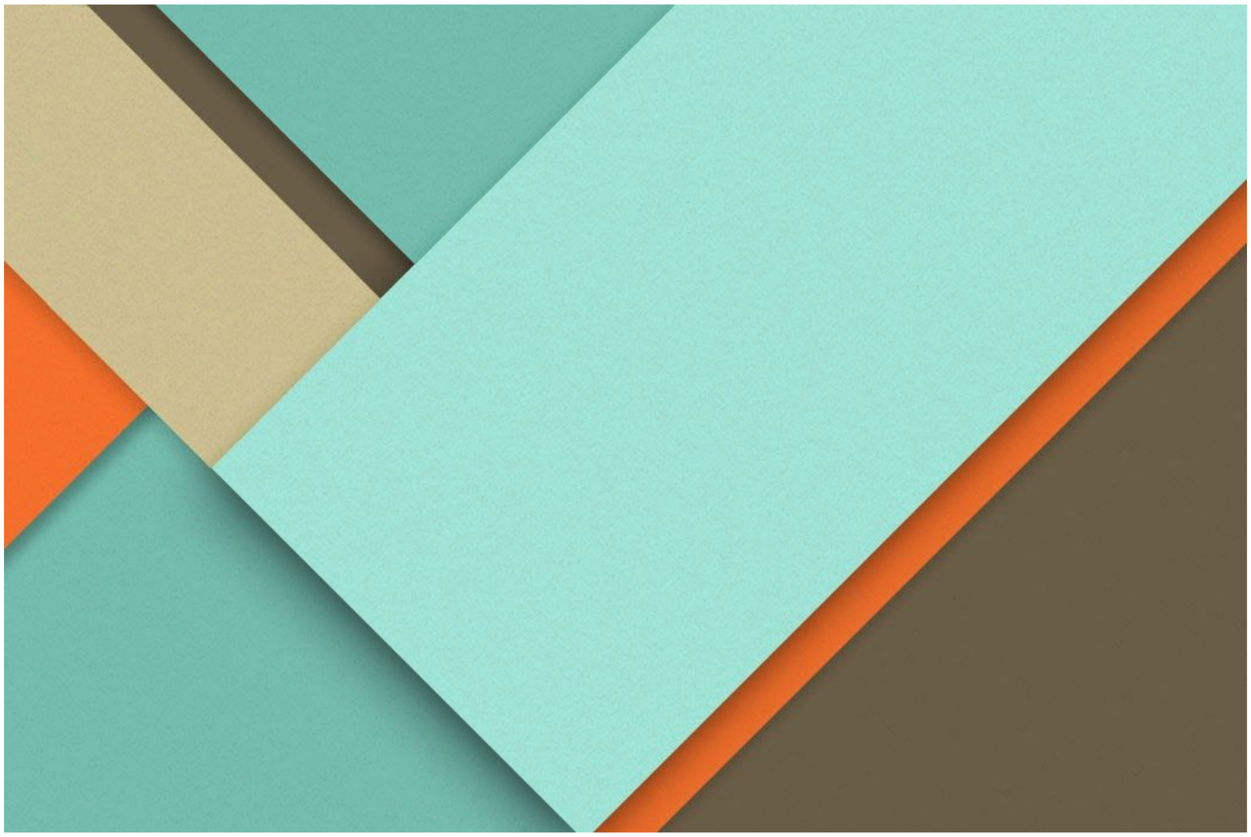


Optimized Image Dehazing

Using Convex Optimization for Fast Image Dehazing

Under the Guidance of Dr. Kapil Ahuja

Jhalak Gupta
170001024
3rd year CSE

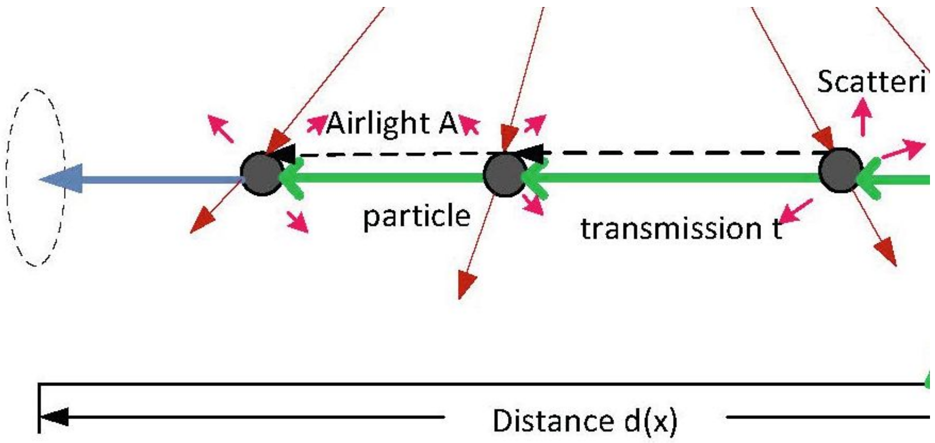
Ashwini Jha
170001012
3rd year CSE

Introduction:

Winter mist and pollution are major reasons for foggy and hazy images. Images taken outdoors often become hazy and vague due to light absorption and light scattering by suspended particles in the atmosphere. Hence, the images under hazy conditions usually lack vivid color and have low contrast, which seriously affects the performance of the photography applications and computer vision. Therefore, restoration of hazy images and recovery of de-hazed and clear images is essential and has caught increased attention in the last few years. Image dehazing is a process of improving and enhancing the images by increasing the image contrast and removing haze from hazy images as per required amount, which is a challenging problem when the depth-of-field information is unknown.

The problem of dehazing images is to process hazy and foggy images to remove the haze effects and recover the original image scene, i.e. image that should have been captured without fog or haze. The image dehazing is a nonconvex problem, as shown by the optical image model, because of the bilinearly coupled haze free image and atmosphere light transmission distribution.

The Hazed Image Model:



The object's direct transmission light is reduced along its original course and is distributed to other directions when passing through the hazy medium. In addition, veiling light scattered from all other directions is added. Thus, the intensity of light I for each pixel x, that reaches the observer, can be formulated as below:

$$I(x) = D(x) + V(x).$$

Where D(x) is the scene radiance and its decay in the medium and V (x) is the veiling light that dominated by the scattering of environmental light towards the camera. D(x) is modelled as:

$$D(x) = J(x)t(x)$$

where, J(x) is the scene radiance for each pixel x, namely the clear image, and t is called the transmission function and expresses the relative portion that has been transmitted between the scene and the camera. In a homogenous atmosphere, the transmission t expressed as:

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

where, β is the positive scattering coefficient depending on the size of the scattering particles and d(x) is the distance from the scene to the camera for each pixel x. Since β and d(x) are positive, we have $t(x) \in (0\ 1]$. Besides, the veiling light V (x) is the main cause of the color shift, which is:

$$V(x) = A(1 - t(x))$$

with $A = [\ A_r\ ,\ A_g\ ,\ A_b\ ]^T$ represents the light in the red, green, blue colour channels. In addition, the atmospheric light A is weighted by $(1 - t(x))$. Combining the equations, we have:

$$I(x) = J(x)t(x) + A(1 - t(x))$$

Image Dehazing:

$$C_{MSE} = \sum_{c\in\{r,g,b\}} \sum_{p=1}^{N} \frac{(J_c(p) - \bar{J}_c)^2}{N} = \sum_{c\in\{r,g,b\}} \sum_{p=1}^{N} \frac{(I_c(p) - \bar{I}_c)^2}{t^2 N}$$

Here C_{MSE} represents the image's Mean Square Error (MSE) contrast. This implies C_{MSE} is inversely proportional to the square of the transmission *t*, so we can select a small value of t to increase the contrast. Therefore, we bring l_1 norm of t into our cost function.

Mathematically we need to solve the following optimization problem:

$$\min_t \lambda||t||_1 + ||t||_{TV}, \quad \text{subject to} \begin{cases} Y(x) = At(x) - q(x) \\ 1 \geq t(x) > 0 \\ J(x) \geq q(x) \geq 0 \end{cases}$$

where

$$q(x) = J(x)t(x)$$

$$Y(x) = A - I(x)$$

$$||t||_{TV} = |\nabla_x t| + |\nabla_y t|$$

Now we need to convert this equation into unconstrained optimization problem.

$$\min_{t,J,q} \quad \lambda|t| + |\nabla_x t| + |\nabla_y t| + \frac{\mu}{2}||Y - At + q||_2^2 + \frac{\epsilon}{2}||q - Jt||_2^2$$

Algorithm:

The basic algorithm for solving the unconstrained optimization problem is given below.

Initialize: $t^0 = w^0 \Leftarrow 1$, $q^0 = J^0 \Leftarrow I$, $d_x^0 = d_y^0 = b_x^0 = b_y^0 \Leftarrow 0$, $\lambda \Leftarrow 0.08$, $\mu = \epsilon = \gamma = \eta \Leftarrow 1$.
While $||t^k - t^{k-1}||_2 > tol$:
1.Update t^{k+1}, q^{k+1} and J^{k+1} using equations (14) (15) and (16),respectively.
2.Update $d_x^{k+1}, d_y^{k+1}, w^{k+1}$:

$$d_x^{k+1} = \text{shrink}(\nabla_x t^{k+1} + b_x^k, \frac{1}{\gamma})$$

$$d_y^{k+1} = \text{shrink}(\nabla_y t^{k+1} + b_y^k, \frac{1}{\gamma})$$

$$w^{k+1} = \text{shrink}(t^{k+1}, \frac{\lambda}{\eta}),$$

with

$$\text{shrink}(x, \gamma) = \frac{x}{|x|} \times \max(|x| - \gamma, 0).$$

3.Update b_x^k, b_y^k:

$$b_x^{k+1} = b_x^k + (\nabla_x t^{k+1} - d_x^{k+1}), \quad b_y^{k+1} = b_y^k + (\nabla_y t^{k+1} - d_y^{k+1}).$$

Implementation:

The basic code of our implementation part is as follows:

```
clear all;
addpath('tv');    %% adding files from the folder “tv”

tic  %% reading start time
I=im2double(imread('Images/y01_photo.png'));
A=air(I,8);

Y=A-I;
[m,n,l]=size(I);

%% initialize
t=ones(m,n,l);
q=I;

tt=ones(m,n); % the two dimension transmission map
dx=zeros(m,n); dy=zeros(m,n); bx=zeros(m,n); by=zeros(m,n);

alpha=0.6; %alpha is the aL2
lambda=1; %lambda is the aTV
mu=1; ref=1.2;
gamma=lambda*ref; %gamma=aTV*ref=lambda*ref, aTV is the lambda
tol=1;

%% main loop
k = 0;
while tol>1e-3
tb=t;
for c=1:3
t_est(:,:,c)=gamma/(2*alpha+mu+4*gamma)*(SA(t(:,:,c))+gamma
*Dxt(dx-bx)+gamma*Dyt(dy-by))+mu/(2*alpha+mu+4*gamma)*(Y(:,:,c)+q(:,:,c))./
A(:,:,c);
end

t_est = max(min(max(max(t_est,eps),1-I./A),1-eps),-Y./(1-A));
tt = max(t_est,[],3);
t=cat(3, tt, tt, tt);
```

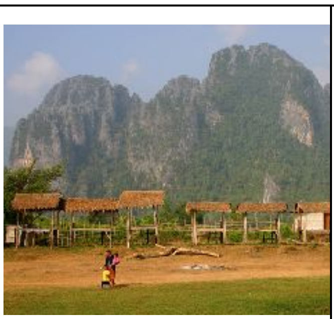
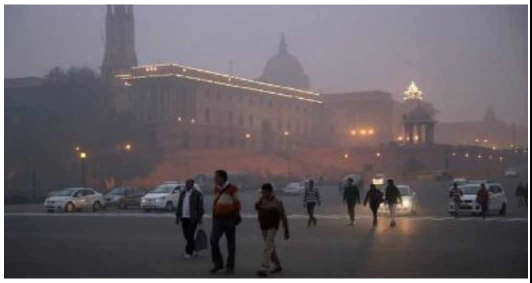
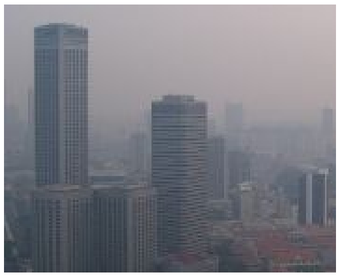
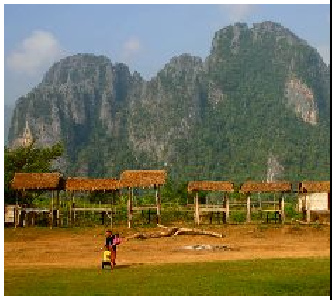
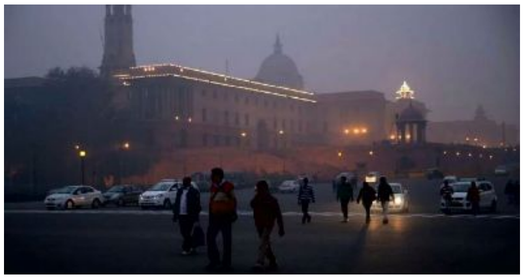
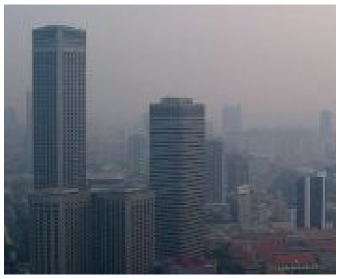
```
q=A.*t-Y;
for c=1:3
dx=shrink(Dx(t(:,:,c))+bx,1/ref); %1/ref=lambda/gamma
dy=shrink(Dy(t(:,:,c))+by,1/ref);
bx=bx+(Dx(t(:,:,c))-dx);
by=by+(Dy(t(:,:,c))-dy);
end
mu=mu*10;
tol=(norm(t(:,:,1)-tb(:,:,1),
1)+norm(t(:,:,2)-tb(:,:,2),1)+norm(t(:,:,3)-tb(:,:,3),1));
% displaying tolerance
%disp(tol);
k = k+1;
%% displaying number of iterations
disp(k);
%J=max(q./t,0);
%toc
%figure, imshow(I), title('hazy image');% the hazy image
%figure, imshow(J), title('clear image');% the reconstructed image
%imwrite(J, 'CO.jpg');
end

J=max(q./t,0);

toc  %% reading and printing execution time till here

%% output the original image (hazed)
figure, imshow(I), title('Original image');
%% output the dehaze image (result)
figure, imshow(J), title('Result image');

imwrite(J, 'result.jpg');
```

Results:

The following table compares the execution time in seconds of some popular image dehazing algorithms and our proposed convex optimization (CO) based algorithm for different sized images.

Algorithm \ SIze	**600*450**	**1024*768**	**1536*1024**
He et al.	10.04	29.46	65.4
Tarel et al.	8.81	68.90	335.79
Meng et al.	4.782	5.60	10.27
Zhu et al.	3.68	4.08	8.08
Convex Optimized	2.25	6.85	12.9

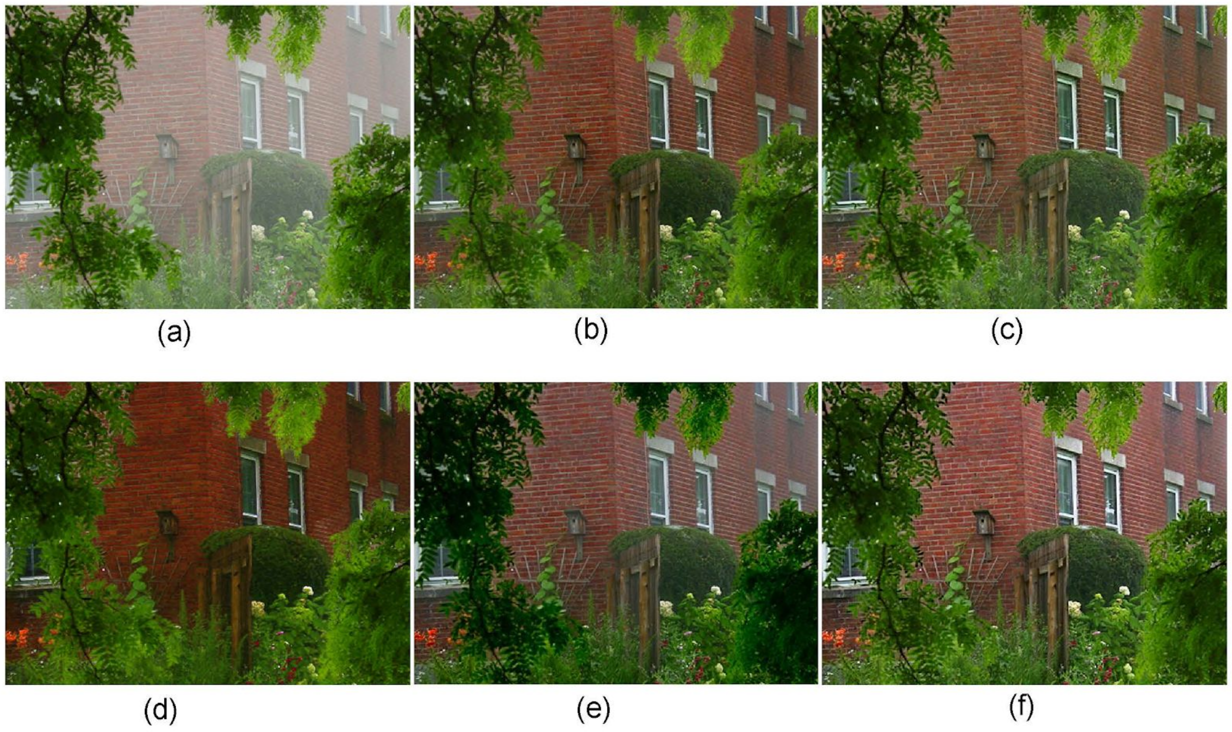
The following are the output images after each iteration using the MATLAB code shared above. The table also includes the execution time for each iteration.

 Original image Time: 0 sec	 Original Image Time: 0 sec	 Original Image Time: 0 sec
 Iteration 1 Time: 0.38 sec	 Iteration 1 Time: 0.72 sec	 Iteration 1 Time: 0.80 sec

Iteration 2 Time: 0.71 sec	Iteration 2 Time: 1.24 sec	Iteration 2 Time: 1.22 sec
Iteration 3 Time: 0.88 sec	Iteration 3 Time: 1.56 sec	Iteration 3 Time: 1.49 sec
Iteration 4 Time: 1.05 sec	Iteration 4 Time: 1.90 sec	Iteration 4 Time: 1.75 sec
Iteration 5 Time: 1.19 sec	Iteration 5 Time: 2.21 sec	Iteration 5 Time: 1.98 sec

Dehazed Image Total iterations: 7 Time: 1.37 sec	Dehazed Image Total iterations: 8 Time: 3.08 sec	Dehazed Image Total Iterations: 8 Time: 2.68 sec

Below are the outputs of different algorithms, which shows our algorithm works pretty good:



Here (a) is the original image, while (b)-(e) are the output images based on other algorithms proposed for image dehazing. (f) is the result of our algorithm using convex optimization.

Conclusion:

The proposed algorithm dehazing results outperform state of the art image dehazing algorithms in the better visual quality of the dehazed images and much faster processing speed.

Also, with more optimization of the algorithm, we may use the technique for dehazing and clearing the foggy and hazy videos.

References:

1. https://ieeexplore.ieee.org/document/7532758 (Convex optimization for fast image dehazing : Jiaxi He, Cishen Zhang, Ran Yang, Kai Zhu)
2. https://link.springer.com/chapter/10.1007/978-3-319-14249-4_9 (Image Dehazing Using Regularized Optimization : Jiaxi He, Cishen Zhang, and Ifat-Al Baqee)
3. https://epubs.siam.org/doi/abs/10.1137/080725891 (The Split Bregman Method for L1-Regularized Problems)
4. https://en.wikipedia.org/wiki/Haze_(optics)