Hardware Implementation of a Fast and Efficient Haze Removal Method

Yeu-Horng Shiau, Hung-Yu Yang, Pei-Yin Chen, *Member, IEEE,* and Ya-Zhu Chuang

***Abstract*—In this letter, a fast and efficient haze removal method is presented. We employ an extremum approximate method to extract the atmospheric light and propose a contour preserving estimation to obtain the transmission by using edge-preserving and mean filters alternately. Our method can efficiently avoid the halo artifact generated in the recovered image. To meet the requirement of real-time applications, an 11-stage pipelined hardware architecture for our haze removal method is presented. It can achieve 200 MHz with 12.8K gate counts by using TSMC 0.13-μm technology. Simulation results indicate that our design can obtain comparable results with the least execution time compared to previous algorithms and is suitable for low-cost high-performance hardware implementation for haze removal.**

***Index Terms*—Hardware architecture, haze removal, real-time.**

I. Introduction

RECENTLY, there has been 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 the intelligent transportation system, 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, the 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, illumination adjustment, and haze removal. In such a real-time surveillance system, a fast-enough and low-complexity solution for those preprocessing algorithms is necessary and must be considered. With the development of very large scale integration technology, hardware implementation for those algorithms becomes a better solution, which can be included in end-user camera equipment to meet the real-time requirement. In this paper, we focus on the development of a fast and efficient haze removal method that is suitable for hardware implementation.

A captured image might be corrupted due to fog or haze. The degraded image substantially loses contrast and reduces the visual quality of the object in the scene. The effect of fog or haze will enormously increase if varied with the distance of the scene points from the camera. To overcome this problem, many researchers have proposed various techniques to enhance the visibility of the blurred image [1]–[12]. These existing methods can be classified into two categories: multiple-image processing [1]–[6] and single-image processing [7]–[12]. Generally, in many cases, it may not be possible to acquire multiple images. Thus, the single-image processing method for haze removal has attracted much attention in recent years. Oakley *et al.* [7] assumed that the airlight is constant over the entire image by finding the minimum value of the global cost function. Kopf *et al.* [8] presented the deep photo system for browsing and editing outdoor photographs by using the geo-referenced terrain and urban models. Tan [9] introduced an automated method to maximize the contrast of images and developed a cost function in the framework of the Markov random fields. Fattal [10] presented a novel transmission estimation method to increase scene visibility and recover haze-free scene contrasts. Tarel and Hautière [11] proposed a novel algorithm for visibility restoration based on a filtering approach. He *et al.* [12] introduced a novel prior (dark channel prior) to remove haze from a single image. However, the haze removal algorithms mentioned above always suffered from the intensity computational problem. In many practical image applications, the exhaustive computation makes the method difficult to meet real-time requirements.

In this letter, a fast and efficient method for haze removal by using the concept of a dark channel is proposed. We employ an extremum approximate method to extract the atmospheric light and apply edge-preserving and mean filters to estimate the transmission. Our method can efficiently avoid the halo artifact generated in the recovered image and obtain the comparable results with the least execution time as compared with previous algorithms. To achieve the goal of real-time haze removal, we present an 11-stage pipelined hardware architecture implemented as an intellectual property core. Our haze removal circuit can be implemented with a dedicated chip

Manuscript received May 17, 2012; revised August 20, 2012 and November 14, 2012; accepted December 19, 2012. Date of publication January 29, 2013; date of current version July 31, 2013. This work was supported in part by the National Science Council, Taiwan, under Grant NSC-101-2221-E-006-151-MY3, and the Ministry of Economic Affairs (MOEA) of Taiwan under Grant MOEA 100-EC-17-A-05-S1-192. This research received funding from the Headquarters of University Advancement at the National Cheng Kung University, which is sponsored by the Ministry of Education, Taiwan, R.O.C. This paper was recommended by Associate Editor R. Lukac.

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

H.-Y. Yang, P.-Y. Chen, and Y.-Z. Chuang are with the Digital IC Design Laboratory, Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 70101, Taiwan (e-mail: yang.hungyu@gmail.com; pychen@mail.ncku.edu.tw; hweijet@gmail.com).

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/TCSVT.2013.2243650

or integrated with other image-processing components, such as noise removal and image enhancement, on a single chip for real-time image applications.

The rest of this letter is organized as follows. In Section II, the background and related algorithm are introduced briefly. The proposed method is presented in Section III. Section IV describes the hardware architecture of our method in detail. Section V illustrates the simulation results and hardware implementation. Conclusions are provided in Section VI.

II. Background

In bad weather conditions, the observed light is absorbed and scattered by turbid mediums, such as atmosphere particles or raindrops. Based on this phenomenon, Koschmieder [13] proposed a simple atmospheric scattering model to represent the formation of a haze image. This model is also employed in many haze removal approaches [2], [3], [7], [9]–[12] and is expressed as

$$\boldsymbol{I}(x) = \boldsymbol{J}(x)t(x) + \boldsymbol{A}(1 - t(x)) \tag{1}$$

where $\boldsymbol{I}$ is the haze image on the three *RGB* color channels, $\boldsymbol{J}$ is the scene without haze, t is the transmission coefficient to describe the percentage of light that can penetrate the haze, and $\boldsymbol{A}$ is the atmospheric light.

Using this atmospheric scattering model to recover the scene $\boldsymbol{J}$, the main challenge of haze removal is to estimate the atmospheric light $\boldsymbol{A}$ and the transmission t from the source image $\boldsymbol{I}$ properly. He *et al.* [12] presented the dark channel prior to estimate the atmospheric light and the transmission. This method can obtain the best recovered scene as compared with other techniques [2], [3], [7], [9]–[11]. First, the dark channel of the input image $\boldsymbol{I}$ is obtained by

$$I^{dark}(x) = \min_{y\in\Omega(x)}\{\min_{c\in\{R,G,B\}} I^c(y)\} \tag{2}$$

where Ω(x) is a local square window of the minimum filter centered at x coordinate. I^c means the R, G, B color channel of the input image $\boldsymbol{I}$. Then, the atmospheric light is estimated from the top 0.1% brightest pixels in the dark channel I^{dark}. According to (1) and (2), the transmission can be computed as

$$t(x) = 1 - \omega \min_{y\in\Omega(x)}\{\min_{c\in\{R,G,B\}} \frac{I^c(y)}{A^c}\} \tag{3}$$

where ω is used to keep a small amount of haze. A^c is R, G, B color channel of the atmospheric light. To avoid the halo artifact generated in the recovered scene, the soft matting method [12] is applied to optimize the transmission. To reduce the running time of this procedure, He *et al.* [14] also presented a guided filter instead of the soft matting to refine the transmission effectively. Avoiding the noise generated in the recovered scene, a lower bound t_0 is used to restrict $t(x)$. Finally, the scene $\boldsymbol{J}$ can be calculated as

$$\boldsymbol{J}(x) = \frac{\boldsymbol{I}(x) - \boldsymbol{A}}{\max\{t(x), t_0\}} + \boldsymbol{A}. \tag{4}$$

However, the haze removal method proposed by [12] requires some extensive and complex computations, such as huge matrix multiplication/division, sort, exponent, and floating-point operation. The experimental results in [12] show that their method takes about 10–20 s to process a 600×400 image on a PC with the Pentium 4 processor. For the practical image applications, this method cannot meet the real-time requirement with a low-speed processor. Thus, we propose a more efficient and low-complexity haze removal method that can obtain good results and satisfy the real-time applications.

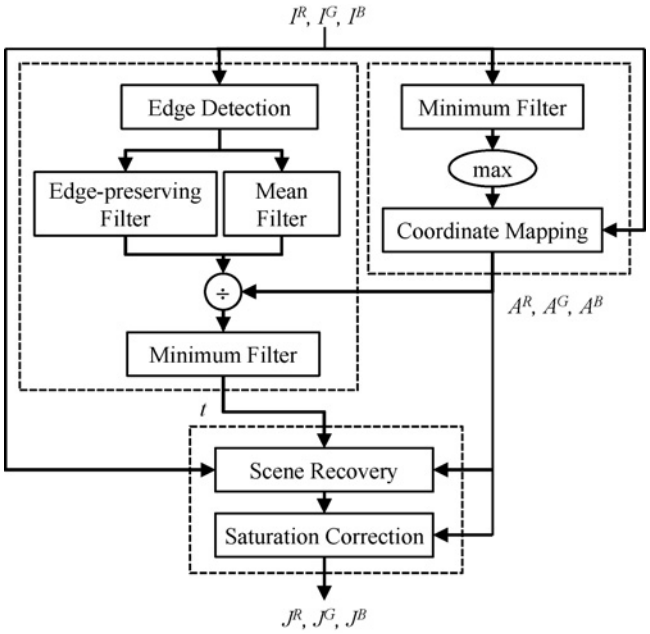

Fig. 1. Flow diagram of the proposed haze removal algorithm.

III. Fast and Efficient Haze Removal Method

The flow diagram of our efficient haze removal algorithm is shown in Fig. 1. Our method contains three procedures: atmospheric light estimation, transmission estimation, and scene recovery. First, an extremum approximate method with a minimum filter is presented to estimate the atmospheric light. Second, we employ an edge detection method to find the contour of objects and apply edge-preserving filter and mean filter alternately to estimate the transmission. Finally, a saturation correction method is proposed to refine the recovered scene. The details of each process are described as follows.

A. Atmospheric Light Estimation

In [12], they picked the top 0.1% brightest pixels from the dark channel and selected a suitable value from these brightest pixels as the atmospheric light. However, the sorting process during this atmospheric light estimation requires a long execution period. The execution time of the sorting process will increase with the size of an input image. Thus, we propose an extremum approximate method that employs a 3×3 minimum filter to obtain the dark channel and finds the pixel with the maximum value in the dark channel as the atmospheric light. The dark channel in our method is calculated as

$$I^{dark'}(i, j) = \min\left\{minI^R(i, j),\ minI^G(i, j),\ minI^B(i, j)\right\} \tag{5}$$

where *min* is the minimum operation to find the minimum value among $minI^R$, $minI^G$, and $minI^B$ at (i, j) coordinate. The values $minI^R(i, j)$, $minI^G(i, j)$, and $minI^B(i, j)$ are obtained as

$$\min I^c(i, j) = \min_{y \in \Omega_{3\times3}(i,j)} \{I^c(y)\} \quad \forall c \in \{R, G, B\} \tag{6}$$

where $\Omega_{3\times3}(i, j)$ is a 3×3 window centered at (i, j) coordinate. When $I^{\text{dark}'}$ is obtained, a maximum value A^{dark} can be determined as

$$A^{\text{dark}} = \max_{(i,j)\in I} \{I^{\text{dark}'}(i, j)\}. \tag{7}$$

This maximum value A^{dark} is chosen as the candidate of the atmospheric light. Assume that the pixel A^{dark} is located at (s, t) coordinate, the corresponding pixel value of I at (s, t) coordinate is found as the atmospheric light. Then, an adjustment parameter σ is applied to refine the atmospheric light. The refined atmospheric light is generalized by

$$A^c = \sigma \times (I^c(s, t)) \quad \forall c \in \{R, G, B\}. \tag{8}$$

The parameter σ can be considered an intensity correction for the recovered scene adjustment. An appropriate value of σ is set to 0.875 that makes the recovered scene more pleasant and makes the method more suitable for low-cost hardware implementation by using shift operation.

B. Transmission Estimation

The assumption for estimating the transmission in the dark channel prior is that the transmission in a local square window $\Omega(x)$ is constant. In fact, the transmission is not always constant in a window, especially in the contour of object. For example, two adjacent pixels may be at quite different distances from the camera, but in (3), the same dark channel is chosen within a window to estimate the transmissions of these two pixels. This means that the pixel at a far distance is recovered by the same transmission as the pixel at a near distance. This erroneous estimation makes the pixel at a far distance yield the halo artifact. To avoid these artifacts, the edge-preserving filter and the mean filter are employed to replace the first minimum operation of (3) according to the gradient. By this method, our approach can preserve edges in detail areas, while still permitting dehazing in smooth areas and minimizing the halo artifact. First, we define a gradient detection as

$$ED(i, j) = \begin{cases} 2, \text{ if } |I^c(i-1, j-1) - I^c(i+1, j+1)| \geq D \text{ or} \\ \qquad |I^c(i-1, j+1) - I^c(i+1, j-1)| \geq D, \\ \quad \text{for } c \in \{R, G, B\} \\ 1, \quad \text{if } |I^c(i-1, j) - I^c(i+1, j)| \geq D \text{ or} \\ \qquad |I^c(i, j-1) - I^c(i, j+1)| \geq D, \\ \quad \text{for } c \in \{R, G, B\} \\ 0, \quad \text{otherwise} \end{cases} \tag{9}$$

where D is a threshold to determine whether there is an edge in a certain direction within a 3×3 window centered at (i, j) coordinate. The value of D is fixed to 80 in the experiments for all results. $ED(i, j) = 2$ means that the diagonal edge is detected. $ED(i, j) = 1$ means that the vertical or horizontal edge is determined. Otherwise, no edge is found. Then, according to the decision of (9), the transmission is estimated as

$$t(i, j) = 1 - \omega' \min_{c\in\{R,G,B\}} \left\{ \frac{P^c_{ED(i,j)}(i, j)}{A^c} \right\}$$

where

$$\begin{aligned} &P^c_0(i, j) \\ &= (I^c(i-1, j-1) + I^c(i-1, j) + I^c(i-1, j+1) \\ &\quad + I^c(i, j-1) + I^c(i, j) + I^c(i, j+1) \\ &\quad + I^c(i+1, j-1) + I^c(i+1, j) + I^c(i+1, j+1))/9 \\ &P^c_1(i, j) \\ &= (I^c(i-1, j-1) + 2 \times I^c(i-1, j) + I^c(i-1, j+1) \\ &\quad + 2 \times I^c(i, j-1) + 4 \times I^c(i, j) + 2 \times I^c(i, j+1) \\ &\quad + I^c(i+1, j-1) + 2 \times I^c(i+1, j) + I^c(i+1, j+1))/16 \\ &P^c_2(i, j) \\ &= (2 \times I^c(i-1, j-1) + I^c(i-1, j) + 2 \times I^c(i-1, j+1) \\ &\quad + I^c(i, j-1) + 4 \times I^c(i, j) + I^c(i, j+1) \\ &\quad + 2 \times I^c(i+1, j-1) + I^c(i+1, j) + 2 \times I^c(i+1, j+1))/16. \end{aligned} \tag{10}$$

In (10), $P^c_{ED(i,j)}$ is the filter for estimating the transmission according to $ED(i, j)$, and ω' is fixed to 0.9375 for all results in this letter. If $ED(i, j) = 1$ or $ED(i, j) = 2$, the edge-preserving filter, P^c_1 or P^c_2, is employed to enhance the contour, respectively. Otherwise, the mean filter P^c_0 is applied to smooth the texture inside the object. The fixed value of ω' is suitable for low-cost hardware implementation by using shift operation.

C. Scene Recovery

Once the atmospheric light and the transmission are estimated, the scene $\boldsymbol{J}$ can be recovered by

$$J^c(i, j) = \frac{I^c(i, j) - A^c}{\max\left\{t(i, j),\ t'_0\right\}} + A^c \quad \forall c \in \{R, G, B\} \tag{11}$$

where t'_0 is a low bound of the transmission and this value is fixed to 0.25 for all results in this letter. Since the color of the scene $\boldsymbol{J}$ recovered by a 3×3 window looks oversaturated, we present a saturation correction method to adjust the scene $\boldsymbol{J}$ to be more pleasant. The correction is defined as

$$\tilde{J}^c(i, j) = (A^c)^\beta \times J^c(i, j)^{(1-\beta)} \quad \forall c \in \{R, G, B\}. \tag{12}$$

The parameter β can be set between 0.1 and 0.3. In this letter, β is fixed to 0.3 in the experiments for all results.

IV. Hardware Architecture

To actually verify the computational complexity of our method, Table I shows the computing time on Cortex-A8 (embedded system processor) and Core 2 Quad processors, respectively. We can observe that it takes about 5.909 s to process a 600×400 color image on Cortex-A8, which is about 50 times slower than the processing time on Core 2 Quad processors. For some real-time image/video applications on the embedded system, a faster execution time is required. Thus, we design a pipelined hardware core of our method to satisfy the requirement for real-time applications on the embedded system.

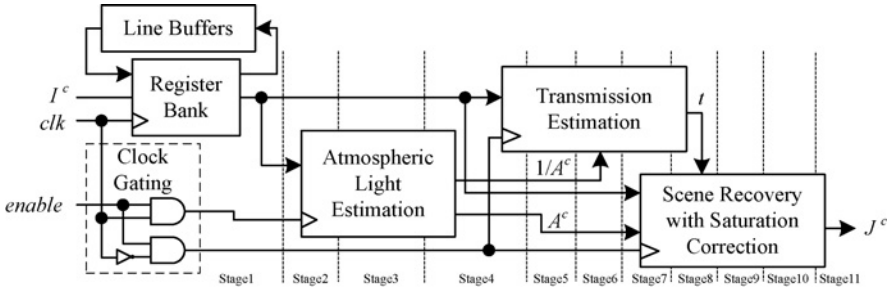

Fig. 2. Block diagram of hardware architecture for our haze removal method.

TABLE I
EXECUTION TIME OF OUR METHOD FOR DIFFERENT PROCESSORS (UNIT: SECOND)

Processor\Image Size	500 × 360	441 × 450	600 × 450	1024 × 768	600 × 400	832 × 556
Cortex-A8	5.249	5.355	8.692	23.883	5.909	14.117
Core 2 Quad	0.151	0.154	0.25	0.687	0.172	0.406

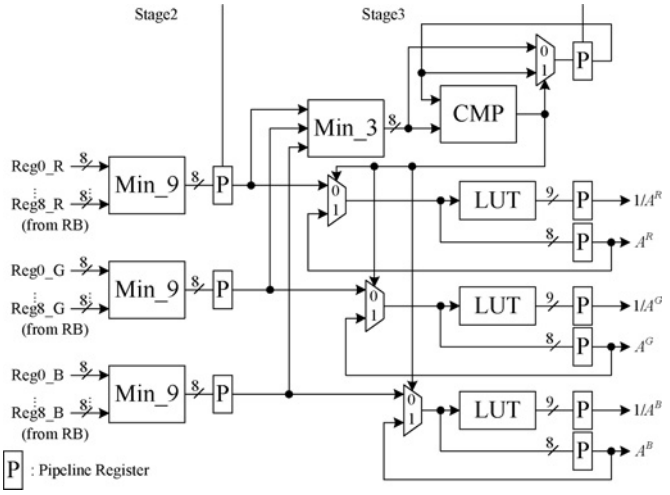

Fig. 3. Architecture of atmospheric light estimation.

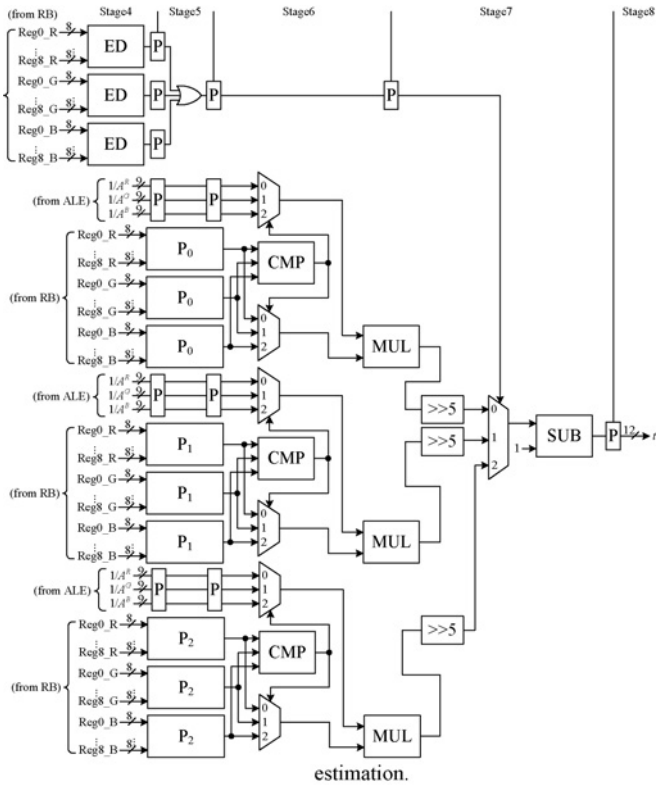

Fig. 4. Architecture of transmission estimation.

The overall block diagram of the hardware architecture for our proposed method is shown in Fig. 2. The architecture is divided into four units: register bank (RB), atmospheric light estimation (ALE), transmission estimation (TE), and scene recovery with saturation correction (SRSC). The RB unit provides nine pixel values of the current 3×3 window for the ALE unit, and the line buffers are used to store the pixel values of two rows in the source image. Since the processes of ALE and TE do not operate concurrently, the clock gating technique is applied to switch ALE and TE for power saving. To improve the performance of our design, the 11-stage pipelined hardware architecture for the haze removal algorithm is presented and implemented as a hardware core. When the start signal is enabled, the hardware core will output the first pixel after 11 clock cycles. Then, it can process one pixel per clock cycle. In total, it would take $(600 \times 400 + 11)$ cycles to process an image with 600×400 pixels. The details of ALE, TE, and SRSC unit are described in the following subsections.

A. ALE Unit

Fig. 3 shows the hardware architecture of ALE where the Min_9 unit determines a minimum value from nine pixel values of the 3×3 window. The Min_3 unit determines a minimum value among three color channels. The CMP unit finds a maximum value as the atmospheric light. The value of $1/Ac$ is first implemented as a look-up table (LUT unit) and then is fed to the TE unit for further execution.

B. TE Unit

Fig. 4 shows the hardware architecture of TE where P_0, P_1, and P_2 units are 3×3 mean filter and two 3×3 edge-preserving filters, respectively. The CMP unit determines the minimum value from three color channels as the candidate of the transmission in the 3×3 window for P_0, P_1, and P_2 unit, respectively. The ED unit detects the edge in a 3×3

TABLE II
COMPARISON OF EXECUTION TIME FOR DIFFERENT METHODS (UNIT: SECOND)

Design	[9]	[10]	[11]		[12]	Proposed	
Processor	Pentium 4	Dual-Core	Dual-Core		Pentium 4	Pentium 4	Dual-Core
Language	N/A	C++	MATLAB	N/A	N/A	C	
Image size	600 × 400	512 × 512	600 × 400		600 × 400	600 × 400	
Execution time	300–420	35	13	0.2	10–20	0.641	0.172

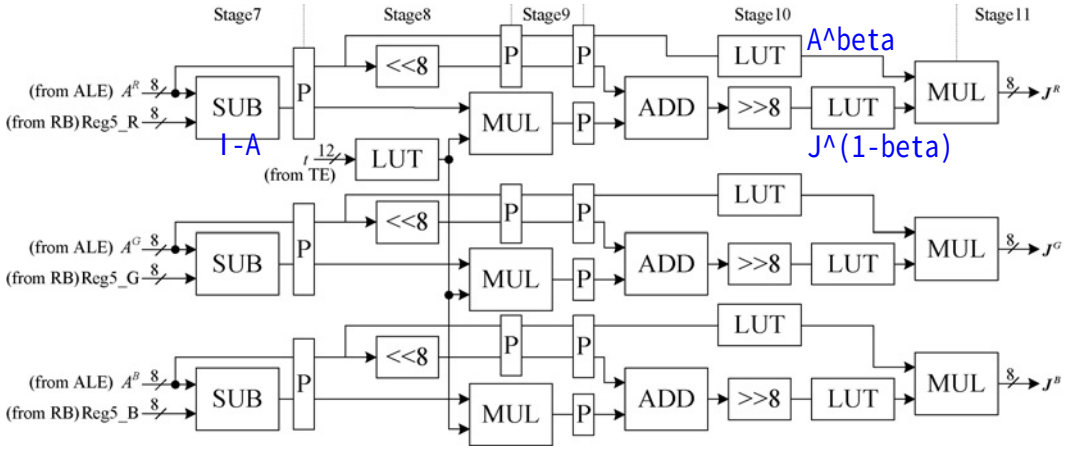

Fig. 5. Architecture of scene recovery with saturation correction.

TABLE III
IMPLEMENTATION RESULTS OF OUR DESIGN

FPGA Device/ ASIC Library	Line Buffer	Total Logic Elements	Flip Flops	Gate Counts	Frequency (MHz)	Throughput (Mpixels/s)
Altera Stratix EP1S10F780C6	6	1607	454	N/A	58.43	58.43
TSMC's 0.13-μm	6	N/A	N/A	12816	200	200

Fig. 6. Simulation results with Tan's work [9] and Tarel's work [11]. (a) Haze image. (b) Tan's result. (c) Tarel's result. (d) Our result.

window and generates the selected signal to determine which filter result is chosen as the transmission.

C. SRSC Unit

Fig. 5 shows the hardware architecture of SRSC. The LUT unit in Stage 8 is a look-up table that provides the value of $1/t$. The LUT units in Stage 10 are applied to obtain the value of $(A^c)^{\beta}$ and $J^c(i, j)^{(1-\beta)}$. Since A^c is already obtained from ALE unit, I^c–A^c can be computed in Stage 7 before the TE unit is finished.

V. SIMULATION RESULTS AND HARDWARE IMPLEMENTATION

To verify the computational complexity, our method was implemented in C language on the 3.0-GHz Pentium 4 and

Fig. 7. Simulation results with Fattal's work [10] and Tarel's work [11]. (a) Haze image. (b) Fattal's result. (c) Tarel's result. (d) Our result.

Fig. 8. Simulation results with Tarel's work [11] and He's work [12]. (a) Haze image. (b) Tarel's result. (c) He's result. (d) Our result.

Dual-Core processors 1024-MB memory, respectively. The source code of [11] implemented in the MATLAB language is provided in the author's website. Table II lists the execution time of these five methods. [11] requires about 13 s since it does not use the compiled MATLAB executable function. Obviously, the proposed method requires a less computing time as compared with other methods [9]–[12].

TABLE IV
POWER CONSUMPTION OF OUR DESIGN WITH DIFFERENT CLOCK PERIOD

Clock Period (ns)	Throughput (Mpixels/s)	Power Consumption (mW)
5	200	11.9
10	100	4.87
20	50	2.41
40	25	1.15
80	12.5	0.58

The hardware architecture of our design is implemented by using Verilog HDL. We use SYNOPSYS Design Vision to synthesize the design with the TSMC 0.13-μm cell library. The layout for the design is generated with SYNOPSYS Astro (for auto placement and routing), and verified by MENTOR GRAPHIC Calibre (for DRC and LVS checks). Finally, SYNOPSYS PrimeTime is employed to measure the total power consumption after auto placement and routing. Synthesis results show that our design contains 12.8K gate counts. It works with a clock period of 5 ns and can achieve a processing rate of 200 Mpixels/s that is fast enough to process a video resolution of QSXGA (2560×2048) at 30 f/s in real time. The power consumption is 11.9 mW with 1.8-V supply voltage. Furthermore, the architecture is also implemented on the Altera FPGA platform for verification. The summary of the implementation results is listed in Table III. To satisfy the different requirement of image applications, we provide the power consumption of our core at different clock period listed in Table IV. In the practical consumer electronics, the processing rate of our circuit can be slowed down to run at low power consumption. Figs. 6–8 show the simulation results compared with Tan's work [9], Fattal's work [10], Tarel's work [11], and He's work [12], respectively. Our design can obtain the comparable results by using the least execution time.

VI. CONCLUSION

In this letter, we presented a fast and efficient haze removal method based on the concept of the dark channel prior. Our method can efficiently avoid the halo artifact generated in the recovered image and obtain the comparable results with the least execution time as compared with previous algorithms. Moreover, we proposed an 11-stage pipelined hardware architecture. It proved to be a good candidate for low-cost haze removal hardware implementation.

REFERENCES

[1] S. G. Narasimhan and S. K. Nayar, "Removing weather effects from monochrome images," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Dec. 2001, pp. II-186–II-193.
[2] S. G. Narasimhan and S. K. Nayar, "Vision and the atmosphere," *Int. J. Comput. Vision*, vol. 48, no. 3, pp. 233–254, Jul. 2002.
[3] S. G. Narasimhan and S. K. Nayar, "Contrast restoration of weather degraded images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 6, pp. 713–724, Jun. 2003.
[4] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, "Instant dehazing of images using polarization," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Dec. 2001, pp. I-325–I-332.
[5] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, "Polarization-based vision through haze," *Appl. Opt.*, vol. 42, no. 3, pp. 511–525, Jan. 2003.
[6] S. Shwartz, E. Namer, and Y. Y. Schechner, "Blind haze separation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2006, pp. 1984–1991.
[7] J. P. Oakley and H. Bu, "Correction of simple contrast loss in color images," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 511–522, Feb. 2007.
[8] J. Kopf, B. Neubert, B. Chen, M. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, and D. Lischinski, "Deep photo: Model-based photograph enhancement and viewing," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 116:1–116:10, Dec. 2008.
[9] R. T. Tan, "Visibility in bad weather from a single image," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2008, pp. 1–8.
[10] R. Fattal, "Single image dehazing," in *Proc. ACM SIGGRAPH*, Aug. 2008, pp. 72:1-72:9.
[11] J.-P. Tarel and N. Hautière, "Fast visibility restoration from a single color or gray level image," in *Proc. IEEE Int. Conf. Comput. Vision*, Sep. 2009, pp. 2201–2208.
[12] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.
[13] W. Middleton, *Vision Through the Atmosphere*. Toronto, ON, Canada: Univ. Toronto Press, 1952.
[14] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. 11th Eur. Conf. Comput. Vis. Part I*, 2010, pp. 1–14.