Interactive (De)Weathering of an Image using Physical Models[*]

Srinivasa G. Narasimhan and Shree K. Nayar
Computer Science Dept., Columbia University, New York, NY 10027
E-mail: {srinivas, nayar}@cs.columbia.edu

Abstract

Images of scenes acquired in bad weather have poor contrasts and colors. It is known that the degradation of image quality due to bad weather is exponential in the depths of the scene points. Therefore, restoring scene colors and contrasts from a single image of the scene is inherently *under-constrained*. Recently, it has been shown that *multiple* images of the same scene taken under different weather conditions or multiple images taken by varying imaging optics can be used to break the ambiguities in deweathering. In this paper, we address the question of deweathering a *single* image using simple additional information provided interactively by the user. We exploit the physics-based models described in prior work and develop three interactive algorithms to remove weather effects from, and add weather effects to, a single image. We demonstrate effective color and contrast restoration using several images taken under poor weather conditions. Furthermore, we show an example of adding weather effects to images. Our interactive methods for image (de)weathering can serve as easy-to-use plug-ins for a variety of image processing software.

1 Need for Interactive Deweathering

Images taken under bad weather conditions such as fog, mist, rain and snow suffer from poor contrasts and severely corrupted colors. In bad weather, the radiance from a scene point is significantly altered due to atmospheric scattering. The amount of scattering depends on the distances of scene points from the observer. Therefore, restoring clear day contrasts and colors of a scene from a single image taken in bad weather is inherently under-constrained.

Recently, there has been considerable research in the vision and image processing communities on color and contrast restoration in bad weather. Deweathering an image has been demonstrated when accurate scene depths are known [6, 8] and when precise information about the atmospheric condition is known [1]. In computer vision, algorithms have been developed to compute scene structure and restore scene contrasts and colors automatically without requiring any information about the atmosphere or scene depths. These algorithms break the ambiguities that exist in deweathering by using multiple images of the same scene taken under different weather conditions [3, 4] or multiple images acquired by varying the imaging optics [7].

In this work, we address the question of how to deweather a *single* image of a scene without using precise weather or depth information. Recall that previous work showed that multiple images of the same scene are necessary to break the ambiguities in deweathering. However, in many cases, it may not be possible to acquire multiple images. For instance, today, there are millions of pictures corrupted by weather, that are taken by amateur and professional photographers, with virtually no information about the depths or the atmosphere tagged to them. Very often, all we may have is a single photograph of a scene that we wish to deweather. In such cases, we will show that using minimal additional input from the user can successfully break the ambiguities in deweathering an image.

We begin by reviewing two scattering models [3, 4] that describe the colors and contrasts of a scene under bad weather conditions. Based on these models, we then present three algorithms to interactively deweather a single image. In all these cases, the user provides simple inputs through a visual interface to our physics-based algorithms for restoring contrasts and colors of the scene. The types of input (for instance, approximate direction in which scene depths increase, or a rough depth segmentation or a region of good color fidelity) may vary from scene to scene, but are easy to provide for a human user. We also use similar interactive methods to add physically-based weather effects to images.

We show several results that illustrate effective deweathering of both color and gray-scale images captured under harsh weather conditions. Our algorithms do not require precise information about scene structure or atmospheric condition and can thus serve as easy-to-use plug-ins for existing image processing software, such as Adobe Photoshop^{TM}. We believe that our interactive methods will make (de)weathering widely applicable.

2 Colors and Contrasts in Bad Weather

In this section, we review two single scattering models that describe colors and contrasts of scene points in bad weather. These models are used in our interactive methods to deweather, and add weather to images.

---

[*] This work was supported by a DARPA HumanID Contract (N000-14-00-1-0916) and an NSF Award (IIS-99-87979).

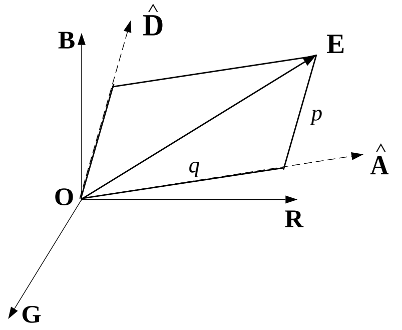


Figure 1: Dichromatic atmospheric scattering model. The color $\mathbf{E}$ of a scene point on a foggy or hazy day, is a linear combination of the direction $\hat{\mathbf{D}}$ of direct transmission (clear day) color, and the direction $\hat{\mathbf{A}}$ of airlight (fog or haze) color.

The dichromatic atmospheric scattering model [3] states that the color of a scene point $\mathbf{E}$ in fog or haze, observed by a color camera, is given by a linear combination of the direction $\hat{\mathbf{A}}$ of airlight (fog or haze) color, and the direction $\hat{\mathbf{D}}$ of the color of the scene point as seen on a clear day (see illustration in figure 1). Mathematically,

$$\begin{aligned} \mathbf{E} &= p\,\hat{\mathbf{D}} + q\,\hat{\mathbf{A}} \\ p &= Re^{-\beta d} \\ q &= E_\infty(1 - e^{-\beta d})\,, \end{aligned} \tag{1}$$

where, E_∞ is the sky brightness, R is radiance of the scene point on a clear day, β is the scattering coefficient of the atmosphere and d is the depth of the scene point. Note that the dichromatic model assumes that the scattering coefficient β is the same for all the color channels. Also, observe that the problem of deweathering an image, by computing clear day colors $p\hat{\mathbf{D}}$ solely from observed color vectors $\mathbf{E}$ is severely under-constrained.

The contrast or monochrome model [4] gives a mathematical expression for the intensity E of a scene point in bad weather, as recorded by a monochrome camera :

$$E = Re^{-\beta d} + E_\infty(1 - e^{-\beta d})\,. \tag{2}$$

As can be seen from both the models, the color and contrast of a scene point degrade exponentially with its depth from the observer[1]. Hence, traditional space invariant techniques for color and contrast enhancement cannot be used to satisfactorily deweather images. In the following sections, we describe our interactive techniques for image deweathering using simple inputs from the user.

3 Dichromatic Color Transfer

Consider a scene with points at different depths but with similar clear day colors. For instance, trees at different distances, or buildings at different depths, may have similar color directions (although their magnitudes may vary) on a clear day. In this scenario, the colors of near scene points are less corrupted by bad weather as compared to the distant scene points. We now describe an algorithm to transfer colors from nearby regions to replace colors of regions that are most effected by bad weather, in a physically consistent manner. In other words, we impose constraints based on the dichromatic model (1) to select colors of near scene points to replace colors of far scene points.

3.1 Interactive Step

Only two manual inputs are necessary for the color transfer algorithm. First, we select a nearby "good" region in the image, where colors $\mathbf{D}$ are not corrupted (or, minimally altered) by bad weather, as shown by the white rectangle in figure 2(a). Then, we mark a region (say, sky) that most resembles the color of airlight, as shown by the black rectangle in figure 2(a)[2]. The average color within this region is computed to estimate the direction $\hat{\mathbf{A}}$ of airlight color.

3.2 Automated Step

For each pixel, with color $\mathbf{E_i}$, in the weather effected region, we search for the best matching color in the "good" region. The search is restricted to a set of pixels in the "good" region that satisfy the dichromatic planarity (1),

$$\mathbf{E_i}.(\mathbf{D} \times \hat{\mathbf{A}}) = \mathbf{0}\,.$$

From this set, we choose a pixel whose color $\hat{\mathbf{D}}_\mathbf{i}$ is farthest (in terms of angle) from the fog color $\hat{\mathbf{A}}$, using

$$min\,\{\hat{\mathbf{D}}.\hat{\mathbf{A}}\}\,.$$

In order to compute the magnitude of the color used to replace the pixel $\mathbf{E_i}$, we use the dichromatic model (1) to decompose the scene color $\mathbf{E_i}$ into two components :

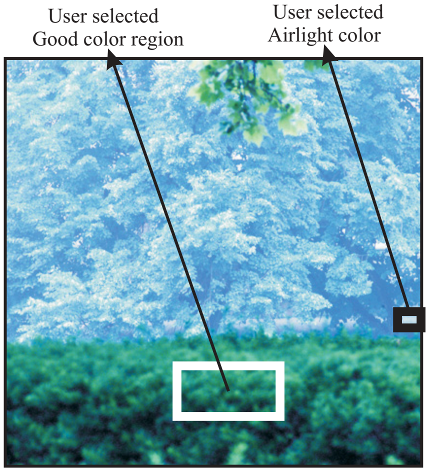
$$\mathbf{E_i} = p\,\hat{\mathbf{D}}_\mathbf{i} + q\,\hat{\mathbf{A}}\,.$$

Finally, we replace the color $\mathbf{E_i}$ of the pixel by the deweathered color, $p\,\hat{\mathbf{D}}_\mathbf{i}$. *Note that the ambiguities in the dichromatic model are broken due to the presence of similar colored scene points at different distances. This algorithm does not require any information regarding scene depths or atmospheric conditions. Further, it*

---

[1] Note that these models are based on single-scattering and hence are not valid for turbulence and aerosol blurring and scattering by pollutants. If the atmosphere is non-homogeneous along the line of sight, β will be a function of depth d. Then scaled depth βd is replaced by optical thickness $T = \int_0^d \beta(x)dx$.

[2] If such a region does not exist in the image, then the user may provide the hue of the sky and assume the sky intensity to be the maximum intensity in the image. Another way of computing the color of airlight is by intersecting dichromatic planes of two different user provided scene colors [3].



(a) Input : Misty image

(b) Output : Color corrected image

Figure 2: Color correction by dichromatic color transfer. (a) Input misty image consisting of green bushes at different distances. A region of "good" color is marked in the white rectangle. A region that most resembles the color of mist is marked in the black rectangle. (b) Colors from the near "good" region are transfered to farther regions. Notice the bluish colors of the farther bushes replaced by greenish colors.

does not assume homogeneity of the atmosphere over the entire field of view. The result of applying this method is shown in figure 2 (b). Notice the significant change in the colors of the far bushes.

4 Deweathering using Depth Heuristics

A limitation of the color transfer method is that all colors in the weather effected region may not have corresponding colors in the "good" color region. In this section, we describe deweathering using heuristics on scene depths. Note that subtle weather effects within small depth ranges are not captured by a camera with limited dynamic range (say, 8 bits). Therefore, precise distances are not required for effective deweathering. Moreover, in many cases, it may be possible to input approximate "trends" in the depths of scene points (say, the direction of increasing depths). For instance, a scene with a street along the viewing direction is common in surveillance or tracking scenarios (see figure 6). The deweathering algorithm is detailed below.

4.1 Interactive Step

We select a region of the sky to obtain the sky intensity E_∞ (and sky color direction $\hat{\mathbf{A}}$, if the input is a color image). Then, the "depth trend" is interactively specified in the following manner. First, we input the approximate location of a vanishing point along the direction of increasing distance in the image (see red circle in figure 4). The distances of the scene points are inversely related to their image distances to the vanishing point. Next, we input the approximate minimum and maximum distances and interpolate distances (say, using a linear or quadratic function) for points in between. For illustration purposes, we used

$$d = d_{min} + \alpha \left(d_{max} - d_{min} \right), \tag{3}$$

where, $\alpha \in (0, 1)$ is the fractional image distance from a pixel to the vanishing point. For $d = d_{max}$, $\alpha = 1$ and for $d = d_{min}$, $\alpha = 0$. The resulting depth trend is shown in figure 3 (a).

4.2 Automated Step

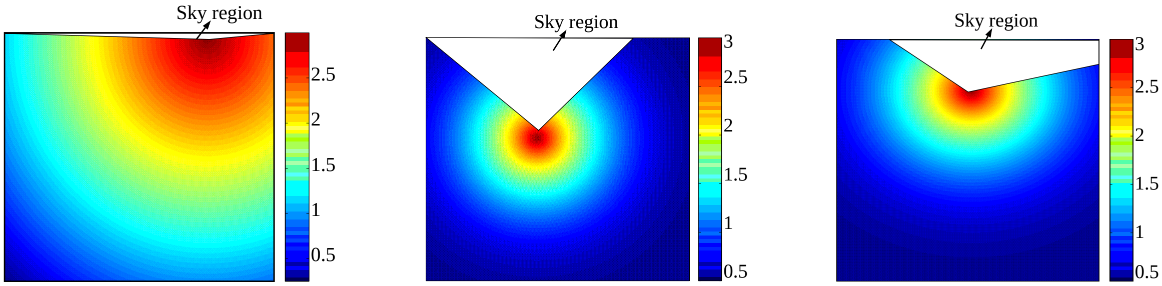
Consider the model given in equation 2. At every pixel, the depth estimate d is known, and the sky brightness E_∞ is measured. Generally, the atmosphere condition remains constant (or varies slowly) over small distance ranges and fields of view that are relevant to computer vision applications. If we assume homogeneity of the atmosphere, then the scattering coefficient β is constant for all pixels in the image. Then, note that different values of the scattering coefficient β produce the effects of different densities of bad weather (moderate, heavy, etc.). Thus, by continuously changing β (imagine a slider in Adobe Photoshop^{TM}), we can progressively estimate the clear day radiances R at each pixel as,

$$R = \left[E - E_\infty (1 - e^{-\beta d}) \right] e^{\beta d} \, . \tag{4}$$

Similarly, note that the dichromatic model (equation 1) can be used to restore colors in an RGB image. Therefore, while the color transfer method can be applied only to color images, this method can be applied to both color and gray-scale images. *In this case, the homogeneity of the atmosphere breaks the ambiguity in deweathering an image.* The results shown in figures 4, 5 and 6 illustrate that approximate depth information can be used effectively for image deweathering.

5 Restoration using Planar Depth Segments

In the previous section, we described an interactive technique where depth trends can be followed. However,



(a) (b) (c)

Figure 3: Depth heuristics used to deweather images shown in figures 4, 5 and 6 respectively. The vanishing point corresponding to the direction of increasing distances is marked. Approximate minimum and maximum distances are input to the algorithm and the intermediate distances are interpolated. The depths are not used for sky regions (empty spaces).

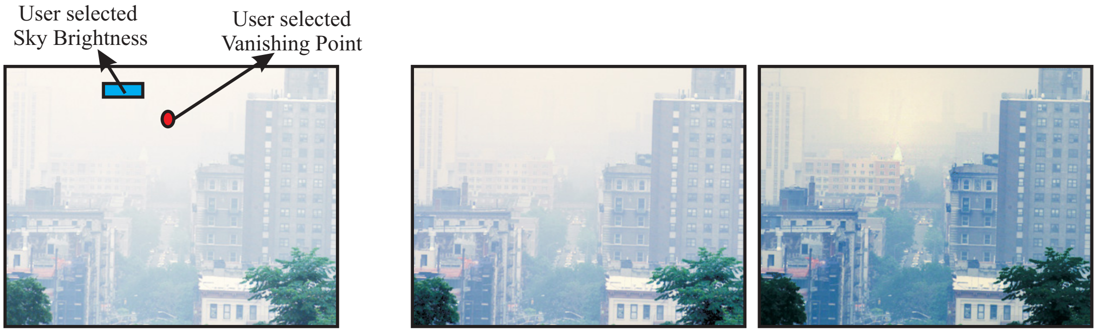
(a) Input : Misty image

(b) Output : Deweathering to different extents (by choosing different values for β)

(c) Zoomed-in regions of (a)

(d) Contrast and color restoration in zoomed-in regions.

Figure 4: Restoring clear day scene colors using depth heuristics. (a) Input image captured in mist. The colors and contrasts of scene points, especially in farther regions, are corrupted severely. (b) Two images illustrating different amounts of mist removed from the image in (a). These images were computed using the depth "trend" shown in figure 3(a). (c) Zoomed in regions selected from (a) at different depths showing different amounts of mist. (d) Corresponding zoomed in regions of the deweathered images. Notice the significant color and contrast enhancement.



(a) Input : Misty image

(b) Output : Deweathering to different extents
(by choosing different values for β)

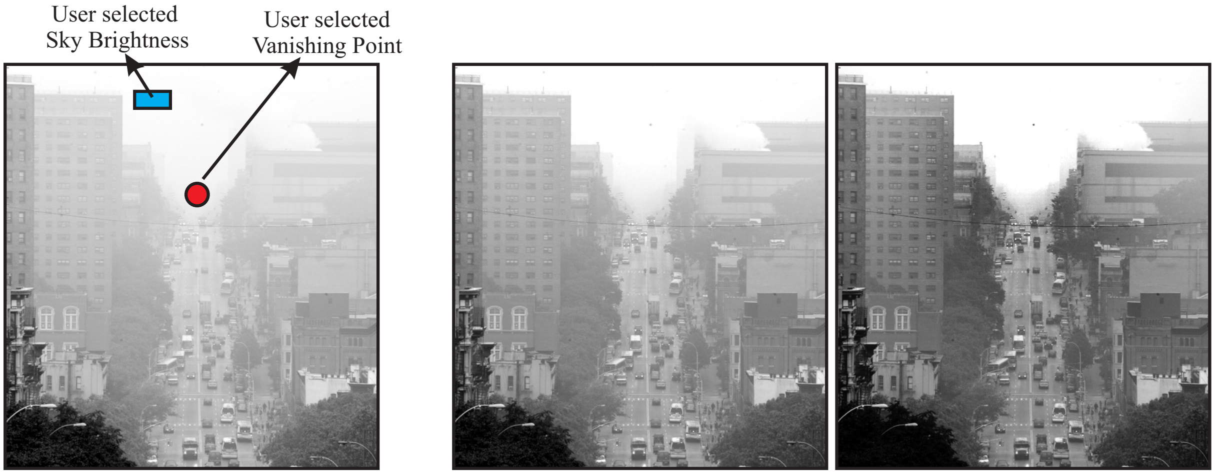
(c) Zoomed-in regions of (a)

(d) Contrast and color restoration in zoomed-in regions.

Figure 5: Restoring clear day scene colors using depth heuristics. (a) Input image captured in mist. The colors and contrasts of scene points, especially in farther regions, are corrupted severely. (b) Two images illustrating different amounts of mist removed from the image in (a). These images were computed using the depth "trend" shown in figure 3(b). (c) Zoomed in regions selected from (a) at different depths showing different amounts of mist. (d) Corresponding zoomed in regions of the deweathered images. Notice the significant color and contrast enhancement.

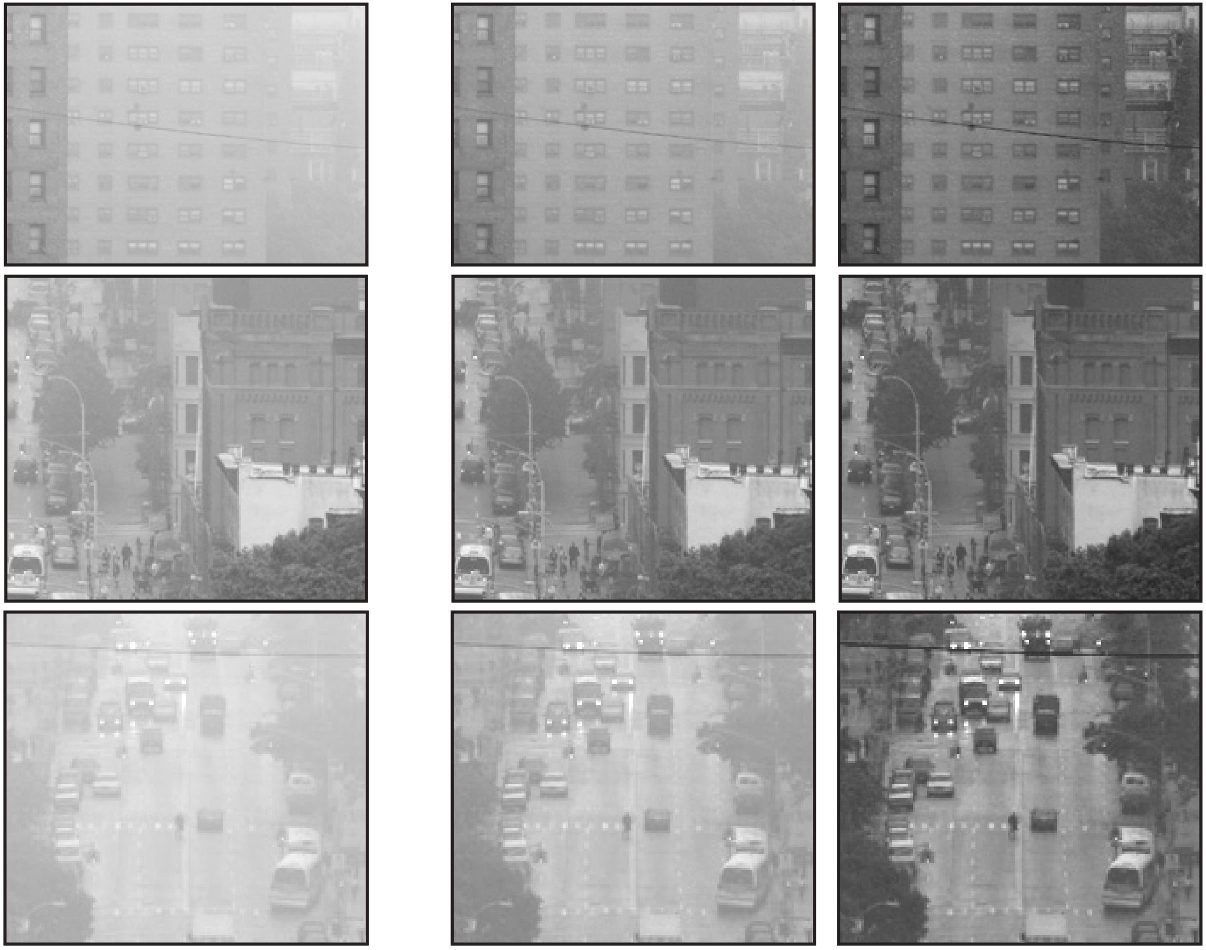
urban scenes with strong depth discontinuities and severe occlusions (induced by different buildings) are not suitable for the previous approach where depth trends were smoothly interpolated. In such cases, it is better to provide a rough depth segmentation of the scene. Recall that precise depth information is not needed to deweather images. For instance, the brightness levels of fog for a frontal planar surface are approximately equal to the brightness levels for a curved surface at the same distance. Thus, planar depth segments should suffice for deweathering in urban scenes (see figure 7 (b)).

The deweathering algorithm is similar to the one presented in the previous section. The depths, however, are provided as approximate planes. In our experiments, we used images from the Columbia Weather and Illumination Database (WILD) [2]. Orthographic depths were obtained from satellite orthophotos (see figure 7 (b)). Once again, the sky brightness E_∞ was measured by selecting a region of the sky. The images can be deweathered by computing clear day scene radiances R or colors $p\hat{\mathbf{D}}$ depending on whether a gray-scale or a color image is input to the algorithm. Results of deweathering a misty scene is shown in figure 7 (c). Notice the significant increase in contrasts of the scene points at various depths. In summary, the above results demonstrate that weather effects can be sufficiently removed from images even when only approximate depths are known.



(a) Input : Foggy image

(b) Output : Deweathering to different extents
(by choosing different values for β)

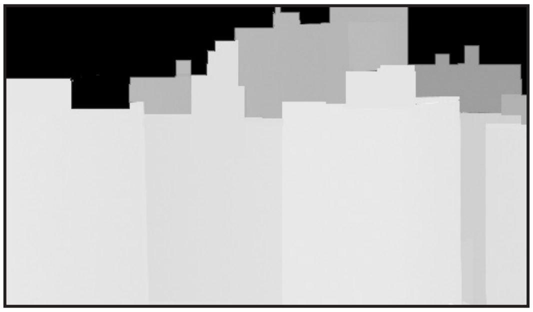


(c) Zoomed-in regions of (a)

(d) Contrast restoration in zoomed-in regions.

Figure 6: Restoring clear day scene contrasts using depth heuristics. (a) Input gray-scale image captured in fog. The contrasts of scene points, especially in farther regions, are degraded severely. (b) Two images illustrating different amounts of fog removed from the image in (a). These images were computed using the depth "trend" shown in figure 3(c). (c) Zoomed in regions selected from (a) at different depths showing different amounts of fog. (d) Corresponding zoomed in regions of the deweathered images. Notice the significant contrast enhancement.

(a) Input: Misty image



(b) Input: Planar depth segments

(c) Output: Contrast enhanced image.

Figure 7: Contrast restoration using planar depth segments. (a) A misty image. (b) Planar depths obtained from satellite orthophotos. (c) Output of the restoration algorithm. Notice the increase in clarity in the farther buildings. The input image and the depths were obtained from the Columbia Weather and Illumination database (WILD).

6 Adding Weather Effects to Images

Thus far we presented algorithms to remove weather effects from a single image. However, in computer graphics applications, it may be required to add atmospheric effects to images. We now demonstrate the addition of weather effects to a single photograph using a manually provided approximate depth map.

We add two types of scattering effects to an image. Note that the intensity of a light ray diminishes when it is scattered by an atmospheric particle. Hence, the number of scattering events each light ray can undergo and still be detectable by the camera, is small. Hence, single scattering models (equations 1 and 2) suffice to add weather effects to images of surfaces in the scene. However, we observe distinct glows around bright light sources in the scene. These glows are due to the multiple scattering of light in the atmosphere. Recently, Narasimhan and Nayar [5] developed an analytic model for the glow around a point light source. In this section, we use this model and present an algorithm to add weather effects to clear day (or night) scenes with both light sources and surfaces.

6.1 Interactive Step

We manually provide depth heuristics and the sky intensities E_∞, as described in the previous sections. Then, we segment the light sources in the image. This step can also be done automatically using a simple thresholding operation on image intensities.

6.2 Automated Step

This step consists of two parts corresponding to the addition of single scattering effects to surfaces and multiple scattering effects to light sources. The pixel intensities of non-light source regions of the scene correspond to the clear day surface radiances R. By substituting different values for the scattering coefficient β in equation (2), we simulate different densities of weather to the region corresponding to surfaces.

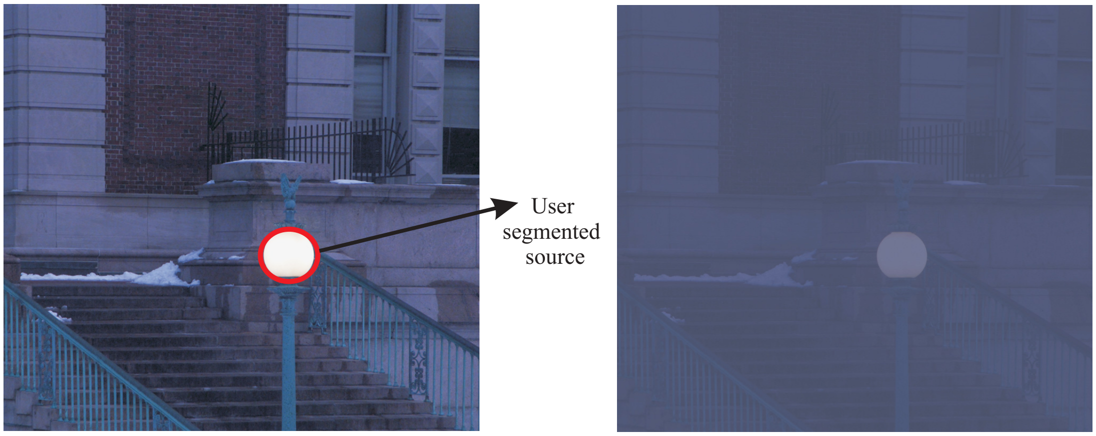
To add multiple scattering effects (glow) around each light source, we perform the following operations. The glow $I(T,\mu)$ around a point light source with radiant intensity I_0 is derived in [5],

$$\begin{aligned} I(T,\mu) &= \sum_{m=0}^{\infty} \left(g_m(T) + g_{m+1}(T)\right) L_m(\mu)\,, \\ g_m(T) &= I_0\, e^{-\beta_m T - \alpha_m \log T} \\ \alpha_m &= m+1 \\ \beta_m &= \frac{2m+1}{m}\left(1 - q^{m-1}\right)\,, \end{aligned} \tag{5}$$

where, μ is the cosine of the ray direction, optical thickness $T = \int_0^d \beta(x)dx$, L_m is a Legendre polynomial of order m, and $q \in [0,1]$ is called forward scattering parameter. Different values of q generate different types of weather conditions (fog, mist, haze, etc.). The glow around a point source can be viewed simply as a point spread function (PSF).

To add a glow around an area light source, the computed PSF is convolved with the shape of the light source. Finally, we combine the contributions of both single and multiple scattering (glows) to render a realistic appearance of a scene in bad weather. We demonstrate this technique using a scene photographed in the evening containing a bright lamp (figure 8). The final expression used to render weather effects to this image is,

$$[Re^{-T} + E_\infty(1 - e^{-T})]\,\delta(1-\mu) + I(T,\mu)\,, \tag{6}$$



(a) Input : Clear evening image

(b) Output : Single Scattering (min T = 1.05)

(c) Output : Mild Mist (q = 0.8, min T = 1.05)

(d) Output :Dense Fog (q = 0.9, min T = 2.0)

Figure 8: Adding weather effects to a photograph. (a) Original Photograph. (b) Hazy image rendered using single scattering (dichromatic model). (c) and (d) Different glows are added to the lamp. Note that multiple scattering effects due to light sources are significant as compared to single scattering effects.

where, δ denotes the Dirac delta function. Two different amounts of mist and fog ($q = [0.8, 0.9]$, min $T = [1.05, 2.0]$) were added to the image in figure 8(a). These results are illustrated in figures 8(c) and (d). Notice the realistic glowing appearance of the lamp and the natural attenuation of the background surface brightnesses. Also, compare the result with only single scattering shown in figure 8(b).

7 Summary

We presented simple interactive tools to remove weather effects from, and add weather effects to, a single image. Our methods do not require precise information about the scene or the weather condition, and do not require changes in weather conditions between image acquisitions. The three methods presented are easy-to-use and can effectively restore clear day colors and contrasts from poor weather images. We also presented an algorithm to add weather effects (both single and multiple scattering) to images. Although we showed four specific examples, it is clear that such simple techniques can be used for most scenes that occur often in practice.

References

[1] N.S. Kopeika. *A System Engineering Approach to Imaging.* SPIE Press, 1998.

[2] S. G. Narasimhan, C. Wang, and S. K. Nayar. All the images of an outdoor scene. *In Proc. ECCV*, 2002.

[3] S.G. Narasimhan and S.K. Nayar. Vision and the atmosphere. *IJCV*, 48(3):233–254, August 2002.

[4] S.G. Narasimhan and S.K. Nayar. Contrast restoration of weather degraded images. *PAMI*, 25(6), June 2003.

[5] S.G. Narasimhan and S.K. Nayar. Shedding light on the weather. *In Proc. CVPR*, 2003.

[6] J.P. Oakley and B.L. Satherley. Improving image quality in poor visibility conditions using a physical model for degradation. *IEEE Trans. on Image Processing*, 7, February 1998.

[7] Y.Y. Schechner, S.G. Narasimhan, and S.K. Nayar. Instant dehazing of images using polarization. *In Proc. CVPR*, 2001.

[8] K. Tan and J.P. Oakley. Physics based apprach to color image enhancement in poor visibility conditions. *JOSA A*, 18(10):2460–2467, October 2001.