Single Image Dehazing

Arico Amaury
BruFace student

Colot Emmeran
BruFace student

Khansir Nima
BruFace student

Abstract—Outdoor images are affected by atmospheric visibility reduction, particularly haze. The hazing effect are influenced by several factors including location, weather, pollution and other geographic and environmental parameters. Visibility degradation is depedent on the distance between the camera and the scene points. Litterature shares various algoritms to dehaze outdoor images. As part of the image processing project, our group will apply an aglorithm preforming a dehazing process requiring one single input developed by scientists from Hebrew University of Jerusalem [1]. By delimiting and exploiting local patches from this single input, the global airlight vector - the atmospheric colour - and the transmission gradient - the hazing reduction coefficient - are extracted. Through this report, we will describe the three steps of the algorithm, validate the dehazing process by showing the after-process results and compare it with the original pictures.

I. Introduction

Haze, visibility reduction phenomenom, is coming from the presence in the atmosphere of particles scattering the ambiant light, attenuing the contrast for outdoor images and corrupting the true radiance of the scenary by a ambiant colour. To dehaze images, majority of the dehazing algorithm is processing image from on its RGB representation. it operate based on the following pixel colour-based image model (1). The algorithm presented in this paper will follow the same operating process.

$$I(x) = t(x)J(x) + (1 - t(x))A \qquad (1)$$

where $I(x)$ is the pixel colour under hazing condition or in other words, the image input, A is the ambiant light colour, $J(x)$ the true pixel radiance, $t(x)$ is the transmission function is the scattering coefficient ranging from 0 (airlight colour) to 1 (true radiance colour), distance-dependent is defined by the equation (2):

$$t(x) = e^{-\beta d(x)} \qquad (2)$$

where β is the wave phase number and $d(x)$ the scene pixel distance from the camera. From the equation (2) one can observe $t(x)$ is dependent on the wavelength and thus be different for each colour component. Nevertheless, the dehazing method assume a constant matting gradient $t(x)$ for each colour channel. The used algorithm is operating under this assumption. The contrast attenuation and the ambiant colour corruption can be highlited by breaking the equation (1) - $t(x)J(x)$, the attenuation component and $(1 - t(x))A$, the ambiant light effect. $I(x)$ being the input, the dehazing process is reduced to find the transmission and the global

airlight parameters to recompose the true radiance scenary. This process can be decomposed in three steps : the airlight vector determination, the airlight magnitude determination and the transmission coefficient calculation.

Even if the algorithm is indeed based on the equation (1), it uses an mathematical relation steming from (1):

$$I(x) = l(x)R_i + c_iA \qquad (3)$$

Where $l(x)$ expresses the shading coefficient related to the angle between the normal direction of the scene surface and the 3D line from the camera to this specific surface, R_i is the true radiance and c_i, the ambiant light coefficient - $(1 - t(x))$. The dehazing method is a local patch-based algorithm meaning the image is decomposed on patches on which information is gleaned after image treatment. The patches should obey to specific conditions : it should be composed of pixels with same true radiance and with constant transmission coefficient. The motivation behind this decomposition lies in the retrieval of the airlight vector. By finding patches composed of pixels fulfilling the previous constraints, we can construct the line $I(x) = l(x)R_i + C$ with C a constant equals to c_iA. The funding idea is that the constructed line will intersect the airligth vector in the RGB frame. Reconstructing lines from multiples patches will lead to the determination of the airlight vector which ultimately will result to the dehazing of the image. The steps will be explained in details in the following sections.

II. Airlight vector determination

The importance of the airlight determination has been introduced in the previous section. We will now move to its computation. the first stage will be focused on the determination of the airligth vector. It can be split in 6 parts :

i *Patch decomposition* As mentionned previously, the airlight vector can be calculated through local patches processing. Thus the initial step is to generate these local patches from the image. Paper [1] advices the usage of 10x10 patches leading to optimal result for most cases and we will follow their methodology. Only patches with smooth variations should be kept avoiding rough colour distorsion which affect the colour line based on the shading coefficient. Therefore, Canny algorithm is run upstream detecting patches with edges and getting rid of them for next steps.

ii *Eigenvalue comparison* Once patches are validated, the colour line slope are calculated through the determination of the eigenvalues. Ensuring the slope is reflecting

the shading coefficient $l(x)$, first the main eigenvalue should be passing a threshold ensuring being high enough - $\lambda_1 > \tau_1$. Secondly, the ratio between the main eigenvalue and the sub ones should be higher than a threshold τ_1 - $\frac{\lambda_1}{\lambda_2} > \tau_2$.

iii *Distance to origin* Another criteria is the distance from the main eigenvector and the origin of RGB space. If the eignevector passes by the origin or close to the origin, it would mean that the pixels within the patches would be not affected by the global airlight and thus would contain no valuable information to reconstruct the airligth vector. The distance from the origin should therefore be higher than a threshold τ_3. The distance from the eingenvector to the origin can be computed through the cross product usage.

$$dist_{0,0} = \frac{||\vec{v_{eig}} \times \vec{\mu}||}{||\vec{v_{eig}}||} \quad (4)$$

where v_{eig} is the eigenvector and μ the centroid of the pixels withtin the local patch. Additional filtering should be done to ensure that all the valid pacthes eigenvector contains only non negative components being not representative of outdoor reflective light $l(x)R_i$.

iv *Eigenvector angles threshold* Ultimately, the crossing eigenvector of two filtered patches will lead to the airlight vector definition. To ensure no false crossing or error generation, the angle between two patch lines should be above $15\deg$ [1].

v *Eigenvector intersection after projection* Intersection computation is done by first computing the plane defined from the patch line and the centroid to origin vector. Once the planes are defined for each patches, each pair of planes intersection are calculated and defines airlight vector candidates :

$$\vec{n_i} = \vec{v_{eig_i}} \times \vec{\mu_i} \quad (5)$$

$$\vec{\hat{A}}_{cand} = \frac{\vec{n_i} \times \vec{n_j}}{||\vec{n_i} \times \vec{n_j}||} \quad (6)$$

where n_i, n_j are normal vector from the centroid x patch line plane and $\hat{A}_{cand}$ the airlight vector.

vi *Minimal distance for $\hat{A}$ determination* Finally, the each potential airlight candidates are compared by computing the Eculidian distance between the candidate and the patch lines. The lowest median distance candidate is selected as the best fit vector. Mathematically, it gives the equation (7).

$$\vec{\hat{A}}_{vector} = argmin_j(median_i(d_{ij})) \quad (7)$$

where d_{ij} is the Euclidian distance between the j-th airlight candidate $\hat{A}_{cand}$ and the i-th patch line.

The threshold τ_1, τ_2, τ_3 are initiated to keep more than 10 patches and are afterwards refined by increasing step by step the thresholds until 10 patches remain. These patches will be used for the airlight candidates determination.

III. AIRLIGHT MAGNITUDE DETERMINATION

NIMA CHAN

IV. TRANSMISSION RANGE DETERMINATION

EMMERAN CHAN

V. VALIDATION

VI. CONCLUSION