Air-Light Estimation Using Haze-Lines

Dana Berman
Tel Aviv University
danamena@post.tau.ac.il

Tali Treibitz
University of Haifa
ttreibitz@univ.haifa.ac.il

Shai Avidan
Tel Aviv University
avidan@eng.tau.ac.il

Abstract

Outdoor images taken in bad weather conditions, such as haze and fog, look faded and have reduced contrast. Recently there has been great success in single image dehazing, i.e., improving the visibility and restoring the colors from a single image. A crucial step in these methods is the calculation of the air-light color, the color of an area of the image with no objects in line-of-sight. We propose a new method for calculating the air-light. The method relies on the haze-lines prior that was recently introduced. This prior is based on the observation that the pixel values of a hazy image can be modeled as lines in RGB space that intersect at the air-light. We use Hough transform in RGB space to vote for the location of the air-light. We evaluate the proposed method on an existing dataset of real world images, as well as some synthetic and other real images. Our method performs on-par with current state-of-the-art techniques and is more computationally efficient.

1. Introduction

Haze and fog have a negative effect on image quality in terms of contrast and color fidelity. Pixel values in hazy images can be modeled as a linear combination of the actual scene radiance and a global air-light. The amount of degradation depends on the distance of the scene point from the camera, and may vary from pixel to pixel. The air-light, on the other hand, is assumed to be global.

Compensating for this degradation, termed *dehazing*, is inherently under-constrained, and the various image dehazing algorithms proposed in the literature differ in the type of prior they use. Some solve for the air-light independently and some assume that the air-light is given as input.

Air-light estimation has not been studied as extensively as priors for dehazing, and, with few exceptions, it is often estimated in an ad-hoc manner. We propose a new algorithm for air-light estimation that relies directly on a new prior, termed *haze-lines*, that was recently introduced [2].

The haze-lines prior is based on an observation made for clear images: the number of distinct colors in clear images is orders of magnitude smaller than the number of pixels. Therefore, pixels in a clear image tend to form a few hundred clusters in RGB space. In hazy images, haze changes the color appearance as a function of object distance. As a result, these clusters become lines in RGB space (termed haze-lines), since each cluster contains pixels from different locations in the scene.

We rely on this prior to estimate the air-light. We treat the pixels of a hazy image as points in RGB space and model their distribution as a set of lines (i.e., the haze-lines) that intersect in a single point (i.e., the air-light).

We use Hough transform to vote on the location of the air-light. In its naïve form, the algorithm is computationally expensive, since there are many air-light candidate locations in 3D RGB space, and for each location we must collect the votes of all pixels in the image.

We propose a couple of approximations to address these issues. First, we reduce the problem from 3D to 2D by considering the projection of pixel values on the RG, GB and RB planes. We combine the votes in the three planes to obtain the final air-light estimation. This has a dramatic effect on the number of air-light candidates we need to sample and evaluate. Second, we cluster all pixels in the image into roughly a thousand clusters. Therefore, we only need to collect votes for a candidate air-light from cluster centers and weigh each vote by the cluster size, rather than collecting votes from all pixels. As a result, our algorithm runs in a matter of seconds, as opposed to minutes in the naïve implementation[1]. We demonstrate our method on a recently proposed data set [1], as well as on other real-world images and synthetic data. Our method is more efficient than state-of-the-art methods (linear vs. quadratic complexity) and performs on-par with them.

2. Related Work

This section concentrates on *air-light estimation* and provides a chronological survey of methods for estimating a single, global, air-light from a single day-light hazy image. Dehazing techniques recently received great attention

---

[1]Code is available: https://github.com/danaberman/non-local-dehazing

in the literature, and readers can find a comprehensive survey of dehazing algorithms by Li et al. [10].

In early works the most haze-opaque pixel was used to estimate the air-light. For example, Tan [16] chose the brightest pixel. Fattal [5] used it as an initial guess for an optimization problem. However, the pixel with the highest intensity might correspond to a bright object rather than to the air-light.

Therefore, He et al. [7] suggest to select the brightest pixel among the pixels that have the top brightest values of the dark channel (the minimal color channel in a small environment). This method is efficient and generally produces accurate results, but it assumes that the sky or another area with no objects in line-of-sight is visible in the image.

Tarel and Hautière [17] first perform white-balance and then the air-light is assumed to be pure white ([1, 1, 1]).

Sulami et al. [15] separately estimate the air-light magnitude and direction. The direction is estimated by looking for small patches with a constant transmission and surface albedo. Each pair of such patches provide a candidate air-light direction as the intersection of two planes in RGB space. The air-light magnitude is recovered by minimizing the dependence between the pixels' brightness and transmission. Fattal [6] uses this air-light estimation [15] for single image dehazing.

The patch recurrence property is an observation that small image patches tend to repeat inside a single natural image, both within the same scale and across different scales. This property is diminished when imaging in a scattering media, since recurring patches at different distances undergo different amounts of haze and have different appearances. Bahat and Irani [1] use this prior to detect differences between such co-occurring patches and calculate the air-light. Both of the methods [1, 6] require finding pairs of patches that satisfy certain conditions. These processes are computationally intensive.
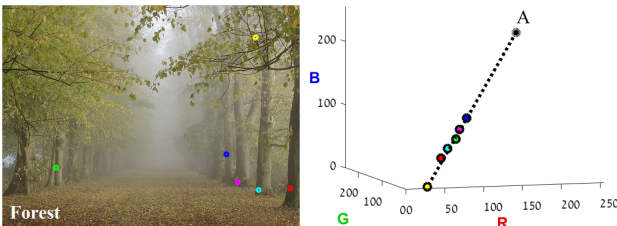

Figure 1. **Haze-Line**. The pixels of the *Forest* image were clustered to haze-lines. Six of the pixels belonging to a single haze-line are marked both in the image plane and in RGB space, with respective colors. They are all located on shaded parts of the trunks and branches, and therefore should have similar radiance. Due to haze they are distributed along a line in RGB space, that is spanned by the air-light A, marked in black, and a nearly haze-free pixel, marked in yellow.

Unlike previous methods, which first estimate the global air-light and use it to estimate the transmission for each pixel, the scheme suggested in [4] first estimates the transmission by DehazeNet, a convolutional neural network, and then the air-light is selected as the brightest pixel whose transmission value is smaller than 0.1. Similarly to [16, 7], it requires a visible area with no objects in line-of-sight.

Alternative scenarios in which the air-light must be estimated include estimation of a single, global air-light from multiple images [14] or recovery of the air-light of a night-time hazy image in the presence of multiple light sources [9].

3. Air-light Estimation Using Haze-Lines

3.1. The Haze-Lines Prior

Hazy images can be modeled as a convex combination of an attenuated signal and the air-light [11]:

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

where x is the pixel coordinate in the image plane, I is the acquired image, J is the scene's unknown radiance, t is the transmission which is related to the distance of the object from the camera, and A is the air-light, which we would like to estimate. Bold letters denote vectors, where x is a 2D vector in the image plane and I, J, and A have three color components (R, G, B).

The color distribution of natural images is sparse, thus clear images can be represented by tight clusters in RGB space. This observation lies at the heart of various image processing applications such as denoising [3] and compression [12]. Since objects with similar colors are often located at different distances from the camera, in the presence of haze these objects will have different transmission values. It follows from the model (Eq. 1) that hazy images can be modeled by lines in RGB space that converge at the air-light coordinate. This assumption is the basis for the non-local single image dehazing [2], where these lines were termed *Haze-Lines*. Formally, Eq.1 can be re-written as:

$$I(x) = t(x) \cdot (J(x) - A) + A \quad . \qquad (2)$$

A haze-line consists of pixels with a similar radiance $J(x)$ but different transmission $t(x)$. Eq. 2 is a line equation in 3D passing through the air-light coordinate A, where $(J(x) - A)$ is the direction, and t is the line parameter. Fig. 1 demonstrates a Haze-Line both in the image plane and in RGB space. To find the haze-lines, the 3D RGB coordinate system is translated so that the air-light is at the origin:

$$I_A(x) = I(x) - A \quad . \qquad (3)$$

The vector $I_A(x)$ can be expressed in a spherical coordinate system:

$$I_A(x) = [r(x), \theta(x), \phi(x)] \quad , \qquad (4)$$

where r is the distance to the origin (i.e., $\|\boldsymbol{I} - \boldsymbol{A}\|$), and θ and ϕ are the longitude and latitude angles, respectively. Haze-Lines consist of points with the same θ and ϕ angles.

3.2. Air-Light Estimation

In [2] the Haze-Lines model is used by assuming a fixed distribution of 3D lines emanating from the air-light 3D coordinate in RGB space. Given the air-light as input, it is possible to dehaze the image.

In this work, we use the same model in the opposite direction. Given a candidate air-light coordinate in RGB space, we model pixels' intensities with a fixed set of lines emanating from the air-light candidate. That is, we wish to model pixels' values by an intersection point (i.e., the air-light) and a collection of lines (i.e., the Haze-Lines). An air-light in the correct RGB location will fit the data better than an air-light in a wrong location.

We use a Hough transform to estimate the air-light. Hough transform is a useful technique to detect unknown parameters of a model given noisy data via a voting scheme. In this case, the voting procedure is carried out in a parameter space consisting of candidate air-light values in RGB space. In particular, we uniformly sample a fixed set of line angles $\{\theta_k, \phi_k\}_{k=1}^{K}$. Given this set, we consider a discrete set of possible air-light values. The distance between a pixel $\boldsymbol{I}(\boldsymbol{x})$ and the line defined by the air-light $\boldsymbol{A}$ and a pair of angles (θ, ϕ) is:

$$d\left(\boldsymbol{I}(\boldsymbol{x}), (\boldsymbol{A}, \phi, \theta)\right) = \|(\boldsymbol{A} - \boldsymbol{I}(\boldsymbol{x})) \times (\cos(\theta), \sin(\phi))\|. \tag{5}$$

A pixel votes for a candidate $\boldsymbol{A}$ only if the distance to one of the lines is smaller than a threshold τ. This threshold is adaptive and depends on the distance between $\boldsymbol{A}$ and $\boldsymbol{I}(\boldsymbol{x})$ to allow for small intensity variations. I.e., instead of working with cylinders (lines with a fixed threshold) we work with cones (lines with a variable threshold). Formally:

$$\tau = \tau_0 \cdot \left(1 + \frac{\|\boldsymbol{I}(\boldsymbol{x}) - \boldsymbol{A}\|}{\sqrt{3}}\right). \tag{6}$$

In addition, we allow a pixel to vote only for an air-light that is brighter than itself. This is due to the fact that bright objects are quite rare, as shown empirically to justify the dark channel prior [7], and usually do not contain information about the haze (e.g., a bright building close to the camera).

Our method can be described as finding the best representation of the pixels' values of a hazy image with air-light $\boldsymbol{A}$ and fixed line directions $\{\theta_k, \phi_k\}_{k=1}^{K}$. This can be formulated as follows:

$$\arg\max_{\boldsymbol{A}} \sum_{\boldsymbol{x}} \sum_{k} \mathbb{1}[d\left(\boldsymbol{I}(\boldsymbol{x}), (\boldsymbol{A}, \phi_k, \theta_k)\right) < \tau] \cdot \mathbb{1}[\boldsymbol{A} > \boldsymbol{I}(\boldsymbol{x})], \tag{7}$$

Algorithm 1 Air-light Estimation

Input: hazy image $\boldsymbol{I}(\boldsymbol{x})$
Output: $\hat{\boldsymbol{A}}$

1: Cluster the pixels' colors and generate an indexed image $\hat{\boldsymbol{I}}(\boldsymbol{x})$ whose values are $n \in \{1, ..., N\}$, a colormap $\{\boldsymbol{I}_n\}_{n=1}^{N}$, and cluster sizes $\{w_n\}_{n=1}^{N}$
2: **for** each pair of color channels $(c_1, c_2) \in \{RG, GB, RB\}$ **do**
3: $\quad$ Initialize accum_{c_1, c_2} to zero
4: $\quad$ **for** $\boldsymbol{A} = (m \cdot \Delta A, l \cdot \Delta A), m, l \in \{0, ..., M\}$ **do**
5: $\quad\quad$ **for** $\theta_k = \frac{\pi}{K}, k \in \{1, ..., K\}$ **do**
6: $\quad\quad\quad$ **for** $n \in \{1, ..., N\}$ **do**
7: $\quad\quad\quad\quad d = |(\boldsymbol{A} - \boldsymbol{I}_n(c_1, c_2)) \times (\cos(\theta_k), \sin(\theta_k))|$
8: $\quad\quad\quad\quad$ **if** $(d < \tau) \wedge (m \cdot \Delta A > I_n(c_1)) \wedge (l \cdot \Delta A > I_n(c_2))$ **then**
$\quad\quad\quad\quad\quad \text{accum}_{c_1, c_2}(k, m, l) += w_n \cdot f(\|\boldsymbol{A} - \boldsymbol{I}_n\|)$
9: $\hat{\boldsymbol{A}} = \arg\max\{\text{accum}_{R,G} \otimes \text{accum}_{G,B} \otimes \text{accum}_{R,B}\}$, where $\otimes$ is an outer product
10: Return

---

where $\mathbb{1}[\cdot]$ is an indicator function that equals 1 if true and 0 otherwise. The term $\mathbb{1}[\boldsymbol{A} > \boldsymbol{I}(\boldsymbol{x})]$ equals 1 if all elements of $\boldsymbol{A}$ are greater than the corresponding elements of $\boldsymbol{I}(\boldsymbol{x})$.

A huge value of $\boldsymbol{A} \gg 1$ might be chosen as the solution, since it maximizes Eq. 7 with all of the pixels in the same large cone. To prevent this, we give a larger weight to values of $\boldsymbol{A}$ that are close to the pixels' values. Formally, we optimize:

$$\arg\max_{\boldsymbol{A}} \sum_{\boldsymbol{x}} \sum_{k} f\left(\|\boldsymbol{I}(\boldsymbol{x}) - \boldsymbol{A}\|\right) \cdot$$
$$\mathbb{1}[d\left(\boldsymbol{I}(\boldsymbol{x}), (\boldsymbol{A}, \phi_k, \theta_k)\right) < \tau] \cdot \mathbb{1}[\boldsymbol{A} > \boldsymbol{I}(\boldsymbol{x})], \tag{8}$$

where $f(y) = 1 + 4 \cdot e^{-y}$ is a fast decaying weight that gives preference to values of $\boldsymbol{A}$ in the vicinity of the pixels' distribution.

3.3. Optimizing Computational Efficiency

The proposed scheme, which includes collecting votes from all pixels for all angles and air-light candidates in the 3D RGB space, is computationally expensive. Therefore, we propose the following approximations, which significantly accelerate the computation while maintaining accuracy. The first, clustering the colors in the image and using the cluster centers instead of all the pixels. The second, performing the voting scheme in two dimensions. The voting is repeated three times, with only two of the (R, G, B) color channels being used each time.

Color clusters. Before we start the Hough voting we quantize the image into N clusters. We do this by converting the RGB image into an indexed image with a unique color palette of length N. This gives us a set of N typical color values, $\{\boldsymbol{I}_n\}_{n=1}^{N}$, where N is much smaller than
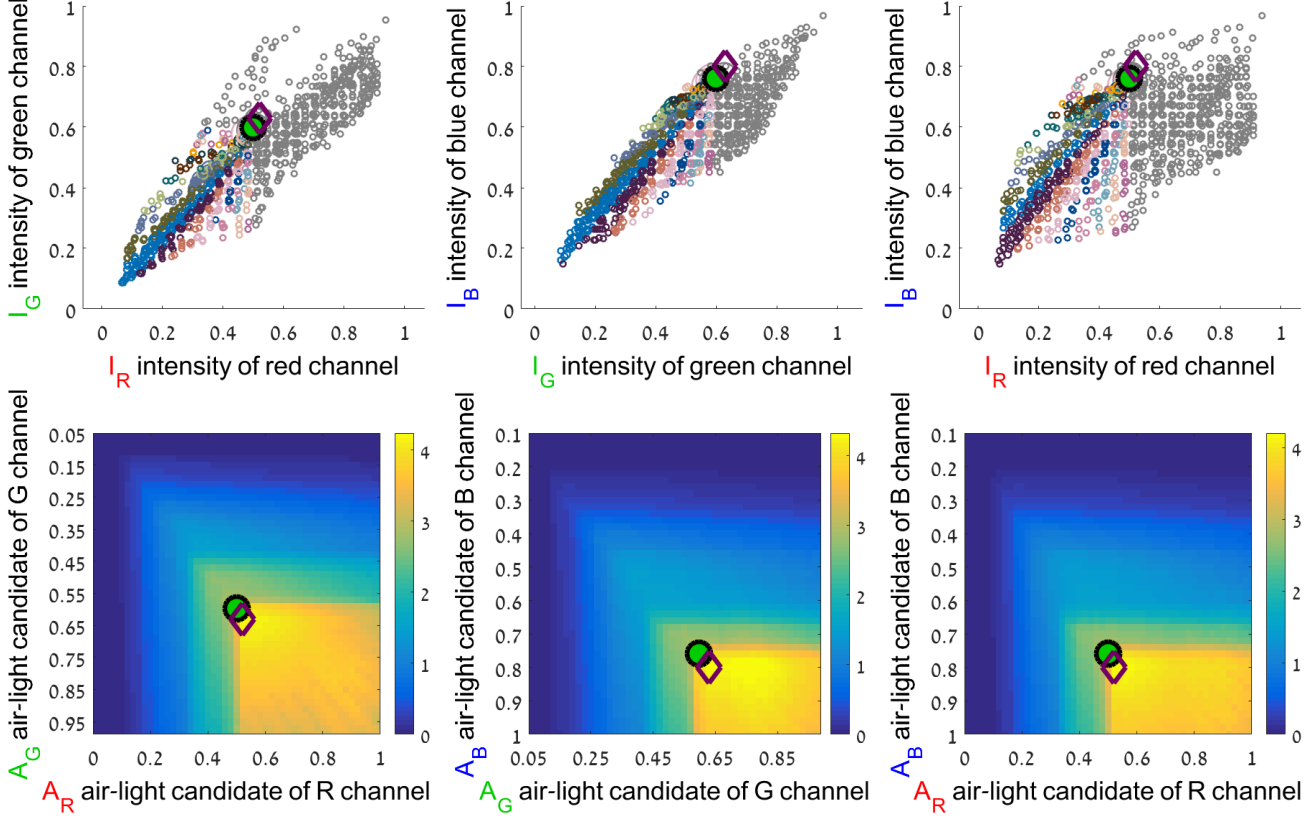
Figure 2. **Hough votes** for the *Schechner* image (Fig. 4). [Top] The color clusters $\{\boldsymbol{I}_n\}_{n=1}^N$ are projected onto 3 different 2D planes. Each cluster n is marked by a circle with a size proportional to w_n. The ground-truth (GT) air-light is marked by a green circle while our estimate is marked by a purple diamond. Each colored cluster votes for the GT value, where different colors indicate different haze-lines. The gray colored clusters do not vote for the GT since the following holds: $\mathbb{1}[\boldsymbol{A} > \boldsymbol{I}_n] = 0$. [Bottom] The three voting arrays, $\mathrm{accum}_{c_1, c_2}$, $(c_1, c_2) \in RG, GB, RB$. Best viewed in color.

the number of pixels in the image. In addition, we have $\{w_n\}_{n=1}^N$, the number of pixels in the image belonging to each cluster. During the Hough voting procedure, each representative color value $\boldsymbol{I}_n$ votes based on its distance to the candidate air-light, and the vote has a relative strength w_n. Therefore, the final optimization function is:

$$\arg\max_{\boldsymbol{A}} \sum_n \sum_k w_n \cdot f\left(\|\boldsymbol{I}_n - \boldsymbol{A}\|\right) \cdot$$
$$\mathbb{1}[d\left(\boldsymbol{I}_n, (\boldsymbol{A}, \phi_k, \theta_k)\right) < \tau] \cdot \mathbb{1}[\boldsymbol{A} > \boldsymbol{I}_n] \quad . \quad (9)$$

Two-dimensional vote. Calculating the full 3D accumulator for all possible air-light values is computationally expensive. Therefore, we perform this calculation in a lower dimension. The accumulator can be seen as the joint probability distribution of the air-light in all three color channels, where the final selected value is the one with the maximal probability. By performing the accumulation two color channels at a time, we calculate three marginal probabilities, where each time the summation is performed on a different color channel. Finally, we look for a candidate air-

light that will maximize the 3D volume created by the outer product of the marginal accumulators.

The proposed method is summarized in Alg. 1.

4. Experiments

We validate the proposed method on a diverse set of images. In all of our experiments we use the following parameters: $N = 1000$, the number of color clusters for each image (some images have less typical colors, resulting in empty clusters and $N < 1000$ in practice); $K = 40$, the number of angles, i.e., haze-lines, in each plane; all of the pixels' intensities are normalized to the range $[0, 1]$, and therefore we set $\Delta A = 0.02$ and $M = \frac{1}{\Delta A}$; the threshold $\tau_0 = 0.02$ determines whether a pixel $\boldsymbol{I}_n$ supports a certain haze-line.

4.1. Algorithm Visualization

Fig. 2 [Top] shows the distribution of the clustered pixels of the *Schechner* image (shown in Fig. 4), in RGB space. We show 2D plots since these projections are used in the
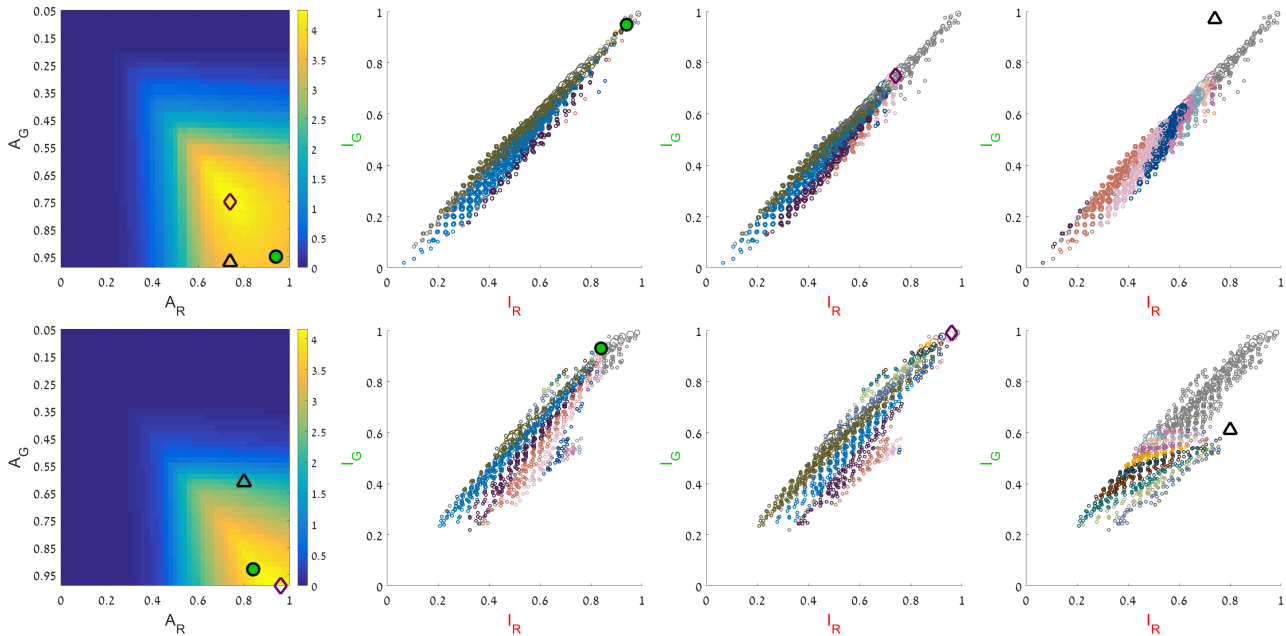
Figure 3. **Votes and distributions in failure cases.** The left column shows $\mathrm{accum}_{R,G}$ for each image, while the other columns show the breakdown of votes for three air-light candidates. Each colored cluster votes for the marked candidate, where different colors indicate different haze-lines, and gray colored clusters do not vote. The air-light candidates are, from left to right: the GT (green circle), our estimate (purple diamond), and a different candidate that received less votes (black triangle). [Top] Votes for *Forest* (Fig. 1), here we have a *magnitude* error. [Bottom] Votes for *Road* (Fig. 4), here we have an *orientation* error. Best viewed in color.

2D voting procedure (step 8 in Alg. 1) as well as provide better visualization. Each cluster n is marked by a circle with a size proportional to w_n. The ground-truth air-light is marked by a green circle while our estimate is marked by a purple diamond. The air-light is pointed at by the strongest haze-lines. Each colored cluster votes for the ground-truth value, where different colors indicate different haze-lines. The gray colored clusters do not vote for the ground-truth since the following holds: $\mathbb{1}[\boldsymbol{A} > \boldsymbol{I}_n] = 0$.

Fig. 2 [Bottom] depicts the three Hough transform arrays, accum_{c_1,c_2}, $(c_1,c_2) \in RG, GB, RB$ as a function of the candidate air-light values. The color-map indicates the number of votes. In this case, the ground-truth air-light had the most votes in all planes (strong yellow color).

Fig. 3 shows two failure cases of our method. On the left $\mathrm{accum}_{R,G}$ is depicted with three different air-light candidates marked by a green circle (the GT value), a purple diamond (our estimation) and a third one, a value that did not receive enough votes (a black triangle). For each of these candidates, the supporting haze-lines are shown.

Fig. 3 [Top] illustrates the analysis of votes for the *Forest* image (Fig. 1). We incorrectly estimate the *magnitude* of the air-light, due to large clusters that were less bright than the air-light. On the right, we demonstrate the votes to a candidate that is far from the pixel distribution, and therefore its votes have a lower weight.

Fig. 3 [Bottom] shows the analysis of votes for the *Road*

image (Fig. 4). We estimate the air-light to have a higher red component (an *orientation* error). This error is caused by bright pixels that have a high red value. The candidate shown on the right, which is marked by a black triangle, is the point of convergence of numerous haze-lines, however it gains fewer votes than both the GT and our estimation as the lines pointing to it do not contain many pixels.

4.2. Natural Images

A diverse set of 40 images was used in [1] to quantitatively evaluate the accuracy of the estimated air-light. This set consists of 35 images that contain points at an infinite distance from the camera, whose color is the air-light $\boldsymbol{A}$. Five additional images were generated by cropping, so that the sky is no longer visible in the image, yet the air-light is known. This procedure verifies the algorithms' robustness in cases where the air-light is not visible (as is often the case, for example in aerial photos). The authors manually marked the distant regions to extract the colors which are used as ground-truth. Even though they graciously sent us their data, we also manually marked regions of extremely distant scene points, in order to evaluate the accuracy of the ground-truth, as well as the accuracy expected from an automatic algorithm. The median L_2 difference between our manual selections was 0.02, which we now consider a lower bound to the accuracy of automatic air-light estimation methods.

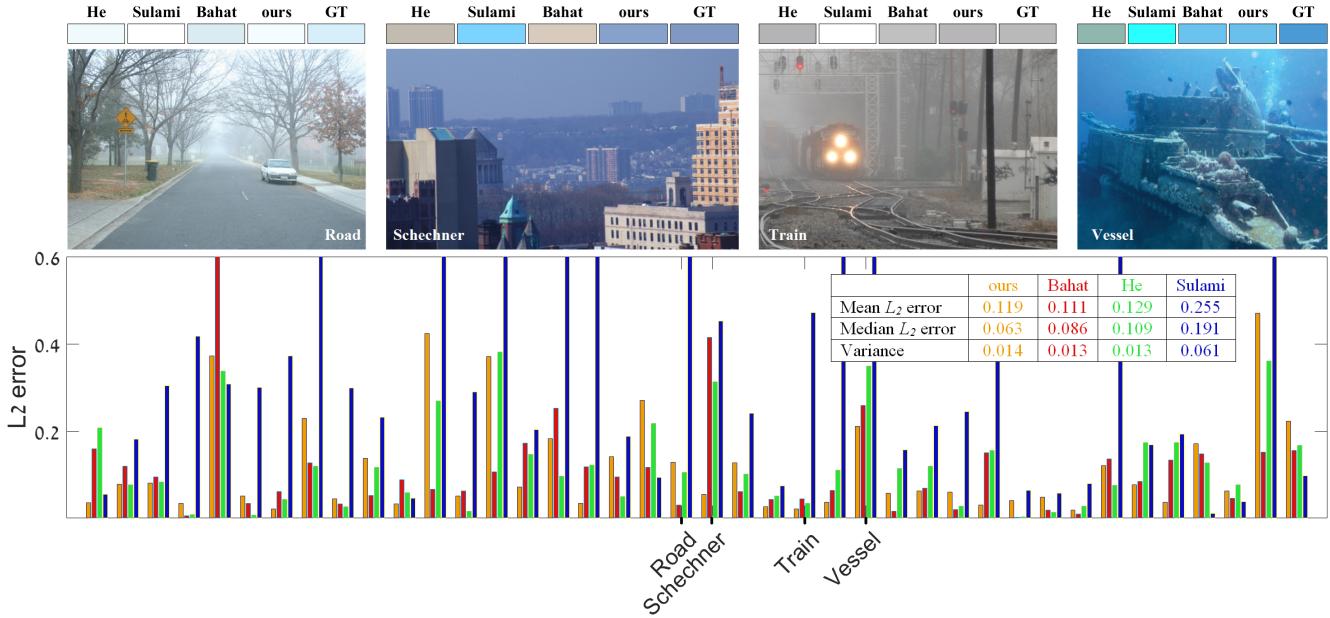|	ours	Bahat	He	Sulami
Mean L_2 error	0.119	0.111	0.129	0.255
Median L_2 error	0.063	0.086	0.109	0.191
Variance	0.014	0.013	0.013	0.061

Figure 4. **Evaluating the accuracy of the estimated air-light on natural images.** Top: Examples of hazy images, along with their manually extracted ground-truth air-light (**GT**), and the results of **Sulami** et al. [15], **He** et al. [7], **Bahat** and Irani [1], and **ours**. Hough votes for *Schechner* and *Road* are depicted in Figs. 2,3, respectively. Bottom: L_2 errors calculated on 40 hazy images (for which the ground truth could be manually reliably extracted).

The complete breakdown of this experiment is depicted in Fig. 4[bottom], where for each image we show the error of our method and the errors of [1, 7, 15]. We also present a table summarizing the errors. Generally, our method and [1] outperform [7] and [15]. Compared to [1], our method results in a lower median error, with slightly higher mean and variance. Interestingly, the performance comparison of both methods is not always consistent, i.e., on some images we perform better than [1], and vice versa. The performance depends on the extent the image adheres to the prior used by each method.

A few example photos are shown in Fig. 4[Top] together with the ground-truth air-light colors and the ones estimated by the four methods. The error bars corresponding to them in Fig. 4[Bottom] are labeled. In the *Road* image our error is larger than [1]. As seen in Fig. 3 this is caused by several bright pixels that have a high red value. In the *Schechner* image our method outperforms all methods. The Hough votes for this image are depicted in Fig. 2. In the *Train* image all methods except [15] perform well. In the *Vessel* image all methods yield relatively high errors. This is probably because the air-light is not uniform across the scene.

To make sure our method is not over-fit to the dataset from [1] we tested it on additional synthetic and natural images, detailed below.

We tested our algorithm on additional challenging natural hazy images that contain a clear patch of sky. First, we estimated the air-light using the entire image, and received an average error of 0.116 (median error 0.091).

Second, in order to test performance in images that do not contain sky patches, we repeated the estimation process with a cropped image. The average error increased to 0.231 (median error 0.25). The images are shown in Fig. 5, where the cropped region is marked by a dotted line. The estimated air-light values of the full and cropped images are shown, as well as the GT value extracted manually from the images. Our sky-less estimations are close to the ones estimated from the full image. The largest error, both before and after cropping, was calculated for the right image on the second row from the top - it had an L_2 error of 0.35.

4.3. Synthetic Images

In [15] hazy images were simulated from haze-free RGB images and their distance maps, gathered from the Lightfields [8] and the Middlebury [13] datasets. The transmission maps were calculated by $t(\boldsymbol{x}) = e^{-\beta d(\boldsymbol{x})}$, and β was chosen such that the most distant object in the scene received $t = 0.1$. The air-light magnitude was uniformly sampled in the range $[0.8, 1.8]$ and the orientation was uniformly sampled from the $10°$ cone around $[1, 1, 1]$. The sampling process was repeated three times for each image and the results are reported in [15]. The simulated hazy images are not available online, therefore we did our best effort to gather the clear images and followed the same protocol for synthesizing hazy images. Since these are not the exact same images, we cannot do a per-image comparison. Instead we report average and median errors in Table 1.

Some of the images in this dataset are indoor images,

	Orientation					Magnitude					l_∞ Endpoint Error				
	He	Tan	Tarel	Sulami	ours	He	Tan	Tarel	Sulami	ours	He	Tan	Tarel	Sulami	ours
Mean	3.218	3.576	3.253	0.581	**0.043**	0.172	0.218	0.412	**0.157**	0.178	0.147	0.177	0.278	**0.103**	0.141
Median	3.318	3.316	3.49	0.22	**0.037**	0.141	0.208	0.393	0.116	**0.095**	0.144	0.178	0.286	**0.077**	0.106

Table 1. **Errors on synthetic images**. Values for He [7], Tan [16], Tarel [17] and Sulami [15] are all taken from [15] for comparison. Our method evaluates air-light orientation significantly better than other methods, while it is comparable in magnitude. It was shown in [15] that estimating the orientation correctly is critical to ensure faithful colors in the dehazed image.
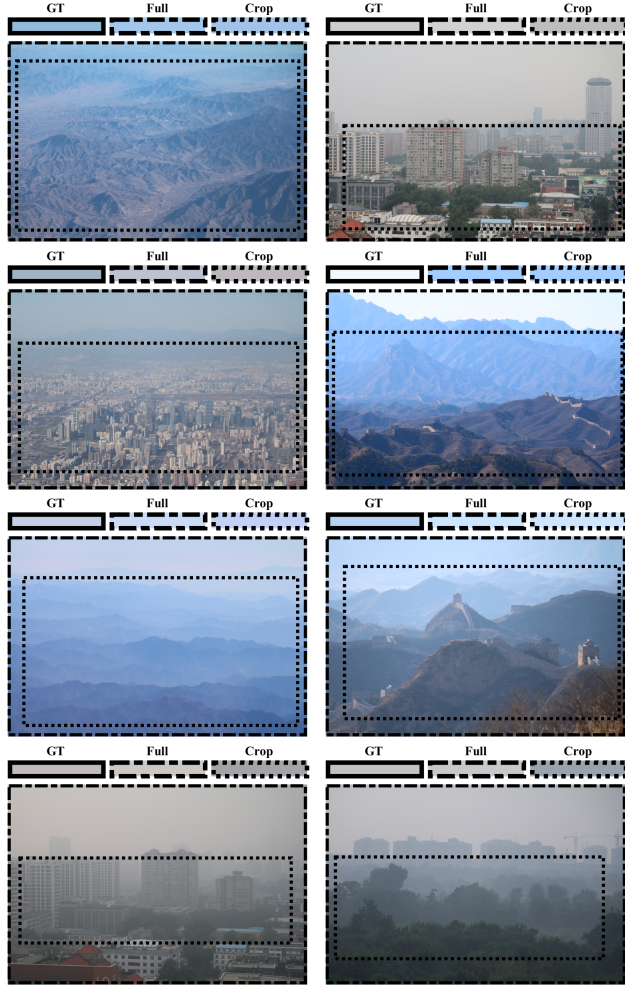

Figure 5. **Algorithm stability without visible sky.** The shown images were given as input to the algorithm twice, with the second time using only a portion of the image (marked by a dotted line), which does not contain a sky region. The estimated air-light colors are shown above the image, as well the GT air-light, which was extracted manually. Photographs courtesy of Dana Arazy.

whose depth distribution is significantly different from that of outdoor images. Despite that, our results are competitive. Specifically, our orientation estimation is the most accurate, which is significant. It has been shown in [15] that estimating the air-light's orientation is more important than its magnitude, since errors in the orientation induce color

Figure 6. **Synthetic images.** Two of the synthetic images tested. The average results on the entire set are reported in Table 1.

distortions in the dehazed image, whereas magnitude errors induce only brightness distortions. Fig. 6 shows two examples of synthetic images used in this experiment.

4.4. End-to-End Dehazing

For completeness, Fig. 7 shows end-to-end dehazing results using both the air-light estimation described in [7] and the proposed method, and the dehazing method [2]. Using different air-light values shows the effect of the estimation on the output dehazed image. The top row shows the *Forest* image, for which our estimated air-light has a magnitude error, as shown in Fig. 3[Top]. This estimation error leads to an error in the transmission map, and some distant regions look faded in the output, as seen in the area circled in black in Fig. 7b. The bottom row shows a successful example where the air-light was accurately estimated for the *Schechner* image. The wrong value estimated by [7] leads to a completely wrong transmission map in Fig. 7c, while the transmission in Fig. 7d approximately describes the scene structure.

4.5. Run-time

The algorithm's run-time depends on the following parameters: the number of pixels in the image P, the number of airlight candidates (in each color channel) M, the number of color clusters N and the number of haze-line orientations K. The conversion from RGB to an indexed image has a run-time complexity of $O(NP)$, while the air-light estimation using the indexed image has a run-time complexity of $O(NKM^2)$.

Notably, the proposed algorithm's complexity is linear

a) air-light estimation [7], dehazing [2]

b) proposed air-light estimation, dehazing [2]

c) air-light estimation [7], dehazing [2]

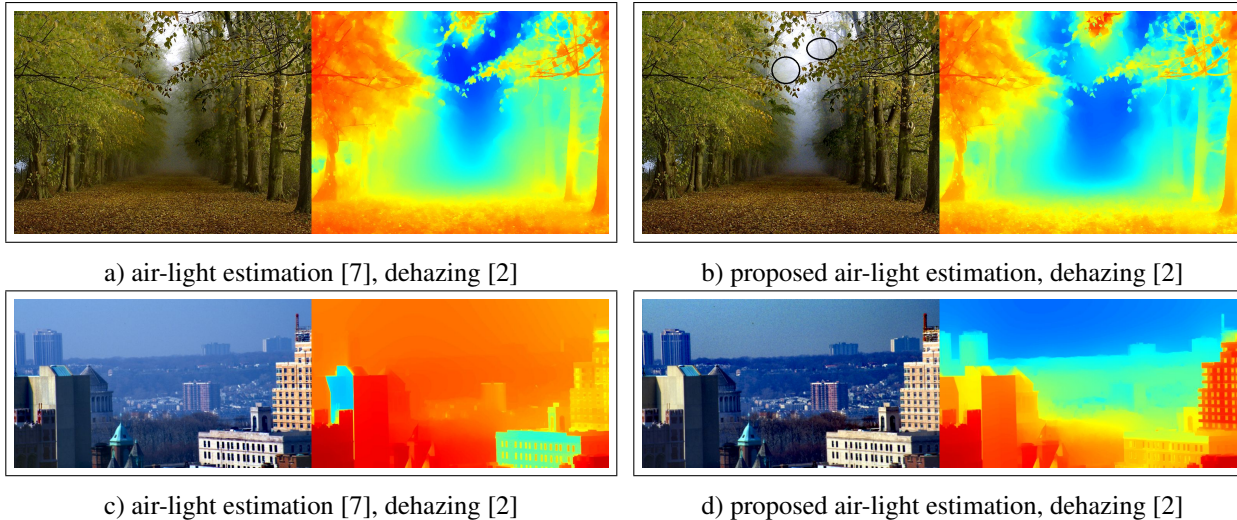d) proposed air-light estimation, dehazing [2]

Figure 7. **End to end dehazing.** Top row: dehazing results of *Forest*. Bottom row: dehazing results of *Schechner*. Left: using the air-light estimated by [7], Right: using the air-light estimated by the proposed method. The dehazing method [2] was used for dehazing.

in the number of pixels in the image, compared to [1, 15] which are quadratic.

As a reference, the run-time of our MATLAB implementation on a desktop with a 4th generation Intel core i7 CPU @3.4GHz and 32GB of memory is on average 6 seconds for a 1Mpixel image.

5. Conclusions

We presented a fast and efficient method for estimating a global air-light value in hazy images. The method is based on the haze-line prior that has recently been introduced. That prior claims that pixels' intensities of objects with similar colors form lines in RGB space under haze. These lines intersect at the air-light color and we take advantage of this observation to find their point of intersection.

We fix a set of line directions and search for a point so that all lines emanating from it, in the given line directions, will fit the data. For that we use Hough transform, where the point with the highest vote is assumed to be the air-light color. As running the Hough transform naïvely is computationally expensive, we proposed two techniques to accelerate the algorithm. One is to work in 2D instead of 3D by projecting pixels' values on the RG, GB and RB planes. The second is by clustering pixels' values, which enables us to collect votes for a candidate air-light only from cluster centers and weigh each vote by the cluster size, rather than collecting votes from all pixels.

The algorithm was evaluated on an existing dataset of natural images, as well as several synthetic images and additional natural images that we gathered. It performs well on all. Our algorithm is fast, and can be implemented in real-time.

References

[1] Y. Bahat and M. Irani. Blind dehazing using internal patch recurrence. In *Proc. IEEE ICCP*, 2016.

[2] D. Berman, T. Treibitz, and S. Avidan. Non-local image dehazing. In *Proc. IEEE CVPR*, 2016.

[3] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Proc. IEEE CVPR*, 2005.

[4] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Trans. on Image Processing*, 25(11):5187–5198, 2016.

[5] R. Fattal. Single image dehazing. *ACM Trans. Graph.*, 27(3):72, 2008.

[6] R. Fattal. Dehazing using color-lines. *ACM Trans. Graph.*, 34(1):13, 2014.

[7] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. In *Proc. IEEE CVPR*, 2009.

[8] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. H. Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32(4):73–1, 2013.

[9] Y. Li, R. T. Tan, and M. S. Brown. Nighttime haze removal with glow and multiple light colors. In *Proc. IEEE ICCV*, 2015.

[10] Y. Li, S. You, M. S. Brown, and R. T. Tan. Haze visibility enhancement: A survey and quantitative benchmarking. *arXiv:1607.06235*, 2016.

[11] W. E. K. Middleton. *Vision through the atmosphere*. Toronto: University of Toronto Press, 1952.

[12] M. T. Orchard and C. A. Bouman. Color quantization of images. *Trans. IEEE Signal Processing*, 39(12):2677–2690, 1991.

[13] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002.

[14] S. Shwartz, E. Namer, and Y. Y. Schechner. Blind haze separation. In *Proc. IEEE CVPR*, 2006.

[15] M. Sulami, I. Geltzer, R. Fattal, and M. Werman. Automatic recovery of the atmospheric light in hazy images. In *Proc. IEEE ICCP*, 2014.

[16] R. Tan. Visibility in bad weather from a single image. In *Proc. IEEE CVPR*, 2008.

[17] J. P. Tarel and N. Hautire. Fast visibility restoration from a single color or gray level image. In *IEEE Proc. ICCV*, 2009.