DVE FINAL PROJECT

Team_28 B09902017 李安傑 B09902128 黃宏鈺

Intruduction

In this project, we try to remove haze in the images by dark channel prior, enhance the color by bright channel prior and improve the preformance on the edge by guided filter.

Dark Channel Prior

Reference

K. He, J. Sun and X. Tang, "Single Image Haze Removal Using Dark Channel Prior," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 12, pp. 2341-2353, Dec. 2011, doi: 10.1109/TPAMI.2010.168.

Intruduction

- **Haze image function**
  An image I can be viewed as weighted average of J and A, where J is the scene radiance, A is the global atmospheric light and the ratio of them is determined as transmittance t :

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

  The reference paper tried to obtain the haze removed image $i.e.$ J by calculating t and A.

- **dark channel**
  For most of the regin that does not belong to sky, some pixel is expected to have very low value at least in one channel, the dark channel is given as :

$$J^{dark}(x) = \min_{y \in \Omega(x)} \left(\min_{c \in \{r,g,b\}} J^c(y) \right) \rightarrow 0$$

  where $\Omega(x)$ means a window centers at x, c means one of the channel in red, blue and green,

- **transmittance**
  Suppose that for every local window $\Omega(x)$, $t(x)$ is a constant $\tilde{t}(x)$ which can be getten as following :

$$\text{given } c \in \{r,g,b\} \rightarrow \frac{I^c(y)}{A^c} = t(x)\frac{J^c(x)}{A^c} + 1 - t(x)$$

$$\min_{y\in\Omega(x)}\left(\min_c \frac{I^c(y)}{A^c}\right) = \tilde{t}(x)\min_{y\in\Omega(x)}\left(\min_c \frac{J^c(y)}{A^c}\right) + 1 - \tilde{t}(x)$$

$$\because \quad \min_{y\in\Omega(x)}\left(\min_c \frac{J^c(y)}{A^c}\right) = 0$$

$$\therefore \quad \tilde{t}(x) = 1 - \min_{y\in\Omega(x)}\left(\min_c \frac{I^c(y)}{A^c}\right)$$

- **global atmospheric light**
  First, pich the top 0.1% brightest pixels in the dark channel which represents the most hazeopaque. Among, these pizels, the pixels with highest intensity in the input image I is selected as the atmospheric light.

Note

- **transmittance of air**
  In the real world, the air is not totally transparent, so $\tilde{t}$ is changed as follow :

$$\tilde{t}(x) = 1 - \omega \min_{y\in\Omega(x)}\left(\min_c \frac{I^c(y)}{A^c}\right), \omega \in [0,1]$$

- **adust small transmittance**
  To avoid divided by zero (or a small number) :

$$\lim_{t(x)\rightarrow 0} J(x) = \lim_{t(x)\rightarrow 0}\left(\frac{I(x)-A}{t(x)} + A\right) = inf$$

  $J(x)$ is adjusted as follow :

$$J(x) = \frac{I(x)-A}{\max(t(x), t_0)} + A$$

Bright Channel Prior

Reference

Sandoub G, Atta R, Ali HA,Abdel-Kader RF. A low-light image enhancementmethod based on bright channel prior and maximumcolour channel.IET Image Process.2021;15:1759–1772.https://doi.org/10.1049/ipr2.12148

Intruduction

The reference paper using the bright channel and maximumcolour channel to enhance low-light images.

- **Image function**
  An normal image I (without haze) can be represented by:

$$I = L \cdot R$$

where I, L, R, represents the measured image, light illumination, and the scene reflectance (the enhamced image we want).
The reference paper tried to use Bright Channel and maximumcolour channel to approach L, and then get R (enhamced image) by devided I with L.

- **bright channel and maximumcolour channel**
  Similar with dark channel in last chapter, the bright channel can be writen as:

$$I^{bright}(x) = \max_{y \in \Omega(x)} \left(\max_{c \in \{r,g,b\}} I^c(y) \right)$$

  Maximumcolour channel is a special case of bright channel with the $Omega(x) = x$, which means it only choose the maximum value from R, G, B of the pixel x itself.
  According to the reference paper, the maximumcolur channel may suffer from the colour distortion, while the bright chennel method considers the local consistency but suffer from the halo artifacts effect.
  So the reference paper combine this two channel into a Fusion-based bright channel, to deal with this trade off.
- **Fusion-based bright channel estimation**
  The reference paper calculate the weight for every pixel x by function:

$$weight_{bright}(x) = (I^{bright}(x) - I^{max}(x))/I^{bright}(x)$$

  And then use this weight to linearly combine two channel into Fusion-based bright channel:

$$\hat{L}(x) = I^{'bright} = I^{bright}(x) \cdot (1 - weight_{bright(x)}) + I^{max}(x) \cdot weight_{bright(x)}$$

  And finally use this $\hat{L}$ and the measured I to compute the enhanced image R:

$$R(x) = I(x)/\max(\hat{L}(x), L_{min})$$

  L_{min} is used to prevent division of the too small $\hat{L}$.
- **Refinement**
  The reference paper next tried to refine the enhanced picture by function:

$$I_{refine}(x,c) = \begin{cases} I'(x,c) + (1 + \sigma(x,c)) \cdot D(x,c), \text{ if } D(x,c) > 0 \\ I'(x,c) \end{cases}$$

  where $D(x,c) = I'(x,c) - avg_{\Omega}(x,c)$ is a color's (R or G or B) light intense differnce between a pixel with its neighbor's average.

Note

After implement this low-light images enhancement with fusion-based method, we also try the Guided Filter with the Bright Channel Prior instead of this fusion method, and it has a much better result. So we finally use the bright channel with Guided Filter as our image enhancement method.

The more detail of the Guided Filter will be described in next chapter.

Guided Filter

Reference

Guided Image Filtering, by Kaiming He, Jian Sun, and Xiaoou Tang, in ECCV 2010 (Oral).

Intruduction

Guided filter is a edge-preserving filter with linear time complexity. With a guided image I and input image p, guided filter trying to get output image q where

$$\begin{cases} q_i & = aI_i + b \\ q_i & = p_i - n_i \end{cases} \Rightarrow n_i = p_i - aI_i + b$$

By minimizing the cost function :

$$E(a_k, b_k) = \sum_{i \in w_k} ((a_k I_i + b_k - p_i)^2 + \epsilon a_k^2)$$

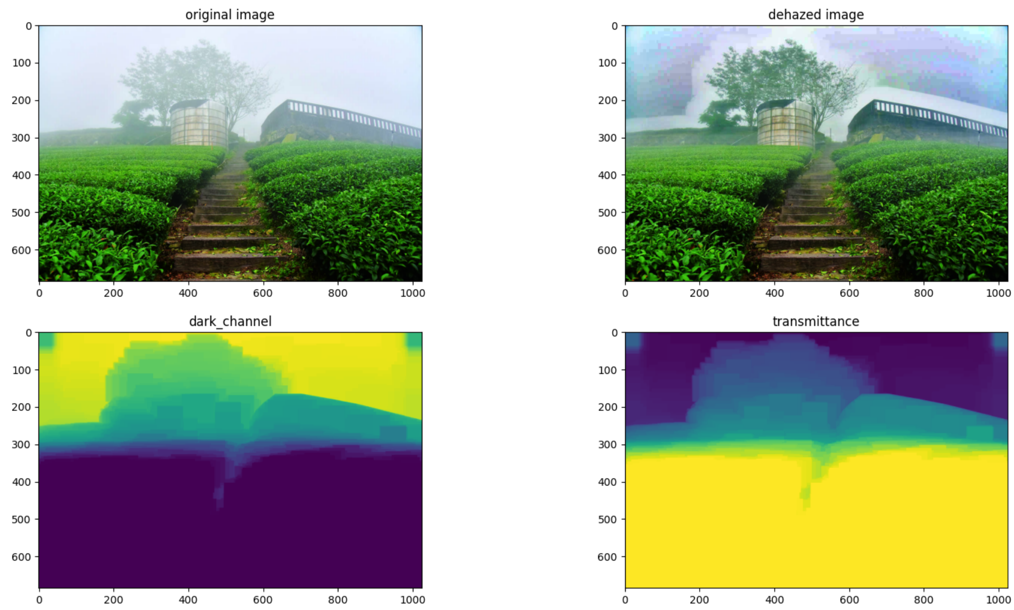
where w_k is a window center on kth pixel, ϵ is used to penalize big a.

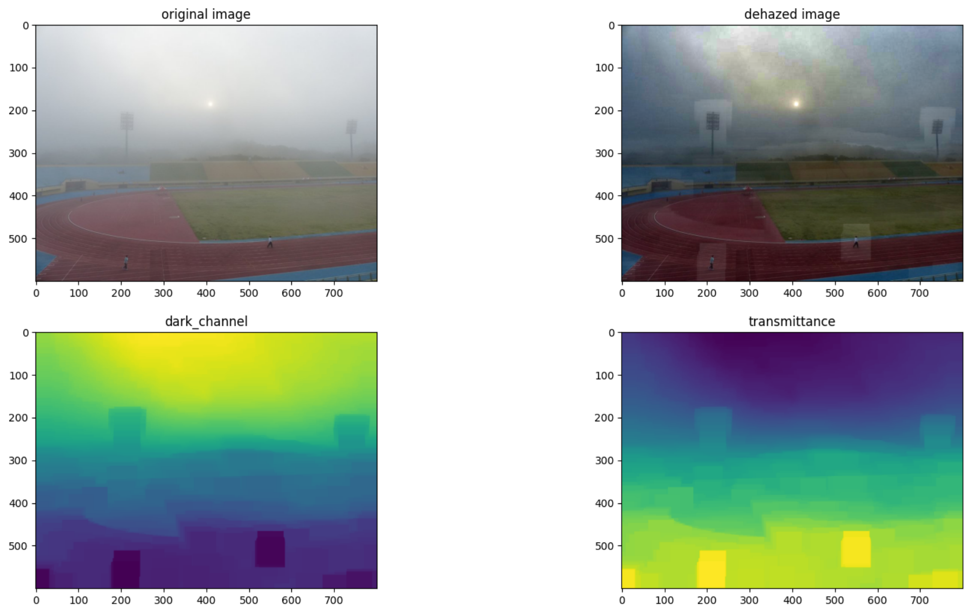
Implement

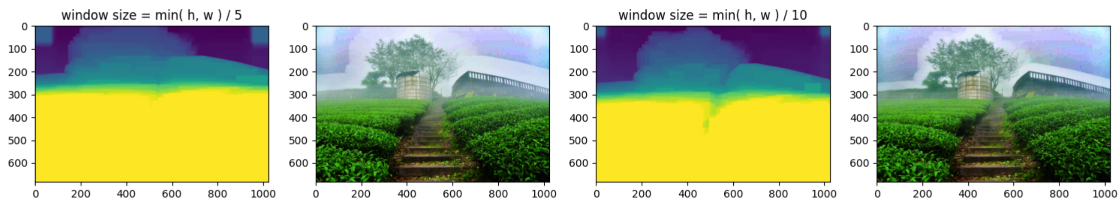
The auther have given their implementation in matlab. I reproduce a verson in python with numpy.

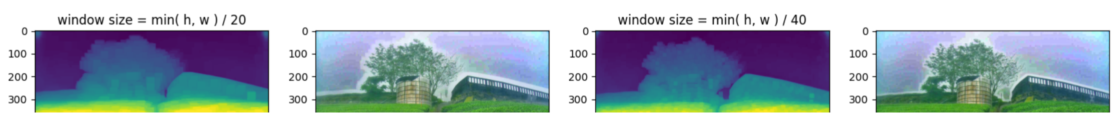
Exprement & Result

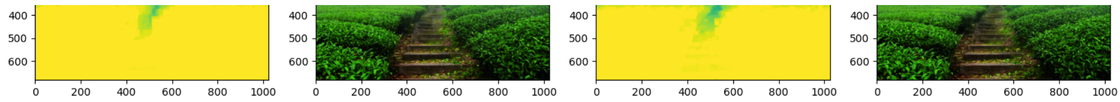
Dark Channel Prior

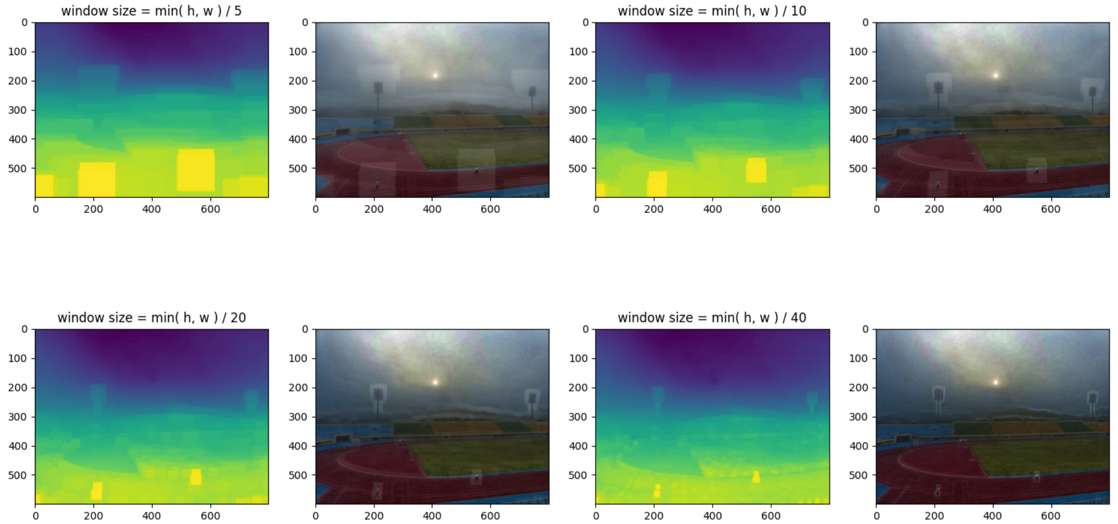

original image
dehazed image
dark_channel
transmittance


original image
dehazed image
dark_channel
transmittance

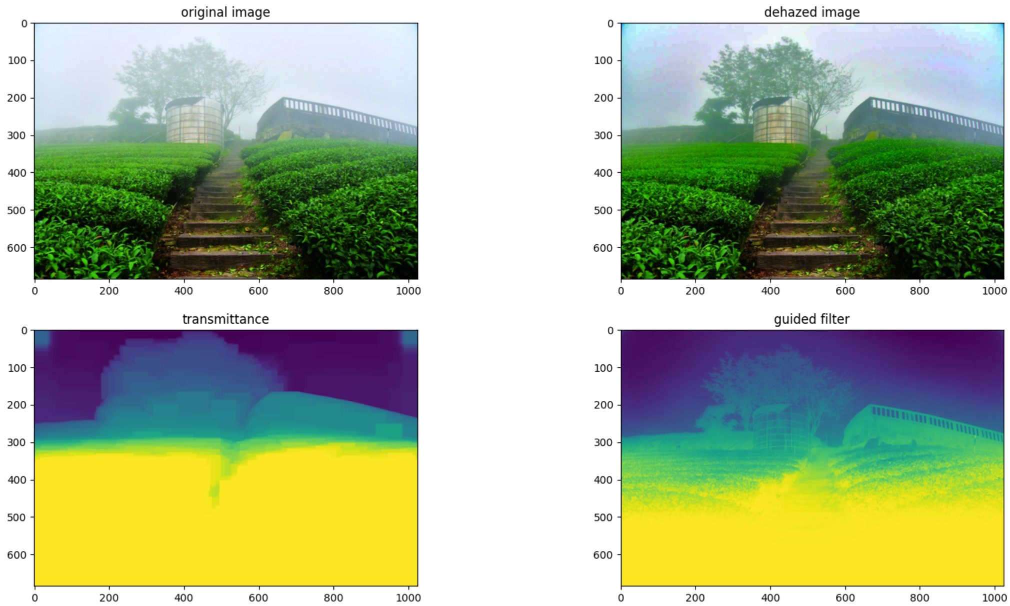

window size = min(h, w) / 5
window size = min(h, w) / 10

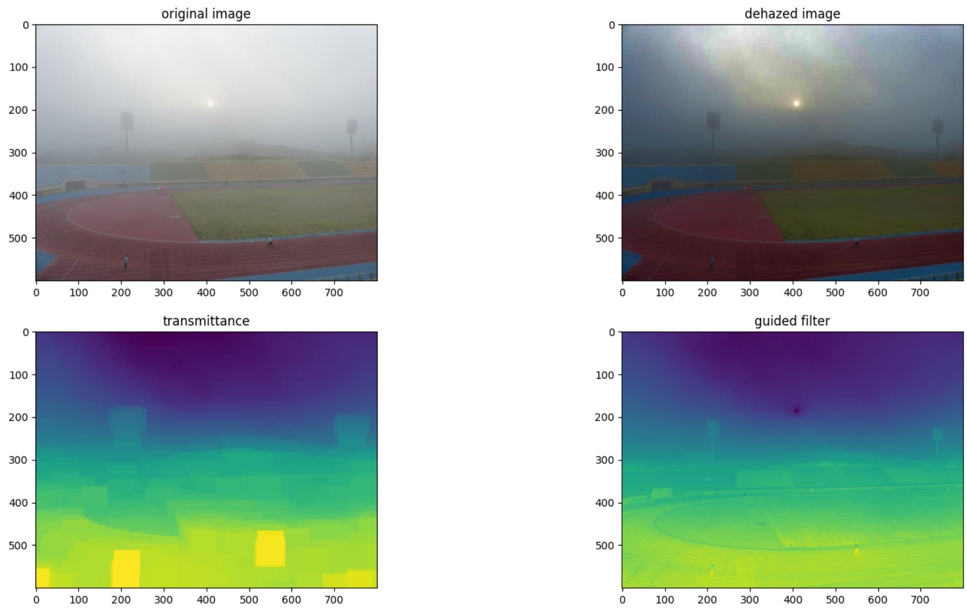

window size = min(h, w) / 20
window size = min(h, w) / 40





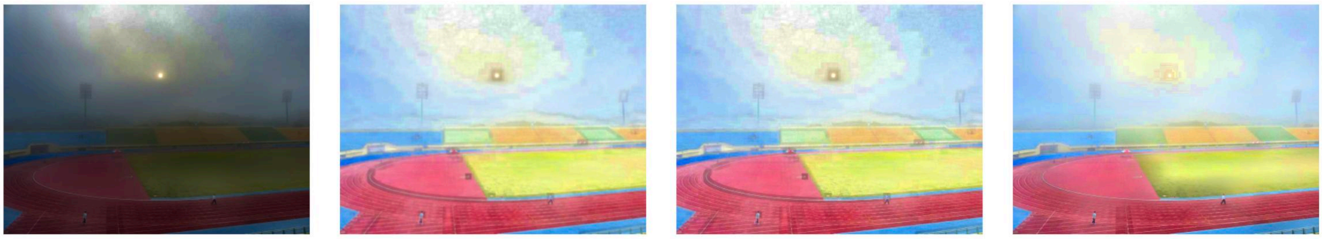
Dark Channel Prior with Guided Filter





Bright Channel Prior (with Guided FIlter)

Original image | Bright Channel | Bright Channel + Refined | Bright Channel + Guided Filter



Original image | Bright Channel | Bright Channel + Refined | Bright Channel + Guided Filter

Original image | Bright Channel | Bright Channel + Refined | Bright Channel + Guided Filter