Fast Single Image Haze Removal Using Dark Channel Prior and Bilateral Filters

Rachel Yuen, Chad Van De Hey, and Jake Trotman
rlyuen@wisc.edu, cpvandehey@wisc.edu, trotman@wisc.edu

UW-Madison Computer Science
1210 W Dayton Street
Madison, WI 53706
http://dehazing.azurewebsites.net

ABSTRACT *In this paper, we propose a simple yet effective improvement to single image haze removal using dark channel prior (He, Sun, and Xiaoou). It is based on the observation that haze removal using dark channel prior accurately removes haze from an image; however, it has a relatively slow computation time. Using the dark channel prior in addition to a series of bilateral filters we can efficiently calculate the amount of haze in an image. Our method computes 100 times faster than the dark channel prior method. Our goal was to improve performance of the dark channel prior algorithm, which ultimately compromises the overall quality of the resulting dehazed image. Results on a variety of outdoor haze, smoke, and steam images demonstrate the power of the proposed prior.*

1. INTRODUCTION

Locations are the most tagged and classified photographs in the world [1]. It is assumed that all of those photographs are taken outdoors, since they are not of objects, artifacts, or people. To some degree each of the outdoor images is polluted with fog, smoke, haze, or smog. Fog is composed of small water droplets suspended in the atmosphere at or near the earth's surface. Due to this property, fog scatters light and makes air much more reflective, which reduces the amount of light reaching a camera.

As subjects become farther from a camera, both color saturation and color contrast drop dramatically, which produce unclear and ill-defined images as shown in the input image in Figure 1. These unclear hazy images degrade the performance of computer vision tasks such as feature detection, accurately computing the energy function of an image used for seam carving, etc.

Figure 1: Smoke removal using a single image. Top: input smoke image. Bottom: image after smoke removal by our approach.

Consequently, there has been much effort for haze removal (i.e. dehazing). To test at the source of the problem, photographers have tried to manually adjust their camera's parameters such as adjusting exposure to reduce the effects of fog. However, the results using this technique have been unsatisfactory, since there is a trade off of either increasing the amount of reflection or making the image dimmer than it originally is by increasing/decreasing the exposure.

Haze removal is a more a complex problem because haze is dependent on an unknown parameter, depth. Computational photography techniques produce far more visually pleasing results. Therefore, many proposed computational methods using multiple images, additional information, or the modern approach—single image input.

Earlier cases i.e. using multiple images and/or additional information successfully removed haze from an image, but using these applications are undesirable due to the additional information that is necessary to run them. Recent findings mostly focus on single image haze removal, which has made significant progress. These methods owe their success to the strong assumptions or priors that are made on haze and haze-free images.

For example, the most highly used method, and the method we use in our research, is the dark channel prior technique [2]. The prior assumes that dark pixels in a hazy image (not within the sky region) have very low intensities in at least one RGB color channel. This information is used to estimate depth based on the comparison between the hazy and clear images. The dark channel prior method has been proved to be a very powerful prior in single image dehazing. Nevertheless, it seems that dark channel prior system still has some limitations in that the dark channel prior becomes invalid when the scene color has low contrast to the haze color (e.g., a snow scene or a white wall). Moreover, the dark channel prior method has an extremely slow soft matting technique and takes about 15 minutes to run on a 400x400 pixel image.

2. MOTIVATION

The dark channel prior method has become a well-adopted algorithm to improve hazy images, and it has been used as the basis of numerous research projects. For example, the dark channel prior is used in a video application to recognize fog based on traffic scene in hazy weather [6]. There have also been extensions of the dark channel prior to improve film and video quality for under water photography [7,8]. There have also been efforts to enhance the current dark channel prior method. For instance, the dark channel prior method was improved through the use of contrast enhancement to enhance color contrast with less color distortion [9].

Still, there has been little effort to improve the computation time of the dark channel prior method. The soft matting function in the dark channel prior algorithm is computationally expensive and causes a bottleneck in the code. For this dehazing method to be a user-friendly application, we believe speed must be improved. We used a series of bilateral filters techniques [3]. Specifically, we design an optimization algorithm that balances between a system of three bilateral filters and the dark channel prior, so that the time to clean up hazy images is improved.

Experimental results show that the proposed method accurately finds areas that have low contrast to the sky region to determine what is hazy and what is not in an image. The produced images have undesirable artifacts; however, our goal was to improve the performance of the dark channel prior method not the quality of the produced image. The results are significantly faster than the conventional dark channel prior method, with speeds now running at about 12 seconds for an 800x600 pixel image. Lastly, methods before claimed that their methods worked on images that were polluted with smoke, fog, haze etc., but those methods never demonstrated experimental results using images other than hazy images. Our method used images that were taken both of fog/haze as well has images that were polluted with smoke and steam.

3. PROBLEM STATEMENT

Objectives
This project is comprised of several components and thus has a number of key objectives.

- Take as input any user-specified RGB source image that is polluted with haze.
- Accurately determine which areas are polluted with haze.
- Dehaze the image using the dark channel prior.
- Complete all computation in a reasonable amount of time (under 30 seconds for an 800x600 pixel image, if possible).

Sub-problems
These broad objectives break down naturally into several sub-problems:

- How can the dehazing method be done as efficiently as possible?
- What kind of bilateral filters do we need to achieve our goal time?

4. RELATED WORK

The model widely used to describe the formation of a haze image is a simple set of equations that calculates the intensity *I* of an image,

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

where *J* is the scene radiance, *A* is the global atmospheric light, and *t* is the medium transmission describing the portion of the light that is not scattered and reaches the camera. The goal of haze removal is to recover *J, A*, and *t* from *I* [4]. In our algorithm we use the dark channel prior to estimate *J(x)t(x)* and bilateral filtering to estimate *A(1-t(x))*.

4.1 Dark Channel Prior

The dark channel prior uses the assumption that within most local non-sky regions there are dark pixels, which have very low intensity in at least one RGB channel [2]. The dark channel for an arbitrary image *J* is given by

$$J^{dark}(\mathbf{x}) = \min_{c \in \{r,g,b\}} \left(\min_{\mathbf{y} \in \Omega(\mathbf{x})} (J^c(\mathbf{y})) \right), \tag{2}$$

where, J^{dark} is the dark channel of image *J*, J^c is a color channel of *J*, and *Ω(x)* is the patch centered at *x*. The dark channel is the outcome of the two minimum operators: $min_{c€(R,G,B)}$ is performed on each pixel and $min_{y€\Omega(x)}$ is a minimum filter. The dark channel prior observation is that if an image *J* is haze-free, then the intensity of *J's* dark channel is equal to zero.

He, Sun and Tang [2] improved atmospheric light estimation using the dark channel prior. They picked the top 0.1% brightest pixels in the dark channel because these pixels are the most haze opaque. Then pixels with the highest intensity in the input image *I* within this group of haze opaque pixels are chosen as the atmospheric light.

The last two steps of the dark channel prior method, before producing the final image, are estimating the transmission and a soft matting technique. We chose to omit the soft matting method due to is slow computation time, and replace it with a series of bilateral filters. Estimating transmission, as a result, needed to be estimated using the new value for *A*.

4.2 Guided Bilateral Filters and Regular Bilateral Filters

A series of bilateral filters were used to complete our algorithm. Specifically, a three-filter system was used: median filter, bilateral filter and joint bilateral filter. The following algorithm was used from [3] to implement this three-step process.

(1) Calculate the min components *W(x)* of *I(x)* and estimate the atmospheric light A;
(2) Yielding initial atmospheric mask *V(x)* using median filtering (Eq. 5);
(3) Filtering *W(x)* using bilateral filter to get the reference image *R(x)* (Eq. 6);
(4) Filtering *V(x)* taking *R(x)* as the reference image to receive corrected atmospheric scattering light $V_R(X)$ (Eq. 7);
(5) Calculate the transmission *t(x)* (Eq. 3) (same as dark channel prior step 3);
(6) Get the recovered image *J(x).*

(3)

Several equations were used to create this algorithm, which are listed below. The key result of taking a series of bilateral filters is to quickly compute an accurate atmosphere mask that is smoother.

4.2.1 Medium Filter:

$$\begin{aligned}
&(1)\ W(x) = min_{c\in\{r,g,b\}}(I^c(x)) \\
&(2)\ B(x) = median_{\Omega}(W(x)) \\
&(3)\ C(x) = B(x) - median_{\Omega}(|W - B|)(x) \\
&(4)\ V(x) = max(min(pC(x), W(x)), 0)
\end{aligned} \tag{4}$$

The goal here is to calculate the atmospheric scattering light *V(x)* where *Ω* is a square window of median filter. First filter *W(x)* using a median filter to receive *B(x)*. Next, to alleviate the affect of contrasted texture for the haze removal, apply the difference of the local mean *B(x)* and local standard deviation of *W(x)*. Finally, multiply *C(x)* by a scale factor $p \in [0, 1]$ to control the strength of visibility restoration.

4.2.2 Bilateral Filter:

$$R(x) = \frac{\sum_{y\in\Omega(x)} f(x-y)\cdot g(W(x)-W(y))\cdot W(y)}{\sum_{y\in\Omega(x)} f(x-y)\cdot g(W(x)-W(y))} \tag{5}$$

The filter is applied to *W(x)* to filter out some texture details, while the edge features can be well preserved.

4.2.3 Guided joint filter:

$$V_R(x) = \frac{1}{k}\sum_{y\in\Omega(x)} f(\|y\|)\cdot g(\|R(y)\|) \cdot h(V(y)-R(y))\cdot V(y) \tag{6}$$

k is the normalizing factor, the range filter kernel $g(\|R(y)\|) = e^{-(R(x)-R(y))2/2\sigma r2}$. The term $g(R(x) - R(y))$ and $h(V(y) - R(y))$ work together to preserve the edge information and remove the useless texture information of the image *R(x),* which produces a more accurate atmospheric mask $V_R(x)$.

4.2.4 Estimating Transmission

Finally, to estimate the transmission, the portion of the light that is not scatter and reaches the camera, Xiao and Gan [3] assumed that the atmospheric light *A* is given and the transmission in a local patch ω is constant. The transmission *t* is estimated through the following:

$$t(x) = 1 - \omega V(x)/A \tag{7}$$

where, $\omega \in (0,1]$ is used to preserve little haze in the distant scene and makes the recovered image more natural. Moreover, it provides the estimation of the transmission.

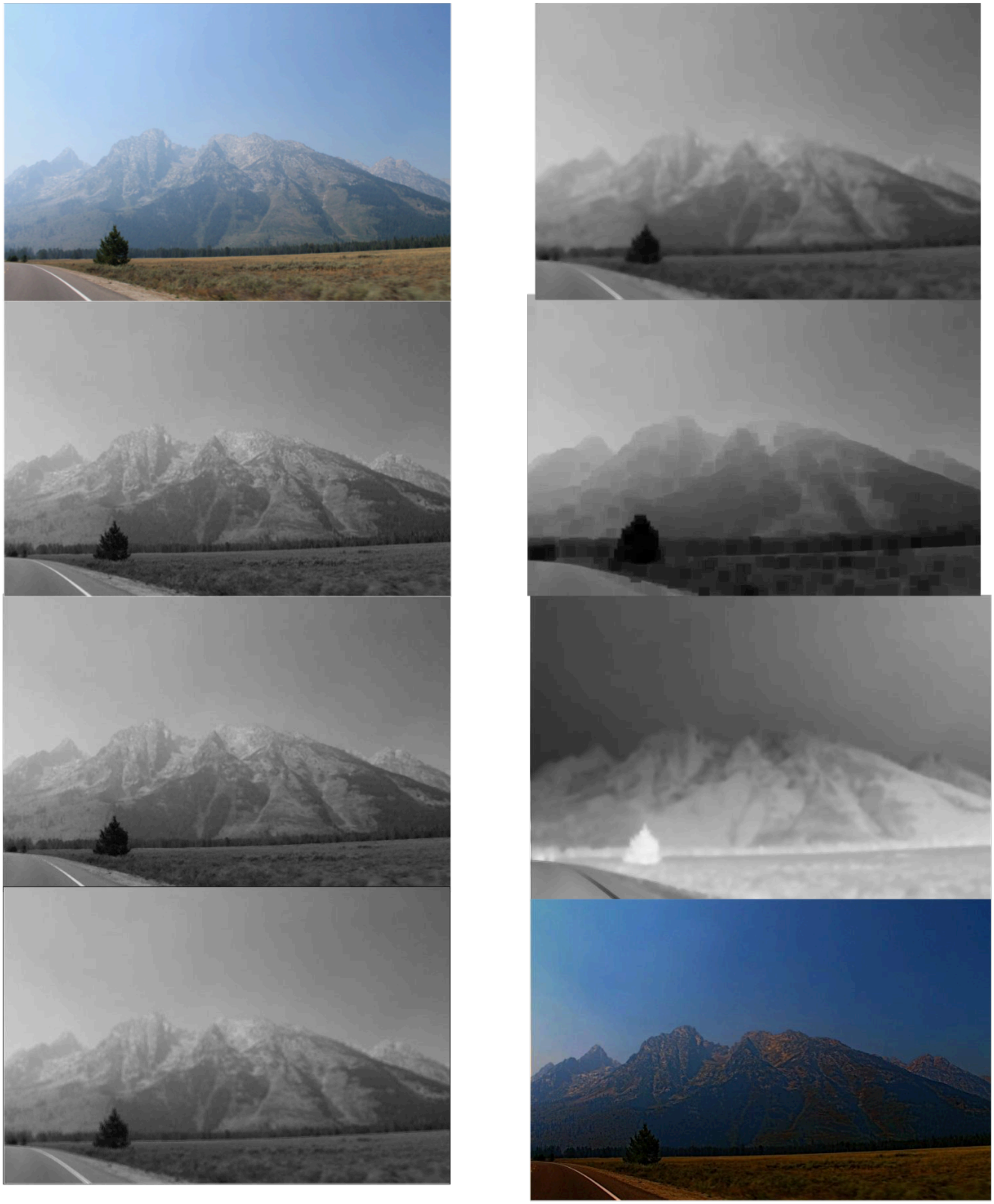

Figure 2: From top to bottom in left column: input smoke image, minimum color channel, custom median filter, and custom bilateral filter. From top to bottom in right column: guided joint bilateral filter, *Jdark*, transmission, and radiance (image after smoke removal by our approach).

5. METHOD AND THEORY

To conduct our experiments we enhanced a dark channel prior Matlab code built by Microsoft Research group in Asia [2]. We also used Matlab code from [4] to apply the guided joint filter in the bilateral filtering algorithm. We decided to continue running the code in Matlab because it tends to be a robust application for computational photography.

We used equation (1) from section 4.1 to execute the dehazing method as a whole, and algorithm (3) to complete the three-step bilateral filtering process. The following breaks down the steps with their corresponding functions that were necessary to develop haze free images.

The test images we used in our experiments were images we took in the field prior to the project. Images are either polluted with haze, steam, or smoke. The input image used in this section was polluted with smoke; however, the results would be similar for a hazy or steamy image. We ran our results on a Mac I5 4GB RAM using the Matlab launcher.

5.1 Bilateral Filtering Functions

First, we applied a series of bilateral filters and regular bilateral filters [3].

5.1.0 Minimum Color Channel

W(x) was calculated using Matlab's minimumChannel function, and it is displayed in Figure 2. The function calculates what is scene content and what has high intensity i.e. smoke. The output is a black and white mapped image that highlights areas of low intensity in white and high intensity in black, as seen in Figure 2.

The dark areas show regions of the original image that were either highly saturated, colorful, or had dark shadows. This is especially prominent for the tree and the shadows on the mountain landscape. The light areas show where the most smoke is found, which is located mainly in the sky region and in the mountain peaks. Peaks that are farther in the background are more polluted with smoke compared to mountains in the foreground because there is more smoke polluted air that light must travel through before reaching the camera. (Note: this same color convention of dark and light pixels will be used for the remaining portion of section 5.1)

5.1.1 Custom Median Filtering

We used steps 2-4 in equation (4) to calculate an initial atmosphere scattering light model, which is obtained through median filtering. This step of the filtering process helps with preserving texture of the background. Input to the function is the minimum color channel *W(x)*. The assumption that was made was a 3x3 median filter using was used to calculate the final result. Output to the function is a black and white image.

Figure 2 shows the output image of the atmospheric vial. Again, the darkest parts of the image are in the foreground, because there is less atmosphere between the object and the camera. The sky has the most haze. The output of the median filter function is doing best to preserve texture in the background. You can see this feature as the mountain background has maintained fine details, verses the tree in the foreground appears blurry.

5.1.2 Custom Bilateral Filtering

Next, we used equation (5) to filter *W(x)* using a bilateral filter to get the reference image *R(x)*. The minimum color channel *W(x)* is also the input for this step. It is refined to generate a new atmosphere mask, which preserves texture in the foreground. Assumption used was a predefined 3x3 patch with distance and intensity filters. Again, the output image to the function is a black and white image illustrating the intensity.

The same conventions apply as far as what is black and white in the image as they were in the median filtering output image. What is different for this filtering technique is that the foreground texture is preserved instead of the background. The image in Figure 2 presents that both the background and foreground are blurry, but the foreground has more detail compared to the background. The tree and the white line on the road are much clearer than the very blurred mountain backdrop.

5.3.3 Guided Joint Bilateral Filter

Finally, we used a De Silva's Matlab function jbfilter2 [4] to solve for the correct atmospheric scattering light, $V_R(X)$. Input to the function is the output images from both the median filtering function and the bilateral filtering function. The assumptions that were used was the half-size of the Gaussian bilateral filter window is size 6, and the standard deviation of the bilateral filter are given by SIGMA = 1000. Details for both input images are used to produce the final image, which is another black and white image demonstrating intensity.

Black and white areas are similar to both input filtering images. What this function does is preserves some detail of both the bilateral and median filter. This can be seen in Figure 2 where both the background and foreground have visible texture.

5.2 Jdark Function

The *Jdark* equation (2) was run after the filtering in our algorithm. The input to the function is minimum color channel $W(x)$. The main goal of the *Jdark* function is to use its output for estimating the atmospheric light A. A 30x30 pixel patch size was used in the equation for an 800x600 pixel image. The output is a black and white mapped image that highlights areas of high intensity in white and low intensity in black, as seen in Figure 2.

Like in the bilateral filtering images, the dark areas show regions of the original image that were either highly saturated, colorful, or had dark shadows. Once more, the tree and the prairie landscape in the foreground are the darkest. The light areas are located mainly in the sky region and in the mountain peaks. The image is very pixilated due to the 30x30 pixel patch size we used.

5.3 Atmosphere Estimation Function

Next, we estimate the atmospheric light using *Jdark* as a heuristic. The use of *Jdark* helps because it measures the levels of whiteness due to smoke and what areas are actually white i.e. a white line on the road or snow. We selected the top 0.1%-0.2% of the brightest pixels in the dark channel. This was done to reduce errors for white objects. Input to the atmospheric function is the output image from the *Jdark* function and the raw input image. The atmospheric function returns 1x3 pixel image with the average atmosphere at each RGB channel.

5.4 Transmission Estimation Function

Fourth, we estimated the transmission of the image using equation (3). Input to the transmission function is the following parameters: the guided joint bilateral output image and the highest value of A from the 1x3 average atmosphere pixel image. A constant value $\Omega(x)$ = .95, which represents the percent of haze kept was input into the function to solve for the transmission. Output to the function is a transmission image and can be seen in Figure 2.

The transmission image is a minimal operation on a channel basis, and gives better results for the sky region. It is essentially an inverse of the minimum channel image *W(x)*. High intensity is in white, which are pixels we want to keep. Therefore, we want to keep the tree, base of the mountain, and the prairie flat land. Low intensity is seen in black, which are areas of haze and what we want to remove. The sky is the darkest, which has the most haze, which was determined in the preceding images. We also want to rid haze on the mountain peaks. The output of this function is used in the recovered radiance step, which finally removes the haze.

5.5 Radiance without Refinement

The last step of our algorithm is to produce the radiance image, which is the final result image of the dehazing method. Input to the function is the atmospheric estimation output, the raw input image, and finally output from the transmission function. As you can see the image successfully removes haze. The sky shows the best demonstration of haze removal, with beautiful color restoration and color saturation. The rest of the image; however, is overly saturated in the shadowed areas and already saturated areas of the original image.

6. EXPERIMENTAL RESULTS

Figure 3 shows our haze removal results. The image recovers depth in the image and accurately removes haze. However, color is over saturated and color contrast is high. Likewise, there are artifacts in the sky region. Our focus for this paper was on reducing the time it took to run the

Figure 3. Haze removal using a single image. Top: input haze image. Bottom: image after haze removal by our approach.

experiment. Therefore, we were happy with the results because there was successful haze removal with a fast computation time.

Figures 1 and Figure 4 also demonstrate successful smoke and steam removal. Smoke produces more of a gray hazy appearance verses the white in fog or haze. This is because smoke is polluted with dark ash particles. Even though there is a difference in the air pollution color, our results prove that the method works for any type of air quality.

Next, Figure 5 shows a comparison of our results to those of the dark channel prior method [2]. Advantages that our code has over the dark channel prior is that it could run on larger images, and produced results more efficiently, see discussion below for more detail. Also, there are not as many artifacts in the output image. Finally, the blending between the sky and mountain is improved. A disadvantage of our method compared to the dark channel prior method is that our method does not remove as much haze as the other method. Second, there are still some artifacts in the haze-region that is likely due to the large level of haze present in the image. A final disadvantage seen between the two images is that there is the color restoration is off. The entire image appears darker than it actually is and more saturated.

He, Sun, and Tang's [2] method has a relatively high computation time. It takes roughly 10-15 minutes to process a 400x400 pixel image on a Mac I5 4GB RAM using the Matlab launcher. The highest time cost is from the soft-matting function. The computational time of our method is significantly shorter with only 12 seconds to process a larger 800x600 pixel image. Therefore, our method computes 100 times faster than the leading haze removal method.

Figure 4. Steam removal using a single image. Top: input steam image. Bottom: image after haze removal by our approach.

Figure 5. Haze removal using a single image. Top: input haze image. Center: image after haze removal by dark channel prior approach. Bottom: image after haze removal by our approach.

7. CONCLUSION

In this paper we have proposed a simple enhancement to the dark channel prior, a series of bilateral filters, for single image haze removal. The algorithm we proposed primarily considers efficiency when dehazing hazy or smoky images. This was done by replacing the soft matting code with the more effective three-filter system code, resulting in quickly determining the atmosphere vial. Combining the bilateral filtering method with the dark channel prior method, single image haze removal becomes faster.

As seen in the experimental results, our proposed approach successfully removes haze that is 100 times faster than the dark channel prior. The time to run the soft matting method is exponential to the size of the image. Now, with bilateral filtering time increases linearly as image size increases, which is ideal. Our test images include a variety of weather conditions including haze/fog, smoke, and steam. Prior methods used only hazy images yet claimed that their methods would work for other atmospheric conditions, and our results actually proves that the algorithm works for those atmospheric conditions such as smoke and steam.

Our work, however, does have some limitations. Replacing soft matting with a series of bilateral filtering causes, in some respects, failure to restore the original scene radiance and color. There are undesirable artifacts especially in the sky region, which were produced from the bilateral filtering method. Also, color contrast and saturation were high in comparison to the original image. In future works, a better estimation of the atmospheric content could be made to improve this quality. To focus on perfection, we would need to refer back to increasing the soft matting speed and improve our atmospheric particle detection.

REFERENCES

[1] B. Sigurbjornsson, and R. van Zwol. Flickr Tag Recommendation based on Collective Knowledge. Refereed Track: Rich Media, pages 327-336, 2008.

[2] K. He, J. Sun, X. Tang, The Chinese University of Hong Kong, and Microsoft Research Asia. Single Image Haze Removal using Dark Channel Prior. IEEE Transaction on Pattern Analysis and Machine Intelligence, 33: 2341-2353, 2011.

[3] C. Xiao and J. Gan. Fast image dehazing using guided joint bilateral filter. Vis Comput, 28: 713-721, 2012.

[4] V. De Silva. Joint Bilateral Filter. Matlab Central, 2010.

[5] R. Fattal. Single Image Dehazing. In *SIGGRAPH*, pages 1-9, 2008.

[6] X. Ji, J. Cheng, J. Bai, T. Zhang, and M. Wang. Real-time enhancement of the image clarity for traffic video monitoring systems in haze. Image and Signal Processing (CISP), 7th International Congress on, pages 11-15, 2014.

[7] Z. Wang, B. Zheng, and W. Tian. New approach for underwater image denoise combining inhomogeneous illumination and dark channel prior. Oceans - San Diego, pages 1-4, 2013.

[8] J. Y. Chiang and Y. C. Chen. Underwater Image Enhancement by Wavelength Compensation and Dehazing. Image Processing, IEEE Transactions on, 21: 1756-1769, 2012.

[9] T. Kil, S. Lee, and N. Cho. A Dehazing Algorithm using Dark Channel Prior and Contrast Enhancement. Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on, pages 2484-2487, 2013.

Code Used and Work Breakdown

We used code from He, Sun, Tang [2] to implement the *Jdark* method and atmospheric estimation in order to complete *J(x)t(x)* in the equation

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

We also used code from De Silva [4] to replace run the guided joint bilateral filtering portion of the bilateral filtering algorithm to complete A(1-t(x)). Both sets of code were ran as is with a few changes to their parameters. For example, *Jdark* is same except experiment patch size to decide how detailed we wanted Jdark to be was adjusted. Atmosphere function stayed same, but we experimented with area of testing. We also tried own function call atm1D, which produced same results so wasn't effective.

The remaining parts of the code e.g. median filtering, bilateral filtering, and transmission function were coded by our group members. We utilized Matlab functions to complete the median filtering function. The transmission function and the bilateral function were calculated using the functions in [2] and [3] and were all original code. We wrote about 150-200 lines of code in MATLAB. Other code that is included in our submission is code that we found online and uses strictly for performance comparisons to our code. It is included to provide references on relative image processing amounts of time, and how were able to decrease total computation time greatly.

We broke the work between the team members evenly. Chad, Jake, and Rachel each wrote parts of the code, with most of the work being done together. Chad wrote the median filtering function and atmospheric function. Jake wrote the bilateral filtering method and found the De Silva [4]. Rachel wrote the color transmission function and the *Jdark* function. Simultaneously while coding we wrote the final report. Rachel Yuen wrote the introduction, motivation, problem statement, and related works sections. Both Jake and Chad contributed to writing the theory, method, experimental results, and conclusion. Rachel also wrote the abstract, Chad wrote the midpoint review check, and everyone contributed to the power point presentation. Finally, Jake created the webpage for our project to be hosted on. Our group was very cooperative and productive, and we enjoyed working on the project as a team.