Day/Night Unconstrained Image Dehazing

Sanchayan Santra, Bhabatosh Chanda
Electronics and Communication Sciences Unit
Indian Statistical Institute
Kolkata, India
Email: {sanchayan_r, chanda}@isical.ac.in

***Abstract*—Images taken under fog or haze have their visibility reduced due to the existence of aerosols in the atmosphere. Image dehazing methods try to recover haze-free versions of these images by removing the effect of haze. Methods proposed till now are exclusively for daytime scene images or for night-time scene. The method we propose here can dehaze an image independent of whether it was captured during the day or night. To achieve this we have relaxed the image formation model to take into account spatially varying atmospheric light that may be present in night-time images. We estimate the contributing airlights and also the patches that are affected by it. We then remove the effect of airlight to obtain haze-free image. We demonstrate the results obtained by using our method on day and night-time images. We have compared out results with that of recently reported methods and the results show the effectiveness of our method.**

***Index Terms*—daytime, night-time, image dehazing**

I. Introduction

An image of an outdoor scene is susceptible to degradation introduced by different weather conditions like fog and haze. Under these conditions, the light gets scattered by the aerosol present in the atmosphere. Due to this phenomenon light loses intensity while reaching the observer and at the same time a semi-transparent layer of atmospheric light covers the scene. The amount of loss is determined by a property of the medium called medium transmittance. As a result the image loses contrast and a color bias is created by a veil of light. This reduces visibility and obstructs view of distant objects.

Image dehazing methods try to recover scene radiance from image(s) taken under these harsh conditions. Now, achieving this from a single image is challenging as the degradation introduced in the scene depends on the distance of an object from the observer. Early methods tried to tackle this problem by using multiple images. For example, images of the same scene under different weather conditions [1], images with different degrees of polarization [2], or a clear day image of the scene in addition to the degraded image [3]. Recently proposed methods try to solve this problem by using stronger assumptions. Tan [4] tried to solve this problem by increasing the image contrast in a spatially consistent way. This is motivated by the observation that non-hazy images have more contrast than hazy ones. Fattal [5] tried to estimate scene radiance assuming that surface shading and medium transmittance are locally statistically uncorrelated. He et al. [6] devised their method by using the observation that in outdoor haze-free images a patch contains some pixels whose intensity is very close to zero in at least one color channel. This is known as dark channel, and with haze the value of the dark channel increases proportionally. Tang et al. [7] tried to dehaze an image in a learning framework. This method assumes that transmittance is independent of scene content and within a small patch it is constant. This proposition is used to synthetically generate hazy patches with different transmittance from haze-free natural image patches. Then a regression model is learned from this data to predict transmittance. The color line dehazing by Fattal [8] exploits the local color line prior to estimate scene transmittance. This method also assumes atmospheric light is known and does not compute it explicitly.

All these methods assume that the atmospheric light is constant throughout the image. This may hold for scenes captured in daylight, but not for night-time scenes. During the night the artificial lights produce non-uniform lighting. Moreover the lights can be of different colors. A few methods have been proposed to deal specifically with night-time haze. Pei and Lee [9] propose to dehaze night-time images using the same imaging model used for daytime dehazing, but with one added preprocessing step. This color transfer preprocessing step tries to fix the color bias due to artificial light by changing the color statistics to that of a target image. Then a modified dark channel prior is used to dehaze the image followed by local contrast enhancement using a bilateral filter. Zhang et al. [10] introduced a new imaging model to account for spatially varying atmospheric light. Their preprocessing step compensates the incident light intensity using retinex and also corrects the colors of incident light before dehazing using dark channel prior. As they have relaxed the model, the atmospheric light is estimated in a local neighborhood instead of being computed globally. Li et al. [11] tried to dehaze night-time images while reducing the halos caused by multiple scattering of light near the light sources.

Here we propose an image dehazing method that can work irrespective of the input image being a day or night-time. This extends our previous work [12] on image dehazing with varying atmospheric light intensity. Also note that the steps of our method are similar to [8], but are modified to suit our problem. The proposed method uses a relaxed imaging equation to properly handle night-time haze images. Our method estimates atmospheric light with different colors and also their influence on different parts of the image. Though our relaxed model is similar to the model used by Zhang et al. [10] to dehaze night-time images, our method dehazes daytime images also using the same model.

The rest of the paper is organized as follows, Section II describes image formation in a haze environment as well as the motivation to relax the imaging equation. Section III outlines the basic assumptions and intuition behind the solution. Section IV describes our method of dehazing. Section V specifies the setup used to generate the results and presents comparison with other methods. Section VI concludes the paper.

II. Background and motivation

Light propagating through a medium gets scattered by the aerosol (e.g. dust, particles and water droplets) present in it. This phenomenon usually is modeled by the following equation [5] [6] [7] [8],

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

Here $I(\mathbf{x})$ is the observed intensity, $J(\mathbf{x})$ denotes light intensity before getting scattered, $t(\mathbf{x})$ represents the transmittance that determines the amount of light reaching the observer after being scattered and 'A' denotes the atmospheric light that gets scattered in the direction of the observer creating a veil of light. For RGB images we can still use this equation where $I(\mathbf{x})$, $J(\mathbf{x})$ and A becomes RGB vectors but $t(\mathbf{x})$ remains a scalar. This works as $t(\mathbf{x})$ does not vary much with wavelength of light in case of fog and haze. Now in this equation 'A' is taken to be a constant assuming the contribution of atmospheric light (diffuse skylight, sunlight and reflected ground light taken together) is uniform in the medium between the observer and the object. But that may not be true in most of the cases. This becomes apparent if sunlight is present in the scene. Another assumption is that the scene is being captured in daytime, otherwise diffuse skylight and sunlight won't be there. Note that light undergoes the same change when fog is there at night, and gets scattered-out from the direction of the observer and scattered-in from other places creating a veil of light. That means when we move from daytime to night-time, the process remains same, but the light sources change. Thus at night-time 'A' can't remain constant as the light sources may not be of the same color or the same intensity. So, the imaging equation need to be relaxed to the following,

$$I(\mathbf{x}) = J(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))A(\mathbf{x}). \tag{2}$$

Here A changes to $A(\mathbf{x})$ to account for the space-variant illumination within an image. However, other things remain the same. So ideally, if we can get $J(\mathbf{x})$ from this equation we will get a dehazed version of the original night-time image. Note that the original equation (eq. (1)) is a special case of equation (2). Therefore we will also be able to dehaze images taken during the day using equation (2).

Now, in the proposed method we use the following form of eq (2) to recover $J(\mathbf{x})$

$$\begin{aligned} I(\mathbf{x}) &= J(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))m(\mathbf{x})\hat{A}(\mathbf{x}) && (3)\\ &= J(\mathbf{x})t(\mathbf{x}) + a(\mathbf{x})\hat{A}(\mathbf{x}). && (4) \end{aligned}$$

That is, we simply decompose $A(\mathbf{x})$ into its magnitude $m(\mathbf{x})$ and direction $\hat{A}(\mathbf{x})$. At each position we need to estimate $\hat{A}(\mathbf{x})$ and $a(\mathbf{x})$. Subtracting the airlight component ($a(\mathbf{x})\hat{A}(\mathbf{x})$) from the input image we get $J(\mathbf{x})t(\mathbf{x})$. We then enhance the image to get haze the free image.

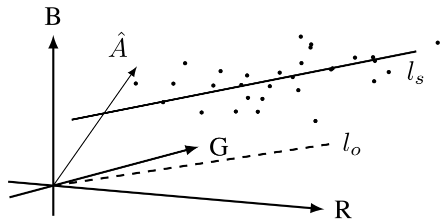

Fig. 1. Colors in a patch as points in RGB space and the fitted line l_s. The original line l_o got shifted in the direction of $\hat{A}$ to make l_s.

III. Solution idea

A. Color line and dehazing

The color line model as described in [13] states that if we take a small patch of a natural image then the colors in that patch ideally lie on a line passing through the origin in the RGB space. But due to scene, sensor and other camera related distortions, the colors spreads out and forms a cluster. This can be seen in the following way. Suppose for the colors within a patch we can write $I(\mathbf{x}) = \mathscr{F}l(\mathbf{x})R$, where $l(\mathbf{x})$ is the shading component and R is surface reflectance vector and $\mathscr{F}$ is constant irradiance. Then we may say that, R provides the direction of the line and $l(\mathbf{x})$ provides the position of color point $I(\mathbf{x})$ in that direction. Now, in the case of hazy images, this line gets shifted in the direction given by $\hat{A}(\mathbf{x})$ due to the additive airlight component (Fig.1). So, if we assume that within a patch the terms $t(\mathbf{x})$, $m(\mathbf{x})$ and $\hat{A}(\mathbf{x})$ are constant and if the line can be estimated from the patch, then within that patch we can compute how much the line gets shifted. So, within a patch equation (3) becomes,

$$I(\mathbf{x}) = J(\mathbf{x})t + (1 - t)m\hat{A}. \tag{5}$$

The constant assumption is valid if the patch is sufficiently small and lies on same object surface. Once we estimate $\hat{A}$ for a patch, we can undo the shift by moving the line (i.e. the corresponding colors in the patch) in the opposite direction. This removes the effect of airlight component. But to be able to do so, we must validate our assumptions beforehand.

B. Estimating $\hat{A}$

As already mentioned in the previous section, from a patch of a natural image we may get a line passing through the origin formed by the RGB vectors of the pixels in the patch. For hazy images this line gets shifted in the direction of $\hat{A}$. So, the plane containing the color line for the patch and the origin also contains $\hat{A}$. Now, if we get two such patch planes that are non-parallel and have the same $\hat{A}$, then this $\hat{A}$ should lie in the intersection of these two patch planes (Fig.2). Now according to our relaxed model (eq (2)) $\hat{A}$ is different for each pixel. However we may safely assume that the number of $\hat{A}$'s is much less than the number of pixels. So, we may group the

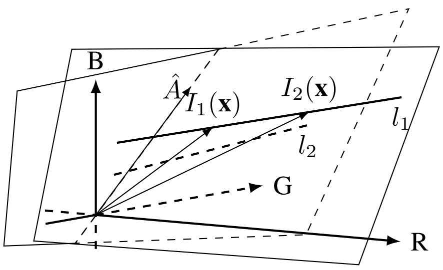

Fig. 2. Fitted lines l_1 and l_2 and the corresponding patch planes obtained from two image patches. $\hat{A}$ lies in the intersection of the planes.

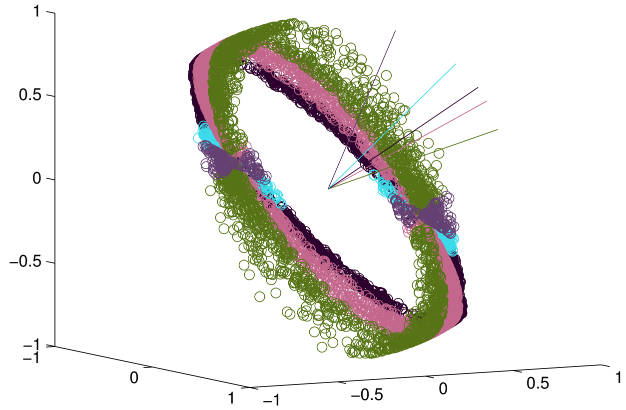

Fig. 3. Normals obtained from image patches plotted as points in RGB space (colored circles) and their associated $\hat{A}$. Each color denotes a group of $\hat{n}$'s and the corresponding $\hat{A}$ is also colored the same.

patches such that within a group all the patches are affected by a single $\hat{A}$. For a given group the $\hat{A}$ can be computed by intersecting the patch planes belonging to that group. Thus we get the $\hat{A}$'s and also the patches affected by it. This information is used to compute the shift of the patch line in each patch.

IV. Proposed Method

A. Line fitting and normal computation

As discussed before, we are supposed to estimate a color line from the RGB vectors in a patch. We first divide the image into patches with 50% overlap. Then on RGB vectors of each patch we apply RANSAC and fit a line. RANSAC provides two points (say I_1, I_2) defining the estimated line and a set of inlier points. Here we also compute the normal ($\hat{n}$) to the plane containing this color line and the origin. The line ($L = \rho D + P_0$) parameters and the normal is computed as follows, $P_0 = I_1, D = \frac{I_2 - I_1}{||I_2 - I_1||}, \hat{n} = \frac{I_1 \times I_2}{||I_1 \times I_2||}$. Note that, in the subsequent, phases given a patch only its inlier points are used for computation. Now to check the validity of our assumptions and discard bad estimates we do the following tests. Accordingly an estimate is accepted as valid if it satisfies the followings.

Number of inliers. Number of inliers found is greater than a percentage (t_i) of total number of points in the patch.

Positive D. All components of D are positive.

Single object. The patch contains points from a single object i.e. an edge is not there. So, the gradient energy over the patch must be low.

Origin far away. The origin in the RGB space is not near the estimated line, i.e. the distance between them is not small.

Adequate intensity variance. Within a patch the color intensity variance is large in the direction of the line. Standard deviation of the projected patch points on the estimated line is used here.

If a patch fails the tests, it is discarded and not used in further computations.

B. $\hat{A}$ computation

We know that $\hat{A}$ lies on the intersection of the patch planes. So we can say that $\hat{A}$ is perpendicular to the normals to the patch planes. Let us consider $\hat{n}$'s as points in the RGB space. If we fit a plane to those points then the unit normal to the fitted plane will be the desired $\hat{A}$. To make the intersection computation more robust we discard some patches Ω based on its dark channel value [6]:

$$Dark(\Omega) = \min_{\mathbf{x}\in\Omega}\Big(\min_{c\in R,G,B} I_c(\mathbf{x})\Big). \qquad (6)$$

A patch (Ω_i) is kept if the following condition is satisfied

$$Dark(\Omega_i) > t_D * \max_{\Omega_j} Dark(\Omega_j), \qquad (7)$$

where $0 < t_D < 1$. Now the problem is that we have to fit an unknown number of planes, and this number varies with the image. To achieve that first we represent the plane using the following form,

$$n_x \cos\theta \sin\phi + n_y \sin\theta \sin\phi + n_z \cos\phi = 0, \qquad (8)$$

where $[n_x, n_y, n_z]^T = \hat{n}$. Then we use Hough Transform to get votes on the parameter values. From the normal to the patch planes we get $\hat{n}$'s, and finally $(\cos\theta\sin\phi, \sin\theta\sin\phi, \cos\phi)$ gives our desired $\hat{A}$ (Fig.3). As $\hat{A}$ indicates a color in RGB space, all of its components will be positive. So, we can say θ and ϕ are in the range $[0°, 90°]$ and we use h_s as the parameter step value in the Hough space. In the computed Hough space we find modes to get candidate $\hat{A}$'s and discard those modes that are below a threshold (t_H) to suppress spurious $\hat{A}$'s. Now two problems arises if we use only the patches that contributed to the modes. First, one patch may vote for more than one of the selected $\hat{A}$. Second, a patch may not vote for any of the selected $\hat{A}$'s. So instead each patch is assigned to one and only one of the $\hat{A}$'s based on the distance between $\hat{A}$ and its $\hat{n}$. We use $|\cos\psi|$ as the distance where ψ is the angle between $\hat{n}$ and $\hat{A}$ and associate a patch to a $\hat{A}$ that gives minimum distance. We then update $\hat{A}$ of each group by computing the intersection of the patch planes. The eigenvector corresponding to the minimum eigenvalue of the covariance matrix of $\hat{n}$'s is used as the solution to the intersection [12]. Now it may happen that for some groups the computed eigenvector does not have all of its component positive. In that case we discard the $\hat{A}$ corresponding to this group and reassign the normals and recompute the eigenvectors. This is repeated till all eigenvectors have positive components.

C. Shift computation and aggregation

In the previous steps we have computed from the patches their associated color lines and $\hat{A}$. We now compute the amount of shift of the line from the origin in the direction of $\hat{A}$. This is obtained by minimizing the error

$$E_{intersect}(\rho, \delta) = \min_{\rho,\delta} ||P_0 + \rho D - \delta \hat{A}||^2, \qquad (9)$$

where δ provides the amount of shift of the line. We validate the estimated shift using the following tests.

Large intersection angle. The angle between $\hat{A}$ and D should be large otherwise the shift estimation is error prone.

Low intersection error. If the value of $E_{intersect}(\rho, \delta)$ as in eq (9) is not close to zero, then the line is still far from the origin after being shifted. So, we need this to be small.

Valid range. From eq (3) we can say that the value of the shift can't be arbitrary as $t(\mathbf{x})$ is between 0 and 1 and $m(\mathbf{x})$ is within a known range. We require the shift to be between 0 and the minimum intensity of the patch.

As we are using overlapping patches it may happen that for a pixel more than one shift value and $\hat{A}$ are computed. So when aggregating the values obtained from the patches, we take the maximum shift value and the corresponding $\hat{A}$ as the aggregated data for a pixel.

D. Interpolation based fill-up

In the aforementioned steps of estimating $\hat{A}$'s and $a(\mathbf{x})$'s from the patches we have discarded quite a few patches where our assumptions failed. So after the aggregation, it is quite likely that at some pixels $\hat{A}$ and $a(\mathbf{x})$ are not available. But we require these values at every pixel to dehaze an image. So, we interpolate values at these pixels before dehazing the image. Here we need to interpolate two quantities: $\hat{A}$ and $a(\mathbf{x})$. Since, $\hat{A}$ is 3×1 vector, we don't interpolate it directly. Instead we denote each one of the $\hat{A}$'s by a label and compute their influence at all the pixels. The influence of each label is obtained by minimizing the function

$$E_{infl}(F) = (F - P)^T(F - P) + \frac{\lambda}{2} F^T L F, \qquad (10)$$

where F is a matrix of size *numpixel* $\times$ *numlabel* with entry $F(i, j)$ denoting the influence of j-th $\hat{A}$ on i-th pixel. P is also a *numpixel* $\times$ *numlabel* matrix with $P(i, j) = 1$ if j-th $\hat{A}$ is assigned to i-th pixel during aggregation. The scalar λ controls the smoothness of the influence. L is the laplacian matrix of the graph constructed from the given image taking each pixel as a node and $1/||I(\mathbf{x}) - I(\mathbf{y})||^2$ as weight of the edge between pixels **x** and **y**. Neighborhood of a pixel is also defined accordingly. The final interpolated $\hat{A}(\mathbf{x})$ is a normalized weighted sum of the $\hat{A}$'s where the weights are the influences.

The interpolation of $a(\mathbf{x})$ is done in a way similar to [12]. $a(\mathbf{x})$ is interpolated by minimizing the function

$$E_{airlight}(a) = (a - \tilde{a})^T \Sigma (a - \tilde{a}) + \alpha a^T L_g a + \beta b^T a, \quad (11)$$

where $\tilde{a}$ is estimated magnitudes of airlight component after aggregation and 'a' is its interpolated value, both in vector form (*numpixel* $\times$ 1). $\tilde{a}$ is zero where the estimate is not there. Σ is a diagonal matrix with its diagonal containing the error variance where $a(\mathbf{x})$ is estimated and 0 otherwise. The error variance is computed in as outlined in [8] by

$$\sigma^2 \frac{||\hat{A} - D(\hat{A}^T D)||^2}{(1 - (\hat{A}^T D)^2)^2}. \qquad (12)$$

Similar to [8] σ is taken as $1/30$. L_g is a laplacian matrix constructed similarly as before, but with a larger neighborhood. Each element of b is $1/||I(\mathbf{x})||$. α and β are scalars controlling the importance of the corresponding terms.

E. Recovery

We have already obtained $\hat{A}$'s and $a(\mathbf{x})$ at each pixel. So, we can compute the airlight component $(a(\mathbf{x})\hat{A}(\mathbf{x}) = (1 - t(\mathbf{x}))m(\mathbf{x})\hat{A}(\mathbf{x}))$ of an image. Subtracting these from the original image we get airlight removed original image:

$$J(\mathbf{x})t(\mathbf{x}) = I(\mathbf{x}) - (1 - t(\mathbf{x}))m(\mathbf{x})\hat{A}(\mathbf{x}). \qquad (13)$$

As our method doesn't compute $t(\mathbf{x})$ explicitly we can't obtain $J(\mathbf{x})$ from this equation directly. Instead we enhance the contrast of this image based on reduction in intensity due to subtraction of the airlight component:

$$R_{im} = \frac{I(\mathbf{x}) - (1 - t(\mathbf{x}))m(\mathbf{x})\hat{A}(\mathbf{x})}{1 - Y((1 - t(\mathbf{x}))m(\mathbf{x})\hat{A}(\mathbf{x}))} \qquad (14)$$

$$Y(I(\mathbf{x})) = 0.2989 I_R(\mathbf{x}) + 0.5870 I_G(\mathbf{x}) + 0.1140 I_B(\mathbf{x}). \quad (15)$$

Though this works well for some images, its good performance can't be guaranteed. Some times the dehazed image turns out dark, and further enhancement becomes necessary.

V. Experiments and Results

We have run our method on a variety of day and night-time images. This includes benchmark images for testing daytime dehazing and night-time dehazing methods. For most of the images the parameter values used are as reported in this section, but some of them may need slightly different values to improve the result. A patch size of 8×8 is used throughout this experiment. For RANSAC we have used code of [14] and 0.02 as its threshold. Percentage of inlier pixels t_i is fixed at 40%. To check the patch coming from a single object 0.06 is used as threshold. To make sure the origin is at a distance a threshold of 0.0005 is used. For checking intensity variance 0.006 is used as threshold. t_D is varied between 0.45 and 0.1 based on average intensity of the image. h_s is set to $3°$ and t_H to 30% of the maximum Hough response. To ensure large intersection angle the threshold is taken as $15°$ and intersection error threshold is kept at 0.05. The parameter λ of eq. (10) is taken to be 1. The results of Li et al. [11] are generated for comparison using the code provided by the authors. The results of Fattal [8] are obtained from the author's website.

The figures 4, 5 and 6 show the comparison of our method with other methods. Table I provides an objective measure [15] called FADE of these images, where a lower value indicates a better dehazed image. For the image in Fig.6 our method

Fig. 4. Miri haze (top) and fog on the bay by DeCenzo (bottom) with results. (Left to right) Original Image, Li et al., Our method, Computed airlight by our method

Fig. 5. Castle (top), tiananmen (middle), ny12 (bottom) image with results. (Left to right) Original image, Fattal color line, Li et al. Our method

Fig. 6. Night road image (top-left) and its dehazed versions: Li et al.(bottom-left), Our method(right)

TABLE I
FADE [15] VALUE OF IMAGES. LOWER VALUE INDICATES LESS FOG.

Image	Time	Original	Li et al.	Fattal	Our
night road	night	0.6753	0.3062	-	**0.2308**
miri haze	night	1.9157	**0.3881**	-	0.4931
fog on the bay	night	1.1555	**0.3799**	-	0.4221
castle	day	1.0507	**0.2212**	0.2457	0.2406
tiananmen	day	1.3454	0.4058	0.4329	**0.3928**
ny12	day	0.7344	0.3017	**0.1715**	0.2057

performs better in terms of score as well as visually. It is evident that the method of Li et al. [11] introduces color bias and noise. Although the FADE score is less for Li et al. [11] for both the images in Fig.4, visually the method performs performs a little better for the first image, but fails badly in the second one. Fig.5 shows comparison for daytime images. Li et al. [11] performs poorly in all 3 cases. Fattal [8] performed better being an exclusively daytime dehazing method. As noted in [8] the method does not work for night scenes as it underestimates transmittance due the existence of artificial lights. But our method works well in both daytime and night-time images as it can be seen from the results. More results can be found in the webpage[1].

VI. Conclusion

We have proposed in this paper a unified dehazing method that works for both night and daytime images. This is achieved by using a relaxed haze imaging model (eq (2)) where the constant atmospheric light assumption is relaxed to a spatially variant one. We find possible directions of the atmospheric light vectors using local color line and Hough transform. These directions are used to calculate airlight component value in each patch of the image. However, some patches, where our assumptions are not valid, are discarded from the processing pipeline. Second, at pixels with an unreliable estimate, both atmospheric light vector directions and airlight component value are interpolated. Now it may happen that due to discretization of the Hough space some atmospheric light direction may be missed. It may also happen, due to the interpolation, that at some places the computed airlight component is less than what it should be and thereby not properly dehazing the input image. Our method does not compute $t(\mathbf{x})$ explicitly instead it computes the whole airlight component $(a(\mathbf{x}) = (1 - t(\mathbf{x}))m(\mathbf{x}))$. For this reason objects with low intensity and color similar to airlight becomes dark after dehazing. Also in the last phase of our method we had to resort to contrast enhancement, though this contrast enhancement procedure is not guaranteed to work satisfactorily in all images.

[1] www.isical.ac.in/~sanchayan_r/day_night_dehaze

References

[1] S. G. Narasimhan and S. K. Nayar, "Contrast restoration of weather degraded images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 713–724, June 2003.

[2] S. Shwartz, E. Namer, and Y. Schechner, "Blind haze separation," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, 2006, pp. 1984–1991.

[3] S. G. Narasimhan and S. K. Nayar, "Vision and the atmosphere," *International Journal of Computer Vision*, vol. 48, no. 3, pp. 233–254, 2002.

[4] R. Tan, "Visibility in bad weather from a single image," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

[5] R. Fattal, "Single image dehazing," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 72:1–72:9, Aug. 2008.

[6] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 12, pp. 2341–2353, Dec 2011.

[7] K. Tang, J. Yang, and J. Wang, "Investigating haze-relevant features in a learning framework for image dehazing," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 2995–3002.

[8] R. Fattal, "Dehazing using color-lines," *ACM Trans. Graph.*, vol. 34, no. 1, pp. 13:1–13:14, Dec. 2014.

[9] S. C. Pei and T. Y. Lee, "Nighttime haze removal using color transfer pre-processing and dark channel prior," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, Sept 2012, pp. 957–960.

[10] J. Zhang, Y. Cao, and Z. Wang, "Nighttime haze removal based on a new imaging model," in *Image Processing (ICIP), 2014 IEEE International Conference on*, Oct 2014, pp. 4557–4561.

[11] Y. Li, R. T. Tan, and M. S. Brown, "Nighttime haze removal with glow and multiple light colors," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 226–234.

[12] S. Santra and B. Chanda, "Single image dehazing with varying atmospheric light intensity," in *2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, Dec 2015, pp. 1–4.

[13] I. Omer and M. Werman, "Color lines: image specific color representation," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, June 2004, pp. II–946–II–953 Vol.2.

[14] P. D. Kovesi, "MATLAB and Octave functions for computer vision and image processing," Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia, available from: http://www.csse.uwa.edu.au/~pk/research/matlabfns/.

[15] L. K. Choi, J. You, and A. C. Bovik, "Referenceless prediction of perceptual fog density and perceptual image defogging," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3888–3901, Nov 2015.