https://communications.science.ankara.edu.tr

Commun.Fac.Sci.Univ.Ank.Series A2-A3
Volume 62, Number 2, Pages 177-188 (2020)
DOI: 10.33769/aupse.811810
ISSN 1303-6009 E-ISSN 2618-6462

*Received by the Editors: October 16, 2020; Accepted: December 04, 2020*

# TOWARDS REAL TIME IMAGE DEHAZING ON ANDROID OPERATING SYSTEM

Yücel ÇİMTAY

HAVELSAN, Signal and Image Processing Group, Ankara, Turkey

ABSTRACT. Haze is one of the most important effects that degrades the quality of the image and video. This diminishes contrast and reduces visual efficiency. The atmospheric light scattering model (ALS) is commonly used for dehazing. There are two unknowns to be measured in this model: atmospheric light and transmission from scene. Such calculations are not easy, and the calculation of atmospheric light is very time consuming. This condition makes it difficult to dehaze in real time. Although dehazing applications have been applied widely for long time, this work is one of the first tries of real time dehazing on android operating system. This is very important in terms of transforming the real time dehazing to mobile platforms. According to the results, this study can run in near real time, and is able to go towards real time by using more powerful hardware.

## 1. INTRODUCTION

Image and video dehazing are crucial for offline and online computer vision applications needed in security, transportation, video surveillance and military areas. Consequently, the number of studies related to image enhancement has steadily increased in recent years [1]. Image dehazing is a kind of image enhancement, but 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 while the distance between sensor and scene is increasing the thickness of haze also increases and the transmission of media decreases. Similarly, 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 image pairs. Image (a) is a hazy

*Keyword and phrases.* Atmospheric light, hazy imagery, depth map, transmission

yucelcimtay@gmail.com
0000-0003-2980-9228

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). It is obvious from the images that the thickness of haze is more in the second image pair. Therefore, haze removal is less effective, and the visual quality is lower.

There are many ways of image dehazing and they can be grouped in three categories which are (1) contrast enhancement [2-5], (2) restoration [6-10] and (3) fusion based [11-15] methods. Contrast enhancement approaches increase the visual image to some extent; however, they cannot fully eliminate the haze. The subcategories of image enhancement models are histogram enrichment which can be applied locally and/or globally, frequency transform methods: wavelet transform, and homomorphic filtering, and Retinex method: single and multi-scale Retinex [16]. Restoration based methods model the image degradation and applies inverse filtering. By this way the lost information is recovered.

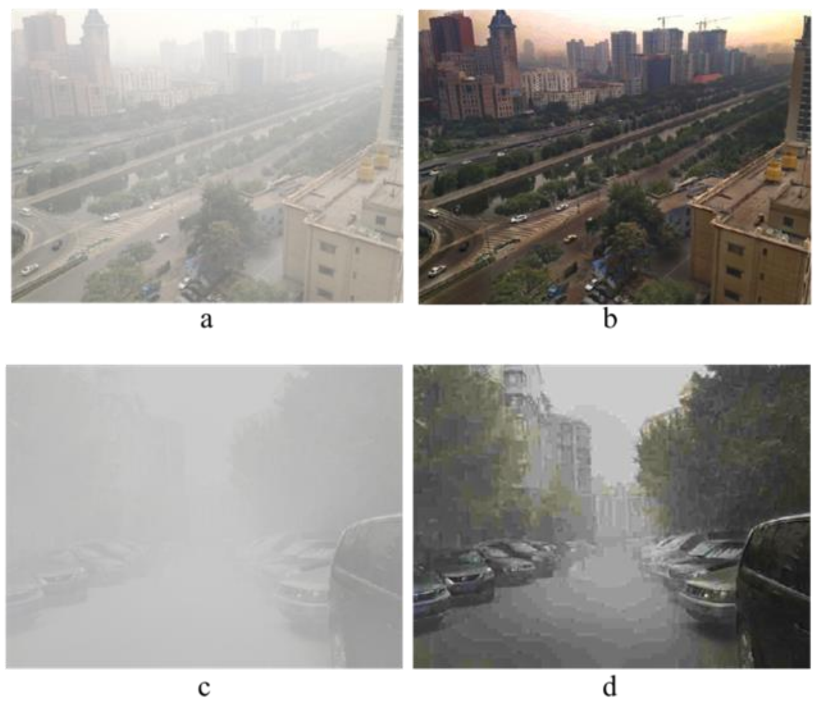


FIGURE 1. Hazy-haze free image pairs

Because this paper is based on the application of image dehazing in real time, we are not going into the specifics for these methods. On the other hand, as the basis of our analysis, we are using the atmospheric light scattering model shown in Figure 2.

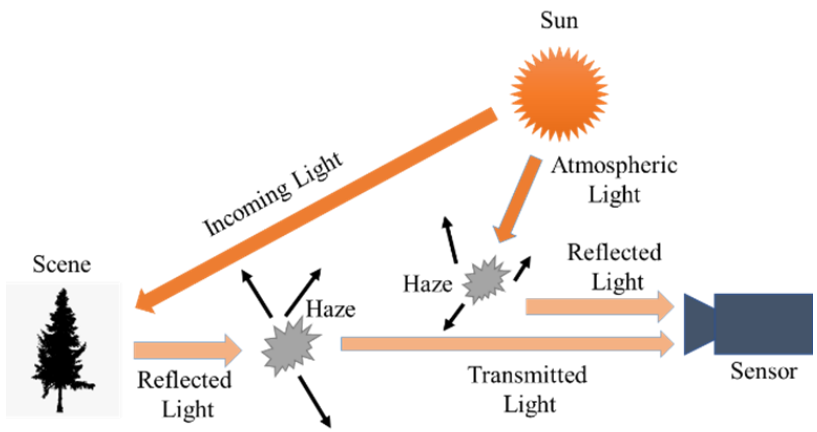


FIGURE 2. Atmospheric light scattering model

Equation 1.1-1.3 express atmospheric scattering model where $I(x,\lambda)$ is the hazy image, $D(x,\lambda)$ is transmitted light through haze (after the reflection from scene) and $A(x,\lambda)$ air light which is the reflected atmospheric light from haze. The sensor integrates the incoming lights and the resultant imagery is the hazy image. In Equation 1.2, $t(x,\lambda)$ is transmission of haze, $R(x,\lambda)$ is the reflected light from the scene and $L_\infty$ is the atmospheric light. Transmission term is expressed as $e^{-\beta(\lambda)d(x)}R(x,\lambda)$ where $d(x)$ is the depth map of scene and $\beta(\lambda)$ is atmospheric scattering coefficient related to capturing wavelength. It can be easily understood from Equation 1.3 is that as the depth from the sensor increases, transmission decreases and vice versa.

$$I(x,\lambda) = D(x,\lambda) + A(x,\lambda) \tag{1.1}$$

$$= t(x,\lambda)R(x,\lambda) + L_\infty\left(1 - t(x,\lambda)\right) \tag{1.2}$$

$$= e^{-\beta(\lambda)d(x)}R(x,\lambda) + L_\infty\left(1 - e^{-\beta(\lambda)d(x)}\right) \tag{1.3}$$

The key point here is, the exact calculation of the term of transmission and atmospheric light. The Dark Channel Prior (DCP) Method [17] is one of the commonly used methods. In simple DCP, the per-pixel dark channel prior is used for haze estimation. On the other hand, for measuring the atmospheric light, quadtree decomposition is applied. Another research that uses the DCP as its basis is [18]. In this study, both per-pixel and spatial blocks are used when calculating dark channel. Recent approaches on image dehazing is mostly based on artificial intelligence models like deep learning [19-21]. In [22] a deep architecture is developed by using Convolutional Neural Networks (CNN) and it adds a new unit to network called "bilateral rectified linear unit". It reports that it achieves superior results compared to previous dehazing studies. The study in [23] employs an end-to-end encoder-decoder CNN architecture to handle haze free images.

There are many successful image dehazing studies in the literature. However, when the focus is real time application, there are always bottlenecks such as the complexity of algorithms, hardware constraints and high financial costs. Nonetheless, there have been several successful studies underway. The study in [24] estimates the atmospheric light by using super-pixel segmentation and applies guidance filter to refine the transmission map. This study mentions that it achieves more accurate results compared to other state of the art models. The study in [25] proposes parallel processing dehazing method for mobile devices and achieves 1.12s for HD (1024x768) imagery on a Windows Phone by using CPU and GPU together. The study in [26] uses DCP but substitutes guided filter with mean filter in order to increase the processing speed. It reports 25 fps over C6748 pure DSP device [27].

The study in [28] converts the hazy RGB image to HSV color 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. The study in [29] conducts 2 level image processing. It first applies histogram enhancement and if the resultant image meets the system requirements then no further action is taken. If it does not, then DCP is used to remove the haze. It saves a lot of time and achieves real time processing.

[30] uses locally adaptive neighborhood and calculates order statistics. By using this information, it produces the transmission map and handles the haze-free image. The study in [31] parallelizes the base Retinex model and decompose the image into brightness and contrast components. For restoration of the image, it applies gamma correction and nonparametric mapping. It reports 1.12ms processing time for 1024x2048 high resolution image on parallel GPU system. The study in [32] 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.

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. This study is one of the first dehazing studies on Android operating system. In this paper we follow DCP-based algorithm [18] on mobile android application.

## 2. Materials and Methods

In this study we concentrate on implementing the algorithm implemented in [18] on Android operating system in real time. This method uses DCP approach, information fidelity, and image entropy 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. Reconstruction is done using Equation 2.

The study in [18] provides very promising accuracy results. The benchmark scores for two different hazy images are given in Table 1 and 2. The images and visual results of different methods are given in Figure 3. According to Table 1 and 2, the comparisons are based on 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.

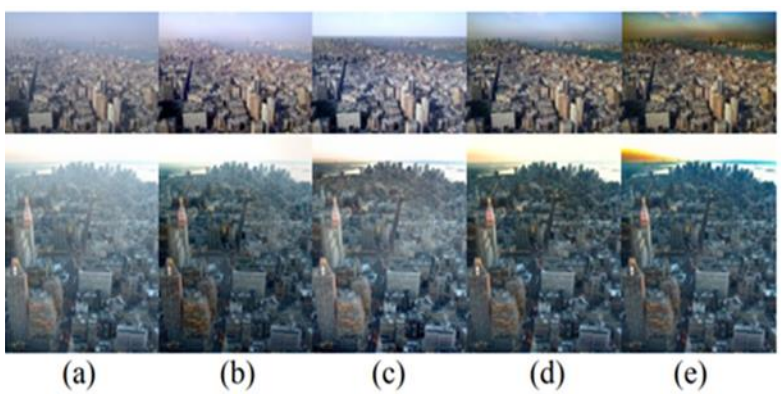


Figure 3. The visual comparison of several methods. (a) Hazy image (b) Study-1's result [33] (c) Study-2's result [34] (d) Study-3's result [35] (e) Result of [18].

The quality of dehazed images improves when q gets smaller and the value of other indicators increases. Although Study-1's method shows good performance in close-distance regions, it is not successful in far-range. Because it cannot remove the haze effectively. As GCF and r scores, Study-2's algorithm provides promising results, however it is not satisfactory for colorfulness and σ scores. In addition, Study-3 has limited performance, since it has good scores only for GCF and σ. The study in [18] is providing better results for overall evaluations.

TABLE 1. Accuracy results for image 1.

| **Index** | **Study-3 [35]** | **Study-2 [34]** | **Study-1 [33]** | **Study [18]** |
|---|---|---|---|---|
| $e$ | 0.02 | 0.02 | 0.11 | 0.32 |
| $r$ | 1.63 | 1.61 | 1.53 | 2.27 |
| $\sigma$ | 0.01 | 1.35 | 1.7 | 0.06 |
| Colorfulness | 963.62 | 455.84 | 652.45 | 1127.42 |
| GCF | 8.63 | 8.53 | 7.87 | 8.49 |

TABLE 2. Accuracy results for image 2.

| **Index** | **Study-3 [35]** | **Study-2 [34]** | **Study-1 [33]** | **Study [18]** |
|---|---|---|---|---|
| e | 0.04 | 0.03 | 0.05 | 0.08 |
| r | 1.39 | 1.4 | 1.28 | 1.41 |
| σ | 0.01 | 0.29 | 9.4 | 0.05 |
| Colorfulness | 509.9 | 390.67 | 387.01 | 706.09 |
| GCF | 6.72 | 6.65 | 5.89 | 6.8 |

The main reason for choosing the study in [18] for real time implementation is its dehazing performance. Since deployment in real time is useless if we're not obtaining good results.

In the literature, as far as we are studying, there is no complete work on dehazing on the android operating system in real time. In this study, we used MATLAB SIMULINK for implementing the image dehazing technique [18]. MATLAB SIMULINK has Android device support for developing and deploying MATLAB codes and MATLAB SIMULINK models [36]. The SIMULINK model we developed is given in Figure 4.

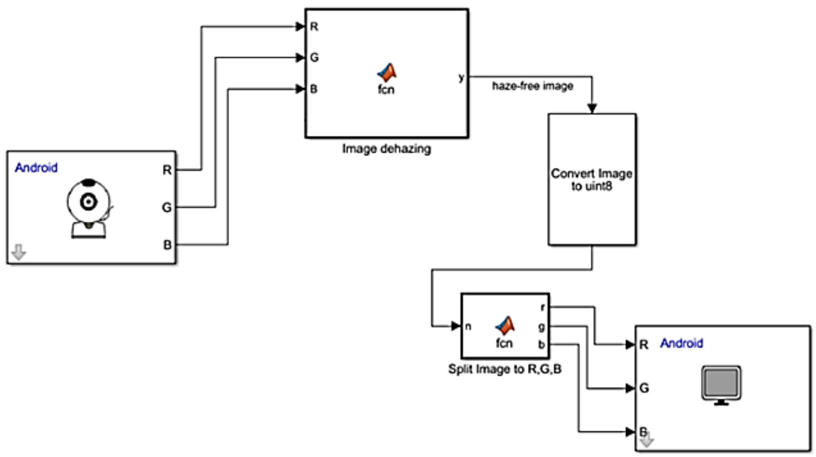


FIGURE 4. SIMULINK Model for Real-Time Implementation

The ‘Android Camera’ block reads live video from device’s camera. The camera resolution can be set by also using this block. Real time video data is fed to the 'Image dehazing' function that runs the dehazing algorithm described in [18]. The next block in Simulink is 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.

We deployed the project on Android device by using ‘Android Studio’ [37]. By the way, The MATLAB codes are transferred to C++ code and a java code is produced for user updates and declare new functions. The android device we used has Qualcomm® Snapdragon™ 665 Octa-core processor, which has frequency up to 2GHz. It has 3GB RAM. The camera’s video resolution is up to 4K at 30 fps.

Figure 5 shows the overall system diagram for real time implementation. Dehazing module reconstructs dehazed image by using camera data and dehazed image data is displayed on device screen.

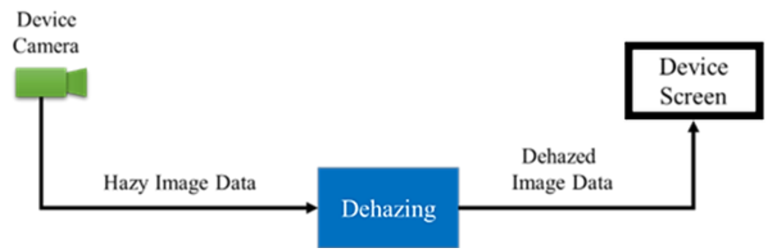


FIGURE 5. Overall System Diagram

## 3. Results

As a real-time experimentation of the study in [18] on Android operating system, this study provides promising processing speed. The results are shown in Table 3 for different imagery resolutions.

TABLE 3. Single Image Processing Time

| Resolution | Proc. Time (sec) per frame |
|---|---|
| 1080p (1920x1080) | 4.36 |
| 720p (1280x720) | 1.95 |
| 480p (864x480) | 0.87 |
| 360p (480x360) | 0.36 |

From Table 3, we can say that the mean processing time for HD imagery is 1.95 seconds per frame. In addition, proposed approach achieves 3fps for 360p video resolution. Figure 6 shows the proposed image dehazing android application (a), hazy image (b) and dehazed image (c). Note that hazy image is displayed on a computer's screen, and the android device's camera is capturing the hazy image from computer's screen. Therefore, the dehazed image on android device's screen may be look low resolution. (despite it is high quality normally).

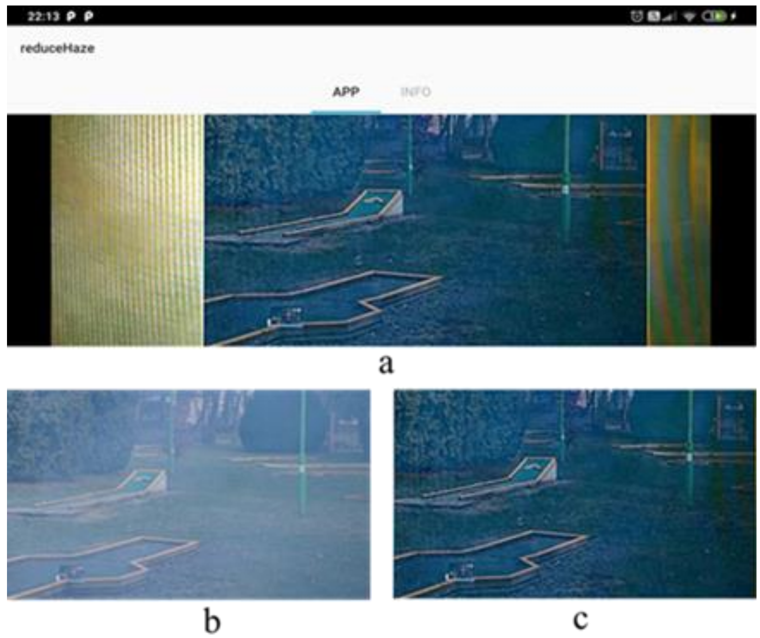


FIGURE 6. Android Dehazing Application (a) Device Screen (b) Hazy Image (c) Dehazed Image.

## 4. CONCLUSIONS

In this study, we implemented an image dehazing method which is described in [18] on Android operating system. The contribution of our study is implementing [18] in real time on Android operating system. We can achieve good processing time results though we use only CPU and a hardware with limited power. Processing time results show that our method can be applied in near real time on android devices. If the system is empowered in terms of hardware specifications, the processing time will decrease automatically, and the system will run in real time.

The future work should be using GPU and/or CPU and GPU together. Furthermore, similar implementation should be done on IOS devices. Another important aspect is that transmission maps may be estimated more accurately by using stereo imaging which enables estimation of depth map. Finally, image dehazing should be implemented on more powerful hardware in order to handle real time processing speed.

## REFERENCES

[1] Wang, W., Yuan, X., Recent advances in image dehazing, *IEEE/CAA Journal of Automatica Sinica*, 4(3) (2017), 410–436.

[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, *Proc. IEEE Int. Conference Acoustics Speech and Signal Processing*, (2010), 1378−1381.

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

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

[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, *Proc. 19th International Conference Geoinformatics*, (2011), 1−4.

[6] Ilgın, H., Akbulut, A., An Artifact Reduction Method For Block-Based Video Coding, *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, 62 (2020 ), 1-13.

[7] Bülbül, A., Haj Ismail, S., Visually Enhanced Social Media Analysis Of Refugees In Turkey, *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, 60 (2018 ), 83-102.

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

[9] Fang, S., Xia, X. S., Xing, H., Chen, C. W., Image dehazing using polarization effects of objects and airlight, *Opt. Express*, 22(16) (2014), 19523−19537.

[10] Galdran, A., Vazquez-Corral, J., Pardo, D., Bertalmio, M., Enhanced variational image dehazing, *SIAM Journal of Imaging Science,* 8(3) (2015), 1519−154.

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

[12] Ancuti, C.O., Ancuti, C., Single image dehazing by multi-scale fusion, *IEEE Transaction on Image Processing,* 22(8) (2013), 3271−3282.

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

[14] Guo, F., Tang, J., Cai, Z.X., Fusion strategy for single image dehazing, *International Journal of Digital Content Technology and Its Applications,* 7(1) (2013), 19−28.

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

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

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

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

[19] Li, J., Li, G., Fan, H., Image Dehazing Using Residual-Based Deep CNN, *IEEE Access,* 6 (2018), 26831-26842.

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

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

[22] Cai, B., Xu, X., Jia, K., Qing, C., Tao, D., DehazeNet: An End-to-End System for Single Image Haze Removal, IEEE Transactions on Image Processing, 25(11) (2016), 5187-5198.

[23] Rashid, H., Zafar, N., Javed Iqbal, M., Dawood, H., Dawood, H., Single Image Dehazing using CNN, *Procedia Computer Science,* 147 (2019), 124-130.

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

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

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

[27] 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

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

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

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

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

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

[33] Fattal, R., Single image dehazing, *Proc. of ACM SIGGRAPH,* 08 (2008).

[34] Kopf, J., Neubert, B., Chen, B., Cohen, M., Cohen-Or, D., Deussen, O., Uyttendaele, M., Lischinski, D., Deep photo: Modelbased photograph enhancement and viewing, *ACM Trans. Graph.,* 27(5) (2008), 1-10.

[35] He, K., Sun J., Tang, X., Single Image Haze Removal Using Dark Channel Prior, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 33(12) (2011), 2341-2353.

[36] Simulink Android Support: Available on: https://www.mathworks.com/hardware-support/android-programming-simulink.html.
[37] Android Studio. Available on: https://developer.android.com/studio.