Project Report : Single Image Haze Removal Using Color Attenuation Prior

Sean Liu, *Student*, *EL-GY 6123 Image and Video Processing*

***Abstract*—Haze removal is very difficult especially from a single image. In this project, I try to dehaze a single haze image using color attenuation prior. And I change the trying model to get a set of new parameters to process the input image. I will clarify this method and introduce how I achieve it and change the model. Last, I will compare the results of several different method.**

Keywords—Dehazing, atmospheric scattering model, depth map.

I. Overview of the Project

I am the only one to do this in the project. So, it's not a team project. My main work here is to understand he principle of haze removal using color attenuation prior, use python to implement the algorithm in the paper and find ways op optimize training model.

II. Theory

A. Atmosopheric Scattering Model

This model is peoposed by McCartney in 1976[1], and is derived by Narasimhan and Nayar later[2][3][4]. The model I use here can be expressed as :

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

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

In this equatuion, **I** is the haze image, **J** is the haze-free image. So, if I get **A** and ***t***, I can process the input haze image to haze-free image. As for ***t***, we can expressed it by *d*, the depth map and a constant, β, which can be seen as 1.0 in the most situation.

And **A**, the atmospheric light value can be estimated by the dollowing equation:

$$\mathbf{A} = \mathbf{I}(\mathrm{x}), \mathrm{x} \in \{\mathrm{x} | \forall \mathrm{y} : \mathrm{d}(\mathrm{y}) \leq \mathrm{d}(\mathrm{x})\}$$

So, the main goal to dehaze an image is to find its depth map.

B. Dark Channel Prior method for haze removal

This method is proposed be Kaiming He in 2011 [5]. It based on a powerful assumption that in most of the local regions which do not cover the sky, some pixels (called dark pixels) very often have very low intensity in at least one color (RGB) channel. Fig. 2 [5] shows several outdoor images and the corresponding dark channels [5]. We can see that the dark channel values in non-haze image are very small, however in haze image they are getting larger when there's haze in that region. Therefore, these dark pixels can directly provide an accurate estimation of the haze transmission [5]. Then we can use the transmission to generate the depth map, which can help us to obtain the dehaze image.

The way we used to decide transmission value ***t*** can be expressed by dark channel assumption as the following equation:

$$\tilde{t}(x) = 1 - \mathrm{darkmap}(x)$$

$$\mathrm{darkmap}(x) = \min_{y \in \Omega(x)} \left(\min_{c} \frac{I^c(y)}{A^c}\right)$$

C. Color Attenuation Prior method for haze removal

This method is proposed by Qingsong Zhu. etc in 2015 [6]. Its assumption is that the depth value of a pixel in an image will increase when there is a larger difference between its brightness value and its saturation value. And this assumption is shown in Fig.1 [6]. We can easily build a linear model like this:

$$d(x) = \theta_0 + \theta_0 v(x) + \theta_0 s(x) + \varepsilon(x)$$

In this assumption, if all three θ parameters are known, *d* is not hard to estimate. So, all we have to do is find a lot of haze images and their depth maps, and then do linear regression on this model.

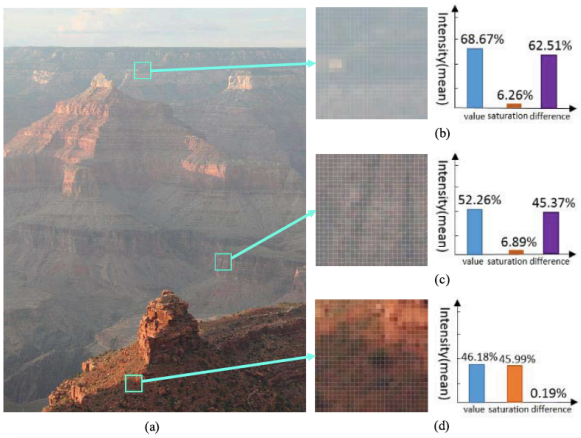


Fig 1. (a) A haze image. (b) The close-up patch of a dense-haze region and its histogram. (c) The close-up patch of a moderately hazy region and its histogram. (d) The close-up patch of a haze-free region and its histogram.

D. Values that influence depth map

In the color attenuation prior method, the author find the brightness value and satuation can influence the depth value. And in the dark channel prior method, the author find transmission can be influenced by dark channel value, the lowest value in RGB channels. And we know that transmission value and depth value can be expressed by each other. So, in this project, I try to find if the result can be better when I put dark channel value into the linear model proposed by Qingsong Zhu. Etc.

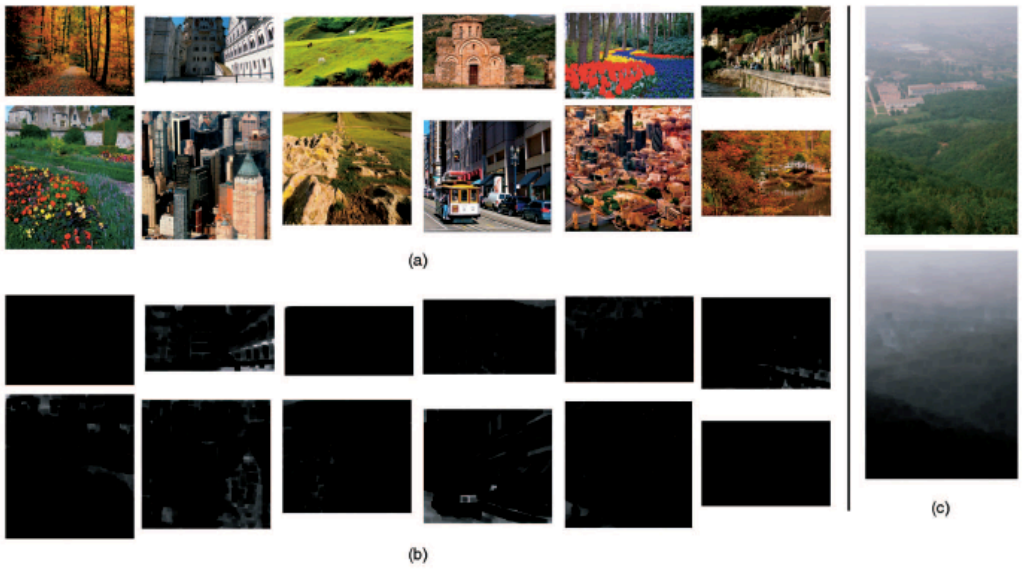


Fig 2. (a) Example images in our haze-free image database. (b) The corresponding dark channels. (c) A hazy image and its dark channel.

III. Project Accomplishment

In this part, I will introduce what I have done in this project and show the results.

A. Training Data Collection

Because what we do is for single haze image, so it's very hard for us to know what exactly the depth map of a haze imagedue due to the fact that there is no reliable means to measure the depths in outdoor scenes. So, I use the method in Zhu's paper to collect the training data[6]. The process can be expressed in Fig 3.

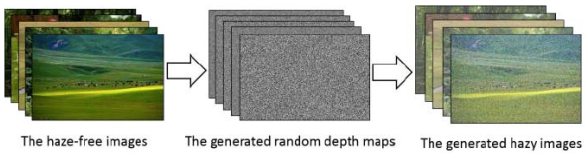


Fig 3. The process of generating the training samples

1) I collect more than 300 haze images from Flickr with the keyword, natural scenery and landscape. These haze-free images can be seen in Fig 4.

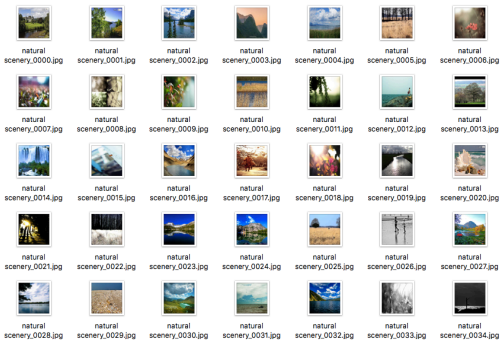


Fig 4. More than 300 haze-free images

2) I generate the random atmospheric light A (k , k , k) where the value of k is around 0.65. And one of these map is shown in Fig 5.

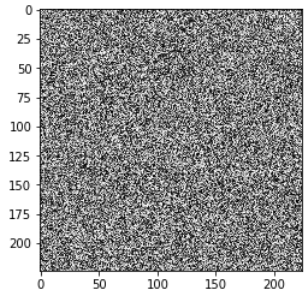


Fig 5. One of those random depth maps

3) Add depth maps to haze-free images using atmosopheric scattering model. And I will get a set of haze images which I know their deps value. They are shown in Fig 6.

Fig 6. Four of those haze images for trining

B. Training model

I also use a linear model to train the data. However I add anothoer input value to the model, so I will get four θ values after linear regression. And in this situation the error ε is pretty, so I ignore the error value when I use the paremeters to recover haze image.

$$d(x) = \theta_0 + \theta_1 v(x) + \theta_2 s(x) + \theta_3 dc(x) + \varepsilon(x)$$

When I train the data, I use LinearRegression function form sklearn to minimize the least square error.Codes in this part can be seen in Fig 7.

```
from sklearn import datasets, linear_model
regr = linear_model.LinearRegression()
regr.fit(X,dr)
print('regr.intercept=',regr.intercept_)
print('regr.coef=',regr.coef_)

regr.intercept= 0.479486076361
regr.coef= [ 0.64872891 -0.88159268 -0.42438411]
```

Fig 7. Python codes to train the model and the four paremeters I got

(a) (b) (c) (d)

Fig 11. (a)Haze image. (b) My result. (c) Result from original color attenuation prior. (d) Result from dark channel prior

C. *Results I got*

I write some python codes to caculate the depths by those for parameters and this function can be seen in Fig 8. And one of the result depth map is shown in Fig 9. Also, I will show two result depth maps recovered by Dark Channel Prior and Original Color Attenuation Prior in Fig 10.

```
def calDepthMap2(I, r):
    hsvI = cv2.cvtColor(I, cv2.COLOR_BGR2HSV)
    s = hsvI[:,:,1] / 255.0
    v = hsvI[:,:,2] / 255.0
    b,g,R = cv2.split(I)
    dc_min = cv2.min(cv2.min(R,g),b)/255
    output =  0.47875879006 + 0.64867465 * v -0.88036*s  -0.42327269*dc_min
    outputPixel = output
    output = scipy.ndimage.filters.minimum_filter(output,(r,r))
    outputRegion = output
    cv2.imwrite("data/outputRegion.jpg", outputRegion*255 )
    return outputRegion, outputPixel
```

Fig 8. Python function to recover depth map.

Fig 9. Mine depth map.

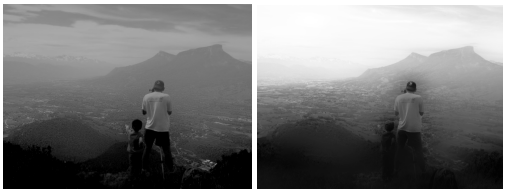
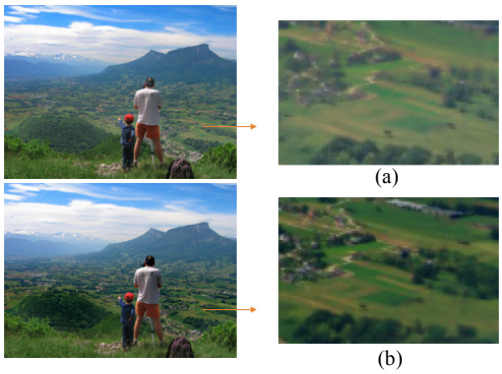


Fig 10. Original color attenuation prior depth map (left) and dark channel prior depth map (right).

Then I use my depth maps to get the transmission maps and put them into the atmosopheric scattering model to obtain haze-free image. I compare the results of those three method in Fig 11.

We can see that both results based on color attenuation method work well in this picture, which has a large sky region. And dark channel prior doesn's work well in those sky region. As for Fig 11(b)(c), we can see some regions that have low depth value in Fig 12. It's clearly that my result works slightly well in those places and recover an image with a better saturation values.



(a)

(b)

Fig 12. Details. (a) Result from original color attenuation prior. (b) My result.

IV. Summary

In this project, I learn a lot. I've know some method to process haze images and compared their results. I change the model in original color attenuation prior method and get some different parameters to recover haze-free image. And in some situation, like the haze picture in Part III, my system works better. But there is still a lot to do for me. I know my system and traning method can not get a good result in some other pictures. And in most pictures without a large sky region, dark color prior method works much better than mine. So, in the future I may try to input more different values and build a nerual network to do this training.

Appendix

1. A folder of sourse code.
2. Two PDF verison of my code.
3. A presentation PPT file.

References

[1] E. J. McCartney, Optics of the Atmosphere: Scattering by Molecules and Particles. New York, NY, USA: Wiley, 1976.

[2] S. K. Nayar and S. G. Narasimhan, "Vision in bad weather," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), vol. 2. Sep. 1999, pp. 820–827.

[3] S. G. Narasimhan and S. K. Nayar, "Contrast restoration of weather degraded images," IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 6, pp. 713–724, Jun. 2003.

[4] S. G. Narasimhan and S. K. Nayar, "Vision and the atmosphere," Int. J. Comput. Vis., vol. 48, no. 3, pp. 233–254, Jul. 2002.

[5] K. He, J. Sun, and X. Tang, "Single Image Haze Removal Using Dark Channel Prior," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 12, pp. 2341–2353, Dec. 2011.

[6] Q. Zhu, J. Mai, and L. Shao, "A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior," IEEE Trans.Pattern Anal. Mach. Intell., vol. 24, no. 11, pp. 3522–3533, Nov, 2015.