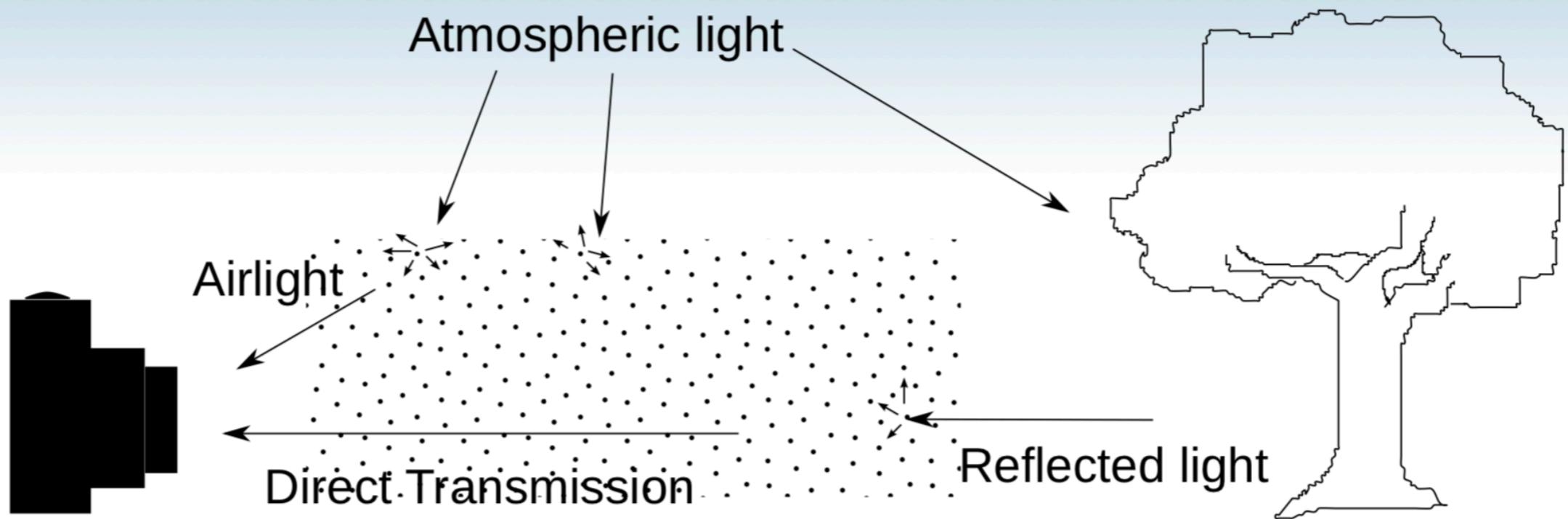


# Single Image Haze Removal Using Dark Channel Prior

Kaiming He, Jian Sun, and Xiaoou Tang, Fellow, IEEE

Самойленко Александр,  
МФТИ 2019

# Imaging model



In haze/fog the image formation equation is given by

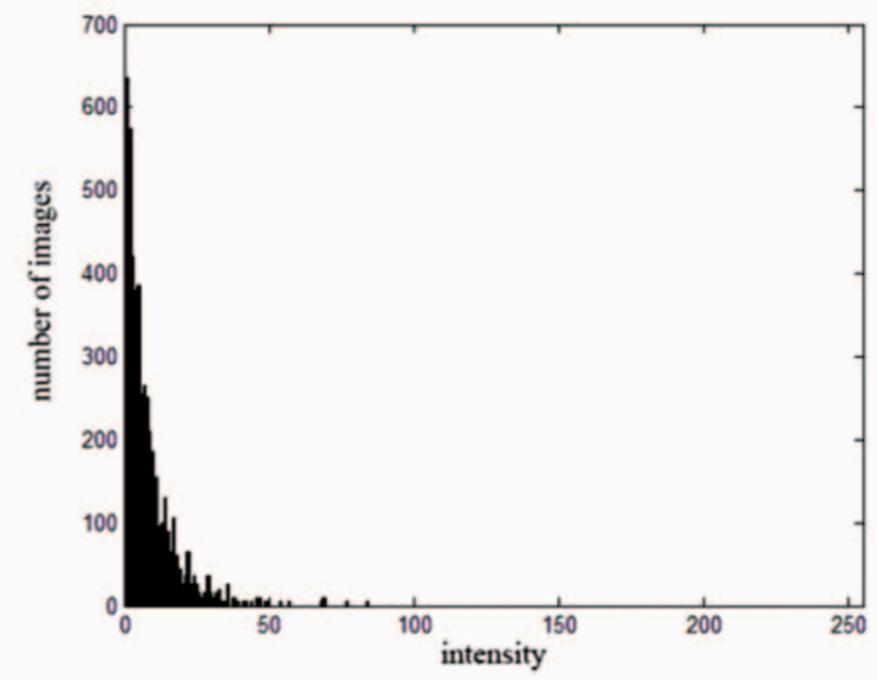
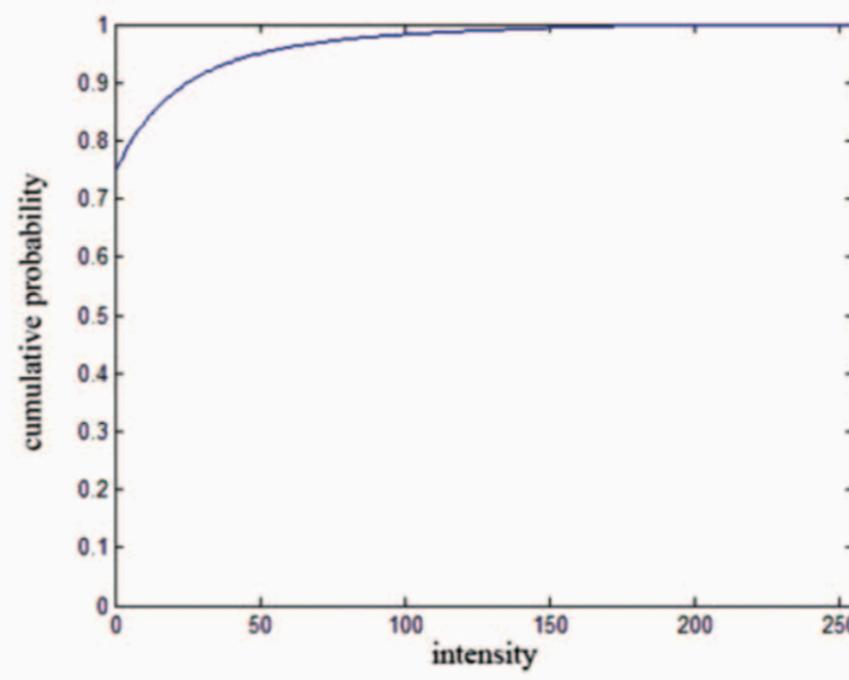
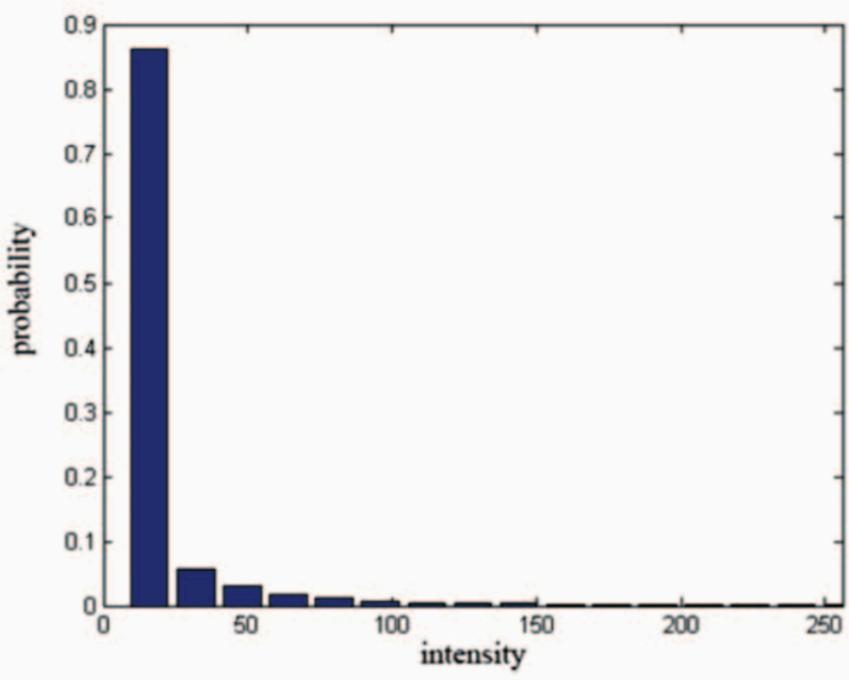
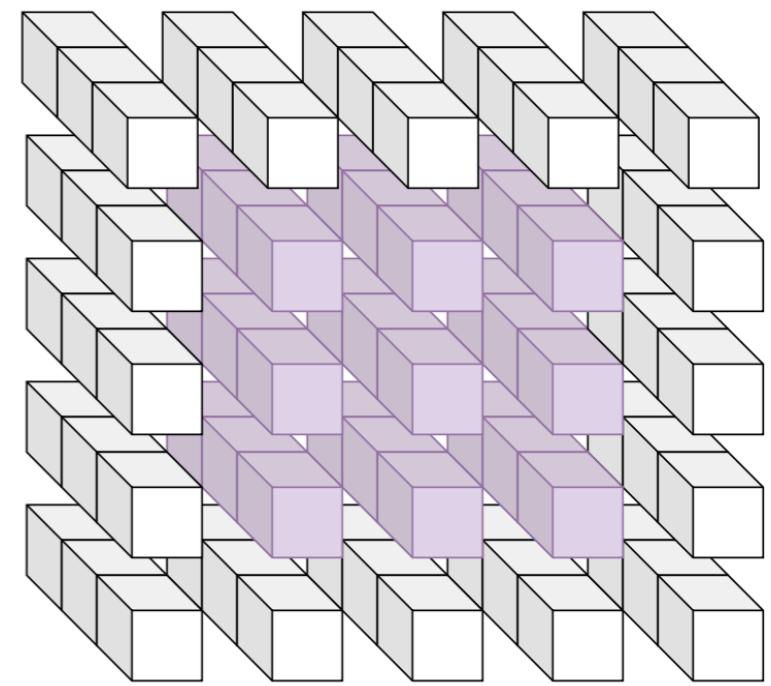
$$I(\mathbf{x}) = \underbrace{J(\mathbf{x})t(\mathbf{x})}_{\text{Direct transmission}} + \underbrace{(1 - t(\mathbf{x}))A}_{\text{Airlight}};$$

# **Dark Channel Prior**

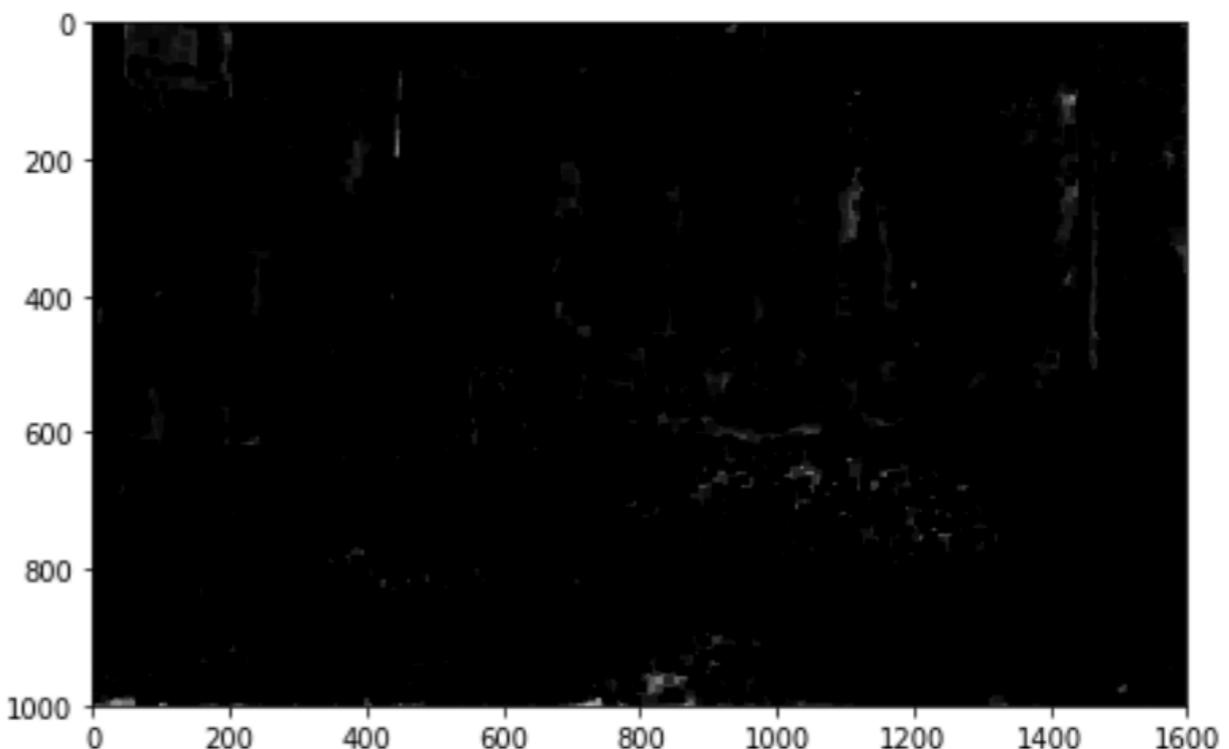
# Dark Channel Prior

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

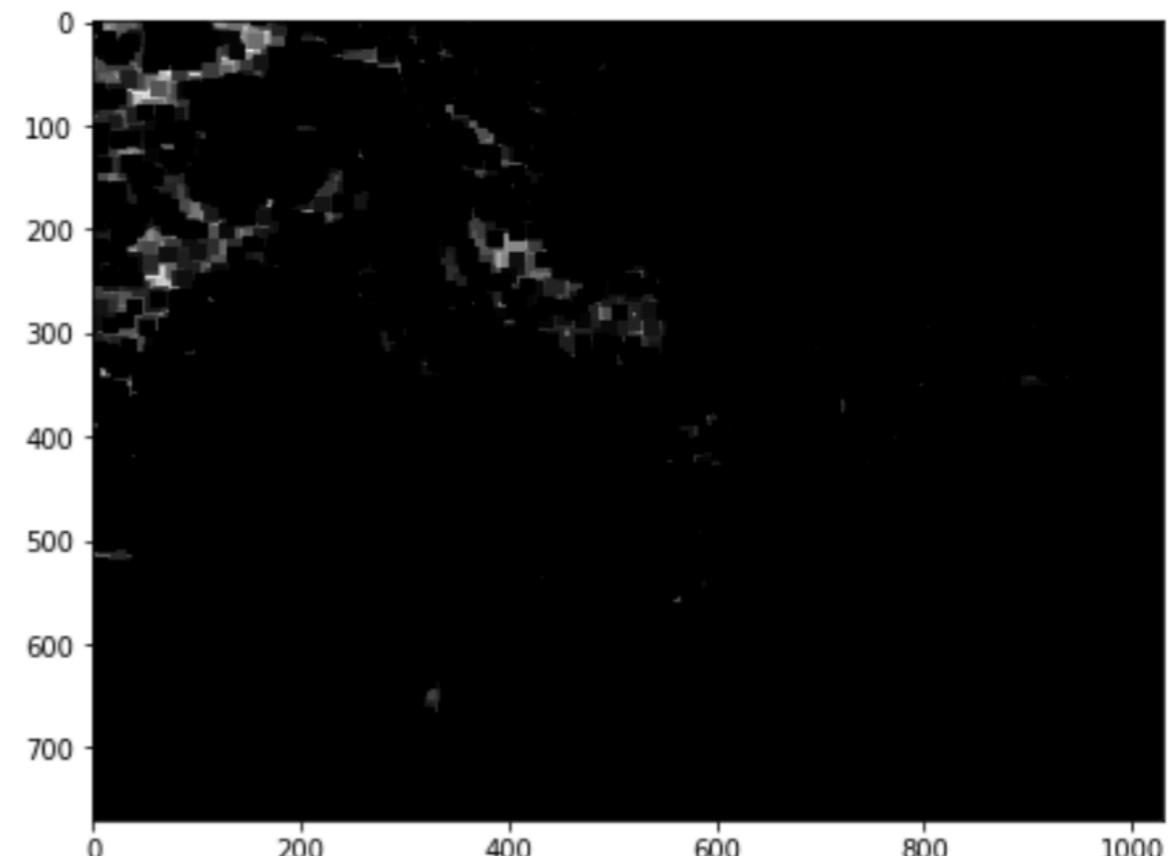
$$J^{dark} \rightarrow 0$$



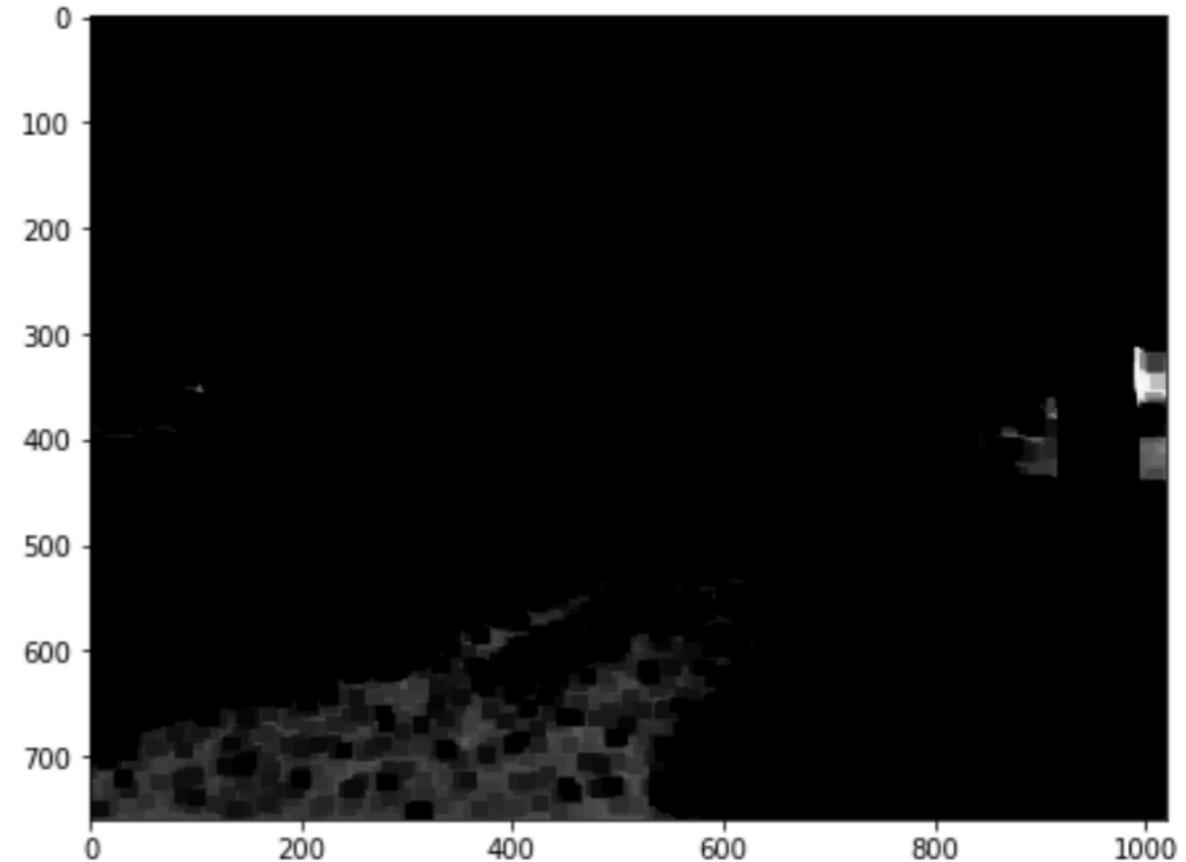
```
1 img0.dark_channel_visualize()
```



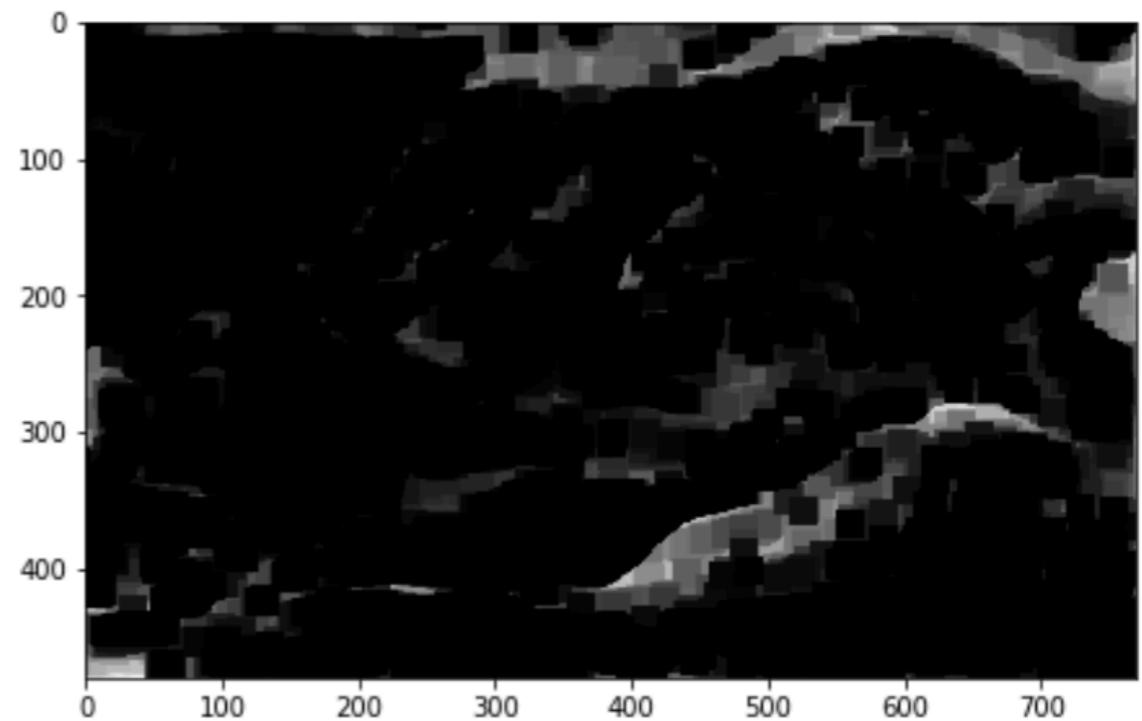
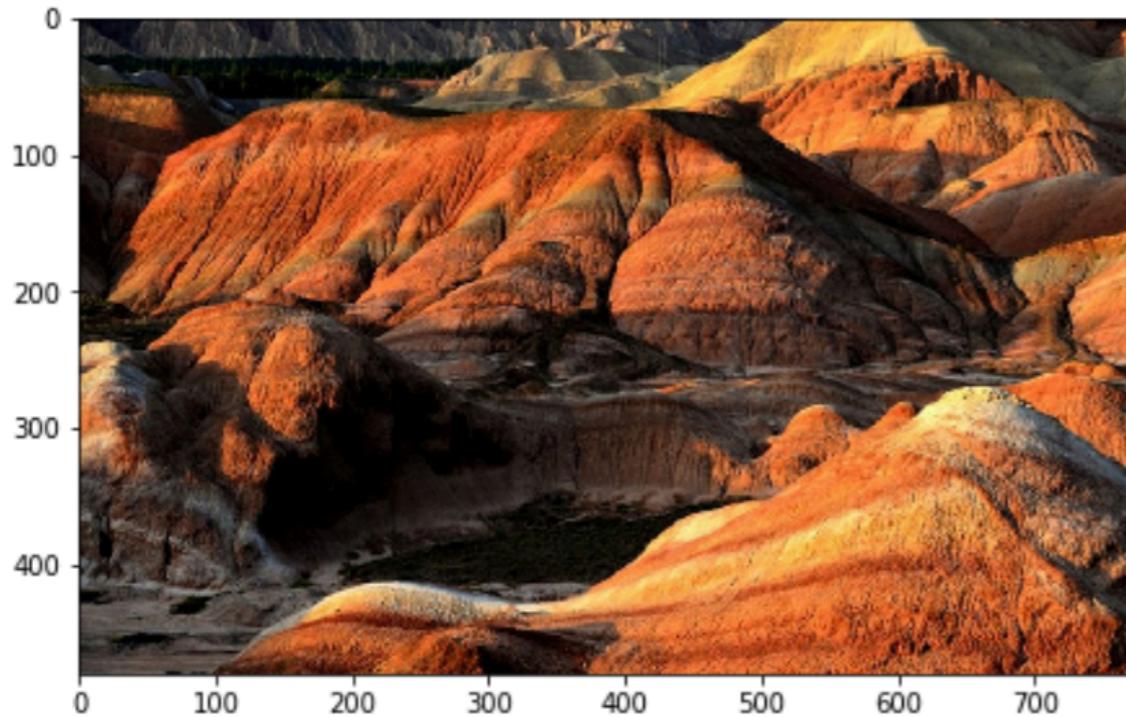
```
1 img1.dark_channel_visualize()
```



```
1 | img2.dark_channel_visualize()
```

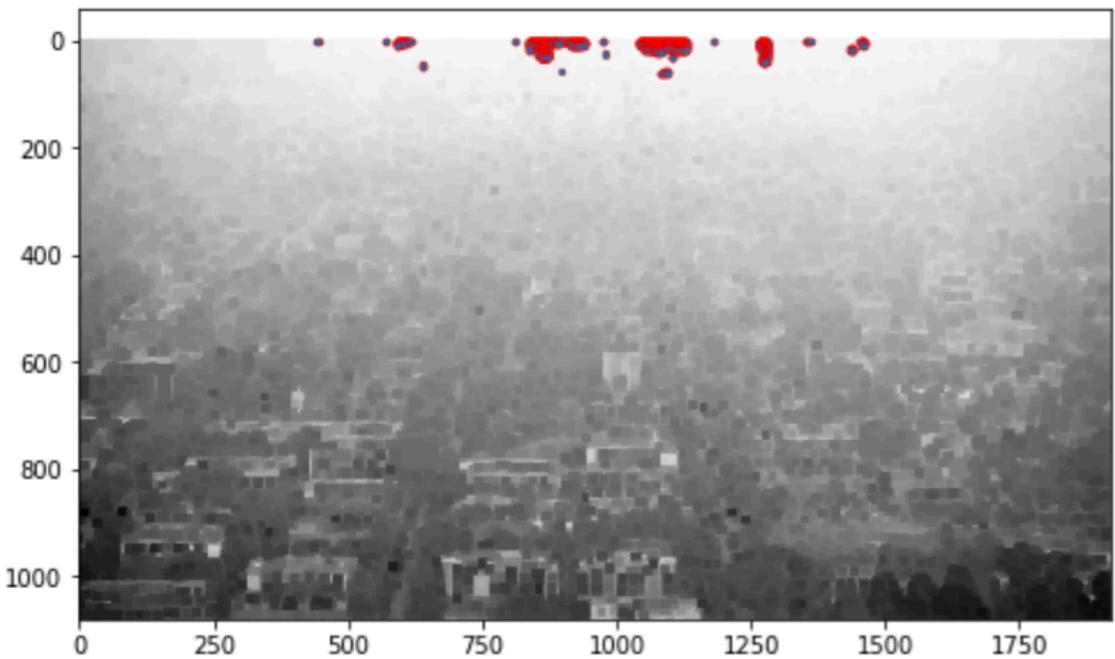


```
1 | img3.dark_channel_visualize()
```

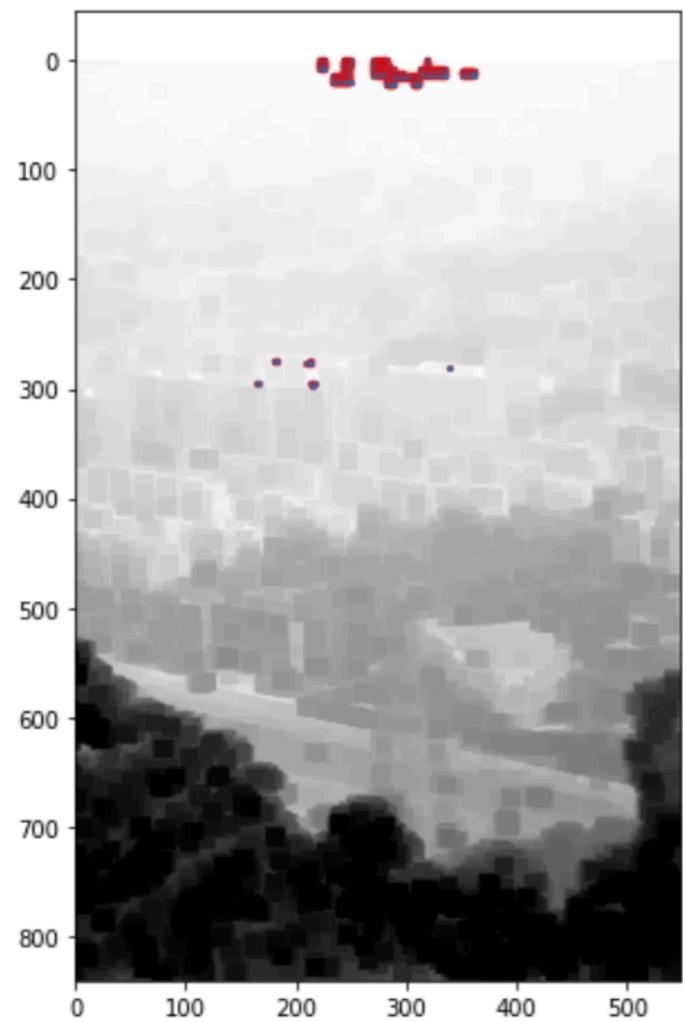


# Atmospheric Light

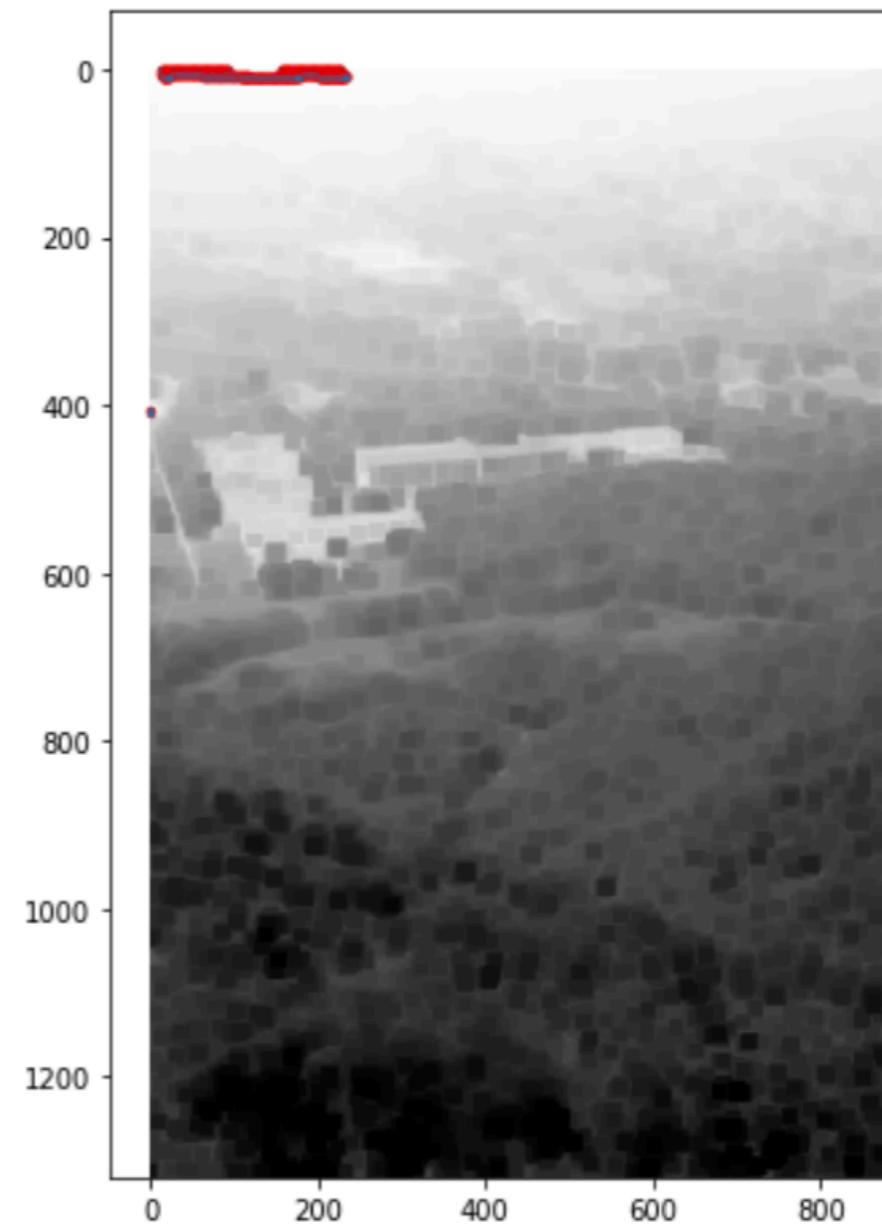
```
1 _ = img8.mark_top_k_percent_pixels()
```



```
1 _ = img9.mark_top_k_percent_pixels()
```



```
1 _ = img8.mark_top_k_percent_pixels()
```



# Transmission

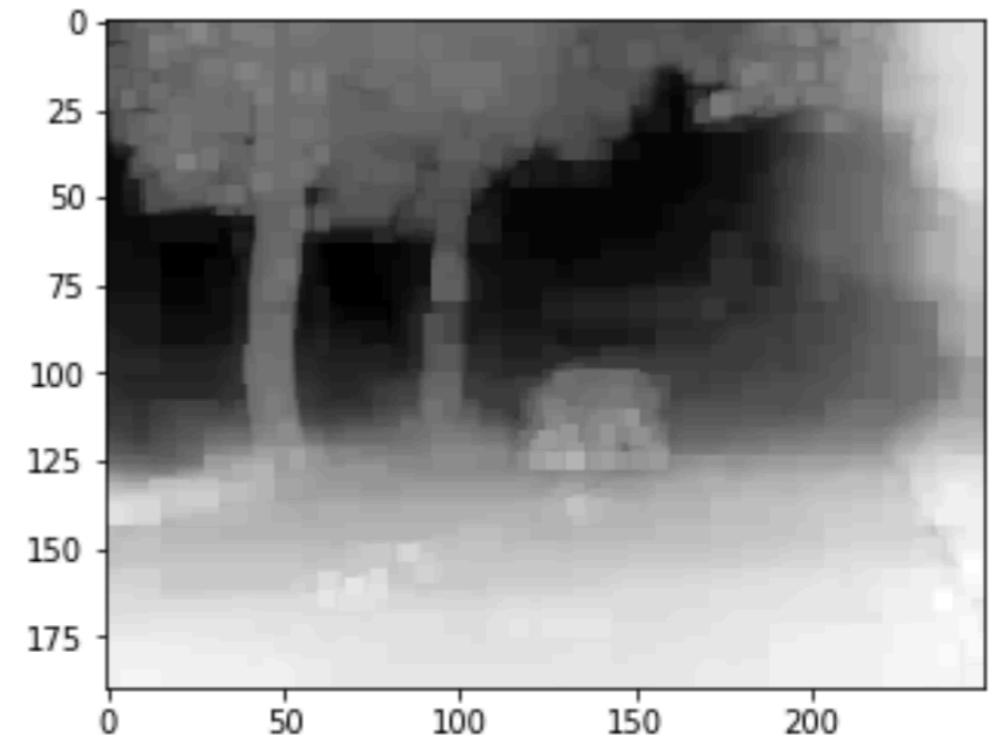
# Transmission

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

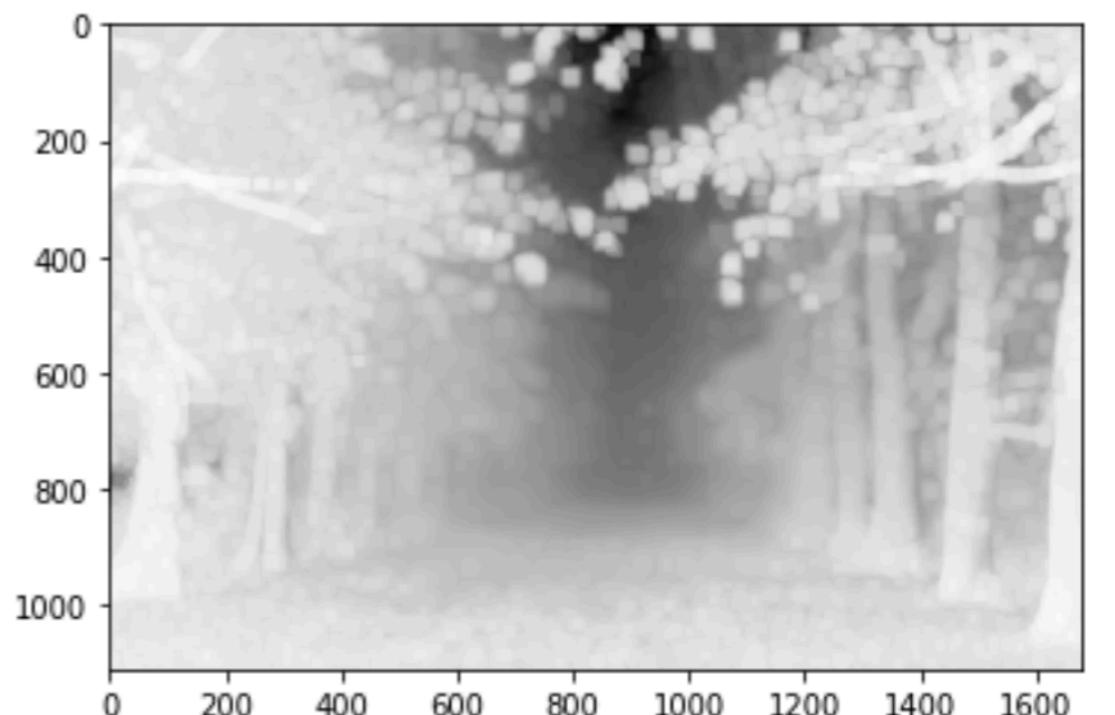
$$\min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_{c \in \{r,g,b\}} \frac{I^c(\mathbf{y})}{A^c} \right) = \tilde{t}(\mathbf{x}) \min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_{c \in \{r,g,b\}} \frac{J^c(\mathbf{y})}{A^c} \right) + 1 - \tilde{t}(\mathbf{x})$$

$$\tilde{t}(\mathbf{x}) = 1 - \omega \min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_{c \in \{r,g,b\}} \frac{I^c(\mathbf{y})}{A^c} \right)$$

```
1 | img6.show_trans_map()
```

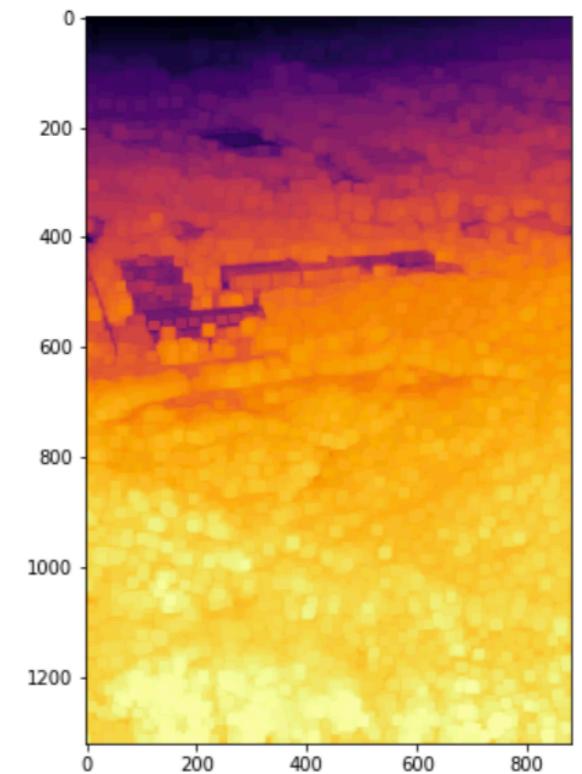
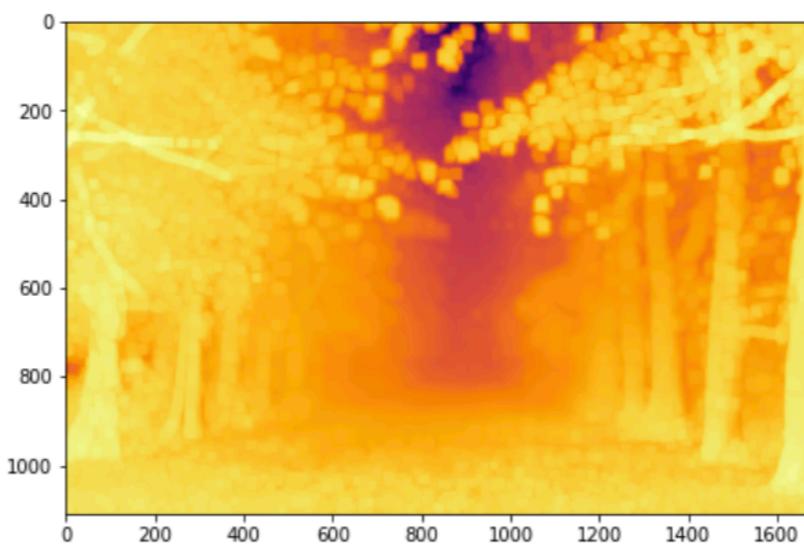
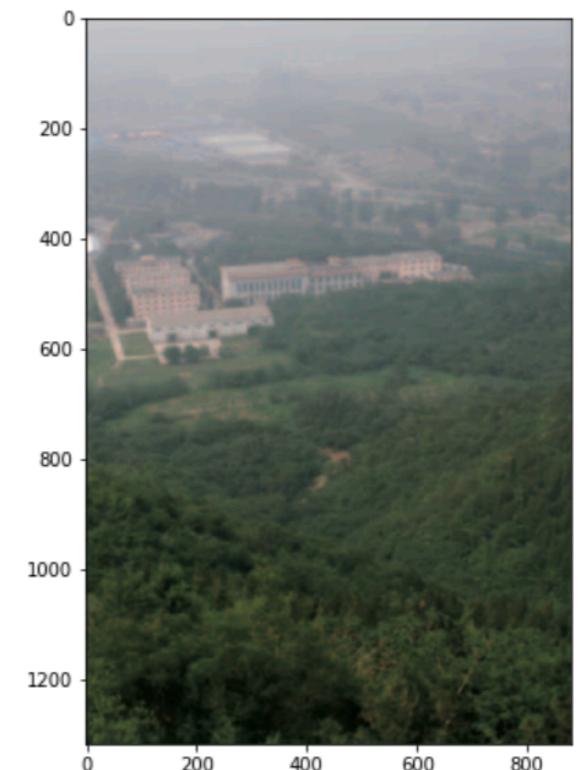


```
1 | img7.show_trans_map()
```

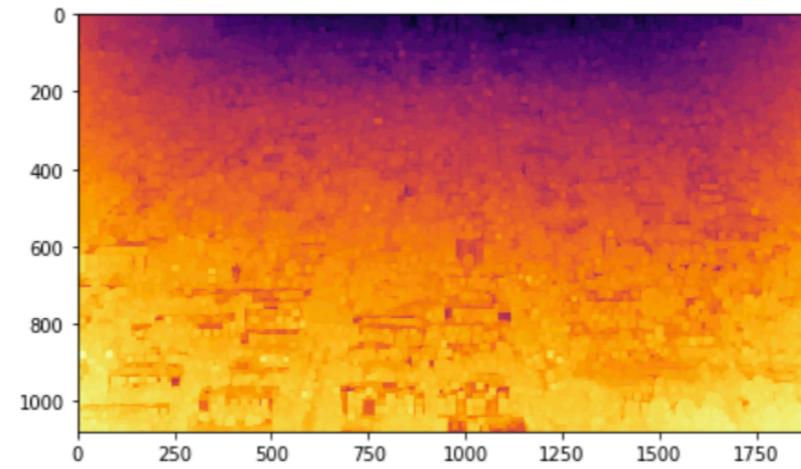
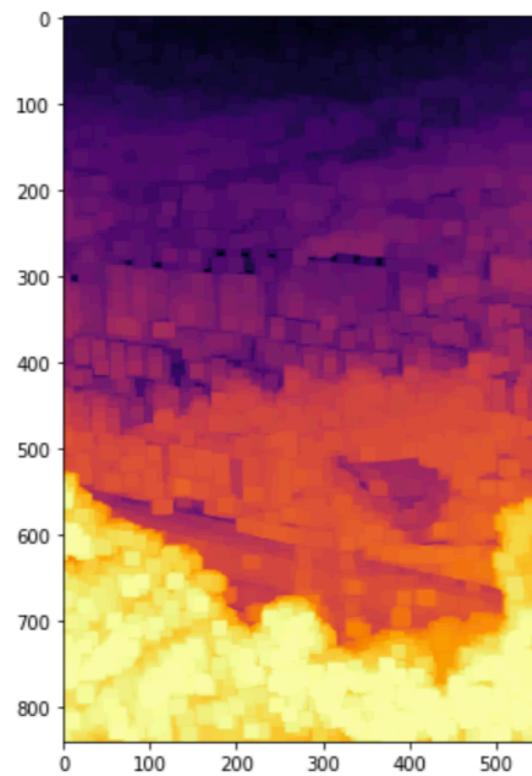
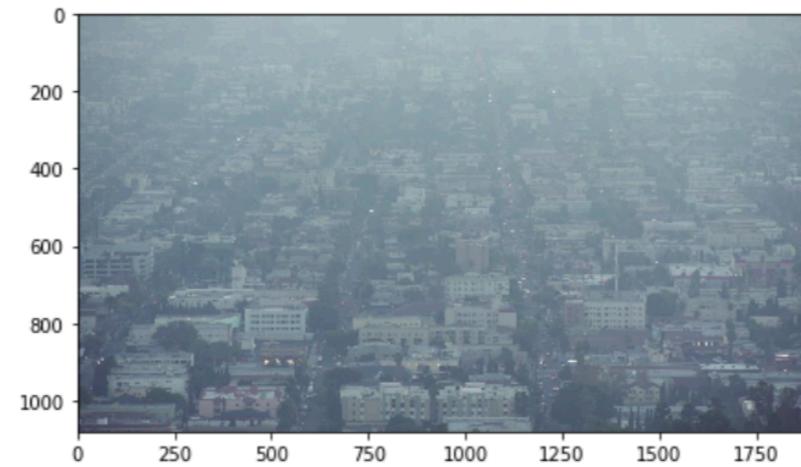


# Transmission + depth map

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



# Transmission + depth map

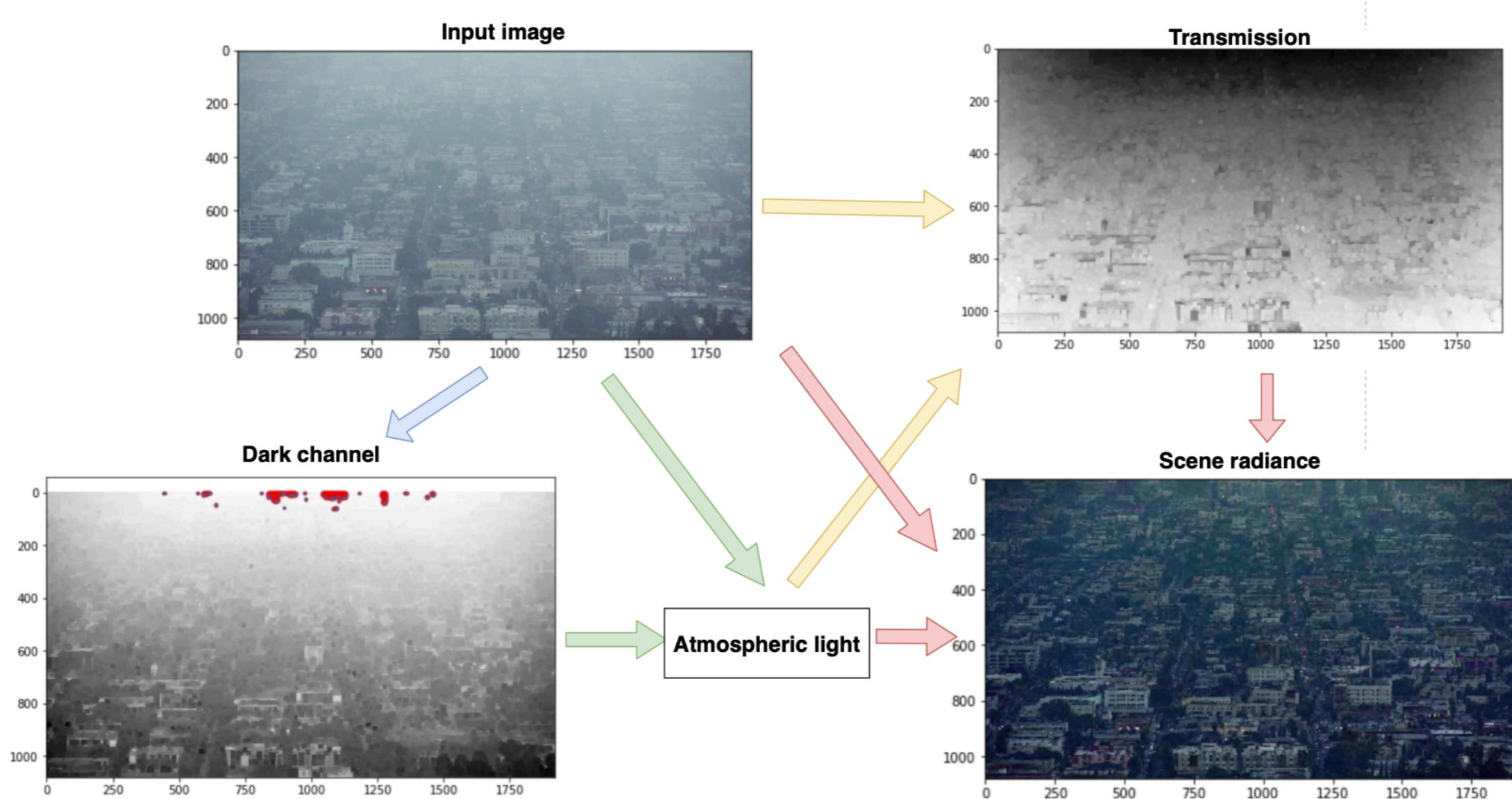


# Recovering the scene radiance

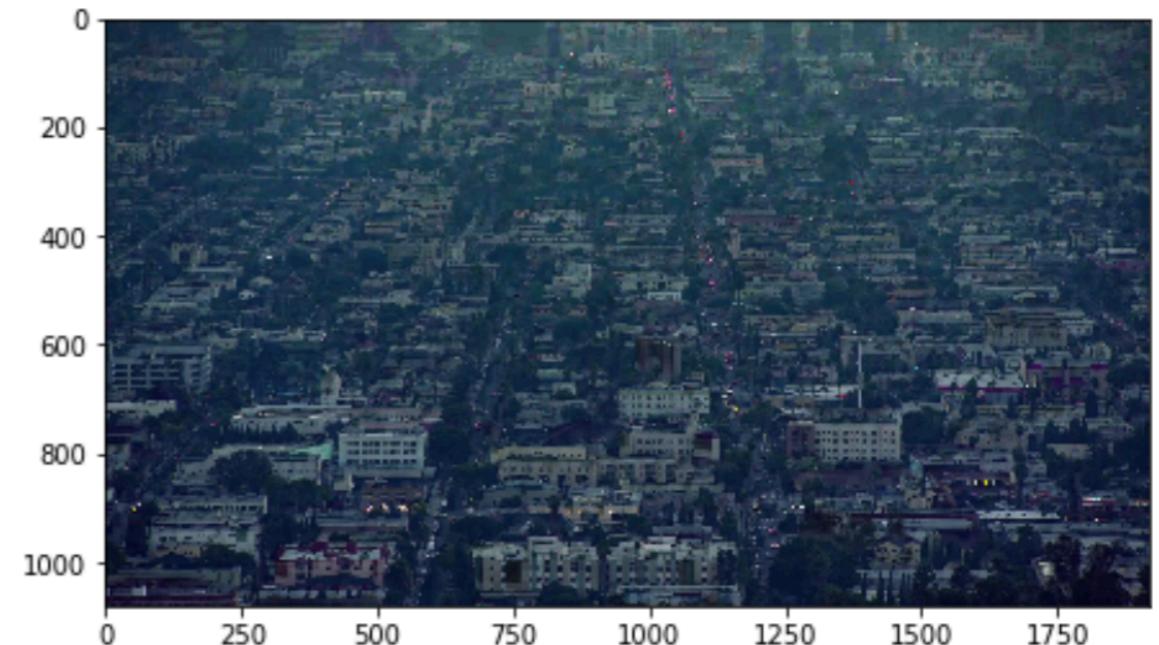
# Recovering the scene radiance

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

$$J^c(\mathbf{x}) = \frac{I^c(\mathbf{x}) - A^c}{\max(t_0, t(\mathbf{x}))} + A^c, \quad c \in \{r, g, b\}$$



```
1 img10 = Antihaze('haze3.png', patch_size=(15,15), omega=0.95)
2 img10.run()
```



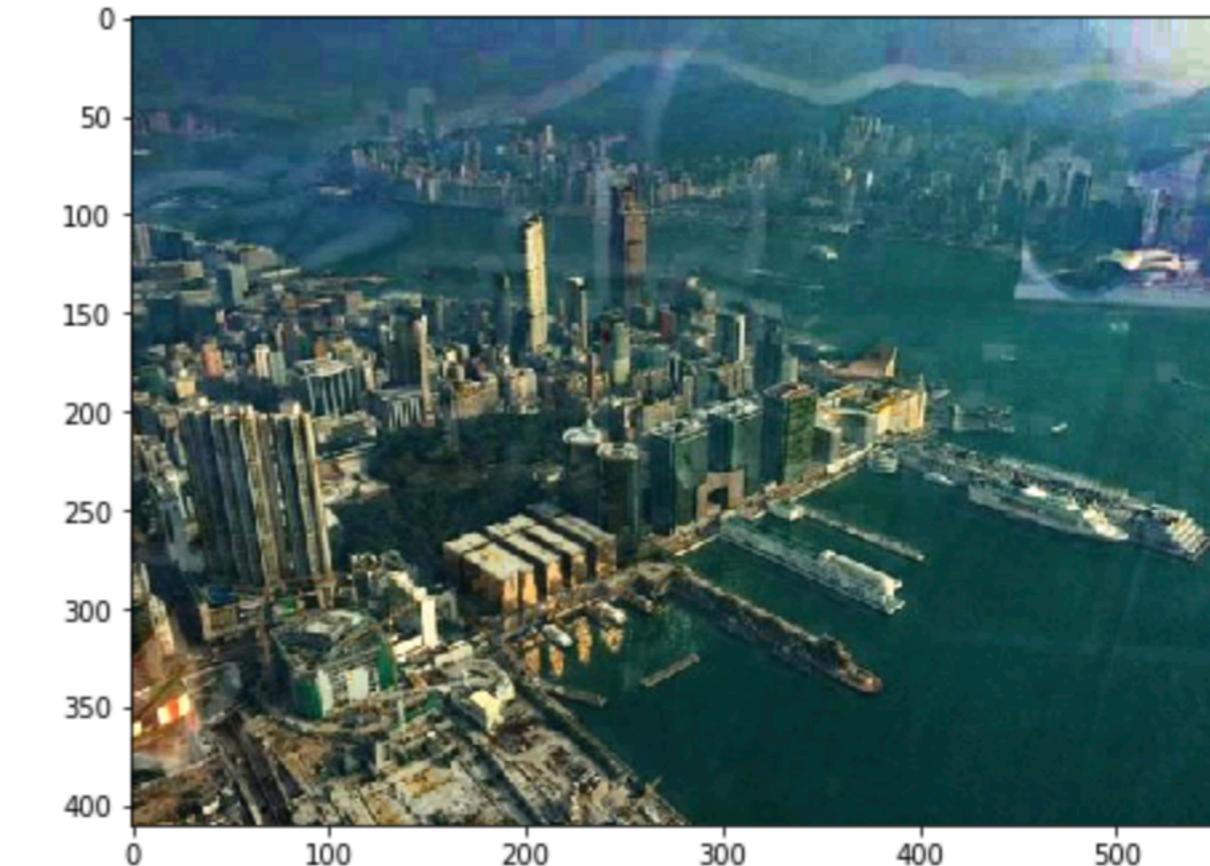
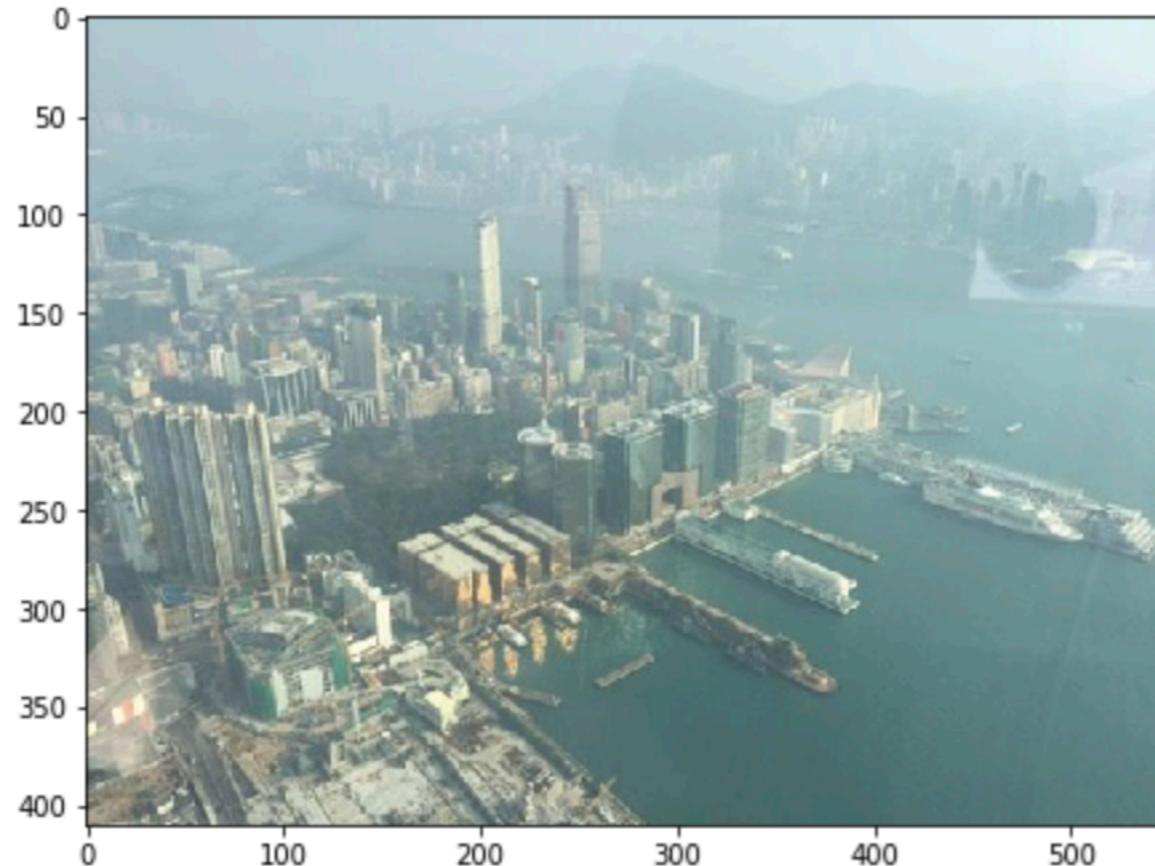
```
1 img11 = Antihaze('img4.jpg')
2 img11.run()
```



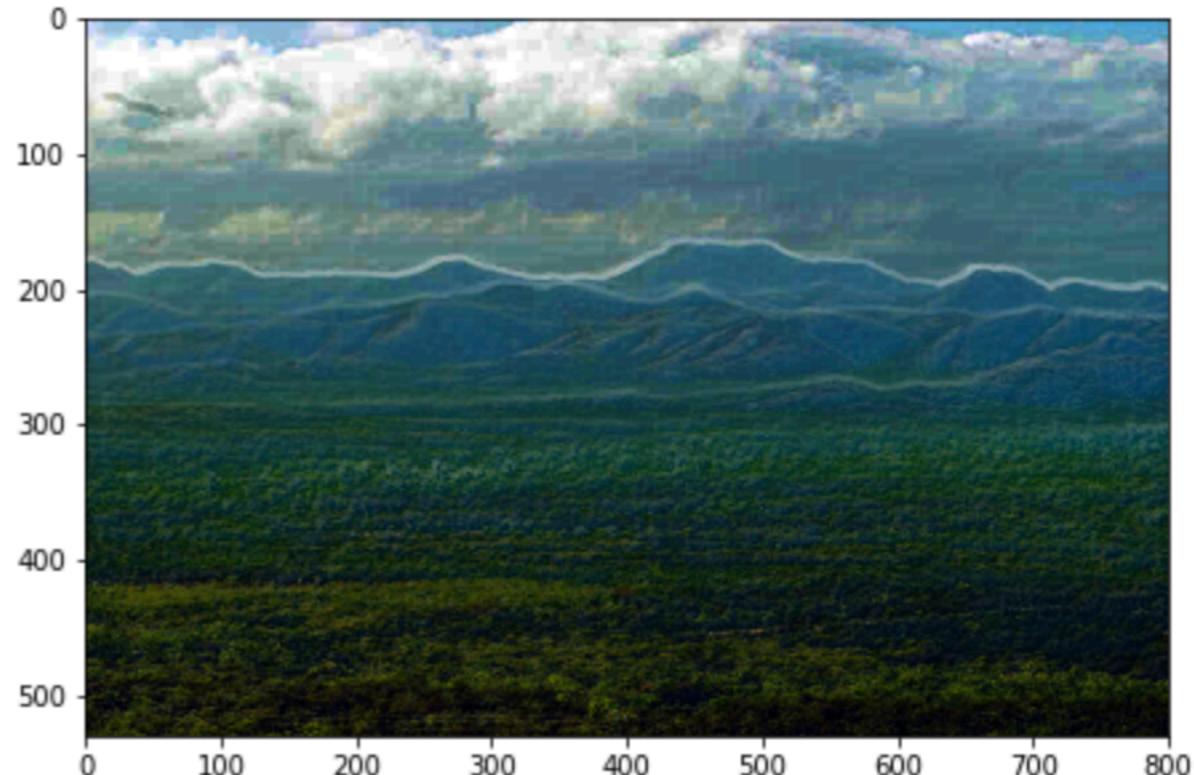
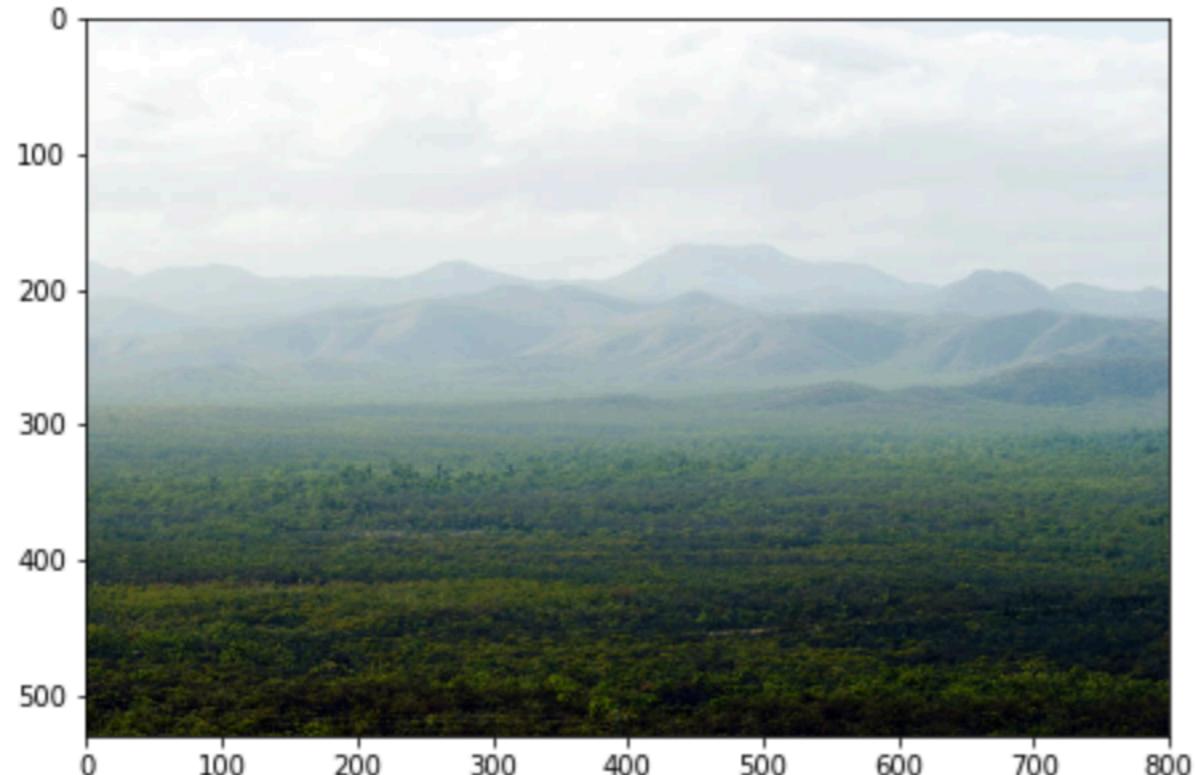
```
1 img12 = Antihaze('haze5.jpg', patch_size=(7,7))  
2 img12.run()
```



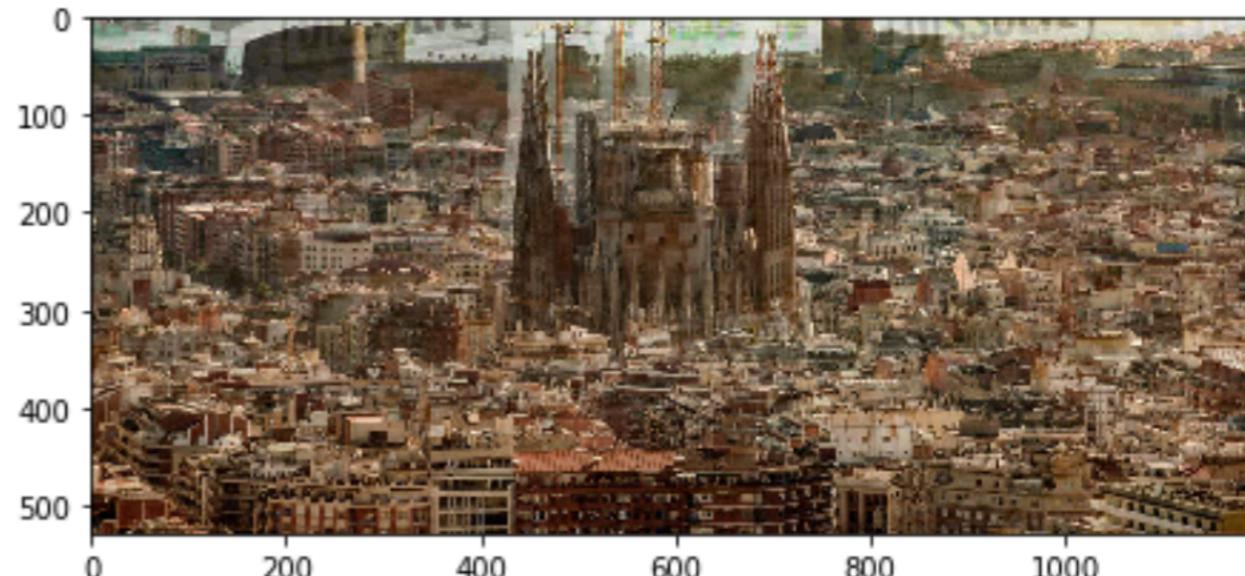
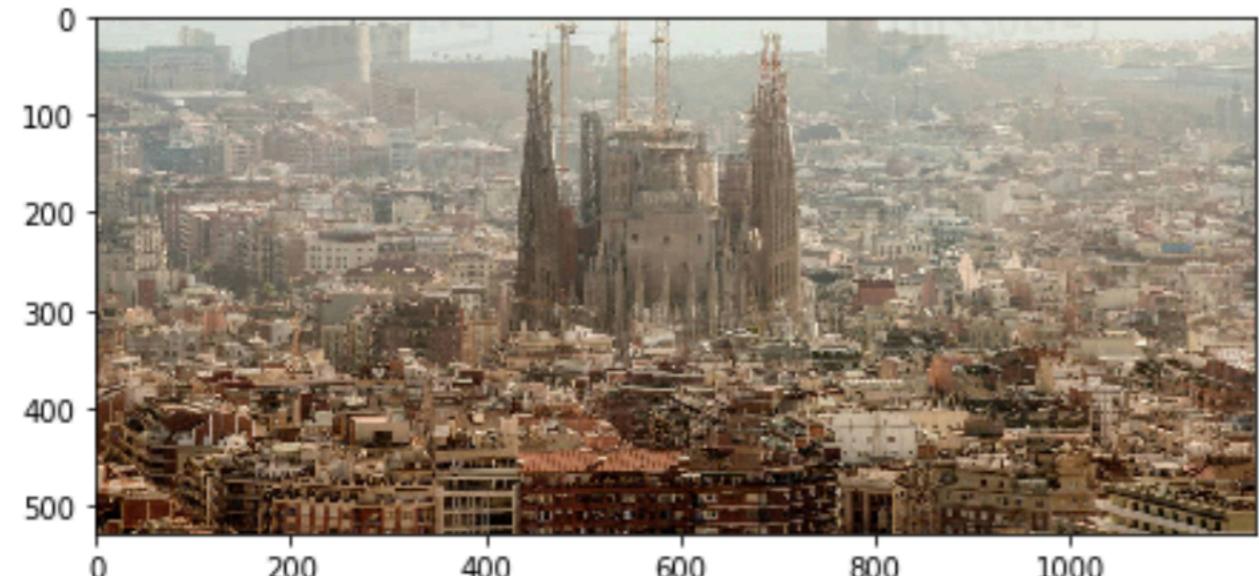
```
1 img12 = Antihaze('haze6.jpg', patch_size=(9,9))  
2 img12.run()
```



```
1 img12 = AntiHaze('haze8.jpg', patch_size=(5,5))  
2 img12.run()
```

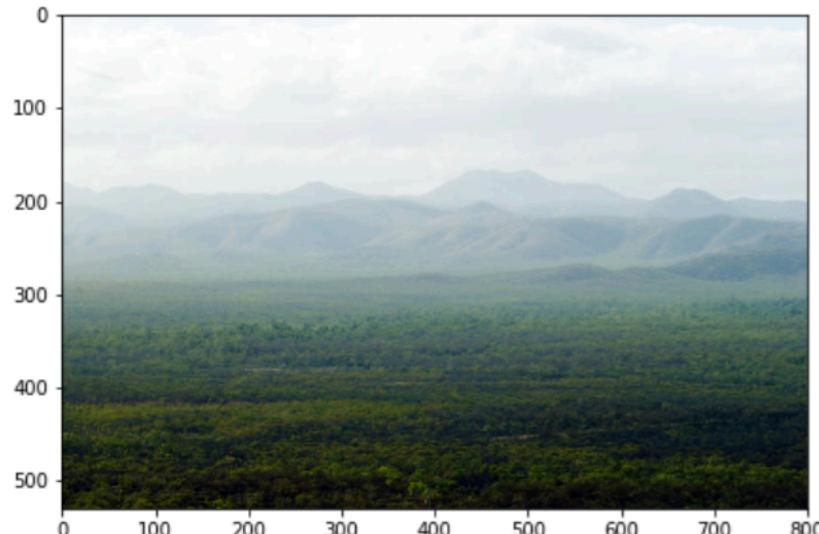


```
1 img12 = AntiHaze('haze9.jpg')  
2 img12.run()
```

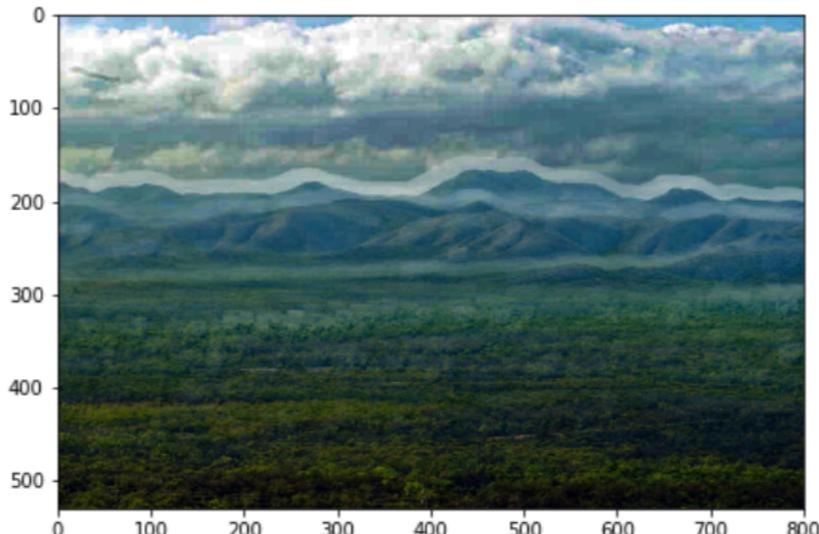


# Patch size

Input image



patch\_size = (15, 15)



patch\_size = (7, 7)

