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 is influenced by several factors including location, weather, pollution level and other geographic and environmental parameters. Visibility degradation depends on the distance between the camera and the scene points. Literature shares various algorithms to dehaze outdoor images. As part of the image processing project, our group will apply an algorithm performing a dehazing process requiring one single input developed by scientists from Hebrew University of Jerusalem [1] [2]. 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 phenomenon, is coming from the presence in the atmosphere of particles scattering the ambient light, attenuating the contrast for outdoor images and corrupting the true radiance of the scenery by a ambient colour. Most dehazing algorithms are performed based on the RGB representation of the images. The haze effect is commonly modelled by the following equation [1]:

$$\mathbf{I}(\mathbf{x}) = t(\mathbf{x})\mathbf{J}(\mathbf{x}) + (1 - t(\mathbf{x}))\mathbf{A} \tag{1}$$

where $\mathbf{x}$ is the pixel position, $\mathbf{I}$ is the pixel colour under hazing condition or in other words, the image input, $\mathbf{A}$ is the ambient light colour, $\mathbf{J}$ the true pixel radiance, t is the transmission coefficient ranging from 0 (airlight colour) to 1 (true colour), which is distance-dependent. It is defined by [2] as:

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

where β is the wave phase number and $d(\mathbf{x})$ the scene pixel distance from the camera. From equation (2), one can observe $t(\mathbf{x})$ is dependent on the wavelength and is thus different for each colour component. Nevertheless, the used algorithm assume a constant matting gradient $t(\mathbf{x})$ for each colour channel. The contrast attenuation and the ambient colour corruption can be highlighted by breaking the equation (1) in two distinct parts:

- $t(\mathbf{x})\mathbf{J}(\mathbf{x})$, the true radiance component,
- $(1 - t(\mathbf{x}))\mathbf{A}$, the ambient light effect.

$\mathbf{I}(\mathbf{x})$ being the input image, the dehazing process is reduced to find the transmission and the global airlight parameters to recompose the true radiance scenery. This process can be decomposed in three steps : determining the airlight vector direction, finding its magnitude and finally computing the transmission coefficient.

II. Airlight vector determination

Even if the algorithm is indeed based on the equation (1), it uses a reformulation of it [1]:

$$\mathbf{I}(\mathbf{x}) = l(\mathbf{x})\mathbf{R}_i + c_i\mathbf{A} \tag{3}$$

Where $l(\mathbf{x})$ 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 ambient light coefficient $(1 - t(\mathbf{x}))$. The airlight determination method described in [1] 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 $\mathbf{I}(\mathbf{x}) = l(\mathbf{x})\mathbf{R}_i + \mathbf{C}$ with $\mathbf{C}$ a constant equals to $c_i\mathbf{A}$. The funding idea is that the constructed line will intersect the airlight 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 importance of the airlight determination has been introduced in the introduction and we will now move to its computation. the first part will be focused on the determination of the orientation of the airlight vector. It can be split in 6 steps:

i *Patch decomposition* As mentioned previously, the airlight vector can be calculated through local patches processing. Thus the initial step is to generate these local patches from the image. The size of those patches is set to 10×10 pixels, which leads to optimal results for most cases [1]. Only patches with smooth variations should be kept avoiding rough colour distortion within a patch 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.

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

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 $15\,\text{deg}$ [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 :

$$\vec{n_i} = \vec{v_{eig_i}} \times \vec{\mu_i} \tag{5}$$

$$\vec{\hat{\text{A}}}_{cand} = \frac{\vec{n_i} \times \vec{n_j}}{||\vec{n_i} \times \vec{n_j}||} \tag{6}$$

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

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

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

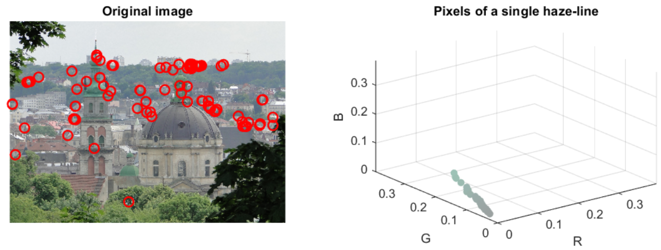

Fig. 1. Haze-line example of an image. The red circles in the original image (left) are the pixels of the haze line. The right plot shows the same pixels in an RGB space centered around the airlight vector.

III. Airlight magnitude determination

NIMA CHAN

IV. Transmission range determination

The method described in [2] is based on a simple assumption: an image without any haze is often composed of a few dominant colours. This allows to regroup the pixels of the input image in a few clusters, each representing one of those dominant colours. Each pixel inside such cluster will be, according to Eq. (1), between the dominant colour and the airlight vector determined previously. The transmission coefficient can then be computed by finding the distance between the pixel and the airlight vector.

The clusters described previously are named *haze-lines* in [2]. and are found by first centering the image around the airlight vector:

$$\mathbf{I}_A(\mathbf{x}) = \mathbf{I}(\mathbf{x}) - \mathbf{A} \tag{8}$$

This means that, when combined with Eq. (1):

$$\mathbf{I}_A(\mathbf{x}) = t(\mathbf{x})\left[\mathbf{J}(\mathbf{x}) - \mathbf{A}\right] \tag{9}$$

And this centered image is then represented in polar coordinates. This means that each pixel x will be represented by its distance to the airlight vector and its longitudinal and azimuthal angle with respect to it:

$$\mathbf{I}_A(\mathbf{x}) = [r(\mathbf{x}), \theta(\mathbf{x}), \phi(\mathbf{x})] \tag{10}$$

In this polar system, pixels with a similar orientation belong to the same haze-line as they would then all have a color between the airlight vector and the dominant color of the haze-line. The clustering is performed by splitting the polar coordinates into 1000 bins (done by the authors of [2] in [3]) and then finding the nearest bin for every pixel. Fig. 1 shows one of the haze-lines together with the reference image. The transmission $t(\mathbf{x})$ is approached by estimating the dominant colour of each haze-line as the one of the pixels with the highest distance to the airlight vector:

$$r_{max,H} = \max_{\mathbf{x}\in H} r(\mathbf{x}) \tag{11}$$

Where H is the haze-line. The transmission coefficient estimate $\tilde{t}(\mathbf{x})$ is then computed as the ratio between the distance of

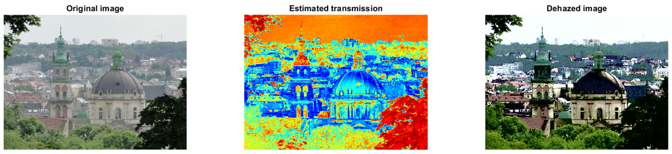

Fig. 2. First transmission estimate. The left image is the original image, the middle one is the transmission coefficient estimate and the right one is the dehazed image using this estimate.

the pixel to the airlight vector $r(\mathbf{x})$ and the maximum distance of the haze-line $r_{max,H}$:

$$\tilde{t}(\mathbf{x}) = r(\mathbf{x})/r_{max,H} \tag{12}$$

This transmission coefficient estimate has a lower bound of $t_{LB}(\mathbf{x})$, which is defined in [2] as:

$$t_{LB}(\mathbf{x}) = 1 - \min_{c \in \{R,G,B\}} \{I_c(\mathbf{x})/A_c\} \tag{13}$$

This lower bound is used to ensure that the radiance $\mathbf{J}$ is not negative.

Using this estimate $\tilde{t}$ would not be very efficient as it would lead to a transmission coefficient that is not continuous across the image. Fig. 2 shows the transmission coefficient computed using the method described above together with the reference image and the resulting dehazed image.

For this reason, the authors of [2] propose to smooth the transmission coefficient estimate by applying an optimization algorithm that aims to find $\hat{t}(\mathbf{x})$ that minimizes:

$$C(\tilde{t}) = \sum_{\mathbf{x}} \frac{\left[\hat{t}(\mathbf{x}) - \tilde{t}(\mathbf{x})\right]^2}{\sigma^2(\mathbf{x})} + \lambda \sum_{\mathbf{x}} \sum_{\mathbf{y} \in N_{\mathbf{x}}} \frac{\left[\hat{t}(\mathbf{x}) - \hat{t}(\mathbf{y})\right]^2}{||\mathbf{I}(\mathbf{x}) - \mathbf{I}(\mathbf{y})||^2} \tag{14}$$

Where $\sigma(\mathbf{x})$ is the standard deviation of $r(\mathbf{x})$ for $\mathbf{x} \in H$, λ is a parameter that controls the importance of the smoothness of the transmission coefficient and $N_{\mathbf{x}}$ is the set of pixels in a 3×3 neighbourhood of $\mathbf{x}$.

The first term of the cost function Eq. (14) keeps the transmission coefficient close to the estimate $\tilde{t}(\mathbf{x})$. The second term penalizes differences between the transmission coefficient of neighbouring pixels, thus enforcing a smoothness constraint. The optimization is done using a weighted least squares optimizer provided in [3] with a weight $\lambda = 0.1$.

V. Validation

VI. Conclusion

References

[1] M. Sulami, I. Glatzer, R. Fattal and M. Werman, "Automatic recovery of the atmospheric light in hazy images," 2014 IEEE International Conference on Computational Photography (ICCP), Santa Clara, CA, USA, 2014, pp. 1-11.

[2] D. Berman, T. Treibitz and S. Avidan, "Non-local Image Dehazing," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 1674-1682.

[3] D. Berman, non-local-dehazing, GitHub repository, 2018. [Online]. Available: https://github.com/danaberman/non-local-dehazing. [Accessed: May 31, 2025].