-user camera equipment, the computational complexity becomes an important issue, particularly for real-time applications.

In this paper, a low-complexity pipelined hardware architecture for a fast illumination adjustment method has been proposed. The experimental results demonstrate that our design requires the least computation load and achieves comparable performance in objective metrics and subjective image quality as compared with other image enhancement methods. To our knowledge, there are few literatures on the hardware design for illumination adjustment. Our low-cost circuit is a good solution for low-cost VLSI implementation for illumination adjustment, which can be integrated with other hardware modules in the SOC system for real-time transportation applications. The design meets the requirement of real-time image/video applications and is suitable to be employed in end-user camera equipment. In the future, the illumination adjustment method may be integrated with a haze detection and a haze removal method. Such integration will increase the recognition rate of the following object recognition of the ITS in bad weather conditions.

REFERENCES

- [1] S. Saponara, L. Fanucci, S. Marsi, and G. Ramponi, "Algorithmic and architectural design for real-time and power-efficient Retinex image/video processing," *J. Real-Time Image Process.*, vol. 1, no. 4, pp. 267–283, Jul. 2007.
- [2] F. Hassan and J. E. Carletta, "An FPGA-based architecture for a local tone-mapping operator," *J. Real-Time Image Process.*, vol. 2, no. 4, pp. 293–308, Dec. 2007.
- [3] S. Marsi and G. Ramponi, "A flexible FPGA implementation for illuminance-reflectance video enhancement," *J. Real-Time Image Process.*, vol. 8, no. 1, pp. 81–93, Mar. 2011.
- [4] Y.-H. Shiau, H.-Y. Yang, P.-Y. Chen, and Y.-Z. Chuang, "Hardware implementation of a fast and efficient haze removal method," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 8, pp. 1369–1374, Aug. 2013.
- [5] D. J. Jobson, Z. Rahman, and G. A. Woodell, "Properties and performance of a center/surround retinex," *IEEE Trans. Image Process.*, vol. 6, no. 3, pp. 451–462, Mar. 1997.
- [6] D. J. Jobson, Z. Rahman, and G. A. Woodell, "A multiscale retinex for bridging the gap between color images and the human observation of scenes," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 965–976, Jul. 1997.
- [7] M. Ogata, T. Tsuchiya, T. Kubozono, and K. Ueda, "Dynamic range compression based on illumination compensation," *IEEE Trans. Consum. Electron.*, vol. 47, no. 3, pp. 548–558, Aug. 2001.
- [8] R. Kimmel, M. Elad, D. Shaked, R. Keshet, and I. Sobel, "A variational framework for retinex," *Int. J. Comput. Vis.*, vol. 52, no. 1, pp. 7–23, Apr. 2003.
- [9] S. Marsi, G. Impoco, A. Ukovich, G. Ramponi, and S. Carrato, "Using a recursive rational filter to enhance color images," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 6, pp. 1230–1236, Jun. 2008.
- [10] C.-T. Shen and W.-L. Hwang, "Color image enhancement using retinex with robust envelope," in *Proc. 16th IEEE Int. Conf. Image Process.*, 2009, pp. 3141–3144.
- [11] T. Arici, S. Dikbas, and Y. Altunbasak, "A histogram modification framework and its application for image contrast enhancement," *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 1921–1935, Sep. 2009.
- [12] S. C. Huang, F. C. Cheng, and Y. S. Chiu, "Efficient contrast enhancement using adaptive gamma correction with weighting distribution," *IEEE Trans. Image Process.*, vol. 9, no. 1, pp. 1032–1041, Mar. 2013.
- [13] T. Celik and T. Tjahjedi, "Automatic image equalization and contrast enhancement using Gaussian mixture modeling," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 145–156, Jan. 2012.
- [14] F. C. Cheng and S. C. Huang, "Efficient histogram modification using bilateral Bézier curve for the contrast enhancement," *J. Display Technol.*, vol. 9, no. 1, pp. 44–50, Jan. 2013.

- [15] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," in *Proc. SIGGRAPH*, 2002, pp. 249–256.
- [16] J. Tang, E. Peli, and S. Acton, "Image enhancement using a contrast measure in the compressed domain," *IEEE Signal Process. Lett.*, vol. 10, no. 10, pp. 289–292, Oct. 2003.
- [17] S. Lee, "An efficient content-based image enhancement in the compressed domain using retinex theory," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 2, pp. 199–213, Feb. 2007.
- [18] J. Mukherjee and S. K. Mitra, "Enhancement of color images by scaling the DCT coefficients," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1783–1794, Oct. 2008.
- [19] E. H. Land and J. J. McCann, "Lightness and retinex theory," *J. Opt. Soc. Am.*, vol. 61, no. 1, pp. 1–11, Jan. 1971.
- [20] N. E. Huang *et al.*, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. R. Soc. A*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.
- [21] S. M. A. Bhuiyan, R. R. Adhami, and J. F. Khan, "Fast and adaptive bidimensional empirical mode decomposition using order-statistics filter based envelope estimation," *EURASIP J. Adv. Signal Process.*, vol. 2008, pp. 1–18, Jan. 2008.
- [22] J. C. Nunes, Y. Bouaouene, E. Delechelle, O. Niang, and Ph. Bunel, "Image analysis by bidimensional empirical mode decomposition," *Image Vis. Comput.*, vol. 21, no. 12, pp. 1019–1026, Nov. 2003.
- [23] J. C. Nunes, S. Guyot, and E. Delechelle, "Texture analysis based on local analysis of the bidimensional empirical mode decomposition," *Mach. Vis. Appl.*, vol. 16, no. 3, pp. 177–188, May 2005.
- [24] Z. Liu and S. Peng, "Boundary processing of bidimensional EMD using texture synthesis," *IEEE Signal Process. Lett.*, vol. 12, no. 1, pp. 33–36, Jan. 2005.
- [25] P.-H. Chen, Y.-C. Yang, and L.-M. Chang, "Illumination adjustment for bridge coating images using BEMD-morphology approach BMA," *Autom. Construction*, vol. 19, no. 4, pp. 475–484, Jul. 2010.
- [26] S.-C. Pei, M. Tzeng, and Y.-Z. Hsiao, "Enhancement of uneven lighting text image using line-based empirical mode decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2011, pp. 1249–1252.
- [27] H.-Y. Yang, S.-Y. Li, P.-Y. Chen, and Y.-H. Shiau, "Efficient color image enhancement based on fast and adaptive bidimensional empirical mode decomposition," in *Proc. Int. Symp. Comput. Scie. Soc.*, 2011, pp. 319–322.
- [28] Y. Lu and M. Dai, "Sort optimization algorithm of median filtering based on FPGA," in *Proc. Int. Conf. Mach. Vis. Human-mach. Interface*, 2010, pp. 250–253.
- [29] Retinex Image Processing. [Online]. Available: <http://dragon.larc.nasa.gov/retinex/pao/news/>
- [30] C. Loeffler, A. Lieenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 1989, vol. 2, pp. 988–991.
- [31] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.
- [32] A. Beghdadi and A. L. Negrata, "Contrast enhancement technique based on local detection of edges," *Comput. Vis. Graph. Image Process.*, vol. 46, no. 2, pp. 162–174, May 1989.
- [33] S. Susstrunk and S. Winkler, "Color image quality on the internet," in *Proc. IS&T/SPIE Electron. Imag., Internet Imagi. V*, 2004, pp. 118–131.
- [34] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2006.



Yeu-Horng Shiau received the B.S. and Ph.D. degrees in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 1997 and 2004, respectively.

He is currently an Assistant Professor with the Department of Electrical Engineering, National Yunlin University of Science and Technology, Yunlin, Taiwan. His research interests include digital system design, very large scale integrated chip design, and video compression.



Pei-Yin Chen (M'08) received the B.S. degree from National Cheng Kung University, Tainan, Taiwan, in 1986; the M.S. degree from Pennsylvania State University, University Park, PA, USA, in 1990; and the Ph.D. degree from National Cheng Kung University in 1999, all in electrical engineering.

He is currently a Professor with the Department of Computer Science and Information Engineering, National Cheng Kung University. His research interests include very large scale integration chip design, video compression, fuzzy logic control, and gray

prediction.



Hung-Yu Yang received the B.S. degree from Chung Cheng Institute of Technology, National Defense University, Taoyuan, Taiwan, in 2003; the M.S. degree from National Taipei University of Education, Taipei, Taiwan, in 2008; and the Ph.D. degree in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 2013.

His current research interests include image processing, very large scale integrated chip design, and embedded systems.



Shang-Yuan Li received the B.S. degree from National Central University, Taoyuan, Taiwan, in 2009 and the M.S. degree in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 2011.

His current research interests include image processing, very large scale integrated chip design, and embedded systems.