Smart and real time image dehazing on mobile devices

Yucel Cimtay (yucelcimtay@gmail.com)
Havelsan A.S

Original Research Paper

Keywords: atmospheric light, hazy imagery, depth map, transmission

Posted Date: February 3rd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-156893/v1

Version of Record: A version of this preprint was published at Journal of Real-Time Image Processing on February 27th, 2021. See the published version at https://doi.org/10.1007/s11554-021-01085-z.

Smart and Real-time image dehazing on mobile devices

Yucel CIMTAY

Image and Signal Processing Group, HAVELSAN A.S

Abstract – Haze is one of the common factors that degrades the visual quality of the images and videos. This diminishes contrast and reduces visual efficiency. The ALS (Atmospheric light scattering) model which has two unknowns to be estimated from the scene: atmospheric light and transmission map, is commonly used for dehazing. The process of modelling the atmospheric light scattering is complex and estimation of scattering is time consuming. This condition makes dehazing in real-time difficult. In this work, a new approach is employed for dehazing in real-time which reads the orientation sensor of mobile device and compares the amount of rotation with a pre-specified threshold. The system decides whether to recalculate the atmospheric light or not. When the amount of rotation is little means there are only subtle changes to the scene, it uses the pre-estimated atmospheric light. Therefore, the system does not need to recalculate it at each time instant and this approach accelerates the overall dehazing process. 0.07s fps (frame per second) per frame processing time (~15fps) is handled for 360p imagery. Frame processing time results show that our approach is superior to the state-of-the-art real-time dehazing implementations on mobile operating systems.

Keywords – atmospheric light, hazy imagery, depth map, transmission.

1. Introduction

Image and video dehazing are crucial for offline and online computer vision applications needed in security, transportation, video surveillance and military. Consequently, the number of studies related to image enhancement has steadily increased in recent years [1]. Image dehazing is a kind of image enhancement, however it varies from others due to changes in image deterioration regarding the scene distance from the observation point and the amount of haze globally and locally. In other terms, as the distance between the sensor and the scene increases the thickness of the haze also increases and the transmission of the media decreases. Likewise, when the density of haze is high and differs locally, the complexity of dehazing process increases. To illustrate, Figure 1 displays two hazy and haze-free (dehazed) image pairs. Image (a) is a hazy image, and (b) is the result of haze removal process applied to (a). Similarly, (c) is the hazy image and (d) is the haze-free pair of (c). Since the thickness of the haze is higher in the second image pair, haze removal operation is less effective, and the visual quality of the dehazed image is poor.

There are many ways of image dehazing and they can be grouped into three categories which are contrast enhancement [2-5], restoration [6-10] and fusion based [11-15] methods. Contrast enhancement approaches aim to improve the visual quality of the hazy images to some degree; however, they cannot eliminate the haze efficiently. The subcategories of image enhancement models are histogram enrichment [16-18] which can be applied locally and/or globally, frequency transform methods: wavelet transform, and homomorphic filtering, and the Retinex method: single and multi-scale Retinex [19]. Restoration based methods focus on recovering the lost information by modelling the image degradation model and applying inverse filtering.

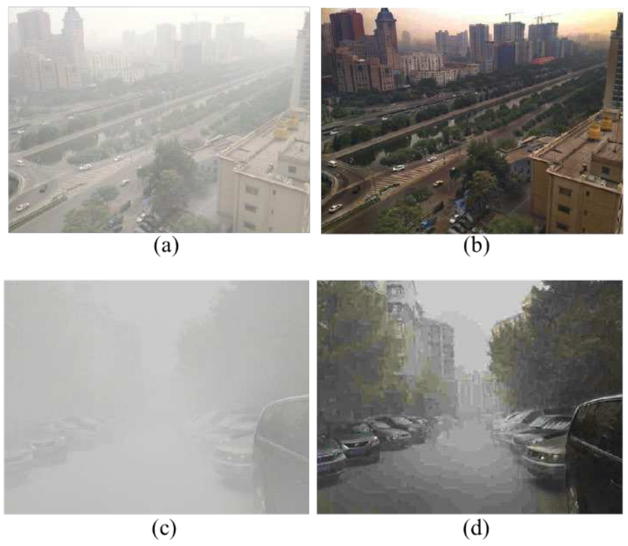

Figure 1 (a) Hazy image (b) Haze-free (dehazed) image of (a) (c) Hazy image (d) Haze-free (dehazed) image of (c)

Since this study is based on the application of image dehazing in real-time, the specifics of dehazing methods will not be covered. On the other hand, ALS (atmospheric light scattering) model which is shown in Figure 2 is used as the basis of our method.

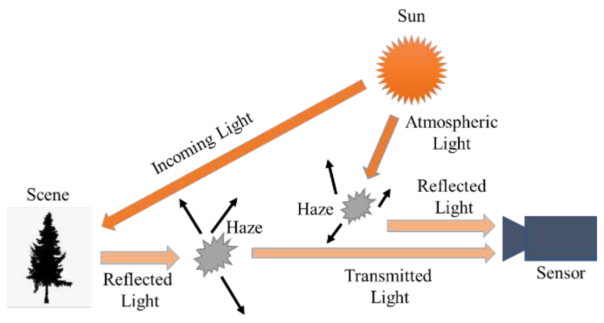

Figure 2 Atmospheric light scattering model

Equations 1-3 which were adopted from the study in [20] express atmospheric light scattering model where $I(x,\lambda)$ is the hazy image, $D(x,\lambda)$ is the transmitted light through the haze (after the reflection from the scene) and $A(x,\lambda)$ is the air light which is the reflected atmospheric light from haze. The sensor integrates the incoming light and the resulted imagery is the hazy image. In Equation 2, $t(x,\lambda)$ is the transmission map of the hazy scene, $R(x,\lambda)$ is the reflected light from the scene and L_∞ is the atmospheric light. The transmission term is expressed as $e^{-\beta(\lambda)d(x)}$ where $d(x)$ is the depth map of the scene and $\beta(\lambda)$ is the atmospheric scattering coefficient with respect to wavelength. It can simply be understood from Equation 3 that, when the depth from the sensor increases, the transmission decreases and vice versa.

$$I(x,\lambda) = D(x,\lambda) + A(x,\lambda) \quad (1)$$

$$I(x,\lambda) = t(x,\lambda)\, R(x,\lambda) + L_\infty(1 - t(x,\lambda)) \quad (2)$$

$$I(x,\lambda) = e^{-\beta(\lambda)d(x)} R(x,\lambda) + L_\infty(1 - e^{-\beta(\lambda)d(x)}) \quad (3)$$

The key point of ALS is the accurate estimation of the transmission and the atmospheric light. DCP (The Dark Channel Prior Method) [21] is one of the most commonly used methods in which the per-pixel dark channel previous is used for haze estimation. At the same time, for measuring the atmospheric light, quadtree decomposition is applied. Another research that uses the DCP as its basis is [22]. In this study, both per-pixel and spatial blocks are used for calculation of the dark channel.

Recent approaches on image dehazing is mostly based on artificial intelligence approaches which mostly use deep learning models [23-25]. In [26] a deep architecture is developed by using CNN (Convolutional Neural Network) and a new unit called "bilateral rectified linear unit" is added to the neural network. It reports that it achieves superior results compared to previous dehazing studies. The study in [27] employs an end-to-end encoder-decoder CNN architecture to handle the haze-free images.

There are many successful image dehazing studies in the literature. However, when the focus is real-time implementation, many bottlenecks such as the complexity of the algorithms, hardware constraints and high financial costs should be considered. Nonetheless, there have been several successful studies underway. The study in [28] estimates the atmospheric light by using super-pixel segmentation and applies a guidance filter to refine the transmission map. It mentions that more accurate results compared to other state-of-the-art models are handled. The study in [29] proposes parallel processing dehazing method for mobile devices and achieves 1.12s per frame processing time for HD imagery on a Windows Phone by using CPU (Central Process Unit) and GPU together. The study in [30] uses DCP but substitutes guided filter with mean filter in order to increase the processing speed. It reports 25 fps over C6748 pure DSP (Digital Signal Processing) device [31].

The study in [32] converts hazy RGB (Red-Green-Blue) image to HSV (Hue-Saturation-Value) colour space and applies a global histogram flattening on value component, modifies the saturation component to be consistent with previous reduced value and applies contrast enhancement on value component. It achieves 90ms dehazing time for HD imagery on GPU (Graphics Processing Unit). The study in [33] conducts 2 level image processing with a smart way. It first applies histogram enhancement and if the resulted image meets the system requirements then no further action is taken. If it does not, then DCP is used to remove the haze. By using a smart way, it saves a lot of time and achieves real-time processing.

The study in [34] uses locally adaptive neighbourhood and calculates order statistics. By using this information, it produces the transmission map and handles the haze-free image. The study in [35] parallelizes the base Retinex model and decompose the image into brightness and contrast components. For restoration of the image, it applies gamma correction and non-parametric mapping and reports 1.12ms processing time for 1024x2048 high resolution image on parallel GPU system. The study in [36] constructs a transmission function estimator by using genetic programming. Then this function is used for computing the transmission map. Transmission map and hazy image are used to obtain the haze-free images. The system runs with high-rate processing time on synthetic and real-world imagery.

Another successful real-time dehazing method is implemented in [37]. A novel pixel-level optimal dehazing criteria is proposed to merge a virtual haze-free image series of candidates into a resulted single hazy-free image. This sequence of images is calculated from the input hazy image by exhausting all possible values of discreetly sampled depth of the scene. The advantage of this method is the computing any single pixel position independent of the others. Therefore, it is easy to implement this method by using a fully parallel GPU system.

The literature is very rich about dehazing the single image and the video. Implementations in real-time are also very interesting. However, real-time processing is very rare on mobile devices such as Android and IOS. The study in [29] implements real-time dehazing on a Windows phone. This study is also one of the benchmark studies in this paper in which the results of the proposed study is compared. In this paper, DCP-based algorithm is implemented on a mobile android operating system with reading the sensor data from the device's orientation sensor. A smart way which determines the run time of re-estimation of atmospheric light is created. If system movement is measured as minor which means that the scene doesn't shift roughly, the previous ambient light is used to dehaze the imagery. If the movement exceeds some predetermined threshold then the estimation will be done once. By using this smart strategy, it is possible to achieve promising time gain in processing. On the other hand, transmission is based on the depth map and minor changes of orientation also lead to major changes on the depth map, so on the transmission map. Therefore, the transmission map is always calculated for each time instant.

The rest of the paper is structured to clarify the details of the proposed approach and its implementation in real-time in section 2. The average real-time processing results and the benchmark table with some other real-time studies are given in Section 3. Section 4 is the final part and some guidelines on some potential future studies relating to real-time dehazing are included.

2. **Proposed Method**

In this study we improve the algorithm introduced in [22] by adding a smart decision method for atmospheric light calculation. DCP approach, information fidelity, and image entropy are used to estimate atmospheric light and map transmission. The steps are prior estimation of the dark channel image, estimation of the atmospheric light, estimation of the transmission, refinement of the transmission with guided filter and reconstruction of the haze-free image by applying Equation 2.

The study in [22] provides very promising accuracy results. The benchmark scores for two different hazy images are given in Table 1 and 2. The images and the visual results of different methods are given in Figure 3. In Table 1 and 2, the comparisons are done based on the colorfulness, GCF (Global Contrast Factor) and visible edge gradient. The visible edge gradient measures the visibility using the restored and hazy images. It has three indicators e, r and σ where e is the amount of visible new edge after dehazing, r is the average ratio of gradient norm values at visible edges, and σ is the percentage of pixels after processing which becomes black or white.

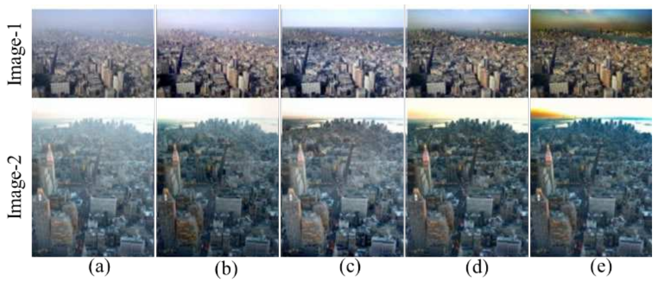

Figure 3 The visual comparison of several methods. (a) Hazy image (b) Fattal's result (c) Kopf's result (d) He's result (e) Park's result.

The quality of dehazed images improves when σ gets smaller and the other indicators gets bigger. Although Kopf's method [39] shows good performance in close-range regions, it is not successful in far-range. Because it cannot remove the haze effectively. As GCF and r scores, Kopf's algorithm provides promising results, however it is not satisfactory for colorfulness and σ scores. In addition, He's method [40] has limited performance, since it has good scores only for GCF and σ. Park's study [22] provides better results for overall evaluations.

Table 1 Accuracy results for image 1

Index	Fattal [38]	Kopf [39]	He [40]	Park [22]
e	0.11	0.02	0.02	0.32
r	1.53	1.61	1.63	2.27
σ	1.7	1.35	0.01	0.06
Colorfulness	652.45	455.84	963.62	1127.42
GCF	7.87	8.53	8.63	8.49

Table 2 Accuracy results for image 2

Index	Fattal [38]	Kopf [39]	He [40]	Park [22]
e	0.05	0.03	0.04	0.08
r	1.28	1.4	1.39	1.41
σ	9.4	0.29	0.01	0.05
Colorfulness	387.01	390.67	509.9	706.09
GCF	5.89	6.65	6.72	6.8

Park's method is successful and effective to be improved for real-time implementation.

In this study, firstly, the amount of time spent for atmospheric light estimation and the other steps of dehazing algorithm is calculated. 50 hazy images are used with various amount of haze and resolution. The atmospheric light estimation step covers most of the processing time spent with a mean percentage of 78%. Therefore, by measuring the orientation and calculating the atmospheric light in a smart manner, the proposed approach presents its value and contribution to related literature.

The overall system diagram for the proposed method is shown in Figure 4. Note that in order to prevent possible synchronization problems, dehazing operation is implemented once atmospheric light, transmission map and camera data is handled. AOO term in Figure 4 stands for 'Amount of Orientation'. Since the device can rotate in 3D space, all possible pitch, yaw and roll angles are checked in the data controller. If any of them is above a predetermined threshold, the atmospheric light and transmission map is recalculated. If not, then the atmospheric light of the previous time instant is used and only the transmission map is calculated. Finally, the dehazing module reconstructs the dehazed image by using camera data, atmospheric light and transmission map. Dehazed image data is displayed on the device screen in real-time.

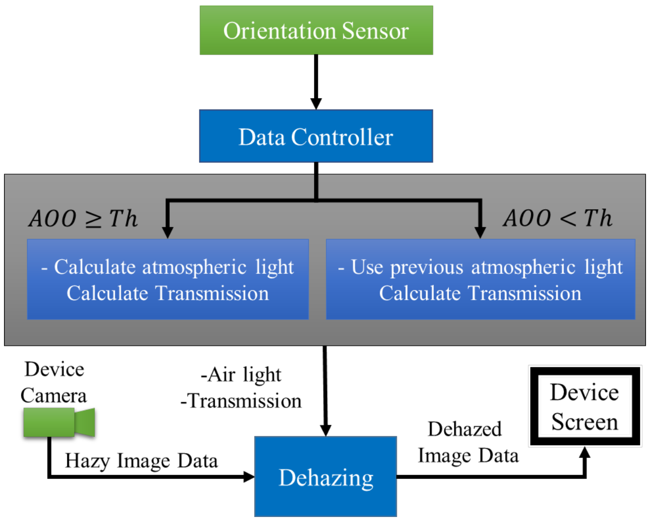

Figure 4 Overall System diagram

The optimal AOO threshold is determined empirically. Determining the optimal threshold is the core of the proposed study. Because, the atmospheric light estimation is the most important step for a high-quality reconstruction process. To determine the optimal AOO for each axis, following steps are applied:

1. The clear imagery of the scene is captured from 2m distance by fixing the place of the android device.

2. The device is rotated up to 20°, only towards one direction, by a step size of 2° on the axis pitch, yaw and roll and the imagery for each step size is captured.

3. Haze is produced by using dry ice and hot water and step 2 is repeated. One example of clear and hazy imagery pair is shown in Figure 5.

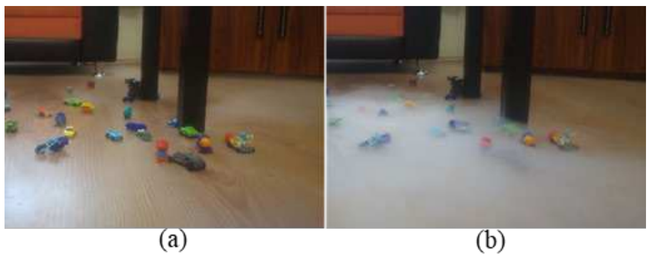

Figure 5 (a) Clear image (b) Hazy image

4. For each hazy image, the haze-free partner is reconstructed both using [22] with calculation of atmospheric light for each step and using the same atmospheric light which was calculated once at beginning. So, we have TS (threesome) of 11 clear, haze free with [22] and haze free with [22] in the case of using the same atmospheric light. The TS images are named as TS-1 (TS_1), TS-2 (TS_2), …, TS-11 (TS_{11}). Threesome members are $TS_{(x)1}$, $TS_{(x)2}$ and $TS_{(x)3}$ respectively where x denotes the threesome index number.

PSNR (Peak Signal to Noise Ratio) which is based on the mean squared error, is one of the mostly used metric for measuring the similarity of the restored image to ground-truth [41, 42]. Therefore, PSNR is used in this study to measure the similarity between the clear image and dehazed image in order to determine an orientation threshold.

5. PSNR between each clear and haze-free images is calculated and named as $PSNR(TS_{(1)1}, TS_{(1)2})$ and $PSNR(TS_{(1)1}, TS_{(1)3})$. For instance, if $PSNR(TS_{(1)1}, TS_{(1)3})$ don't reduce by 20% compared to $PSNR(TS_{(1)1}, TS_{(1)2})$, then the next threesome is processed and same calculation is done for $PSNR(TS_{21}, TS_{22})$ and $PSNR(TS_{21}, TS_{23})$ and so on. Note that for each following threesome, the atmospheric light which was calculated in the dehazing process of $TS_{(1)3}$ is used for reconstruction of $TS_{(x)3}$.

6. When the $PSNR(TS_{(x)1}, TS_{(x)3})$ drops 20% below of $PSNR(TS_{(x)1}, TS_{(x)2})$, we choose the optimal rotation value as the rotation value of the image $TS_{(x-1)1}$. An example of threesome is given in Figure 6. This shows an example of the dehazing results where the $PSNR(TS_{(x)1}, TS_{(x)3})$ drops 20 % below of $PSNR(TS_{(x)1}, TS_{(x)2})$.

The change of PSNR values for yaw, roll, pitch axis with respect to threesome index is given in Tables 3-5. These tables show the PSNR as the rotation of the device changes. Starting from zero, for each 2° change of orientation, a new dehazed image is reconstructed and $PSNR$ between dehazed image pairs is re-calculated. The orientation angle is increased by 2° at each step and since the PSNR tolerance is chosen as 20%, this is continued up to the $PSNR(TS_{(x)1}, TS_{(x)3})$ drops 20 % below of $PSNR(TS_{(x)1}, TS_{(x)2})$. This procedure is repeated for each of yaw, roll and pitch axis.

Table 3 PSNR values for each threesome w.r.t yaw

x	Yaw angle (°)	$PSNR(TS_{(x)1}, TS_{(x)3})$	$PSNR(TS_{(x)1}, TS_{(x)2})$
1	0	23.6	23.6
2	2	21.3	22.5
3	4	22.7	24.1
4	6	23.8	25.2
5	8	19.6	23.4
6	10	19.3	23.9
7	12*	19.5	24.0
8	14	18.1	23.8

* Optimal angle

Table 4 PSNR values for each threesome w.r.t roll

x	Roll angle (°)	$PSNR(TS_{(x)1}, TS_{(x)3})$	$PSNR(TS_{(x)1}, TS_{(x)2})$
1	0	23.6	23.6
2	2	21.6	23.1
3	4	20.2	22.8
4	6	19.7	24.3
5	8	19.2	24.6
6	10*	19.3	23.9
7	12	18.7	23.5

* Optimal angle

Table 5 PSNR values for each threesome w.r.t pitch

x	Pitch angle (°)	$PSNR(TS_{(x)1}, TS_{(x)3})$	$PSNR(TS_{(x)1}, TS_{(x)2})$
1	0	23.6	23.6
2	2	22.4	25.3
3	4	21.3	24.7
4	6	20.5	23.2
5	8	20.0	25.4
6	10*	19.8	24.8
7	12	18.1	24.1

* Optimal angle

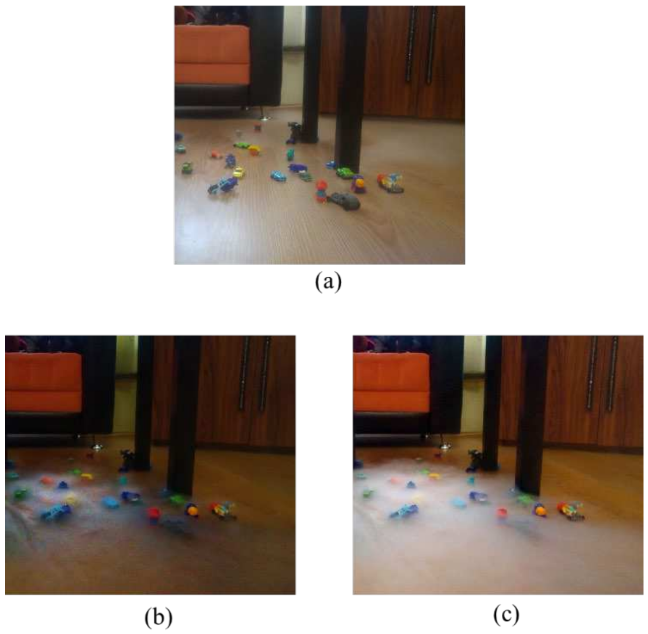

Figure 6 An example threesome. (a) Clear Image (b) Result of direct application of Park's method (c) Result of proposed approach. (PSNR is just below the threshold).

7. The same procedure is repeated for each pitch, roll and yaw axis. The founded optimal rotation angles for 3 different scenes are given in Table 6.

For instance, for scene-1 the optimal angles for pitch, roll and yaw are 10°, 10° and 12° respectively. Similarly, they are 10°, 12°, 14° for scene-2 and 12°, 12° and 12° for scene-3. Therefore, the most common optimal angle (12° for this case) for pitch, roll and yaw among scenes is chosen as the optimal angle for 20% PSNR tolerance.

Table 6 Optimal rotation angles for pitch, roll and yaw

Scenes	Pitch (°)	Roll (°)	Yaw (°)
1	10	10	12
2	10	12	14
3	12	12	12

According to Table 6, the optimal rotation angle is 12°.

2.1 **Design on MATLAB and Deploying on Android OS**

In the literature, to now, there is no complete work on dehazing on the Android operating system in real-time. In this study, MATLAB SIMULINK is used for implementation of the proposed method. MATLAB SIMULINK has Android device support for developing and deploying MATLAB codes and MATLAB SIMULINK models [43]. The SIMULINK model developed is given in Figure 7.

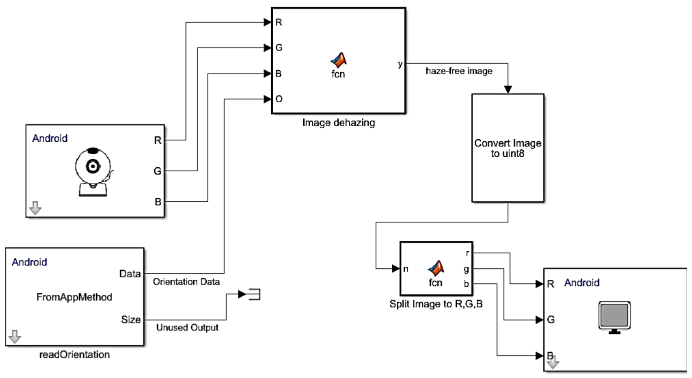

Figure 7 SIMULINK Model for Real-Time Implementation

A Simulink block called "FromAppMethod" is used which named as 'readOrientation' and coded in Android studio. This function reads and outputs the android device's actual and preceding time orientation data in real-time. Since the 'size' output is not needed, it is terminated. On the other hand, the 'Android Camera' block reads live video from the device's camera. The camera resolution can be set by also using this block. Real-time video and orientation data are fed to the 'Image dehazing' function that compares the previous and current orientation data and runs the proposed dehazing algorithm. The next block in Simulink is the image type conversion block which converts its input's type to double. 'Image splitting' block splits the RGB image to its color components R, G and B. Then these components are displayed on device screen by using 'Android Video Display' block.

This project is deployed on an Android device by using 'Android Studio' [44]. By the way, The MATLAB codes are transferred to C++ code and a Java code is produced for user updates and declaring new functions. The android device used has Qualcomm® Snapdragon™ 665 Octa-core processor, which has frequency up to 2GHz. It has 3 GB RAM. The camera's video resolution is up to 4K at 30 fps.

The pseudo code of the proposed method is:

```
def dehaze (hazyImg, airlight, transMap, $\beta$)
    dehazedImg=(hazyImg-airlight*(1-$e^{-\beta*\text{transMap}}$))/ $e^{-\beta*\text{transMap}}$
    return dehazedImg

def estimateAtmLight(hazyImg)
    //estimation of atmospheric light
    return airlight
def estimateTransMap (hazyImg)
    //estimation of transmission map
    return transMap
```

```
//Setting rotation threshold value as the optimal threshold handled
//empirically.
Threshold = optimalThreshold
$\beta$ = 0.25 //scattering coefficient
hazyImg = readImage ()
airlightinit = estimateAtmLight (hazyImg)

while True
    hazyImg = readImage ()
    orientation = readOrientation ()
    //check if orientation > Threshold
    if orientation > Threshold
        airlight = estimateAtmLight (hazyImg) //atm. light estimation
        transMap = estimateTransMap(hazyImg)
        dehazedImg = dehaze (hazyImg, airlight, transMap, $\beta$)
        display(dehazedImg)
        airlightinit = airlight
    else
        airlight = airlightinit //no estimation of airlight
        transMap = estimateTransMap (hazyImg)
        dehazedImg = dehaze (hazyImg, airlight, transMap, $\beta$)
        display(dehazedImg)
```

3. **Results**

Theoretically, the most time-consuming part of a dehazing algorithm is the estimation of atmospheric light. Therefore, in this study, the author focuses on finding a logical way of not estimating the atmospheric light for each time instant and a novel approach which is based on reading the orientation sensor of a mobile device and measuring the amount of orientation prior to the dehazing operation is proposed. The atmospheric light has not very high variance in many scenes. It may change from time to time according to the weather conditions however, for a specific time interval it does not change so much within some specific rotation limits. This assumption is validated in this study by applying it on 3 different scenes and the detail procedure is introduced in the 'Proposed Method' section. Therefore, determining a smart way to measure the rotation of the device enables to reduce the processing time. This is achieved by estimating the atmospheric light when required instead of estimating it for each time instant. By that way, this study provides promising real-time processing speed. The mean results are shown in Table 7 for different imagery resolutions.

Table 7 Real-Time Processing Speed

Resolution	Proc. Time (sec) per frame $AOO \geq Th$	Proc. Time (sec) per frame $AOO < Th$
1080p (1920x1080)	4.36	0.88
720p (1280x720)	1.95	0.42
480p (864x480)	0.87	0.19
360p (480x360)	0.36	0.07

From Table 7, the mean processing time for HD (High Definition) imagery changes from 0.42 to 1.95 seconds per frame. This mean time depends on the amount of movement of the device. If the amount of movement is high which means for a specific time interval the device is rotated much above the optimal rotation angle, then the number of recalculation of atmospheric light will be high and the processing time per frame will increase towards 1.95s. However, if the rotation is less, then the mean processing time will go down to 0.42s.

Secondly, as the resolution of imagery increases, since the number of pixels increases, the processing time per frame also increases automatically.

Another important point is the threshold for atmospheric light re-estimation. There is an optimal value of the threshold which depends on the day of the year, cloud rate and other environmental effects. In our tests, the threshold is empirically set to 12° optimally which keeps the reconstructed image visual quality and PSNR. Note that this threshold was determined empirically and should be set depending on the conditions at the time of dehazing.

The processing time results of this study is compared with the results of the studies in [29, 30, 32, 35]. The benchmark results are given in Table 8.

Note that the studies [30], [32] and [35] are not applied on mobile operating systems/devices. Therefore, the processing time results are generally better due to the more powerful hardware they use. Although they are not directly comparable with the proposed method, since they are also useful studies in real-time implementation context, their results are also included in this study and benchmark table for introducing the current level of real-time dehazing on mobile based operating systems compared to non-mobile systems.

Table 8 Benchmark results for processing time per frame

Studies	1080p (1920x1080)	720p (1280x720)	480p (864x480)	Mobile based
[29] with CPU	105.50	39.27	14.98	Yes
[29] with (CPU+GPU)	3.22	1.12	0.72	Yes
[32] with GPU	NA	0.09	NA	No
[35] with parallel GPU	0.001	NA	NA	No
[30] on DSP	0.04	NA	NA	No
Proposed Method	0.88	0.42	0.19	Yes

As it is explained in detail in the previous sections, the proposed method gets use of the reality that atmospheric light is not very variant in a specific time interval. Therefore, measuring the current variance of it at the time of dehazing and specifying the threshold rotation angle is the novelty of our method. The proposed method reads the orientation sensor of the mobile device and decide to recalculate the atmospheric light or not. Once the rotation angle exceeds the optimal threshold rotation angle, the atmospheric light is recalculated. Otherwise, the previously calculated atmospheric light is used for dehazing. Since the most time-consuming part in dehazing is the estimation of atmospheric light, the proposed approach is successful in terms of accelerating the dehazing process. From Table 8 it can be observed that the proposed method is more successful among the other studies which implement dehazing on mobile operating systems. It is better than the study in [29] both for CPU case and CPU and GPU together case.

4. **Conclusion and Future Work**

In this study, a new image dehazing method which is based on measuring the change of the scene by reading the device orientation sensor in real-time and a mechanism to re-calculate the atmospheric light is implemented. Since, the change of the scene has not high variance in many real-time dehazing applications, this study gets use of defining some reconstruction error toleration for dehazed image. This study proves that it is possible to handle high quality dehazed images by skipping the calculation of the atmospheric light for some time instant up to exceeding a pre-defined orientation threshold. Since the most time-consuming part of image dehazing is atmospheric light calculation, proposed approach accelerates the overall process and reduce the processing time for each frame. This enables to dehaze in real-time. By keeping the visual quality of the reconstructed image, promising image processing time results are achieved despite limited power of hardware and only CPU is used. The results are superior or on par with the other state-of-the-art real-time dehazing applications. Processing time results show that proposed method can be applied in real-time on the devices which have android operating system. If the system is empowered in terms of hardware specifications, then the processing time will decrease dramatically.

The future work should be based on using GPU and/or CPU and GPU together. On the other hand, more powerful hardware devices should be used. Furthermore, similar implementation should be done on IOS devices. Another important point is that transmission maps may be estimated by using stereo imaging which enables more accurate estimation of the depth maps.

The next work will be based on using deep learning models and deploying the model on Android devices. This will most probably increase the visual quality besides increasing the processing speed.

References

[1] Wang W., Yuan X.: Recent advances in image dehazing. In IEEE/CAA Journal of Automatica Sinica, vol. 4, no. 3, pp. 410–436 (2017)

[2] Jia Z., Wang H. C., Caballero R., Xiong, Z. Y., Zhao J. W., Finn, A.: Real-time content adaptive contrast enhancement for see-through fog and rain. In Proc. IEEE Int. Conference Acoustics Speech and Signal Processing, pp. 1378−1381 (2010)

[3] Al-Sammaraie, M. F.: Contrast enhancement of roads images with foggy scenes based on histogram equalization. In Proc. 10th International Conference on Computer Science & Education, pp. 95−101 (2015)

[4] Kim J. H., Sim, J. Y., Kim, C. S.: Single image dehazing based on contrast enhancement. In Proc. IEEE International Conference Acoustics, Speech and Signal Processing, pp. 1273−1276 (2011)

[5] Cai W. T., Liu Y. X., Li, M. C., Cheng L., Zhang, C. X.: A self-adaptive homomorphic filter method for removing thin cloud. In Proc. 19th International Conference Geoinformatics, pp. 1−4 (2011)

[6] Tan, K., Oakley, J. P.: Physics-based approach to color image enhancement in poor visibility conditions. In Journal of the Optical Society of America, vol. 18, no. 10, pp. 2460−2467 (2001)

[7] Tang, K. T., Yang, J. C., Wang, J.: Investigating haze-relevant features in a learning framework for image dehazing. In Proc. IEEE Conference Computer Vision and Pattern Recognition, pp. 2995−3002 (2014)

[8] Gibson, K. B., Belongie, S. J., Nguyen, T. Q.: Example based depth from fog. In Proc. 20th IEEE International Conference on Image Processing, pp. 728−732 (2013)

[9] Fang, S., Xia, X. S., Xing, H., Chen, C. W.: Image dehazing using polarization effects of objects and airlight. In Opt. Express, vol. 22, no. 16, pp. 19523−19537 (2014)

[10] Galdran, A., Vazquez-Corral, J., Pardo, D., Bertalmio, M.: Enhanced variational image dehazing.: SIAM Journal of Imaging Science, vol. 8, no. 3, pp. 1519−154 (2015)

[11] Son, J., Kwon, H., Shim, T., Kim, Y., Ahu, S., Sohng, K.: Fusion method of visible and infrared images in foggy environment. In Proc. International Conference on Image Processing, Computer Vision, and Pattern Recognition, pp. 433−437 (2015)

[12] Ancuti, C. O., Ancuti, C.: Single image dehazing by multi-scale fusion. In IEEE Transaction on Image Processing, vol. 22, no. 8, pp. 3271−3282 (2013)

[13] Ma, Z. L., Wen, J., Zhang, C., Liu, Q. Y., Yan, D. N.: An effective fusion defogging approach for single sea fog image. In Neurocomputing, vol. 173, pp. 1257−1267 (2016)

[14] Guo, F., Tang, J., Cai, Z. X.: Fusion strategy for single image dehazing. In International Journal of Digital Content Technology and Its Applications, vol. 7, no. 1, pp. 19−28 (2013)

[15] Zhang, H., Liu, X., Huang, Z. T., Ji, Y. F.: Single image dehazing based on fast wavelet transform with weighted image fusion. In Proc. IEEE International Conference on Image Processing, pp. 4542−4546 (2014)

[16] Simi V.R., Edla D.R., Joseph, J., and Kuppili, V.: Parameter-free Fuzzy Histogram Equalisation with Illumination Preserving Characteristics Dedicated for Contrast Enhancement of Magnetic Resonance Images, Applied Soft Computing, vol. 93, (2020)

[17] Joseph, J., and Periyasamy R.: A fully customized enhancement scheme for controlling brightness error and contrast in magnetic resonance images, Biomedical Signal Processing and Control, vol. 39, pp. 271-283, (2018)

[18] Joseph, J., Sivaraman, J., R. Periyasamy and Simi V.R.: An objective method to identify optimum clip-limit and histogram specification of contrast limited adaptive histogram equalization for MR images, Biocybernetics and Biomedical Engineering, vol. 37, issue 3, pp. 489-497, (2017)

[19] Hao, W., He, M., Ge, H., Wang, C., Qing-Wei G.: Retinex-Like Method for Image Enhancement in Poor Visibility Conditions. In Procedia Engineering, Volume 15 (2011)

[20] Wang, Wencheng & Chang, Faliang & Ji, Tao & Wu, Xiaojin. (2018). A Fast Single-Image Dehazing Method Based on a Physical Model and Gray Projection. IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2794340.

[21] Kaiming, H., Jian, S., Xiaoou, T.: Single Image Haze Removal Using Dark Channel Prior. In IEEE Transactions on pattern analysis and machine intelligence. (2011)

[22] Park, D., Park, H., Han, D. K., Ko, H.: Single image dehazing with image entropy and information fidelity. In IEEE International Conference on Image Processing (ICIP), pp. 4037-4041, (2014)

[23] Li, J., Li, G., Fan, H.: Image Dehazing Using Residual-Based Deep CNN. In IEEE Access, vol. 6, pp. 26831-26842 (2018)

[24] Li, C., Guo, J., Porikli, F., Fu, H., Pang, Y.: A Cascaded Convolutional Neural Network for Single Image Dehazing. In IEEE Access, vol. 6, pp. 24877-24887 (2018)

[25] Haouassi, S., Di, W.: Image Dehazing Based on (CMTnet) Cascaded Multi-scale Convolutional Neural Networks and Efficient Light Estimation Algorithm. In Applied Sciences (2020)

[26] Cai, B., Xu, X., Jia, K., Qing, C., Tao, D.: DehazeNet: An End-to-End System for Single Image Haze Removal. In IEEE Transactions on Image Processing, vol. 25, no. 11, pp. 5187-5198 (2016)

[27] Rashid, H., Zafar, N., Javed Iqbal, M., Dawood, H., Dawood, H.: Single Image Dehazing using CNN. In Procedia Computer Science, Volume 147, pp. 124-130 (2019)

[28] Hassan, H., Bashir, A.K., Ahmad, M. et al.: Real-time image dehazing by superpixels segmentation and guidance filter. In Journal of Real-Time Image Proc (2020).

[29] Yuanyuan, S., Yue. M.: Single Image Dehazing on Mobile Device Based on GPU Rendering Technology. In Journal of Robotics, Networking and Artificial Life (2015)

[30] Lu, J., Dong, C. DSP-based image real-time dehazing optimization for improved dark-channel prior algorithm. Journal of Real-Time Image Processing (2019)

[31] C6748 pure DSP device data sheet : Available on: https://www.ti.com/lit/ml/sprt633/sprt633.pdf?ts=1597690676332&ref_url=https%253A%252F%252Fwww.google.com%252F

[32] Vazquez-Corral, J., Galdran, A., Cyriac, P. et al. A fast image dehazing method that does not introduce color artifacts. In Journal of Real-Time Image Processing, Vol 17, pp. 607–622 (2020)

[33] Yang, J., Jiang, B., Lv, Z. et al. A real-time image dehazing method considering dark channel and statistics features. In Journal of Real-Time Image Processing, Vol 13, pp. 479–490 (2017)

[34] Diaz-Ramirez, V.H., Hernández-Beltrán, J.E. & Juarez-Salazar, R. Real-time haze removal in monocular images using locally adaptive processing. In Journal of Real-Time Image Processing, Vol 16, pp. 1959–1973 (2019)

[35] Cheng, K., Yu, Y., Zhou, H. et al. GPU fast restoration of non-uniform illumination images. In Journal of Real-Time Image Processing (2020)

[36] Hernandez-Beltran, J., Diaz-Ramirez, V., Juarez-Salazar, R.: Real-time image dehazing using genetic programming. In Journal of Optics and Photonics for Information Processing, Vol 13, (2019)

[37] Zhang, J., Hu, S. A GPU-accelerated real-time single image de-hazing method using pixel-level optimal de-hazing criterion. J Real-Time Image Proc 9, 661–672 (2014). https://doi.org/10.1007/s11554-012-0244-y

[38] Fattal, R.: Single image dehazing. In Proc. of ACM SIGGRAPH 08 (2008)
[39] Kopf, J., Neubert, B., Chen, B., Cohen, M., Cohen-Or, D., Deussen, O., Uyttendaele, M., Lischinski, D.: Deep photo: Modelbased photograph enhancement and viewing. In ACM Trans. Graph., vol. 27, no. 5, pp. 1–10 (2008)
[40] He, K., Sun J., Tang, X.: Single Image Haze Removal Using Dark Channel Prior. In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 12, pp. 2341-2353 (2011)
[41] Simi, V.R., Edla, D.R., Joseph, J. and Kuppili, V.: Analysis of Controversies in the Formulation and Evaluation of Restoration Algorithms for MR Images, Expert Systems with Applications, vol. 135, pp. 39-59, (2019).
[42] Kuppusamy, P.G., Joseph, J. and Sivaraman, J.: A customized nonlocal restoration scheme with adaptive strength of smoothening for MR images, Biomedical Signal Processing and Control, vol. 49, pp. 160-172, (2019)
[43] Simulink Android Support: Available on: https://www.mathworks.com/hardware-support/android-programming-simulink.html
[44] Android Studio. Available on: https://developer.android.com/studio

Figures

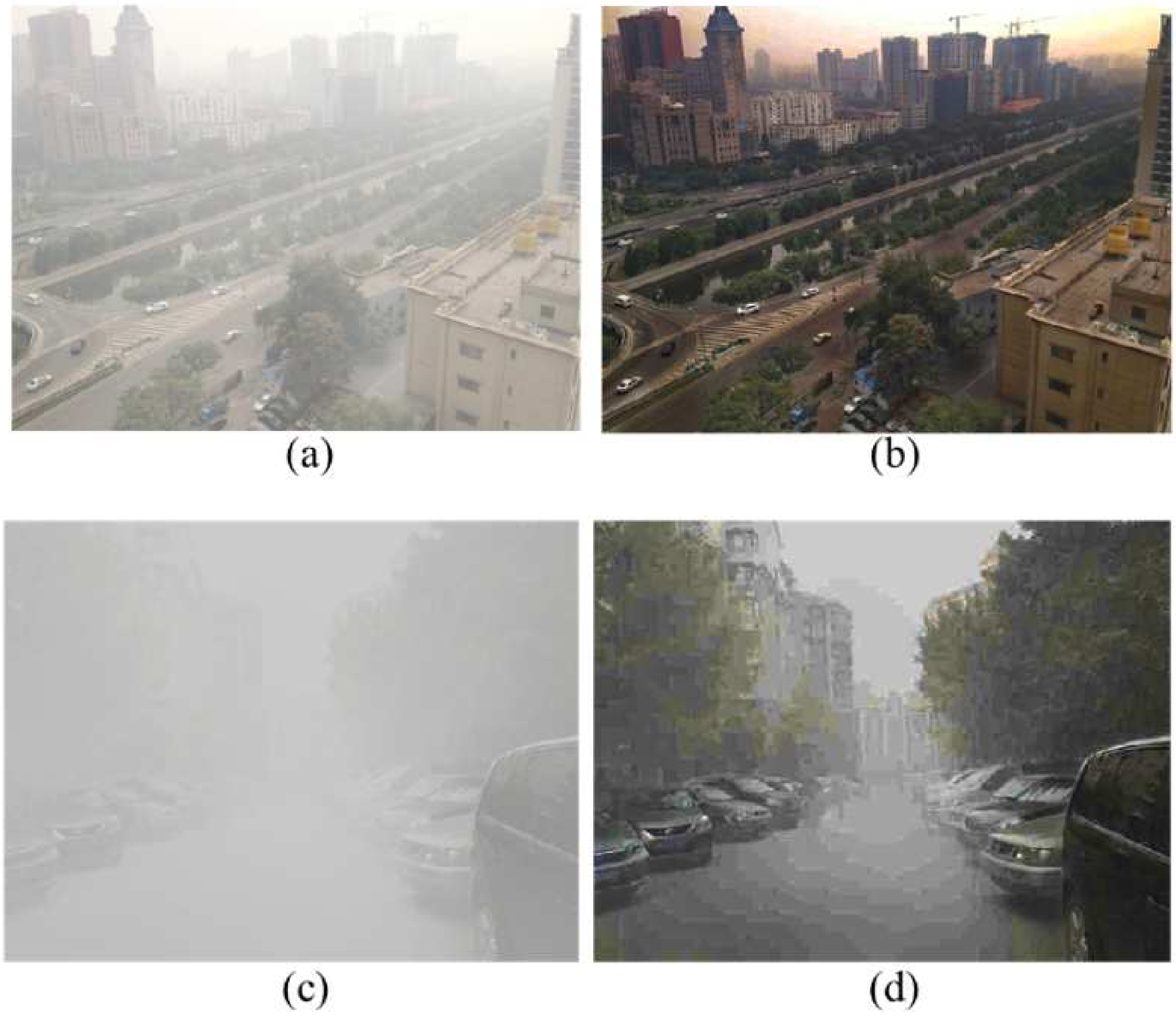

Figure 1

(a) Hazy image (b) Haze-free (dehazed) image of (a) (c) Hazy image (d) Haze-free (dehazed) image of (c)

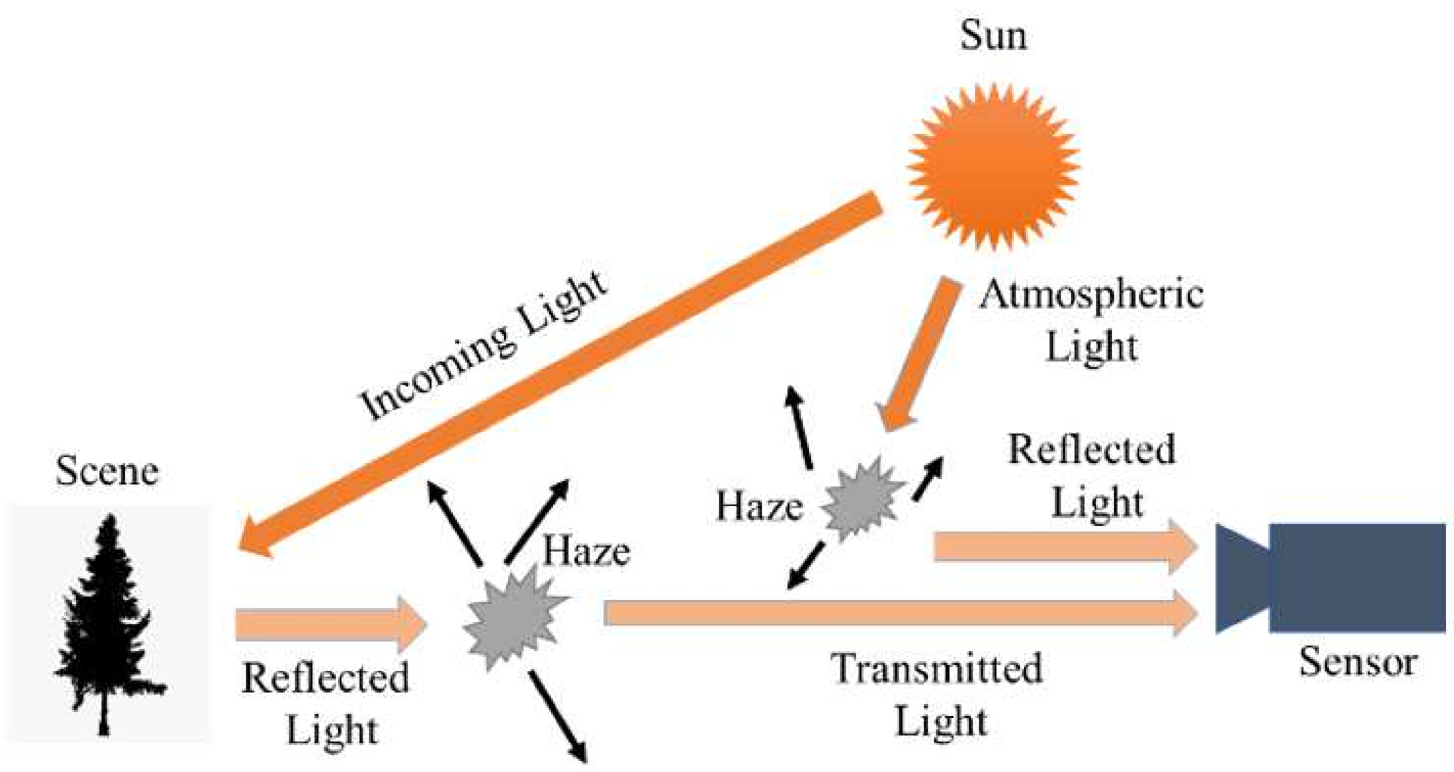

Figure 2

Atmospheric light scattering model

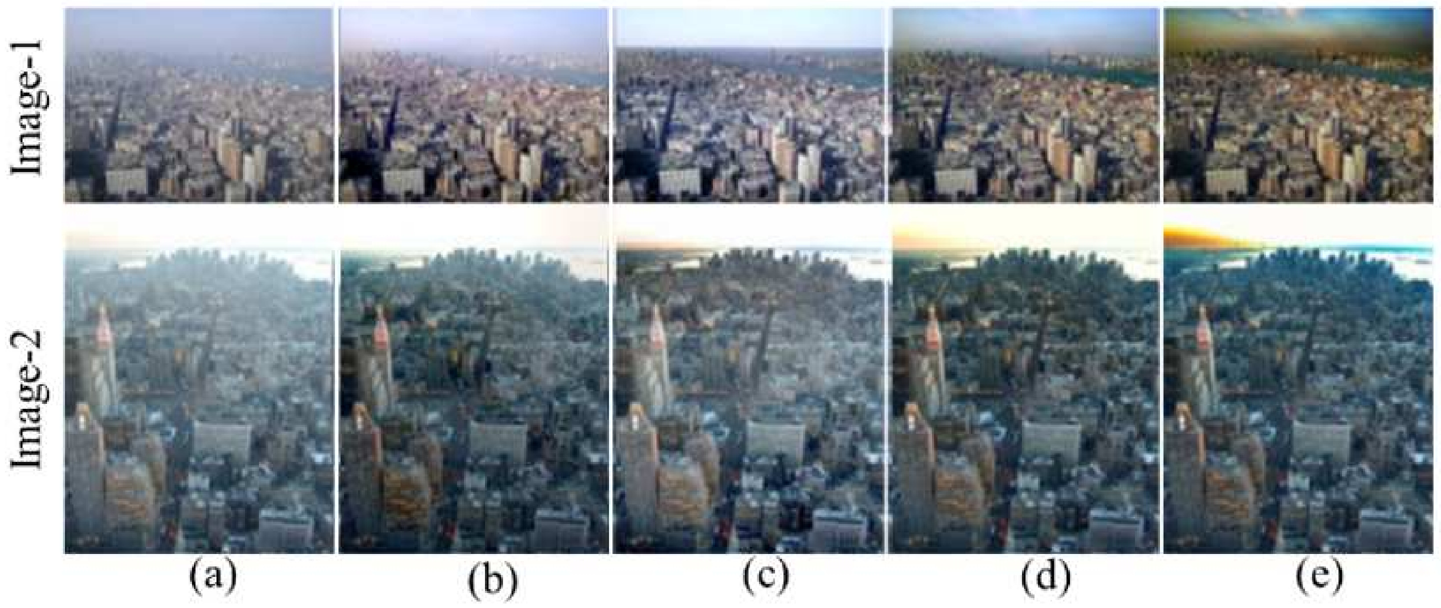

Figure 3

The visual comparison of several methods. (a) Hazy image (b) Fattal's result (c) Kopf's result (d) He's result (e) Park's result.

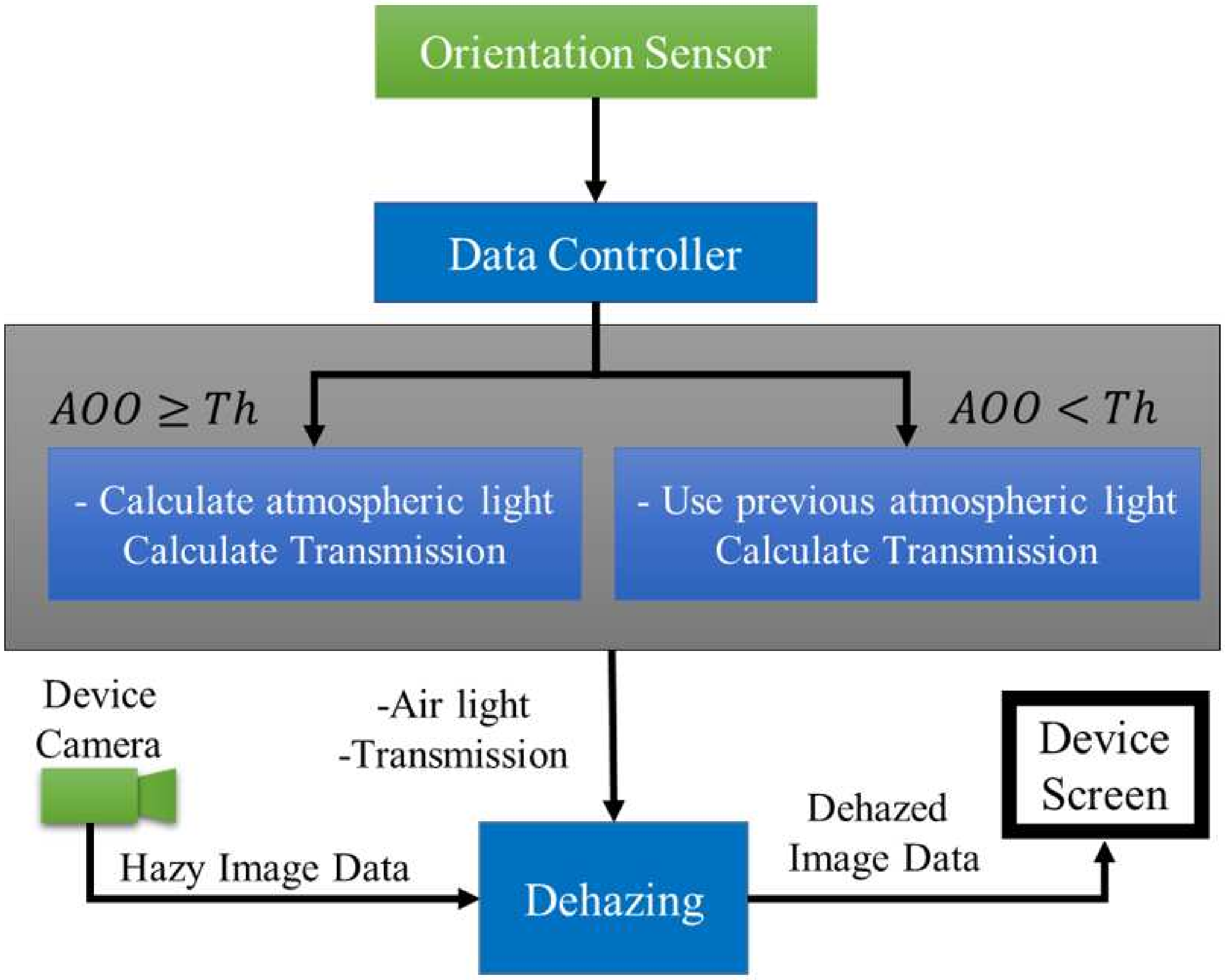

Figure 4

Overall System diagram

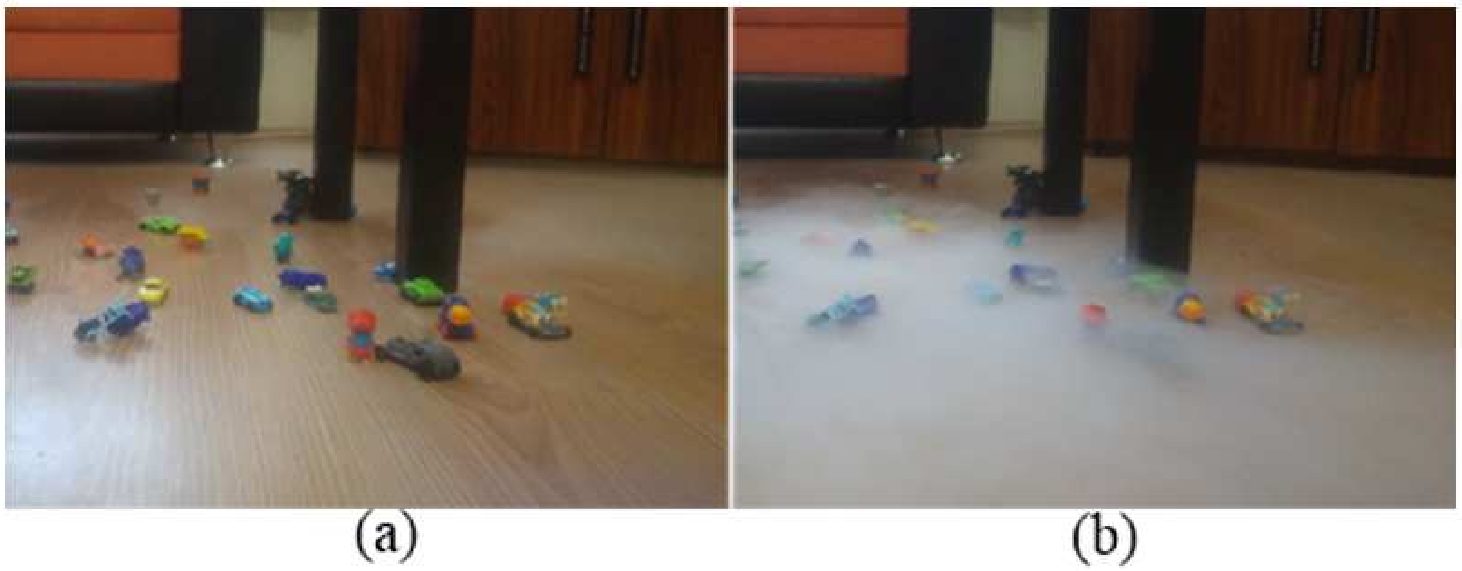

Figure 5

(a) Clear image (b) Hazy image

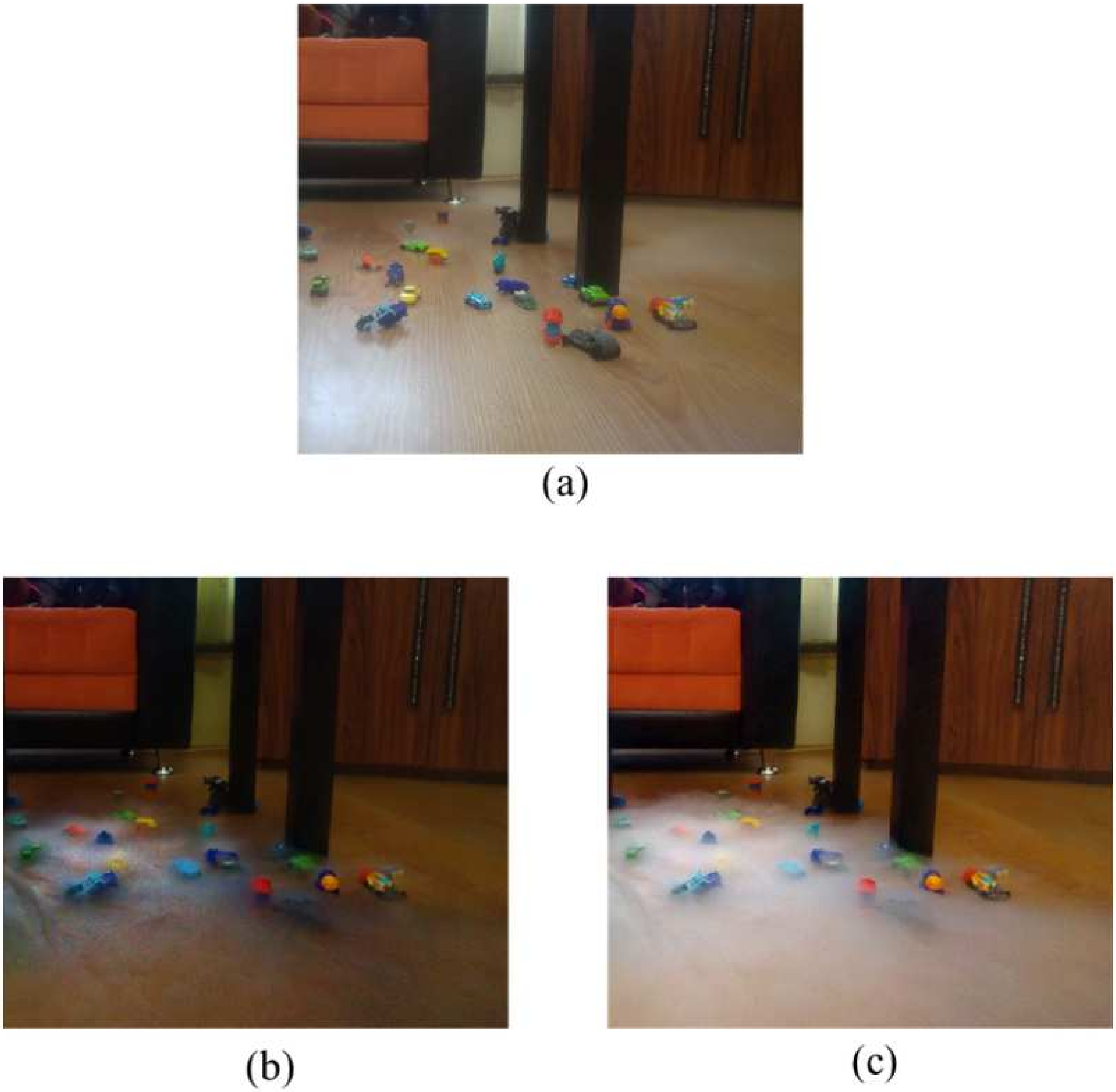

Figure 6

An example threesome. (a) Clear Image (b) Result of direct application of Park’s method (c) Result of proposed approach. (PSNR is just below the threshold).

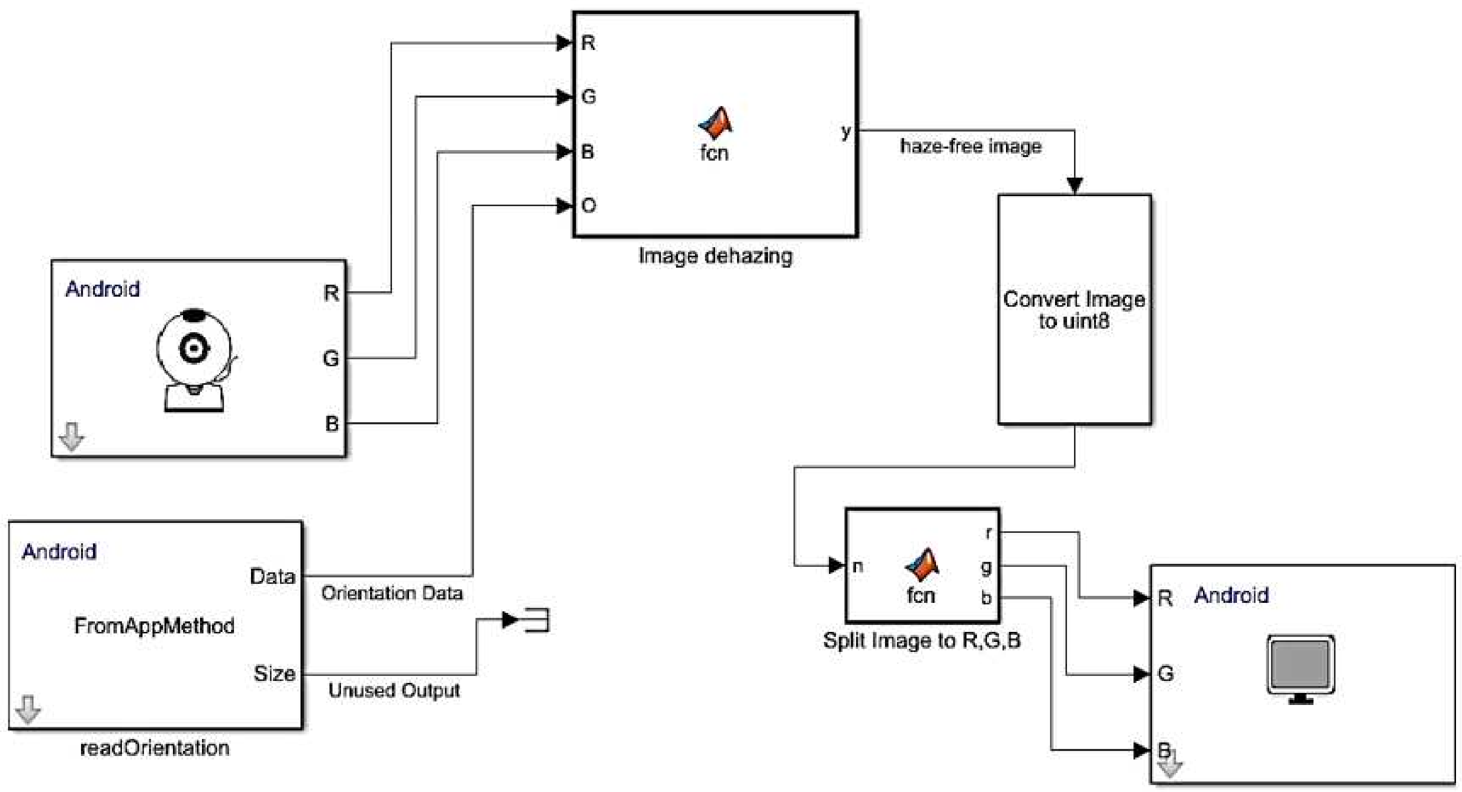

Figure 7

SIMULINK Model for Real-Time Implementation