Digital Image Processing

Final Project

A Fast Image Dehazing Algorithm Using Morphological Reconstruction

Shervin Halat

98131018

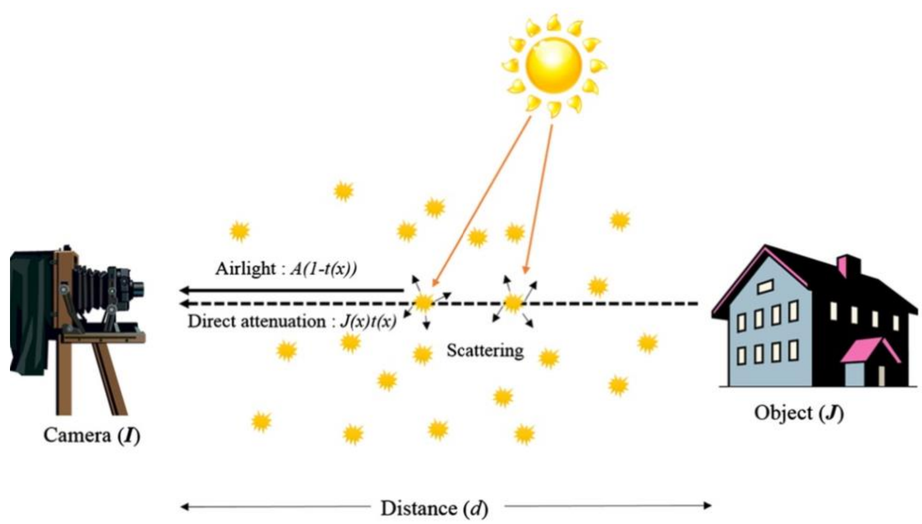
Introduction

Image denoising, dehazing and deblurring is an old task overcame by many methods proposed in the past. To be precise, Image dehazing has some limitations. The limitations include a need for an external depth estimator or other sensors to measure a new data type to ensure perfect dehazing. Although, in most of the applications, the camera image needs to be dehazed with single image as the input. To resolve this problem, some scientists proposed the concept of dark channel prior (DCP) which states that natural images (excluding the sky) have a least a channel with an intensity close to zero. This phenomenon then helps the recovery of the transmission function of the environment with the existence of haze in the non-sky regions. In the following paragraph, the proposed method in the paper will be explained in detail.

Methodology

Atmospheric scattering Model is indicated in the following formula and picture:

$$I(\chi) = J(\chi)t(\chi) + A\left(1 - t(\chi)\right)$$



In this formula A Is the atmospheric light, t is the transmission function which in haze conditions is defined by depth and scattering coefficient. To recover J we need to have t and A.

Due to DCP, the following formula should indicate zeros values in the natural conditions in non-sky regions

$$I^{dark}(\chi) = \min_{C \in R,G,B}\left(\min_{y \in \Omega(\chi)}\left(I^c(y)\right)\right)$$

In the aforementioned equation for first a minimum of Although, Due to the existence of haze I^{dark} is non zero in haze regions, and Atmospheric light can be estimated from the maximum intensity of this dark channel prior image which corresponds to transmission value close to zero. As mentioned, A is approximated due to the following equation.

$$A = \max \sum_{C=1}^{3} I^{C} \left(\arg \max_{\chi \in (0.1\% * h * w)} \left(I^{dark}(\chi) \right) \right)$$

Now, if we had t(x) we could have approximated non-hazy image with the following equation:

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

This paper proposes a novel morphological method to calculate DCP image and t(x), this method is demonstrated in the following diagrams:

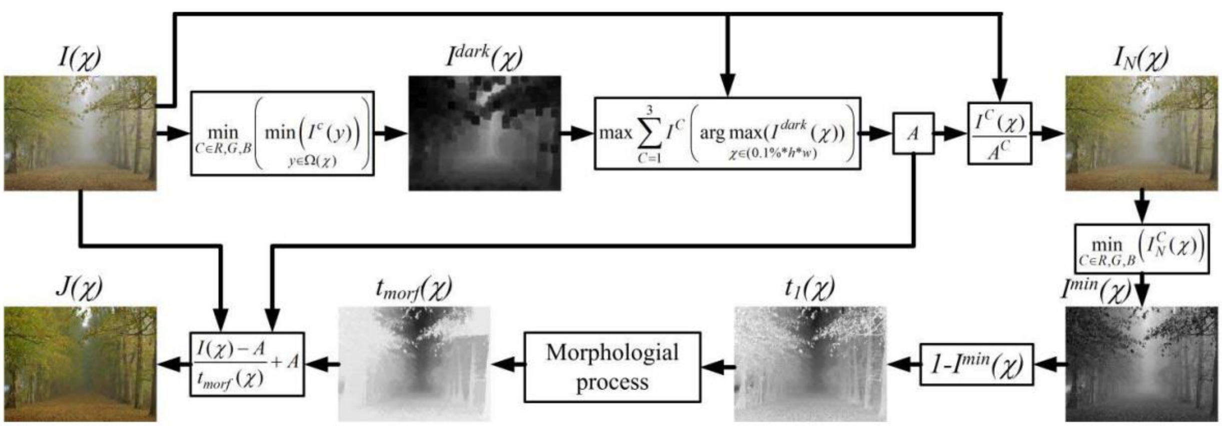


Fig. 3. Flowchart of the proposed algorithm.

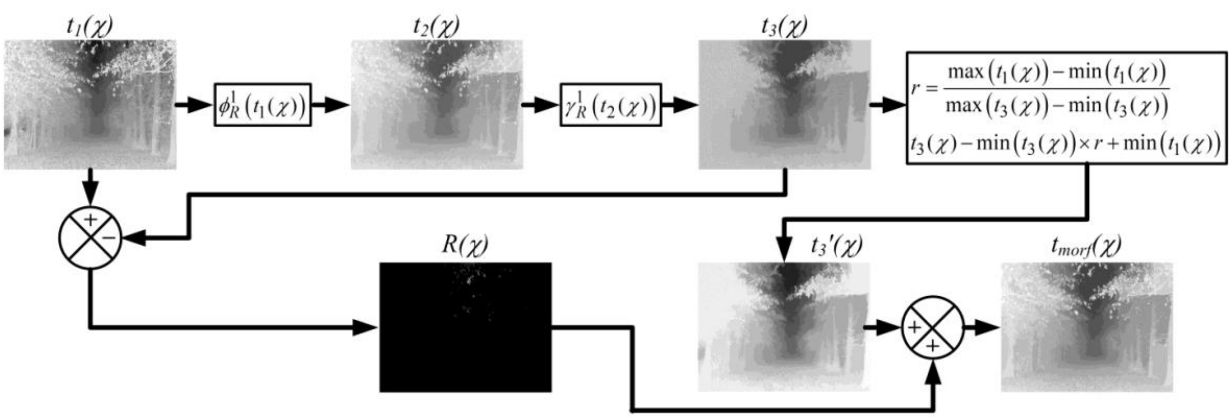


Fig. 4. Morphological process.

Where ϕ and γ are morphological closing and opening reconstruction operators discussed in the course (binary versions discussed, although the grayscale version is not so different), formulated as the following:

$$\gamma_R^{(n)}(I) = R_I^{\delta}[\varepsilon^{(n)}(I)]$$
$$\phi_R^{(n)}(I) = R_I^{\varepsilon}[\delta^{(n)}(I)]$$

Where grayscale erosion and dilation are ε and δ and R is defined on geodesic dilation and erosion, defined as the following:

$$\varepsilon_g^{(1)}(I) = \varepsilon^{(1)}(I) \vee G$$

$$\varepsilon_g^{(n)}(I) = \varepsilon_g^{(1)}[\varepsilon_g^{(n-1)}(I)]$$

$$\delta_g^{(n)}(I) = \delta_g^{(1)}[\delta_g^{(n-1)}(I)]$$

$$\delta_g^{(1)}(I) = \delta^{(1)}(I) \wedge F$$

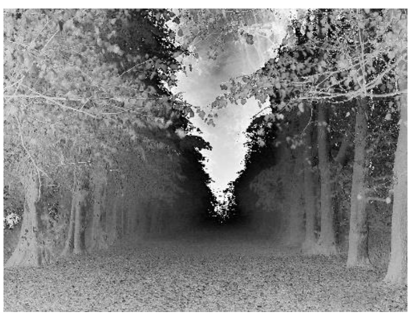
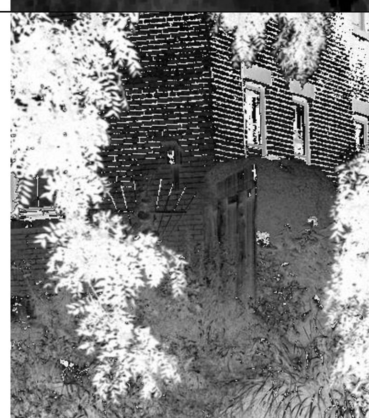
The superscript corresponds to the number of iterations and F is the mask. As a result for example for the image t_2, the mask is t1 and marker is the grayscale dilated t1 and from that point with geodesic erosions, we calculate the t_2 image.

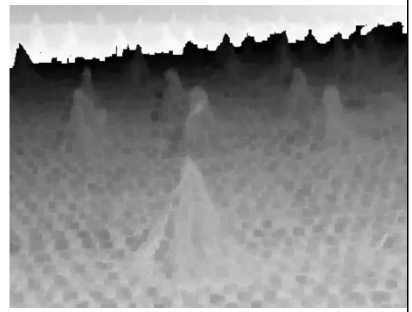
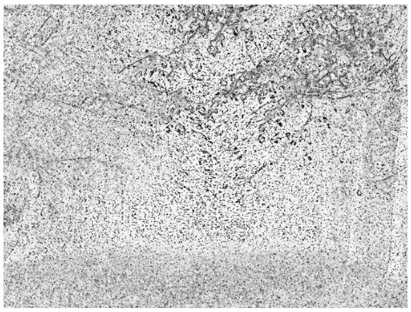
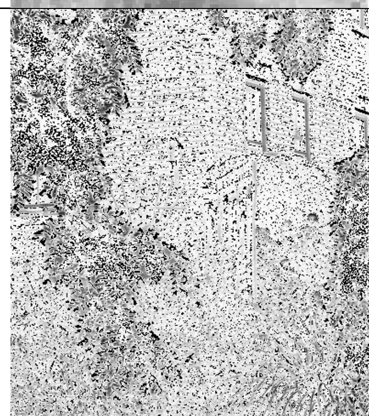
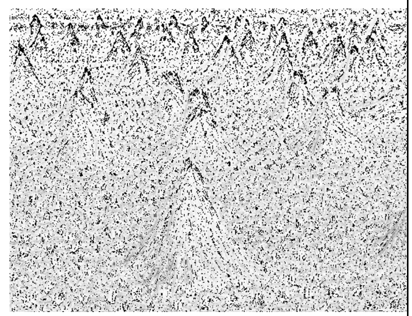
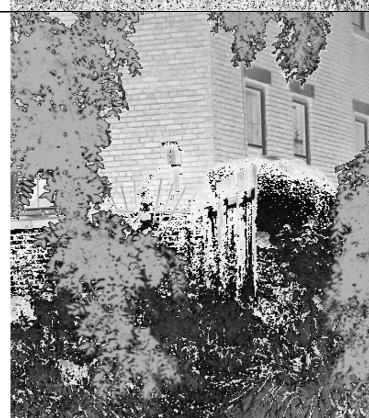
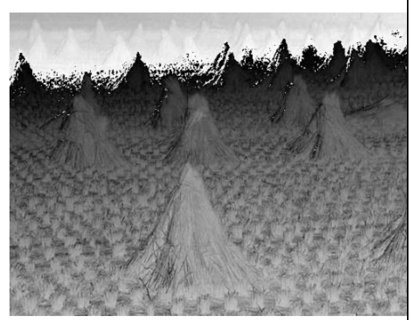
Also, the dark channel image is obtained via erosion of minimal channel image (erosion acts as local minimizer).

The only trick in this algorithm is to properly manage color and grayscale images!

Implementation and Results

As For the purpose of implementation, we used OpenCV and python. The geodesic erosion and dilation functions were implemented as functions. Other morphological operations were also used as the proposed method indicated. The basic element for all morphological elements was chosen to be rectangular with the shape of (7, 7). This leads to better overall results. In the following, the performance of the implemented algorithm on 3 images of Dehaze_cl dataset is depicted. The performance of the implemented code is equal (slightly better due to some corrections in the color management and usage of element with size 7) to the performance depicted in the paper by authors.

Input image			
DCP			
T1			
T2			

T3			
R			
Transmissi on			
Dehazed image			

Conclusion

The proposed dehazing algorithm is truly fast on high-level languages like python. Although, it lacks the state-of-the-art performance proposed in some papers. To cut the story short, the proposed algorithm is a very fast and efficient and effective algorithm to dehaze a single image in the natural environment. This achievement is truly valuable for embedded system image processing. May other complex and compute-intensive image processing algorithms thrive like this one in the near future.