

**DEPT. OF CEN**

**AMRITA SCHOOL OF AI**

**B. TECH**

**CSE-AI**

**SEMESTER-5**

**21AIE303**

**SIGNAL AND IMAGE PROCESSING**

**END SEMESTER PROJECT**

**TEAM MEMEBERS:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **NAME** | **ROLL.NO** |
| **1.** | **MARREDDY MOHIT SASANK REDDY** | **CB.EN.U4AIE21031** |
| **2.** | **YARRAM SRI SATHWIK REDDY** | **CB.EN.U4AIE21077** |
| **3.** | **P.SAI VIJAY KUMAR** | **CB.EN.U4AIE21040** |
| **4.** | **B. HARISH BALAJI** | **CB.EN.U4AIE21007** |

**Abstract**

This study explores the effectiveness of two dehazing methods—DCP Morphology and CAP HLP—alongside the integration of Contrast Limited Adaptive Histogram Equalization (CLAHE) for image enhancement. DCP Morphology utilizes mathematical morphology for haze reduction, while CAP HLP employs haze-line and colour attenuation priors. The addition of CLAHE aims to further improve local contrast. Metric analysis, including PSNR, SSIM, and VIF, is conducted to quantitatively assess the performance of these techniques. Results from diverse hazy images provide insights into the strengths and limitations of each method, offering practical guidance for image dehazing in various applications.

**Problem Statement:**

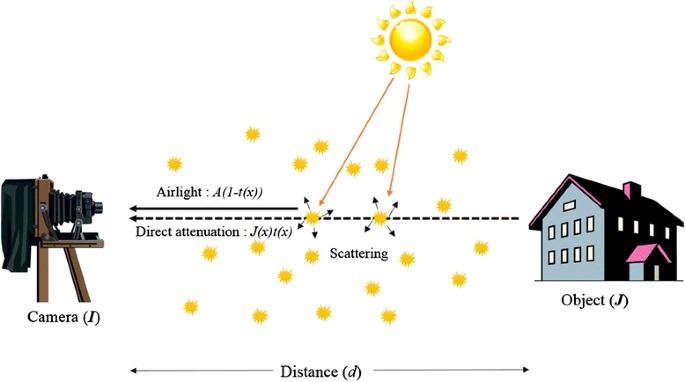
Addressing the challenge of hazy image degradation in computer vision applications, the study aims to assess the comparative effectiveness of DCP Morphology and CAP HLP dehazing techniques. The investigation includes the integration of Contrast Limited Adaptive Histogram Equalization (CLAHE) to enhance the dehazing process.

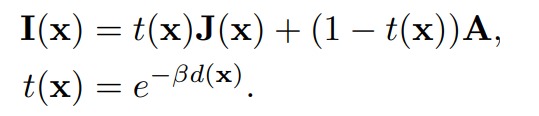
**Goal:**

The research goal is to quantitatively evaluate the performance of DCP Morphology and CAP HLP, individually and in combination with CLAHE, in improving image visibility and quality under hazy conditions. The findings aim to guide the selection and optimization of dehazing methods for enhanced computer vision in challenging environmental scenarios.

**Introduction:**

The observed hazy image, I(x) ∈R N×3, is a convex linear combination of the haze-free scene radiance, J(x), and the atmospheric light component, A, called the air light; usually represented as a constant 3-vector in RGB space, A =(Ar, Ag, Ab).The transmission map coefficients, t(x) ∈ RN control the relative force of each component, in each pixel in the image, x ∈ RN . The transmission is a function of the depth, d(x), of the scene from the observer. Our goal in single-image-dehazing is to obtain the haze-free scene radiance, J(x). To do so, however, one needs to solve a set of 3N equations (only I(x) is given), with 4N + 3 unknowns (J(x), t(x), A). Thus, additional prior knowledge of the images in question is needed.





**Methodology:**

**Dark Channel Prior:**

This algorithm has three main objectives:

* Estimate atmospheric light
* Find Transmission map
* Smoothen Transmission map using regularization

The dark channel prior is a method used in image dehazing to estimate the atmospheric transmission map, which is used to remove haze from an image. The basic idea behind the dark channel prior is that, in most natural images, there is a small patch of pixels that have very low intensity values in at least one color channel. This patch of pixels is called the dark channel, and it is used to estimate the atmospheric transmission map

I(x) = t(x)J(x) + (1 − t(x))A

This is the equation represents the image captured where I(x) is Image captured J(x) is the actual scene radiance without haze A is atmospheric light and t(x) is transmission

The atmospheric light is a scalar value representing the amount of light that is scattered by the atmosphere and reaches the camera. In the dark channel prior algorithm, atmospheric light is calculated from the highest intensity value of the pixels in the dark channel. Once the dark channel of an image is calculated, where each pixel represents the minimum intensity value of a patch centered on that pixel, the atmospheric light is calculated by taking the highest intensity value of the pixels in the dark channel.

In the dark channel prior (DCP) method, the transmission map is estimated by first calculating the dark channel of the image. The dark channel is a 2D array, where each pixel represents the minimum intensity value of a patch centered on that pixel. The patch size is typically 3x3, 5x5, or 7x7 pixels. With the atmospheric light, the transmission map can be calculated using the formula:

t(x) = 1 - min(I(x))/A

Where x is a pixel in the image, I(x) is the intensity of the pixel and A is the atmospheric light Later we perform regularization and use the equation below to find the scene without haze

J(x) = I(x) − A/( [max (t(x), µ)]^δ) + A

Where µ is constant 0.001 to avoid division by zero

J(x) = I(x) − A [max (t(x), µ )]^δ + A

**Dark Channel Prior using Morphology:**

This algorithm has three main objectives:

* Estimate atmospheric light
* Find the transmission map
* Refine the transmission map

Atmospheric light and the transmission map are computed using concepts similar to Dark Channel Prior. But to refine the transmission map, we used the concepts of morphology.

**Refining the transmission map:**

* Perform closing operation on initial transmission map and reconstruct the image. This operation removes the small dark elements from the image.
* Perform opening reconstruction on the transmission map. This operation removes small objects which are smaller than structuring element. On doing the this some small useful data might be lost in order to save it we store the removed objects.
* Recover ranges from original image and add the removed small objects in order to get the refined transmission map
* Feed the atmospheric light and refined transmission amp to ASM model to get the Dehazed image.

**Colour attenuation prior using Haze lines prior**:

This algorithm is mixture of CAP and Haze line prior algorithms.

The main objective of this algorithm:

* Find the scene depth
* Find the scaterring coefficient
* Estimate the air Light How to find scene depth:

In order to find the scene depth we use CAP algorithm. This algorithm states that:

‘If the thickness of the haze increases the scene depth also increases’.

Hence the scene depth of the image will be linearly dependent on the difference between brightness and saturation in the image. Using this we estimate the scene depth.

How to find scattering coefficient:

In general scattering coefficient is kept constant as in most of the images the haze is evenly spread. But in case of satellite images the haze is inconsistent and unevenly spread. So the scattering coefficient must keep changing. Scattering coefficient directly depends on scene depth. Hence with an increase in scene depth, scatering coefficient increases exponentially.

How to find airlight:

In order to estimate the airLight we use Haze lines Prior. In this algorithm, we map the pixels in the image to RGB colorspace. Then the model takes these pixels as lines and these lines converge at a point. This point is airLight. Then we use a feature extraction model in order to find the origin point for this airLight.

After finding scene depth,scaterring coefficient and estimating the airlight, we send these values to ASM model in order to get the dehazed image.

**CLAHE Algorithm and Mix CLAHE:**

This algorithm defines two functions, clahe and clahe2, which can be used to apply the Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm to an image.

The clahe function takes three parameters:

image: The input image, which should be a 2D array.

clipping\_limit: A float value that sets the limit for contrast enhancement. Higher values allow for more contrast enhancement, but can also amplify noise in the image. The default value is 2.0. grid\_size: A tuple that sets the size of the contextual regions, which are the small regions of the image that the algorithm adjusts the contrast of separately. The default value is (8, 8).

The function first gets the dimensions of the input image, and then divides the image into nonoverlapping contextual regions based on the grid\_size parameter. It then creates an output image (clahe\_image) that is the same size as the input image, but filled with zeros.

The function then performs CLAHE on each grid. For each grid, it:

* Gets the grid region of the image by slicing the input image with the grid's row and column indices
* Gets the histogram of the grid region, which is a distribution of the number of pixels of each intensity level in the grid region
* Clips the histogram by a certain value. Clipping the histogram means that it limits the number of pixels in the histogram that can have a high intensity level. This is done to avoid amplifying noise in the image. The clip limit is determined by the clipping\_limit parameter, which is a float value. The clip limit value multiplied by the total number of pixels in the histogram gives the maximum number of pixels that can have high intensity levels.
* Calculates the average number of pixels to redistribute, which is the total number of pixels in the histogram minus the number of clipped pixels, divided by the number of intensity levels.
* Redistributes the clipped pixels by adding them to the neighboring intensity levels. This is done by starting from the highest intensity level and adding the clipped pixels to the intensity levels in descending order until all the clipped pixels have been added.
* Calculates the cumulative density function (CDF) of the histogram, which is a distribution of the cumulative number of pixels of each intensity level in the grid region
* Maps the original pixel values to the new values using the CDF. This is done using the numpy's interp() function, which maps the original pixel values to new values according to the CDF.
* Inserts the grid region back into the output image by slicing the clahe\_image with the grid's row and column indices.
* After all the grids have been processed, the function returns the output image, clahe\_image, which contains the result of the CLAHE algorithm applied to the input image.

Now we get the output that is dehazed image

**Working of Mix CLAHE:**

This works similar to CLAHE but works in HSV space, HSV stands for Hue, Saturation, and Value. It is a color space that represents colors using three channels: Hue, Saturation, and Value. Hue represents the color itself, and is typically represented as a value between 0 and 360. Saturation represents the intensity of the color, and is typically represented as a value between 0 and 100. Value represents the brightness of the color, and is typically represented as a value between 0 and 100. The clahe2 function first converts the image from the BGR color space to the HSV color space using the cv2.cvtColor() function. This allows the function to separate the color information (Hue and Saturation) from the brightness information (Value). By isolating the brightness information, the function can apply the CLAHE algorithm specifically to the V channel of the image, which represents the Value or the brightness. This can help to remove haze more effectively since haze mainly affects the visibility by reducing the brightness of the image. After applying the CLAHE to the V channel of the image, the function converts the image back to the BGR color space using the cv2.cvtColor() function. Finally, it applies the CLAHE on each channel of the image using the clahe function, which helps to further improve the contrast and visibility of the image. The final output of the function is the image with CLAHE applied to each channel.

**The two parameters that control the dehazing:**

The clip\_limit parameter sets the limit for contrast enhancement. A higher value allows for more contrast enhancement, but can also amplify noise present in the image. A lower value will limit the contrast enhancement and will result in a less contrasted image. Increasing the clip\_limit value allows more pixels to be included in the histogram thus resulting in a more contrasted image but it also amplifies noise in the image. The grid\_size parameter sets the size of the contextual regions, which are the small regions of the image that the algorithm adjusts the contrast of separately. A smaller grid size will result in more regions, and therefore more local adjustments to the contrast of the image. A larger grid size will result in fewer regions, and therefore fewer local adjustments to the contrast of the image. Increasing the grid size will result in more general contrast enhancement while decreasing the grid size results in more local contrast enhancement.