DOI:10.28974/idojaras.2023.4.2

IDŐJÁRÁS

Quarterly Journal of the Hungarian Meteorological Service
Vol. 127, No. 4, October – December, 2023, pp. 447–457

Evaluating dehazing techniques on artificial and satellite land surface images

András Fridvalszky*, **Balázs Tóth**, and **László Szécsi**

Department of Control Engineering and Information Technology,
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics
Műegyetem rkp. 3, H-1111, Budapest, Hungary

**Corresponding author E-mail: fridvalszky@iit.bme.hu*

(Manuscript received in final form September 12, 2023)

Abstract— Many image-based recognition tasks are highly susceptible to different types of natural phenomena like foggy weather, snow, or rain. The participating media will likely obscure important details necessary for these algorithms to work correctly. Still, these aspects could be recovered in certain situations with prior information about the underlying light interactions. This could be done with certain heuristics or with the nowadays popular deep-learning based methods. In this paper, we review and compare the results of two approaches to remove or scale down the effects of foggy weather. We also examine how these results can be applied to high resolution satellite images of land surfaces.

Key-words: dehazing, fog, clouds, satellite images

1. Introduction

Fog removal or dehazing is a popular problem nowadays. One example of usage is autonomous vehicles and the corresponding computer vision task, as the artificial intelligence must operate the vehicle with high safety even in the worst visibility conditions. Instead of trusting the machine learning algorithms to solve these problems using diverse data sets, we can help these algorithms by formulating a separate problem. That is to produce a clear output image without the visibility impairing effect from a single input image. It is also important that we must do this with high performance on multiple images, as this is just basically a preprocessing step. The same idea also applies to satellite images. These images are often used to survey, observe, and analyze agricultural areas or vegetation changes, but foggy weather or clouds can be detrimental to their usefulness.

In this paper, we are not introducing another novel algorithm to solve this problem, as there are existing approaches with multiple different ideas. Instead, we turn our attention to the evaluation and analysis of a deep-learning based solution, comparing it to a more traditional idea. We also describe the synthetic training data generation used to train the neural network and how it affects the results.

2. Fog removal techniques

Now we shall discuss the previously mentioned approaches. The first method is based on the work of *He et al.* (2010) and uses a dark channel prior to estimate the contribution of haze that is present on the image. The other technique uses a convolutional neural network to compute an image without fog from the original input picture. It was designed and proposed by *Li et al.* (2017).

2.1. Approach with dark channel prior

Haze (or fog) reduces the light radiance L reflected off surfaces according to the haze transparency (also called transmission). It also adds its contribution to the image, called airlight (denoted by A). The result of radiance reduction and added color is that the original object's radiance is faded, and image value I is different from surface radiance L. Dehazing aims to reconstruct the original surface color L from I as if haze was not there. Dehazing is an ill posed problem, because there is no information in the image about the haze and airlight and where they can be seen. Thus, certain assumptions must be made in order to recover this missing information from the image. In the implemented method, the most fundamental assumption is the “dark channel existence”, which means that the neighborhood of every pixel contains a pixel that should have zero radiance without haze on the examined wavelength. This assumption is reasonable because of the high

frequency of highly saturated color objects, and also because of the existence of shadows. The approach is defined as follows:

$$D = min_B(I) \ , \tag{1}$$

where *D* is the dark channel value, *I* is the image value, *B* is the color channel of RGB image.

Unlike in the original paper proposing the dark channel prior, we evaluate the dark channel and the airlight separately on the red/green/blue wavelengths. This approach is less robust than the original but allows us to handle colored haze, fog, and homogeneous smoke.

The dark channel, according to our assumption, should be zero in haze-free, real-life images. In the case of haze, the dark channel is modified by the airlight, which allows us to estimate the airlight and the haze contributing to the image. More formally, we assume that the minimum radiance must be the haze contribution in any neighborhood.

Knowing the airlight, the transmission can be estimated, which is further refined by the color variation of the original image with the guided filter. Finally, we can subtract the haze and amplify the remaining colors. In the following subsections, these steps are analyzed in detail.

2.1.1. Dark channel computation

The minimum radiance in the neighborhood of each pixel is called the dark channel. The neighborhood is defined by a box filter. For efficiency reasons, we exploit the separability of minimum and average filters, i.e., the 1D versions are executed two times, once for horizontal, and once for vertical direction. This way, the complexity can be reduced from N^2 to *2N*, where *N* is the edge size of the box.

2.1.2. Airlight estimation

The second step is to find out the airlight, i.e., the atmospheric illumination. If no object were visible in a given direction, then we would see this airlight. So pixels are candidates for showing the airlight if their dark channel value *D* is high, i.e., the neighborhood is not dark, and its intensity *I* is also high. Based on these two parameters, we obtain a single comprehensive parameter *F* that describes both components as

$$F = \frac{2DI}{D+I} \ , \tag{2}$$

and select the average intensity *I* of pixels that have the highest *F* parameter:

$$A = \mathrm{avg}_{F \text{ in top } 0.1\%}(I) \ . \tag{3}$$

Our implementation builds up the histogram of the comprehensive parameter *F* on the GPU (graphic processing unit) and selects the top elements on the CPU (central processing unit). As histogram building requires atomic additions, it is done in two steps. First, GPU multiprocessors create partial histograms in their shared memories, and then these partial histograms are merged into a global histogram stored in the main memory. Having read the histogram back to the CPU memory, we read its bins starting from the highest intensity until 0.1% of the number of pixels are included. The airlight is the average of the intensities belonging to this top 0.1% category.

2.1.3. Approximate transmission estimation

Having the airlight *A*, we can determine the transparency, i.e., transmission *t* of the haze at each pixel. Note that, in the dark channel, we only have the contribution of the airlight, and the own contribution of the surface is assumed to be zero, i.e.,

$$D = (1 - t)A\,. \tag{4}$$

From this, the transmission *t* associated with this pixel is

$$\mathrm{t} = 1 - \frac{\mathrm{D}}{\mathrm{A}}\,. \tag{5}$$

Removing all haze typically results in unnatural images, so we introduce a removed haze parameter ω of [0%, 100%] with a default value as 95%. So the transmission is obtained as

$$t = 1 - \omega\frac{D}{A}\,. \tag{6}$$

The transmission computed this way suffers from resolution problems since the dark channel describes a neighborhood, so does transmission *t*. However, at object boundaries the depth value and consequently the transmission can change abruptly, so the edges of the transmission map must be corrected. For this, a guided filter is implemented that corrects the transmission map using the second derivative of the original image. We use two different guided filter implementations here depending on the size of the filtering kernel because of performance considerations. The first one uses on-the-fly box filtering, the second uses integral images aka Summed Area Tables or (SATs). The second derivatives are computed by the following formula:

$$\begin{aligned} I_{\mathrm{grad2}}[x,y] = \quad & |I[x+1,y] - 2I[x,y] + I[x-1,y]| + \\ & |I[x,y+1] - 2I[x,y] + I[x,y-1]|\,, \end{aligned} \tag{7}$$

where x and y represent the image coordinates.

2.1.4. Recovering the original image

Having obtained the refined transmission with guided filtering, the final image is recovered. If the original radiance is L, then we would see I through haze of airlight A and transmission t:

$$I = tL + (1 - t)A \ . \tag{8}$$

From this, original intensity L is

$$L = A + \frac{I-A}{t}. \tag{9}$$

Eq.(9) is numerically unstable for small transmission values t, therefore, we limit it with a t_{min} minimum value, which also limits the power of recoverability. Its reciprocal is called amplification and is 20 for 16 bit images and 10 for 8 bit images by default.

Additionally, only for the floating point implementation, the limited transmission is exponentiated differently on the three color channels to compensate the bluish scattering of air. The exponents are $1+3r$ on red, $1+2r$ on green, and $1+r$ on blue, where r is the blue removal parameter.

2.2. Deep learning based approach

With recent advancements and success of neural networks and deep learning in multiple fields, it is reasonable to try to use them for dehazing (*Yang et al*., 2018; *Song et al*., 2017). In our implementation, the main idea is that instead of an end-to-end network, we are using a transformed atmospheric scattering equation and we incorporate it directly into the model (*Li et al.,* 2017). The network has a standard convolutional structure, and it is relatively small (less than 2000 parameters). The training generally converges with at most 10 epochs. Details of the network can be found in the cited publication. The main problem is obtaining the training data, which we will address in the following section.

3. Training data acquisition

To adequately train the neural network, we need a large amount of training data. The data set should contain image pairs in the form of pictures with and without fog. Collecting a large enough real-life data set is clearly an unrealistic goal. It would be inevitable that the image pairs would have differences, that are unrelated

to the fog, like disappearing objects. It is a much more plausible idea to synthetically generate the images. If we consider that the network's goal is to remove the effect of the fog, and we can properly model its behavior in our simulation, then it is reasonable to expect the trained network to also work on real-life images as well.

We used the Sponza scene with an artificially added inhomogeneous fog to generate 10.000 images (with 800×600 pixels resolution). The images were stored in a 16-bit format motivated by the discussion in *Section 2.1*. We will examine the algorithm behind the fog simulation and visualization in *Section 3.1*.

This is not the case with satellite images which are taken regularly above the same area. They can also be aligned reasonably well with each other based on camera parameters, current time, and position of the satellite. Images are from the Sentinel-2 satellites with a 10m/pixel resolution. We manually selected pairs of aligned images which were taken shortly after each other, so changes to the landscape were minimal at this precision. One was a clear shot without obscuring participating media, while the other had partially transparent clouds. The images were sliced into square tiles of size 256×256 pixels. A pair of training images is shown in *Fig. 1*.

Fig. 1. A pair of images used to train the neural network.

3.1. Fog generation

Our goal is to simulate and visualize fog in a physically correct way. The implemented algorithm is based on the works of *Wronski* (2018). The process can

be broken down into three parts: fog simulation and storage, light propagation and processing, and finally usage during object rendering.

The algorithm accounts for extinction, out-scattering, and in-scattering of light with a single scattering model. It is a volumetric approach, thus it can work with inhomogeneous fog. The basic idea is to use raymarching, where each view ray is sampled throughout the volume. Extinction and out-scattering are calculated by sampling and accumulating the fog, while we can take into account the in-scattering by accumulating the incoming light for each light source. The problem is that for general camera orientations and positions, the ray marching process can be too slow on the GPU.

To mitigate this issue, we are using voxels aligned with the camera frustum. For the first part, we can use a compute shader to write the density of the fog into each voxel. The fog could be modeled by various methods (e.g., a particle system, grid based simulation, etc.), but in our case we simply used an animated simplex noise. In the second part, ray marching must be done. However, because we are using the previously described data structure, we can do it with one compute shader pass, where each invocation works on blocks of pixels instead of individual rays.

Incoming light is simulated by using shadow maps to accumulate light in the voxels (in the first part), and then it is propagated in the second part (while properly accounting for extinction). Multiple scattering could be handled by an iterative algorithm that propagates light in all direction (*Premože et al.,* 2004) inside the volume in a separate pass, but we did not implement this.

In the end, we have a data structure where every voxel stores an approximation of accumulated fog density from the direction of the camera and the in-scattered radiance from nearby light sources (towards the camera). Using the data structure now is simple, because we can use linear filtering to sample it in the fragment shader during object rendering. Reflected radiance is decreased according to the accumulated fog density, while in-scattering is simply added to the final radiance. The results are shown in *Fig 2*.

Fig. 2. A pair of rasterized images: original and a version with fog.

An important observation is that during camera movement, temporal aliasing can occur as the frustum-aligned voxels are moving. To mitigate this, it is important to maximize the utility of the allocated memory. By the nature of the phenomena, voxels that are close to the observer has larger effect on the results than those that are further away. By using a denser resolution closer to the camera, we can improve the visuals of the fog. Thus, we used an exponential depth distribution instead of a linear one.

The algorithm does not account for blocking geometry, so in certain scenarios fog and light bleeding can occur. This could be addressed by a kind of adaptive voxelization, but it is unclear, how it would work together with the camera frustum alignment.

4. Results

Our results for synthetic images are shown in *Fig 3*. Here we are showing the trained model beside the dark channel prior approach. We can see that this model can successfully decrease the foggy effect. Colors are recovered, and the bluish tint now appears in the correct place. Compared to this model, the dark channel prior technique removes more haze from the image, but some artifacts are left behind. In the second row, an incorrect lightening occurs on the left part of the image. The borders have some artifacts too, and they also appear in the corners of the geometry. In the beginning we used three channel training data, but this resulted in the image shown in *Fig. 4*. We are still investigating this kind of false coloring artifact. We believe that this was caused by our unbalanced training data. To prevent this, we used only the intensity of the ground truth images during the training. This way the network would not prefer one color over the other.

Fig. 3. From left to right: hazy image, dark channel prior, neural network model, ground truth.

Fig. 4. Results with one channel (left), and with three channels (right) training data.

We also tested these methods on some real-life images. Results are shown in *Fig. 5*. The dark channel prior again has some serious artifacts, generally around edges and on the sky, but it also successfully recovers details that are unseen in the original image. Similarly to the previous comparison, the neural network model decreases the effects of the fog, but now more problems appear. Closer objects, where the fog has no effect, have some serious artifacts like whitening or unnatural tint.

Fig. 5. From left to right: hazy image, dark channel prior, neural network model.

We also trained two separate networks on satellite images. It was the same network discussed earlier, except that one incorporated the atmospheric scattering equation as before (*Network A*), while the other omitted this and followed an end-to-end approach (*Network B*). Results are shown in *Fig. 6*. The difference between *Network A* and *B* is that the latter replaces the fully covered (by clouds) parts of the image with generic background that is similar to the general tone. It also applies a considerable blur. *Network A* keeps the opaque clouds intact but tries to tone down the effect of the transparent ones. Still, these are recognizable in the results. Compared to this, the traditional approach provides a vastly different result. The clouds are more prominent, and many artefacts appear.

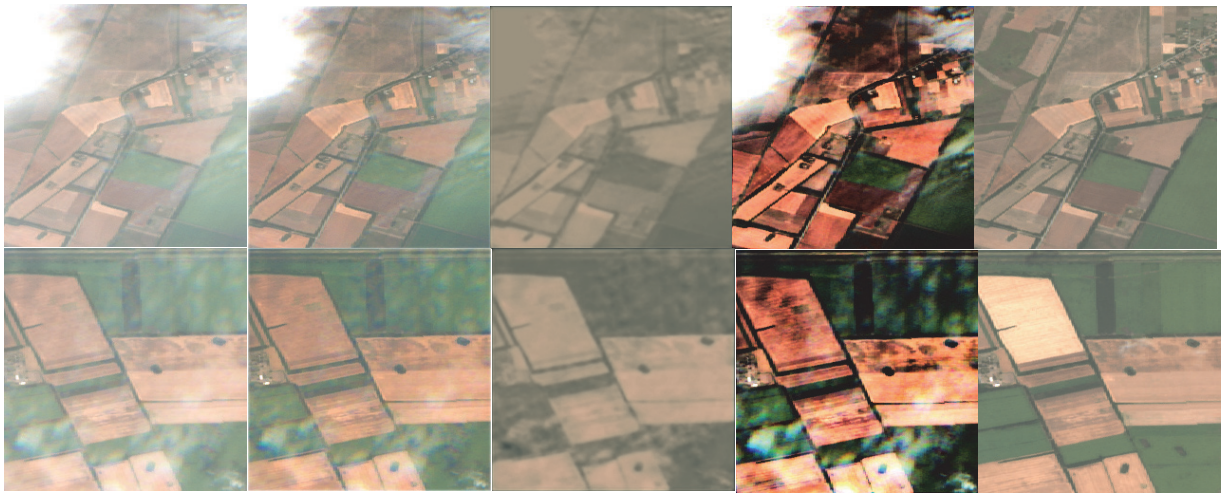

Fig. 6. From left to right: hazy image, Network A, Network B, dark channel prior, ground truth.

5. Conclusion and future work

We have discussed two models for fog removal and a technique for artificial fog rendering. Based on our results, the neural network approach has potential to be used as a dehazing tool, but the variety and balance of the training data is paramount, therefore, we will enhance these aspects for future research. The combination of neural networks and the dark channel prior based method could be also viable, but the performance will likely suffer.

The results on satellite images are promising for the following reasons. An important use-case for these images is to follow the changes in vegetation and in

the landscape. Clouds obscure parts of the image, but generally we have an idea about what we expect to see under them based on previous images of the same area (without clouds). This leads to the following problem: we would like to *predict* the missing parts of the image based on prior information (previous images). It comes naturally to use neural networks to solve this problem, and our results suggest (especially Network B) that a small network can accomplish this with proper additional inputs.

Acknowledgments: The authors would like to express their gratitude towards *Dr. Kálmán Kovács*, *Dr. Dániel Kristóf,* and the *Lechner Knowledge Center* for providing advice and support with the acquisition of satellite images. The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program. The GPUs have been donated by NVIDIA.

References

He, K., Sun, J. and *Tang, X.,* 2010: Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence, 33*(12), 2341–2353. https://doi.org/10.1109/CVPRW.2009.5206515

Li, B., Peng, X., Wang, Z., Xu, J., and *Feng, D.,* 2017: Aod-net: All-in-one dehazing network. Proceedings of the IEEE international conference on computer vision, 4770– 4778. https://doi.org/10.1109/ICCV.2017.511

Premože, S., Ashikhmin, M., Tessendorf, J., Ramamoorthi, R., and *Nayar, S.,* 2004: Practical rendering of multiple scattering effects in participating media. Proc. of Eurographics Symposium on Rendering, 2(10.2312):363–374. https://doi.org/10.2312/EGWR/EGSR04/363-374

Song, Y., Li, J., Wang, X., and *Chen, X.,* 2017: Single image dehazing using ranking convolutional neural network. *IEEE Transactions on Multimedia 20*(6), 1548–1560. https://doi.org/10.1109/TMM.2017.2771472

Wronski, B., 2018: Volumetric Fog and Lighting. GPU Pro 360 Guide to Lighting.

Yang, D. and *Sun, J.,* 2018: Proximal dehaze-net: A prior learning-based deep network for single image dehazing. Proceedings of the European Conference on Computer Vision (ECCV), 729–746. https://doi.org/10.1007/978-3-030-01234-2_43