J Real-Time Image Proc (2019) 16:1117–1128
DOI 10.1007/s11554-016-0603-1

ORIGINAL RESEARCH PAPER

High-speed video haze removal algorithm for embedded systems

Mostafa M. El-Hashash[1] · Hussein A. Aly[1]

Received: 8 September 2015 / Accepted: 17 May 2016 / Published online: 4 June 2016

Abstract In this paper, we implemented a video haze removal system that runs in real time and can be integrated in a vehicle dashboard to assist drivers. We used a heterogeneous TI TMS320DM6446 platform to distribute the algorithmic tasks among ARM, DSP, and VICP cores. Our processing algorithm uses the dark channel prior and has been applied to video. It has two algorithmic components which are the operation on multi-scale resolution and a new reconstruction formula for recovering the scene radiance to save computational time without sacrificing quality. We achieved eight frames per second at 720 × 480 video frame resolution.

Keywords Real-time video processing · Multi-core embedded system · DM6446 · Haze removal · Dark channel prior

1 Introduction

Modern video, audio, and multimedia applications for mobile smart devices and onboard of vehicles that require real-time processing rely on multi-core embedded systems for their high data rate processing capabilities and portability. Near future will bring more occurrences of these systems with the "Internet of Things". In this paper, we show how to remove haze, a phenomenon which degrades visibility, from videos. It is caused by particles or water droplets in the atmosphere that absorb and scatter the light that travels from the source to the observer as stated by McCartney [16]. Haze affects drivers as it changes the colors of the objects and attenuates the scene radiance. Moreover, haze affects the performance of autonomous vehicles which depend on detecting objects in the scene such as the road lanes. Thus, removing haze from videos will have a great impact on enhancing the performance of these systems and will provide a great help for drivers as well. However, the implementation of haze removal algorithms requires many calculations and large memory. These come as barriers to real-time implementation on embedded systems.

In this paper, the TMS320DM6446 multi-core digital media system on chip (DMSoC) [26] is used to provide an onboard portability for the embedded system and high speed in removing haze from videos. The TMS320DM6446 consists of ARM926EJ-S core as a general purpose processor (GPP), TMS320C64x+ as a digital signal processor (DSP) core, and a video and image coprocessor (VICP).

The most commonly used haze imaging model, also called image degradation model, proposed by McCartney [16] is:

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

where $\mathbf{I}$ is the observed scene, $\mathbf{J}$ is the scene radiance, t is the medium transmission, $\mathbf{A}$ is the atmospheric light, and $\mathbf{x} = (x, y)$ are the horizontal and vertical positions of the pixel. The term $\mathbf{J}(\mathbf{x})t(\mathbf{x})$ is the direct attenuation of the scene radiance, and the term $\mathbf{A}(1 - t(\mathbf{x}))$ is the added atmospheric light which distorts scene colors. Haze removal problem aims at recovering $\mathbf{J}$ given only the observation $\mathbf{I}$ blindly with the absence of knowledge about

✉ Mostafa M. El-Hashash
m.el-hashash@hotmail.com

Hussein A. Aly
haly@ieee.org

[1] Computer Department, Military Technical College, Cairo, Egypt

both **A** and t. Therefore, solving Eq. (1) is an ill-posed inverse problem according to Hadamard [10].

The main challenge lies in the ambiguity of the problem as the number of equations is less than the number of unknowns. Though, there are an infinite number of solutions given only the degraded or hazy image **I**. There are two main approaches to solve this ambiguity by either acquiring more knowledge about the scene or imposing *a priori* constraints. For instance, some haze removal algorithms capture multiple images of the same scene under different settings to assist the recovery process such as the algorithms proposed by Narasimhan and Nayar [17–21], Shwartz et al. [22], Kopf et al. [14], and Feng et al. [9]. However, it is difficult to obtain extra images in practice in many consumer video cameras.

The second approach is to impose extra constraints using knowledge or assumptions which are called priors such as the algorithms proposed by Tan [23], Fattal [8], He et al [11], and Ke and Chen [12]; thus, practically we only require one image. Generally, a prior can be a statistical property, rule, or heuristic assumption. The quality of the results is often determined by the extent to which the prior is valid.

Wu et al. [28] have surveyed haze removal algorithms. Among the different methods, single image haze removal without additional imaging hardware can operate using most consumer video cameras. A comparison of the execution time for these methods, as reported in their papers, is shown in Table 1. The methods proposed by He et al. [11] and Ke and Chen [12] are based on the dark channel prior that have lower execution time compared to the other methods.

We rely our work on the dark channel prior because it is fast and effective in most scenes. However, it may not work for some particular cases such as saturated objects or objects which are inherently similar to the atmospheric light and without shadow. Our focus is to design a fast algorithm to present a real-time haze removal system.

Previous work based on the dark channel proposed by Khodary and Aly [18] and El-Hashash et al. [12] developed haze removal for video on TMS320DM6446 platform and achieved 0.5 and 3 frames per second (fps) respectively for video frames of 720 × 480 pixels. The idea of multi scale resolution was introduced in [7] to expedite computations. In this paper, we made the processing across resolution scales and propose a fast reconstruction equation. We added clear analysis from a signal processing perspective to justify our algorithm.

This paper is organized as follows: The dark channel prior-based algorithms are reviewed in Sect. 2. In Sect. 3, our algorithm using the dark channel prior and its implementation technique on the target platform are presented. In Sect. 4, experimental results and performance evaluation are demonstrated. Finally, the paper is concluded in Sect. 5.

2 Background

The dark channel prior-based algorithms are reviewed in Sect. 2.1 and the implementation technique for the target platform is presented in Sect. 2.2.

2.1 Dark channel prior-based algorithms

He et al. [11] defines the dark channel as:

$$J^{\text{dark}}(\mathbf{x}) = \min_{c\in\{r,g,b\}}\left(\min_{\mathbf{y}\in\Omega(\mathbf{x})}(I^{c}(\mathbf{y}))\right), \tag{2}$$

where J^{dark} is the estimated dark channel of **I**, I^c is a color channel (red, green, blue) of **I**, and $\Omega(\mathbf{x})$ is a local patch around the pixel at **x**. According to He et al. [11], the dark channel is almost zero for haze-free images as shown in Fig. 1, where we have the dark channel for two images: a hazy and a haze-free one. In Fig. 1a, b, it can be seen that the objects that are clearly visible in **I** have zero-valued dark channel such as the rocks in the front. The atmospheric light is computed using J^{dark} by finding pixels with the highest 0.1 % values as in Eq. (3). These pixels are usually most haze opaque. Some haze removal methods estimate the airlight by selecting the pixel that has the maximum intensity, which is not valid in many scenes (for example, if there is an object that has higher intensity than the airlight). The indices of the highest 0.1 % pixels are stored in **m** as follows:

$$\mathbf{m} = \arg\max_{\tilde{\mathbf{x}}\ \ 0.1\%}(J^{\text{dark}}(\tilde{\mathbf{x}})). \tag{3}$$

Then, the sum of the three color channel values in $\mathbf{I}(\mathbf{m})$ is calculated. Finally, the pixel corresponding to the maximum sum is selected as **A** as follows:

Table 1 Execution times of different single image haze removal methods on personal computers

Method	Resolution	CPU	Clock speed	Time	Pixel/second
Tan [23]	600 × 400	Double processors of Pentium 4	N/A	5–7 min	571–800
Fattal [8]	512 × 512	Intel Pentium Dual Core	1.6 GHz	35 s	7490
He et al. [11]	600 × 400	Intel Pentium 4	3.0 GHz	10–20 s	12,000–24,000
Ke and Chen [12]	600 × 400	Intel Core i5	3.2 GHz	760 ms	315,789

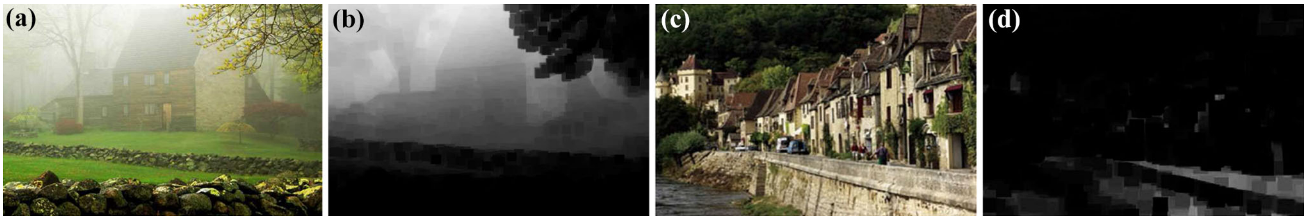


Fig. 1 **a** Hazy image and its dark channel in **b**, **c** haze-free image and its dark channel in **d**

$$\mathbf{A} = \arg\max_{\tilde{\mathbf{I}}} \left(\sum_{c \in \{r,g,b\}} \tilde{I}^c(\mathbf{m}) \right). \tag{4}$$

Using **I** and the estimated **A**, the normalized dark channel is defined by:

$$J_N^{\text{dark}}(\mathbf{x}) = \min_{c \in \{r,g,b\}} \left(\frac{I^c(\mathbf{x})}{A^c} \right), \tag{5}$$

then the coarse transmission $\tilde{t}$ is assumed to be:

$$\tilde{t}(\mathbf{x}) = 1 - (\omega \times \min_{\mathbf{y} \in \Omega(\mathbf{x})} (J_N^{\text{dark}}(\mathbf{y}))), \tag{6}$$

where ω is the aerial perspective factor and is set to 0.95 in He et al. [11]. The transmission map should be object based at specific scene depth. There are two approaches to obtain an object-based transmission map. One way was performed by He et al. [11] where they assumed the objects to be patches (15 × 15) followed by removing the blocking artifacts by edge-directed smoothing operation (soft matting method by Levin et al. [15]). Alternatively, Ke and Chen [12] approached the problem starting from dense field based on pixels which results in noisy incoherent transmission map that is smoothed by a moving average filter. Both approaches provide almost the same result that is called refined (enhanced) transmission map t in [11]. The enhanced transmission t and **A** are used to recover **J** as:

$$\mathbf{J}(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x}) - \mathbf{A}}{\max(t(\mathbf{x}), t_0)} + \mathbf{A}, \tag{7}$$

where t_0 is set to 0.1 in He et al. [11] to avoid division by zero for areas where the value of the transmission map is below 0.1. As from Eq. (1), when $t \to 0$, $\mathbf{I} \to \mathbf{A}$ and in this case it is very hard to reconstruct **J** because it is severely attenuated by the small value of t. Hence, in the reconstruction they stop further amplification of **A** by setting t_0 to 0.1.

Soft matting requires extensive computations and memory, and thus Ke and Chen [12] calculated the dark channel by only selecting the value of the minimum color channel for the pixel. They used Eqs. (3) and (4) to estimate **A**, then Eqs. (2) and (6) in [12] became:

$$J^{\text{dark}}(\mathbf{x}) = \min_{c \in \{r,g,b\}} (I^c(\mathbf{x})), \tag{8}$$

$$\tilde{t}(\mathbf{x}) = 1 - (\omega \times J_N^{\text{dark}}(\mathbf{x})). \tag{9}$$

A smoothing operation is used instead of soft matting and performed for $\tilde{t}$ by a moving average filter to obtain t_s reducing the computational cost. Finally, a gamma correction function is applied to the smoothed transmission to avoid over haze removal and preserve a natural look for the scene:

$$t(\mathbf{x}) = (t_s(\mathbf{x}))^{\gamma}, \tag{10}$$

where Ke and Chen [12] set γ to 0.6 to enhance the contrast of the scene. A sample resulting transmission map using [12] is shown in Fig. 2b which looks very comparable to Fig. 2a of He et al. [11] but at reduced computations and without large memory requirements.

2.2 Implementation technique for the DM6446

The main cores of the TMS320DM6446 are the ARM, DSP, and VICP. The ARM9 GPP core at 300 MHz handles tasks related to both the peripherals and the control

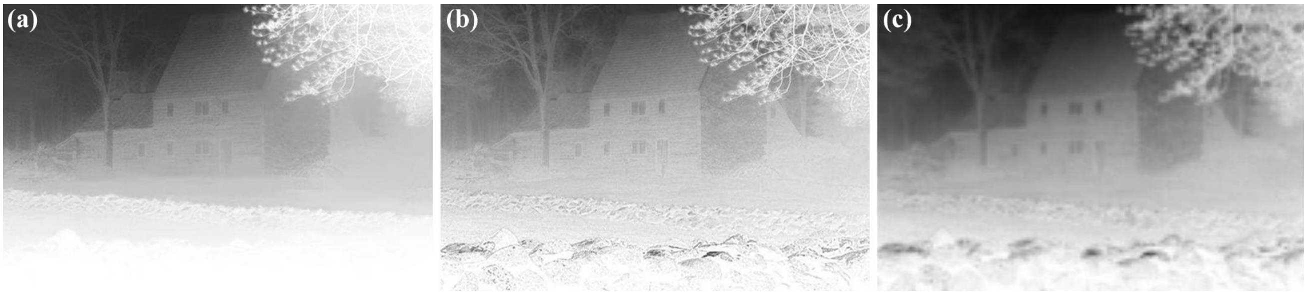


Fig. 2 Transmission map of image in Fig. 1a using different haze removal methods: **a** using [11], **b** using [12], **c** using our algorithm

application and uses embedded Linux as a real-time operating system (RTOS). The C64x+ DSP core at 600 MHz handles vector processing (such as convolution) operations efficiently. The VICP hardware accelerator at 405 MHz is effective in enhancing DSP performance. VICP takes over the execution of varied computationally intensive tasks [26] that have multi-input multi-output nature such as color conversions. The DSP core uses DSP/BIOS [24] which is a RTOS provided by TI, and runs the desired algorithm. The ARM and DSP cores communicate with each other via DSP/BIOS Link [6] on shared internal bus as shown in Fig. 3. Implementation of applications on this heterogeneous system is performed using the codec engine framework [4], which is a set of APIs used to instantiate and run eXpress DSP algorithm interoperability standard (XDAIS) [25]. To support digital media encoders, decoders, and codec extensions for video, imaging, speech, and audio (VISA), codec classes provide the XDAIS-DM interface [29] which specifies APIs for digital media codecs. For instance, the DSP and the VICP can run the haze removal algorithm which should meet the XDAIS-DM before it is instantiated and executed by the ARM using the VISA APIs. The block diagram for the DM6446 is detailed in Fig. 3.

3 Haze removal system

We describe our haze removal algorithm in Sect. 3.1, while in Sect. 3.2, we provide our software architecture for the algorithmic tasks distributed among the different cores.

3.1 Implemented haze removal algorithm

We implemented Ke and Chen [12] algorithm on DM6446 where the execution time for each task is shown in Table 2. Equations (9) and (10) are the bottleneck tasks and consume 93 % of the total time. The DSP core TMS320C64x+ has a fixed point architecture. Floating point operations will generally deteriorate the system performance (frames per second (fps)). Hence, we focus on redesigning Eqs. (9) and (10) by our new method as shown later in Eq. (15). The implemented algorithm

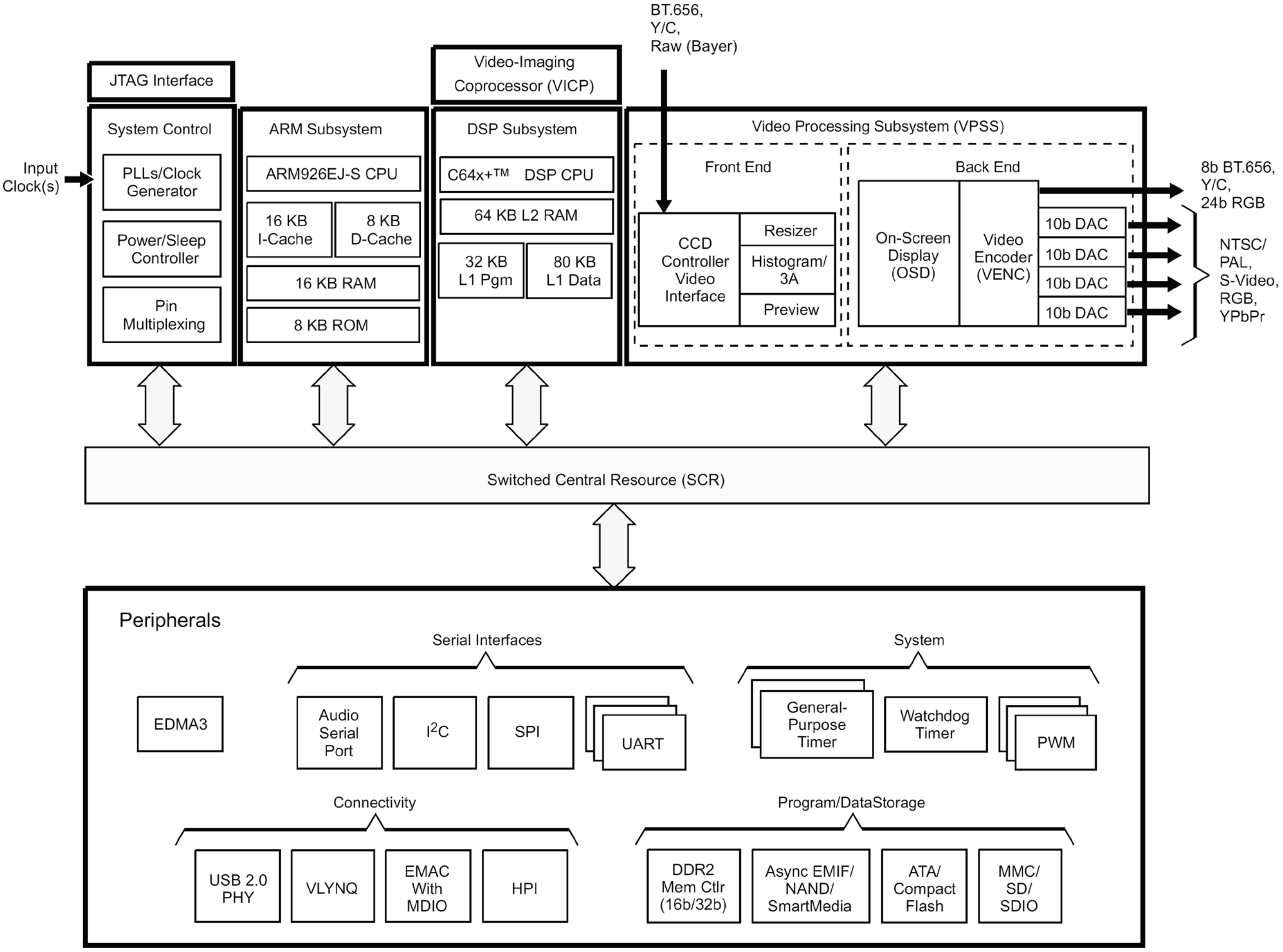


Fig. 3 TMS320DM6446 functional block diagram [26]

Table 2 Time profiling for Ke and Chen algorithm [12]

Task	Execution time (ms)
Convert from YC_bC_r to RGB	12
Find the dark channel J^{dark} (8)	12
Find airlight $\mathbf{A}$ (3) and (4)	10
Find normalized dark channel J_N^{dark} (5)	12
Find the coarse transmission map $\tilde{t}$ (9)	430
Apply moving average filter to obtain t_s filter	4
Apply gamma correction function to obtain t (10)	1433
Recover the scene radiance $\mathbf{J}$ (7)	75
Convert from RGB to YC_bC_r	12
Total	2000

consists of two components; the first is a new technique to replace Eq. (9) and the smoothing operation using the moving average filter by Ke and Chen [12]. The second component provides a new approach to recover the scene radiance.

Regarding the first component, we need to reduce the execution time consumed by Eq. (9) and a good smoothing filter for $\tilde{t}$ instead of the moving average filter used by Ke and Chen [12], specifically, a filter with minimum ripples in the stop band. Several candidate filters with large taps could be used. However, the vectorized multiply and add instruction of the DSP cannot be efficiently used as the number of coefficients to be loaded are limited. Additionally, filtering with moving average or bilinear filters have optimized hardware implementation on the DSP of the DM6446. We use the separable bilinear filter twice. The first use is as an anti aliasing filter $h_a(\mathbf{x})$ in a downsampling of J_N^{dark} by a factor d in each dimension to obtain $J_N^{\text{dark}\downarrow}$. The second use is as a bilinear interpolation filter $h_i(\mathbf{x})$ in a subsequent upsampling of $J_N^{\text{dark}\downarrow}$ by a factor d in each dimension to obtain the smoothed normalized dark channel J_{Ns}^{dark}. Although this idea was presented in [7], we give in this paper detailed justification. Without loss of generality, we can use a simple filter such as bilinear for both h_a and h_i.

To compare between the smoothing using moving average filter $h_b(\mathbf{x})$ by Ke and Chen [12] and our method using only the downsampling followed by upsampling, we cast our analysis in the frequency domain where $T_1(\mathbf{u})$ and $T_2(\mathbf{u})$ are the discrete Fourier transform of the smoothed transmission maps of the method by Ke and Chen [12] and ours, respectively. They can be written as:

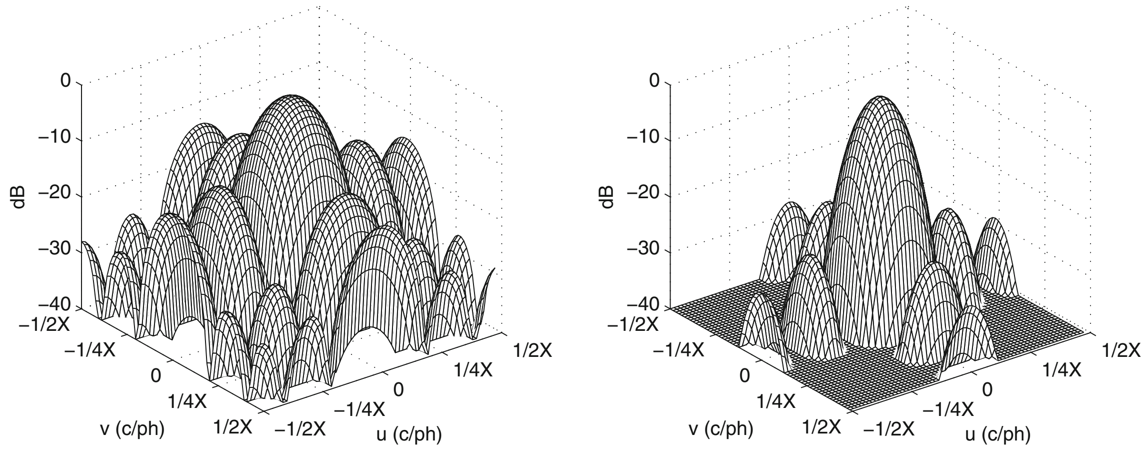
$$T_1(\mathbf{u}) = (H_b\tilde{T})(\mathbf{u}), \tag{11}$$

where $\tilde{T}$ is the fourier transform of the coarse transmission map which is smoothed by a moving average filter whose frequency response is $H_b(\mathbf{u})$:

$$H_b(\mathbf{u}) = \frac{\sin(\pi ud)}{d\sin(\pi u)}\frac{\sin(\pi vd)}{d\sin(\pi v)}, \tag{12}$$

where the vector $\mathbf{u} = [uv]^{\mathrm{T}}$ has the horizontal and vertical frequencies, and d is the size of the moving average and can determine the cut-off frequency $\mathbf{u}_c = [\pm\frac{1}{2d}\pm\frac{1}{2d}]^{\mathrm{T}}$. In our algorithm, as will be described later, we do not need to calculate the transmission map explicitly. Our implicit smoothing of $\tilde{t}$ using downsampling followed by upsampling by factor d in each dimension can be written as:

$$\begin{aligned} T_2(\mathbf{u}) &= H_i(\mathbf{u})\sum_{k=0}^{d^2-1}(H_a\tilde{T})(\mathbf{u}+\mathbf{s}_k) \\ &= (H_aH_i\tilde{T})(\mathbf{u}) + H_i(\mathbf{u})\sum_{k=1}^{d^2-1}(H_a\tilde{T})(\mathbf{u}+\mathbf{s}_k), \end{aligned} \tag{13}$$



(a) **(b)**

Fig. 4 Frequency response of H_b: **a** of a 5 × 5 moving average filter in [12] **b** corresponding bilinear filter

where $\mathbf{s}_k$ are the coset representatives of the high-resolution reciprocal lattice in the lower resolution reciprocal lattice as detailed by Aly and Dubois [1]. $k = 1, 2, \ldots, d^2 - 1$ represents the aliasing effect due to downsampling in the frequency domain. The first term is an implicit smoothing filter whose frequency response is $H_a H_i(\mathbf{u})$, while the second is the aliasing components.

Since H_b, H_a, and H_i are approximations of a low-pass filters with cut-off frequency at $\mathbf{u}_c = [\pm\frac{1}{2d} \pm \frac{1}{2d}]^{\mathrm{T}}$, then the product $H_a H_i(\mathbf{u})$ in Eq. (13) is considered to be an approximation of a low-pass filter as well. Illustrations using moving average and bilinear filters for H_a, H_i are shown in Fig. 4a, b, and the product $H_a H_i(\mathbf{u})$ in Fig. 5a, b for the moving average and bilinear prototype H_a and H_i. It can be seen that at the pass band $\mathbf{u}_p \in [-\frac{1}{2d}, \frac{1}{2d}]^2$, the frequency response of $H_a H_i(\mathbf{u})$ is similar to a smoothing filter such as $H_b(\mathbf{u})$. At the stop band $\mathbf{u}_s \in [-\frac{1}{2}, \frac{1}{2}]^2 - \mathbf{u}_p$, we see that $H_a H_i(\mathbf{u})$ has no ripples or attenuated ones as compared to $H_b(\mathbf{u})$ in both cases of moving average and bilinear filters. This makes the implicit filter $H_a H_i(\mathbf{u})$, a good smoothing filter better than $H_b(\mathbf{u})$. For sake of comparison only with the other methods, we apply Eq. (9) and our implicit transmission map is shown in Fig. 2c which appears to be smooth in regions with the same scene depth without ringing around edges.

The second component of our algorithm deals with the reconstruction in Eq. (7) and the preceding steps in Eqs. (9) and (10). The three equations involve massive floating point operations. We designed a new method to recover the scene. We substitute the cumulative effect of $\frac{1}{(1-\omega h_b * J_N^{\mathrm{dark}})^\gamma}$ in Ke and Chen [12], where * denotes convolution, by a first-order polynomial approximation $1 + KJ_{Ns}^{\mathrm{dark}}$ as follows:

$$\mathbf{J}(\mathbf{x}) = (\mathbf{I}(\mathbf{x}) - \mathbf{A})(1 + KJ_{Ns}^{\mathrm{dark}}(\mathbf{x})) + \mathbf{A}, \qquad (14)$$

where K is a constant. Thus, the scene radiance $\mathbf{J}$ is recovered by:

$$\mathbf{J}(\mathbf{x}) = \mathbf{I}(\mathbf{x}) + (\mathbf{I}(\mathbf{x}) - \mathbf{A})(KJ_{Ns}^{\mathrm{dark}}(\mathbf{x})). \qquad (15)$$

This is illustrated in Fig. 6a where we plotted the two functions ($\frac{1}{(1-\omega h_b * J_N^{\mathrm{dark}})^\gamma}$ and $1 + KJ_{Ns}^{\mathrm{dark}}$). We empirically found the best estimate for the factor K in Eq. (15) to be 1.5. We focus on the interval where the values for $J_{Ns}^{\mathrm{dark}} \in [0.2, 0.6]$ as this interval contain the majority of pixels values in J_{Ns}^{dark} as shown in the histogram in Fig. 6b, c for J_{Ns}^{dark} for two hazy images (Forest and Train).

This reconstruction equation reduces the computational cost of the algorithm as there is no need for calculating the transmission map or the gamma correction function, and the scene is recovered directly using J_{Ns}^{dark}.

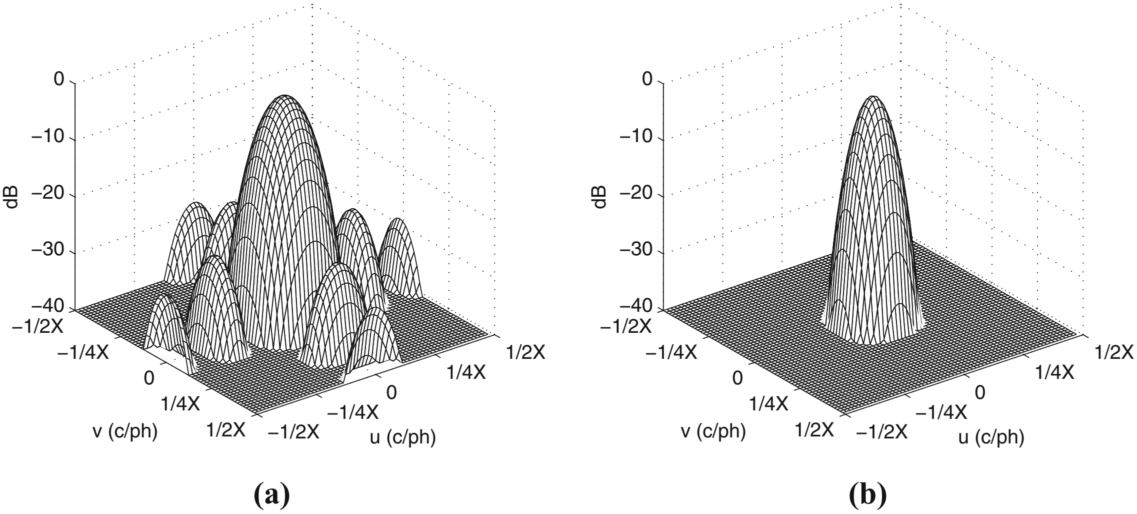
Thus, our haze removal algorithm is as follows:

Algorithm 1 Our haze removal algorithm

```
Find the dark channel $J^{dark}$ (8)
Find the air light $\mathbf{A}$ (3) and (4)
Find the normalized dark channel $J_N^{dark}$ (5)
Downsample the normalized dark channel $J_N^{dark\downarrow}$
Upsample the normalized dark channel $J_{Ns}^{dark}$
Recover the scene radiance $\mathbf{J}$ (15)
```

3.2 Implemented software architecture on TMS320DM6446

We analyzed the tasks of our haze removal algorithm and matched each with a suitable core (ARM, DSP, VICP). This matching was based on the nature of each core. For



(a) **(b)**

Fig. 5 Frequency response of the implicit filter $H_a H_i$: **a** using moving average for both H_a, H_i **b** using bilinear filter for both H_a, H_i

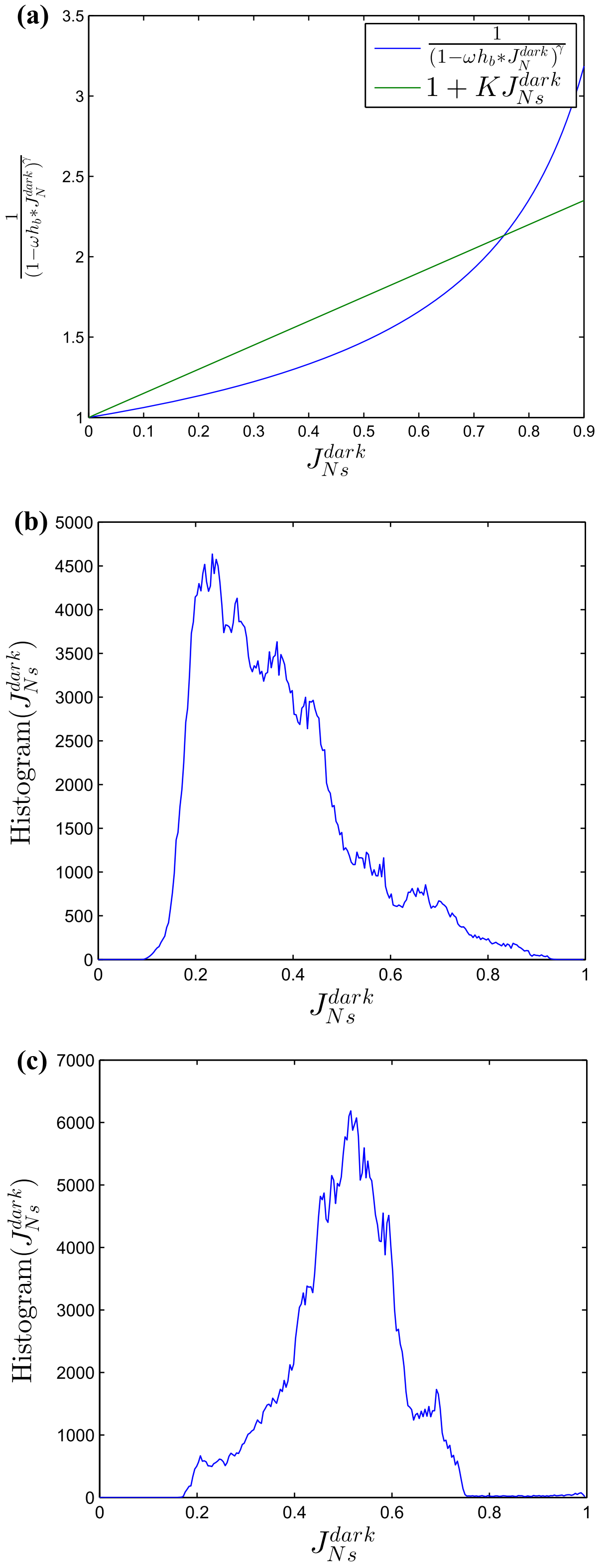


Fig. 6 **a** Scene reconstruction functions ($\frac{1}{(1-\omega h_b * J_N^{\text{dark}})^{\gamma}}$), ($1 + KJ_{Ns}^{\text{dark}}$), $K = 1.5$, **b**, **c** histogram for J_{Ns}^{dark} for Forest and Train hazy images, respectively

instance, VICP has multi-input multi-output (MIMO) nature thus color conversions $RGB \longleftrightarrow YC_bC_r$ are assigned to it. This color conversion is necessary as video data is presented in the NTSC by the TVP5146 decoder in YC_bC_r at 4:2:2 format, while our haze removal algorithm based on the dark channel operates on the *RGB* format. On the other hand, DSP has vector processing nature so convolution tasks such as downsampling are directed to it. The complete list of tasks and core assignments are shown in Table 3.

The implemented software architecture, shown in Fig. 7, is as follows: An application runs on the ARM9 core that has two threads. The first thread is the control thread that deals with the peripherals through the input/output drivers. The second one is the haze removal thread that creates, processes, controls, and deletes the haze removal algorithm as a codec through the VISA interface [3]. The DSP core runs the haze removal codec that is implemented as XDAIS-DM [4] and a codec engine server [5] configured to make it available for the ARM application. The DSP core initializes and controls the VICP using the VICP signal processing library [27]. Furthermore, fixed point data type is used instead of floating point data without affecting the accuracy as we still use four bytes for data representation, and the division operations are converted into multiplications as stated by Bovik [2]. Moreover, shortcut arithmetic operations are used to minimize memory movements.[1] The implemented software architecture of the algorithmic blocks distributed among the multi-cores is detailed in Fig. 7.

4 Experimental results and performance evaluation

Since implementation of the method by He et al. [11] on TMS320DM6446 is not feasible, due to soft matting requirements, and in order to maintain a benchmark for comparisons of both quality and run time, we present two sets of results. First in Sect. 4.1, results for single images on a personal computer (PC) compared to the methods by He et al. [11] and Ke and Chen [12]. Then, in Sect. 4.2 we present our video execution time compared to the methods by Ke and Chen [12], Khodary and Aly [13], and El-Hashash et al. [7] on TMS320DM6446.

4.1 Benchmark results on personal computer

The implementation of the methods by He et al. [11], Ke and Chen [12], and our algorithm has been made using

[1] As in Eq. (15) implemented in C language as $\mathbf{I}+=(\mathbf{I}-\mathbf{A})(KJ_{Ns}^{\text{dark}})$, where $\mathbf{I}$ at the L.H.S represents $\mathbf{J}$.

Table 3 Distribution of our algorithmic tasks among different cores

Task	ARM	DSP	VICP	Execution time (ms)	Complexity	Bound
Get video frame	✓					I/O bound
Convert from YC_bC_r to RGB			✓	12	$O(n)$	Cache bound
Find the dark channel J^{dark} (8)			✓	12	$O(n)$	Cache bound
Find the airlight **A** (3) and (4)		✓		10	$O(n)$	Cache bound
Find the normalized dark channel J_N^{dark} (5)			✓	12	$O(n)$	Cache bound
Downsample normalized dark channel $J_N^{\text{dark}\downarrow}$		✓		6	$O(n)$	Cache bound
Upsample normalized dark channel J_{Ns}^{dark}		✓		6	$O(n)$	Cache bound
Recover the scene radiance J (15)		✓		54	$O(n)$	Cache bound
Convert from RGB to YC_bC_r			✓	12	$O(n)$	Cache bound
Display video frame	✓					I/O bound
Total				124		

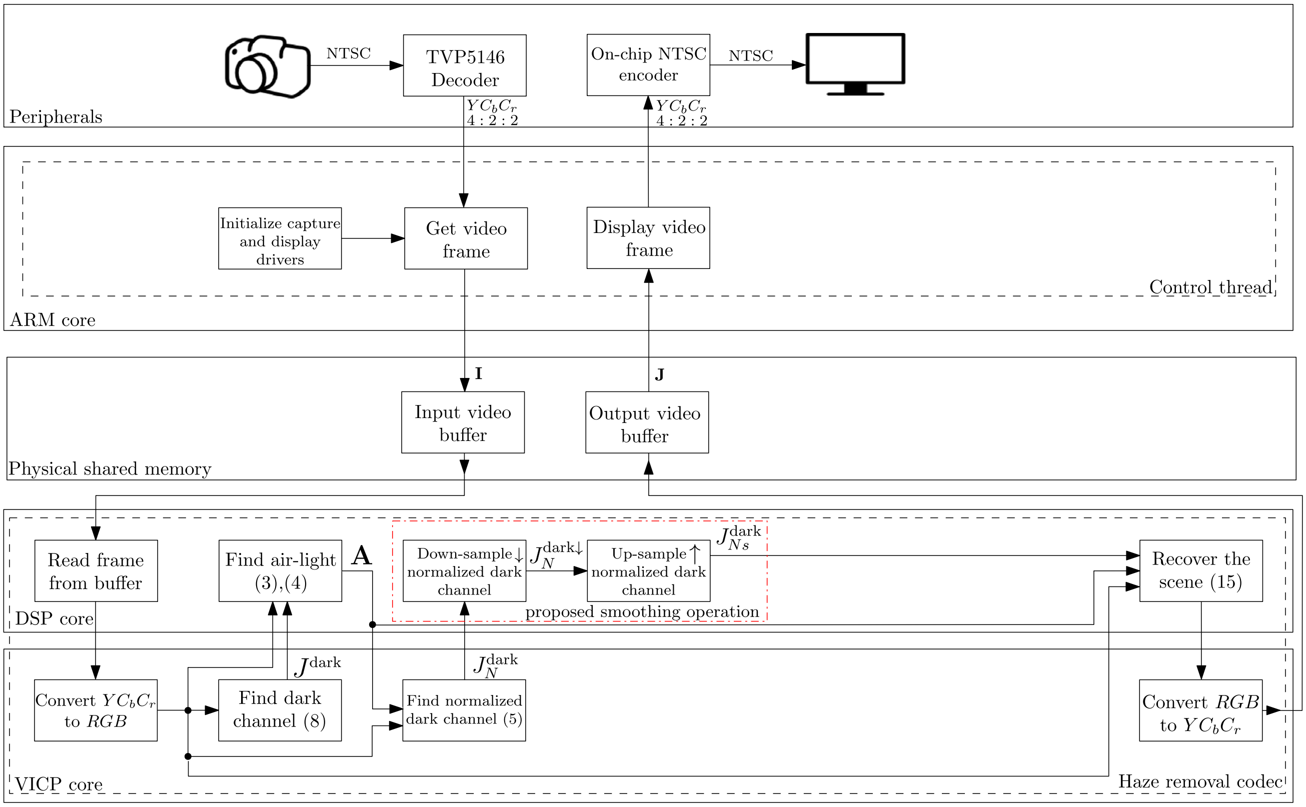


Fig. 7 The implemented haze removal software architecture on the target heterogeneous multi-cores

MATLAB 8.3 on core i5 at 2.27 GHz PC with 4GB RAM using 720 × 480 images. Results for samples of the used test data set are shown in Fig. 8. The execution times for sample images are calculated and shown in Table 4. The Sobel edge detection for the sample hazy images and the images with removed haze are shown in Fig. 9. The reconstructed images contains more edges than the hazy ones.

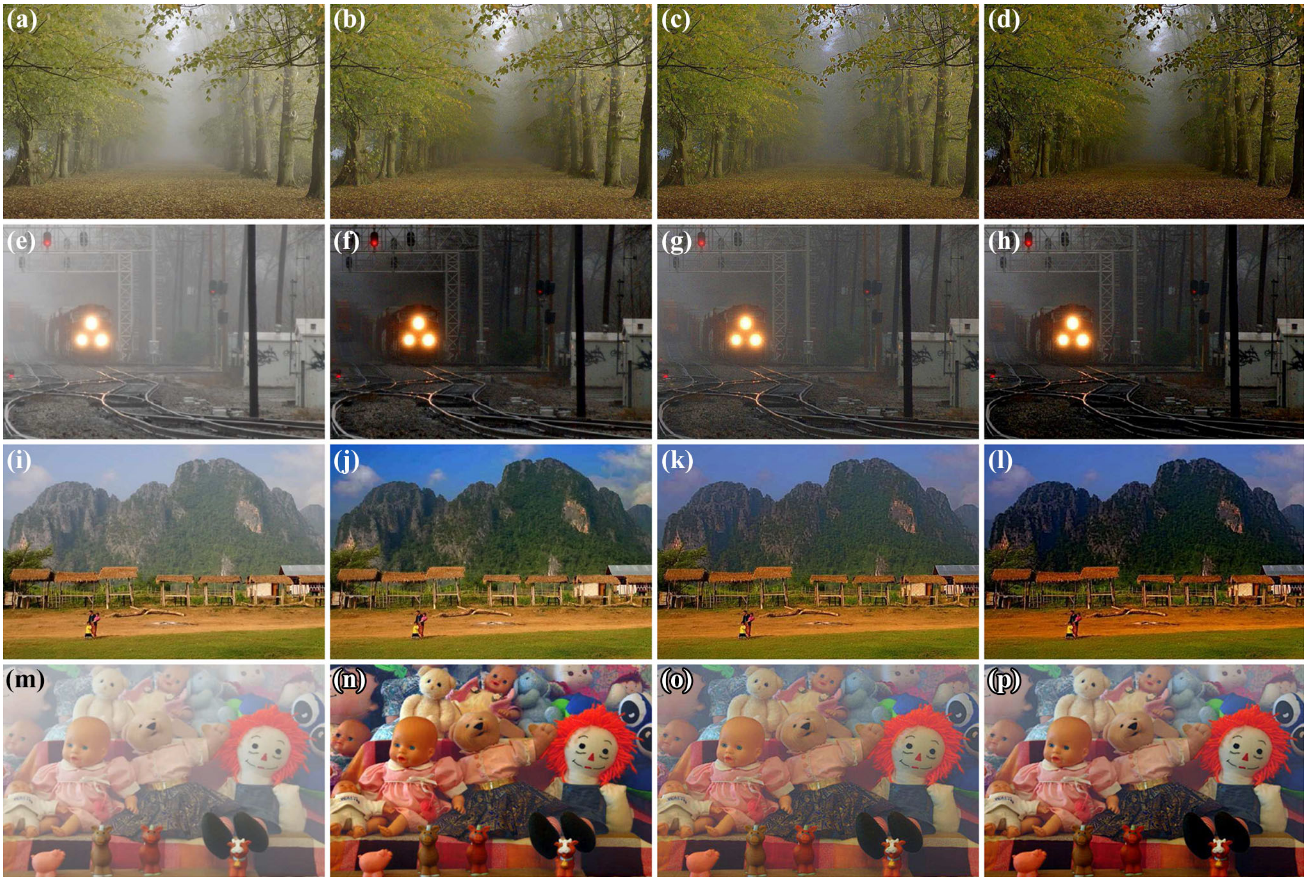


Fig. 8 *First column* hazy image **a** Forest, **e** Train, **i** Mountain, **m** Toys, *second column* **b**, **f**, **j**, **n** haze removal using He et al. [11], *third column* **c**, **g**, **k**, **o** haze removal using Ke and Chen [12], *fourth column*: **d**, **h**, **l**, **p** haze removal using our algorithm

Table 4 Execution times in seconds for images in Fig. 8 all of size 720 × 480 using different haze removal algorithms on personal computer (best value in bold)

Image	He et al. [11]	Ke and Chen [12]	Ours
Forest	69.5	1.2	**0.9**
Train	69.1	1.2	**0.9**
Mountain	70.4	1.3	**0.9**
Toys	71.2	1.2	**0.9**

Best values are shown in bold

Our algorithm achieves the lowest execution time compared to the methods in [11, 12] on the PC. Moreover, our algorithm provides better haze removal visual results in most scenes compared to the other methods.

4.2 Performance of the implemented algorithm under embedded environment

We tested the method of Khodary and Aly [13], El-Hashash et al. [7], and additionally implemented the method of Ke and Chen [12] on TMS320DM6446 to provide comparisons. The used input video stream on the target platform was NTSC for all three methods. We have deployed the system onboard and test-driven it in hazy weather, and sample results are shown in Fig. 10. Areas of heavy haze are reconstructed by our algorithm without spurious colors as in [12, 13]. The implemented system provides a 16-fold increase in frame rate compared to [12, 13] and approximately a three fold increase compared to [7] as shown in Table 5 and achieving a high speed of 8 fps.

5 Conclusions

In this paper, a video haze removal system on heterogeneous multi-cores has been implemented. The system uses a newly designed haze removal algorithm based on the dark channel prior. Our algorithm has the lowest execution time compared to the related work in literature while maintaining the haze removal image quality. Our algorithm is developed on the DM6446 DMSoC, and the algorithmic

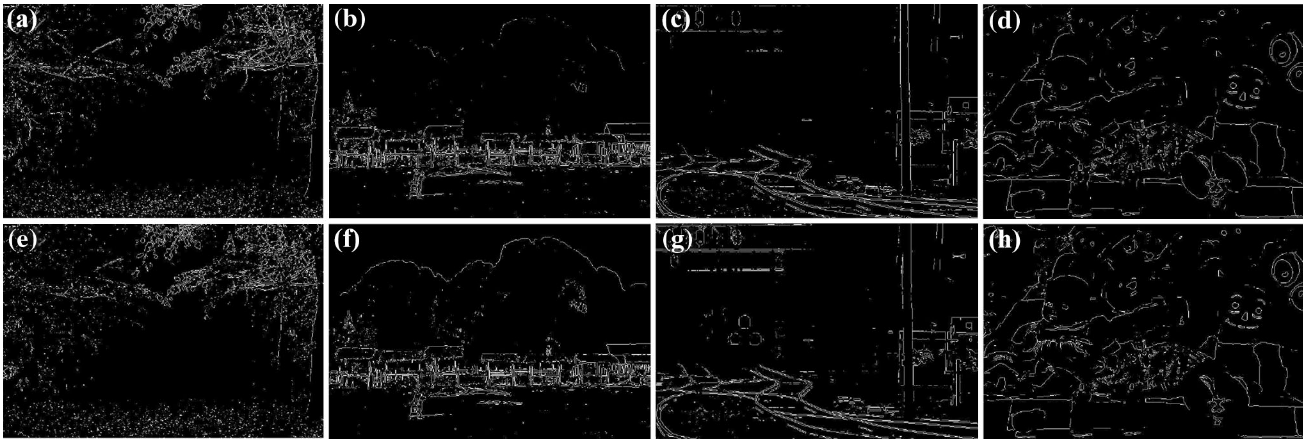


Fig. 9 *First row* **a**, **b**, **c**, **d** sobel edge detection for hazy images, *second row* **e**, **f**, **g**, **h** sobel edge detection for images with haze removed using our algorithm

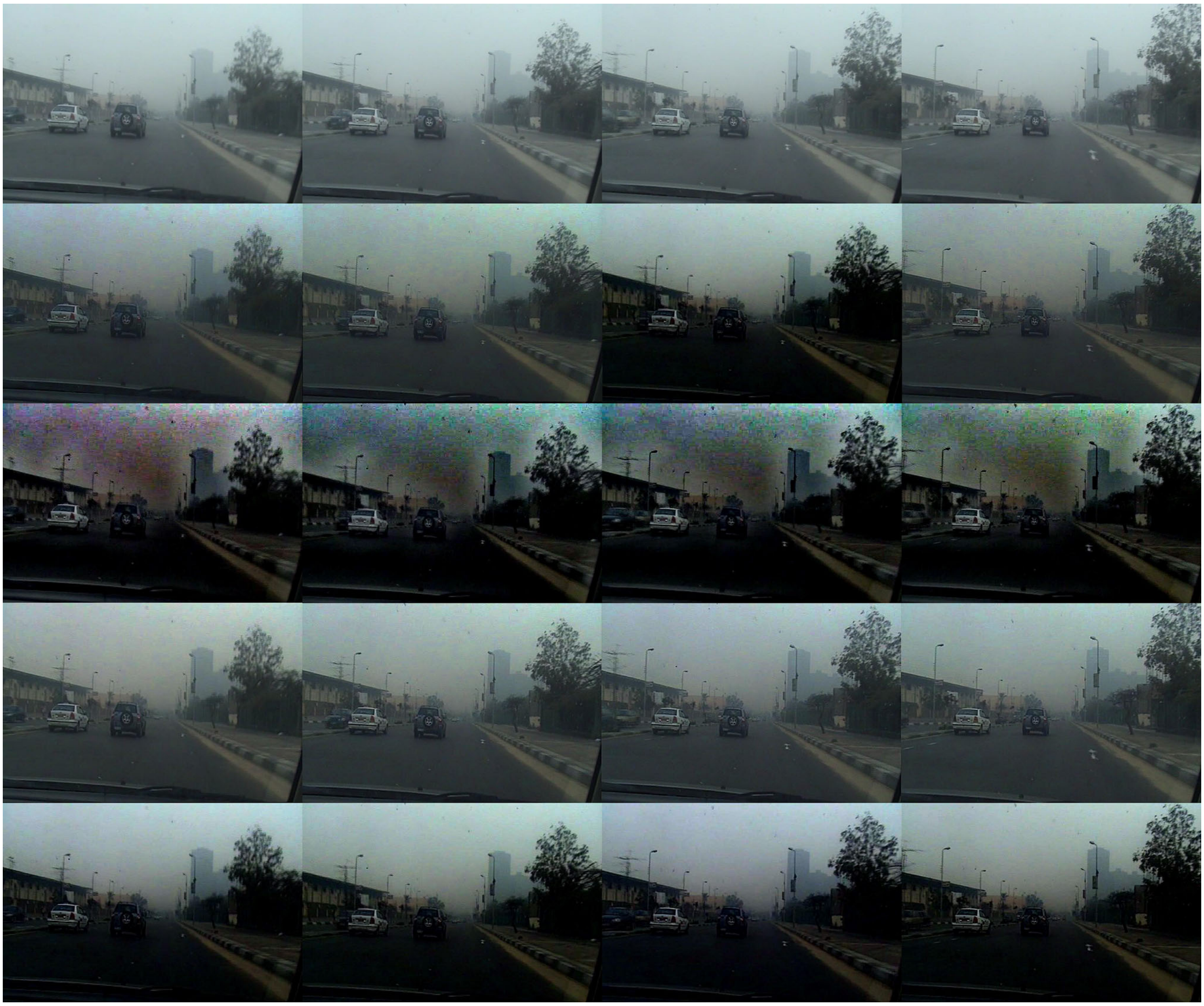


Fig. 10 *First row* hazy video frames, *second row* haze removal using Ke and Chen [12], *third row* haze removal using Khodary and Aly [13], *fourth row* haze removal using El-Hashash et al. [7], *fifth row* haze removal using our system onboard

Table 5 Frame rate (fps) of different haze removal systems on embedded platform (best value in bold)

System	Frame rate
Ke and Chen [12]	0.5
Khodary and Aly [13]	0.58
El-Hashash et al. [7]	3
Our system	**8**

blocks (tasks) were distributed among the ARM, DSP, and VICP cores. The time consuming bottleneck tasks were analyzed and redesigned in our algorithm. This provided us with 8 fps for full TV resolution (720 × 480). This was achieved by using the two components of the smoothing operation at lower resolution scale and the new reconstruction formula.

References

1. Aly, H.A., Dubois, E.: Specification of the observation model for regularized image up-sampling. IEEE Trans. Image Process. **14**(5), 567–576 (2005). doi:10.1109/TIP.2005.846019
2. Bovik, A.: The Essential Guide to Video Processing. Academic Press, London (2009)
3. Codec engine application developer user's guide. Technical Report SPRUE67, Texas Instruments (2006)
4. Codec engine algorithm creator user's guide. Technical Report SPRUED6C, Texas Instruments (2007)
5. Codec engine server integrator user's guide. Technical Report SPRUED5B, Texas Instruments (2007)
6. DSP/BIOS link user's guide. Technical Report Version 1.65, Texas Instruments (2006)
7. El-Hashash, M.M., Aly, H.A., Mahmoud, T.A., Swelam, W.: A video haze removal system on heterogeneous cores. In: Proceedings IEEE Global Conference on Signal and Information Processing, pp 1255–1259 (2015). doi:10.1109/GlobalSIP.2015.7418399
8. Fattal, R.: Single image dehazing. ACM Trans. Graph. **27**(3), 72:1–72:9 (2008). doi:10.1145/1360612.1360671
9. Feng, C., Zhuo, S., Zhang, X., Shen, L., Susstrunk, S.: Near-infrared guided color image dehazing. In: Proceedings on IEEE International Conference on Image Processing, pp. 2363–2367 (2013)
10. Hadamard, J.: Lectures on Cauchy's Problem in Linear Partial Differential Equations. Dover Pub. Inc, New York (1952)
11. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE Trans. Pattern Anal. Mach. Intell. **33**(12), 2341–2353 (2011). doi:10.1109/TPAMI.2010.168
12. Ke, N., Chen, J.: Real-time visibility restoration from a single image. In: Proceedings on IEEE International Conference on Image Processing, pp. 923–927 (2013)
13. Khodary, A.G., Aly, H.A.: A new image-sequence haze removal system based on DM6446 DaVinci processor. In: Proceedings IEEE Global Conference on Signal and Information Processing, pp. 703–706 (2014). doi:10.1109/GlobalSIP.2014.7032209
14. Kopf, J., Neubert, B., Chen, B., Cohen, M., Cohen-Or, D., Deussen, O., Uyttendaele, M., Lischinski, D.: Deep photo: model-based photograph enhancement and viewing. ACM Trans. Graph. **27**(5), 116:1–116:10 (2008). doi:10.1145/1409060.1409069
15. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 228–242 (2008). doi:10.1109/TPAMI.2007.1177
16. McCartney, E.: Optics of the Atmosphere: Scattering by Molecules and Particles. Pure and Applied Optics Series. Wiley, New York (1976)
17. Narasimhan, S., Nayar, S.: Chromatic framework for vision in bad weather. In: Proceedings IEEE Conference on Computer Vision Pattern Recognition, vol. 1, pp. 598–605 (2000). doi:10.1109/CVPR.2000.855874
18. Narasimhan, S., Nayar, S.: Contrast restoration of weather degraded images. IEEE Trans. Pattern Anal. Mach. Intell. **25**(6), 713–724 (2003a). doi:10.1109/TPAMI.2003.1201821
19. Narasimhan, S., Nayar, S.: Interactive deweathering of an image using physical models. In: IEEE Workshop on Color and Photometric Methods in Computer Vision (2003b)
20. Nayar, S., Narasimhan, S.: Vision in bad weather. In: Proceedings IEEE Conference on Computer Vision, vol. 2, pp. 820–827 (1999). doi:10.1109/ICCV.1999.790306
21. Schechner, Y., Narasimhan, S., Nayar, S.: Instant dehazing of images using polarization. In: Proceedings IEEE Conference on Computer Vision Pattern Recognition, vol. 1, pp. 325–332 (2001). doi:10.1109/CVPR.2001.990493
22. Shwartz S, Namer E, Schechner Y.: Blind haze separation. In: Proceedings IEEE Conference on Computer Vision Pattern Recognition, vol. 2, pp. 1984–1991 (2006). doi:10.1109/CVPR.2006.71
23. Tan, R.: Visibility in bad weather from a single image. In: Proceedings IEEE Conference on Computer Vision Pattern Recognition, pp. 1–8 (2008). doi:10.1109/CVPR.2008.4587643
24. TMS320 DSP/BIOS user's guide. Technical Report SPRU423B, Texas Instruments (2002)
25. TMS320 DSP algorithm standard API reference user's guide. Technical Report SPRU360E, Texas Instruments (2007)
26. TMS320DM6446 digital media system-on-chip (Rev. e). Technical Report SPRS283E, Texas Instruments (2007)
27. VICP signal processing library for DM6446, DM6441, DM647, and DM648 user's guide (Rev. e). Technical Report SPRUGJ3E, Texas Instruments (2010)
28. Wu, D., Zhu, Q., Wang, J., Xie, Y., Wang, L.: Image haze removal: status, challenges and prospects. In: Proceedings IEEE International Conference on Information Science and Technology, pp. 492–497 (2014). doi:10.1109/ICIST.2014.6920524
29. XDAIS-DM (digital media) user's guide. Technical Report SPRUEC8, Texas Instruments (2007)

Mostafa M. El-Hashash received the B.Sc. (very good with honors) in Computer Engineering from the Military Technical College (MTC), Egypt, in 2010 (first rank in computer department group 2010). He received his M.Sc. degree in Electrical Engineering from the MTC in 2015. He worked as a teaching assistant at the Department of Computer Engineering at MTC since 2013, where he taught variety of computer engineering courses. His research interests are in the field of embedded computer systems, and digital image and video processing. He is a member of Egypt Engineers Syndicate.

Hussein A. Aly received the B.Sc. (excellent with honors) in Computer Engineering and the M.Sc. degree in Electrical Engineering from the Military Technical College (MTC), Egypt, in 1993 and 1997, and the Ph.D. in Electrical Engineering from the University of Ottawa in 2004. He is Associate Professor with the Computer Department at MTC and the Chief of Department (2010–2013). He was a visiting professor in the Electrical Engineering Department at the University of Rochester (September 2012–March 2013). His research interests are in image sampling theory and sampling structure conversion. His current research is focused on high-quality image magnification, interpolation of color filter array data, the application of total variation for image processing, data fusion, video steganography, and embedded computer systems for video processing. While at the University of Ottawa, he participated in a project on image magnification funded by the Royal Canadian Mounted Police (RCMP). Dr. Aly is a senior member of the IEEE (since 2011), Society for Optics Photonics (SPIE), and Egypt Engineers Syndicate.