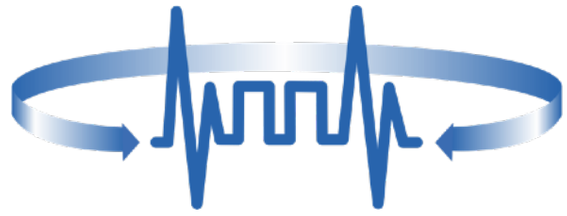
Single Image Haze Removal Using Dark Channel Prior

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

Усеинов Булат
ФУПМ МФТИ
2019

Glossary



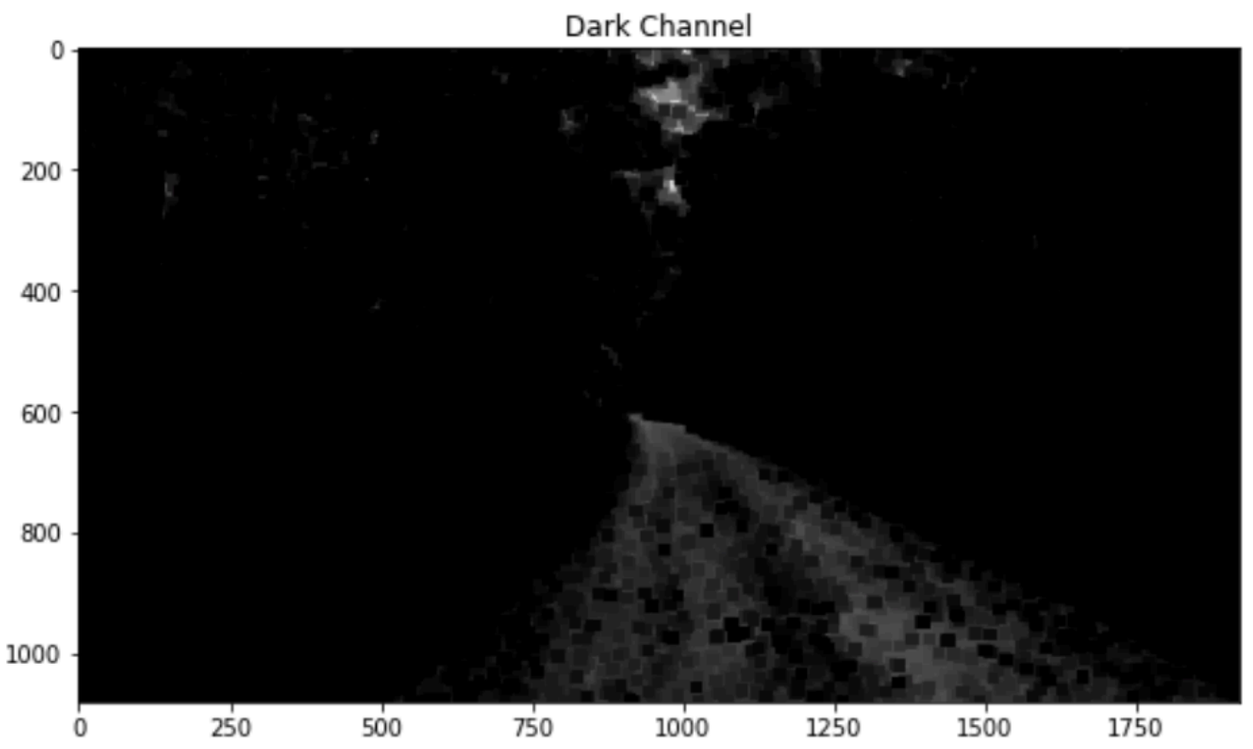
- Airlight - естественное световое загрязнение или засветка атмосферы.

- Radiance - свет, отраженный от объектов в поле зрения.

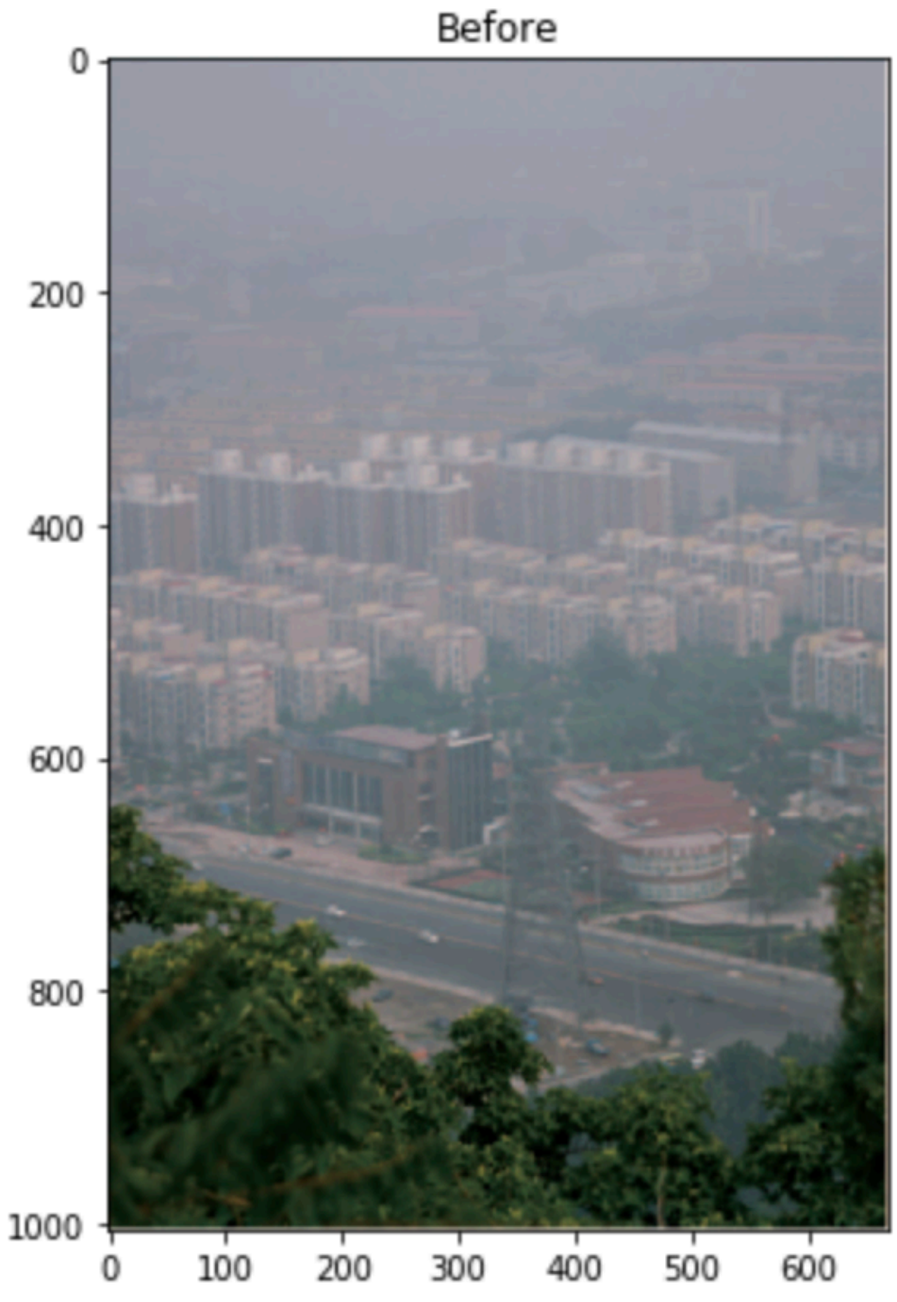
- Transmission - порция света, которая не преломилась и попала в объектив камеры.

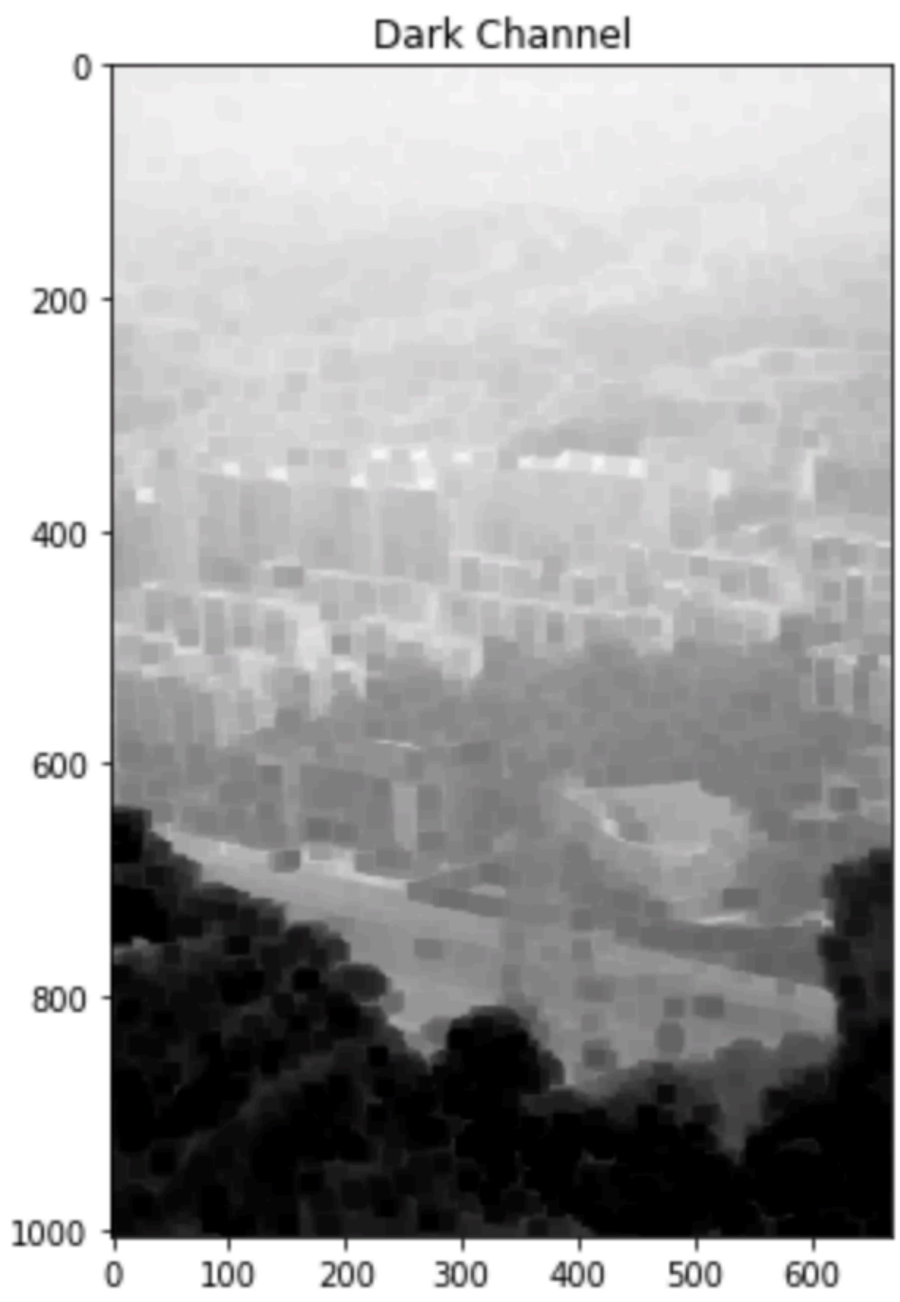
- Patch - область картинки, по которой происходит расчет.

Dark Channel Prior

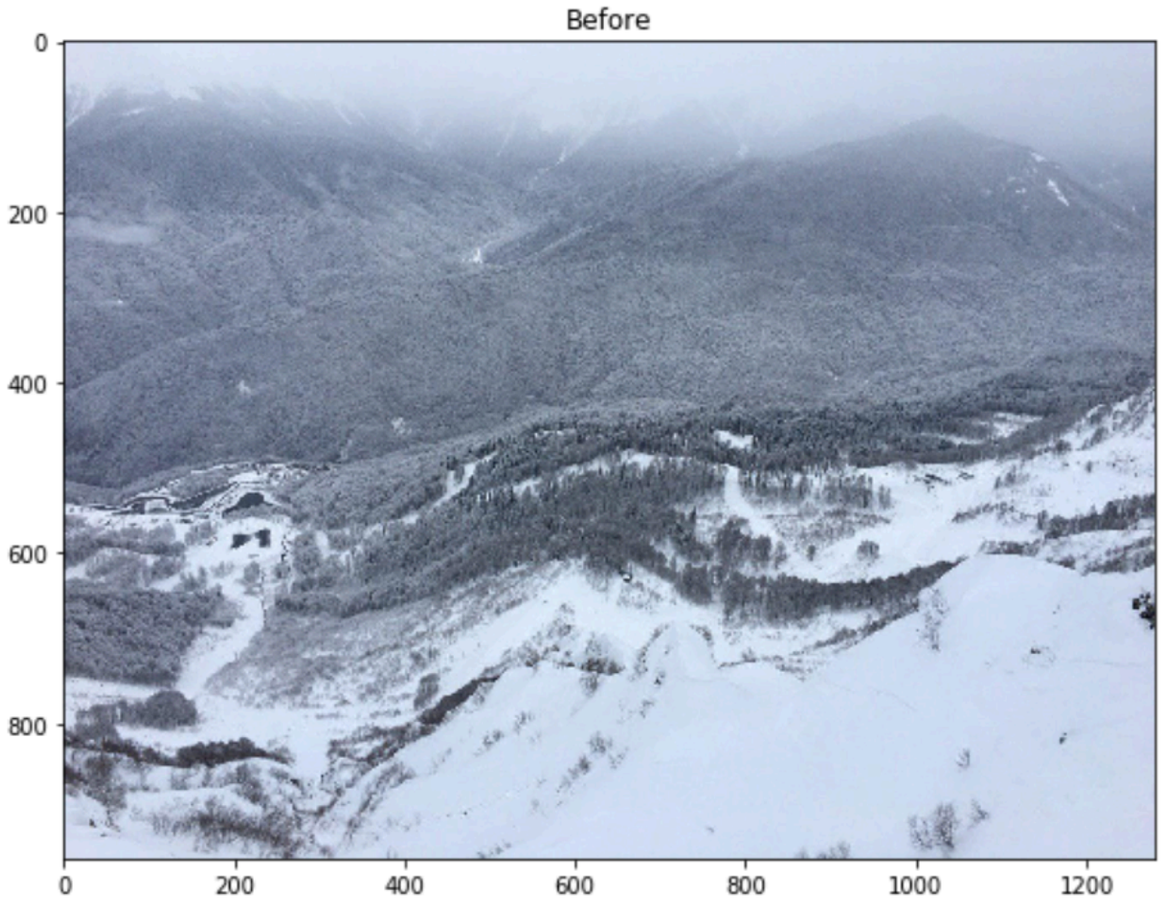


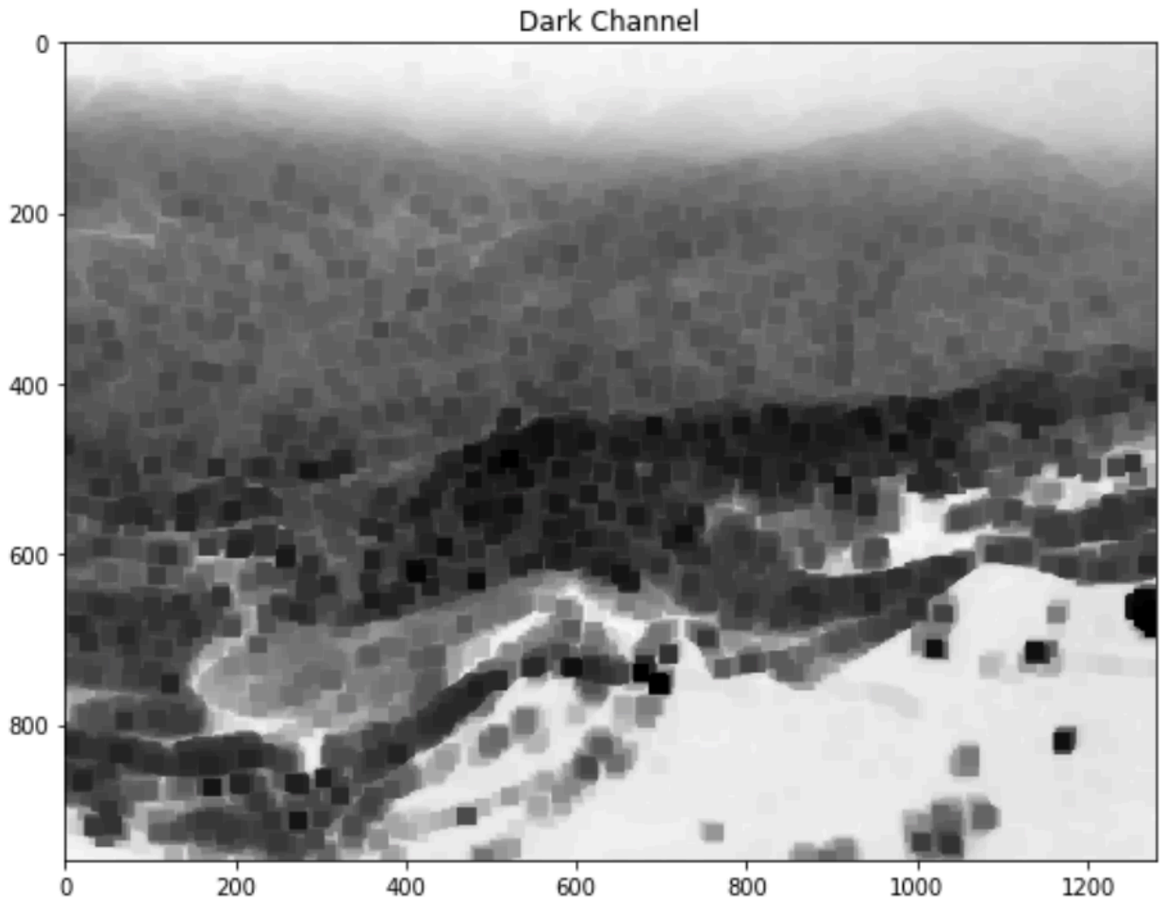
Dark Channel Prior





Dark Channel Prior





Model

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

$I(x)$ – **Input Image**

$J(x)$ – **Scene Radiance**

A – **Atmospheric Light**

$t(x)$ – **Transmission**

$A(1 - t(x))$ – **Airlight**

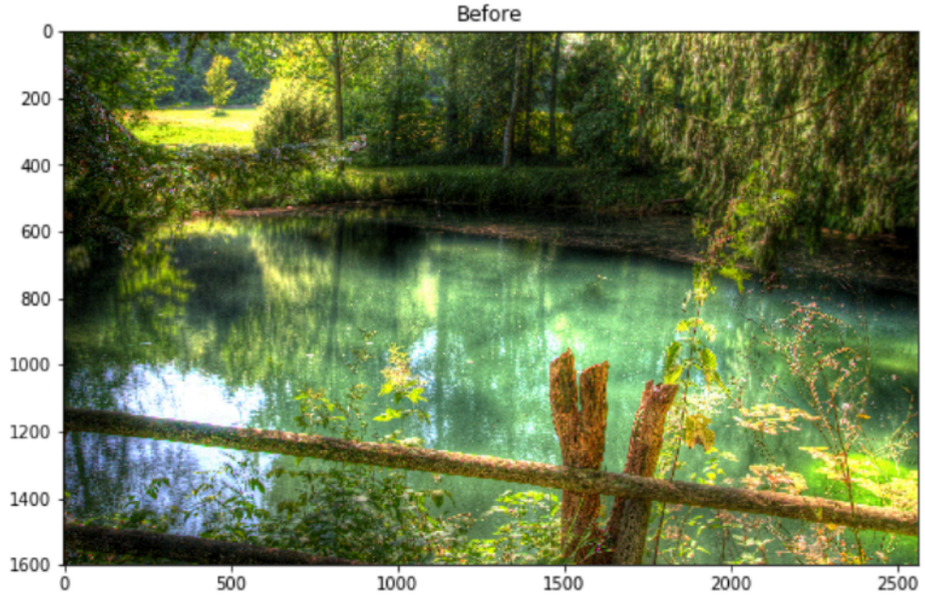
Dark Channel

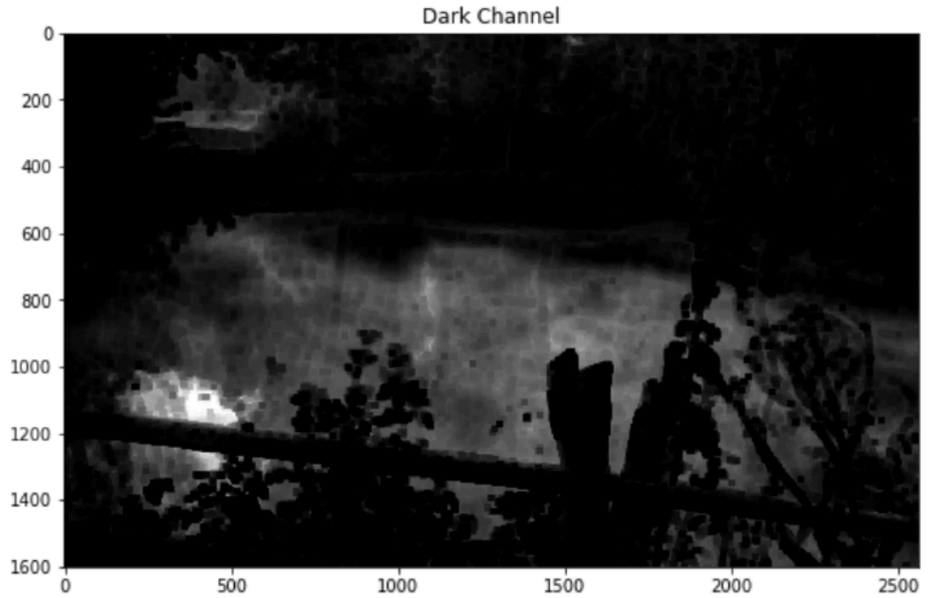
$$J^{dark}(x) = \min_{y \in \Omega(x)} \left(\min_{c \in \{R,G,B\}} J^c(y) \right)$$

$\Omega(x)$ – **local patch centered at** x

Dark Channel Prior

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





Atmospheric Light

$$J(x) = R(x)A$$

$$I(x) = R(x)At(x) + A(1 - t(x))$$

+

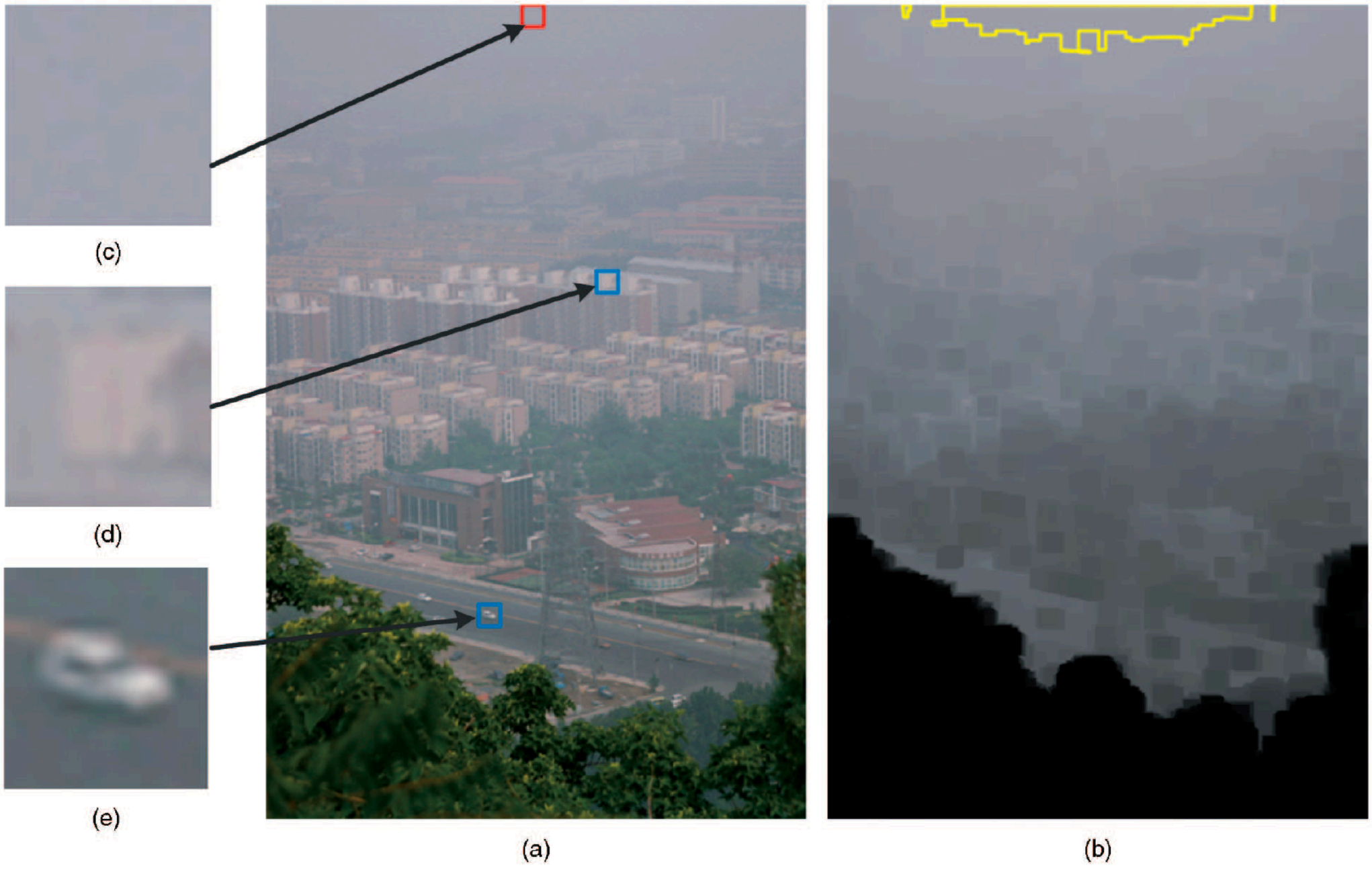
Sunlight (S)

=

$$J(x) = R(x)(A + S)$$

$$\boxed{I(x) = J(x)At(x) + J(x)St(x) + A(1 - t(x))}$$

Atmospheric Light



https://ieeexplore.ieee.org/abstract/document/5567108

Transmission

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

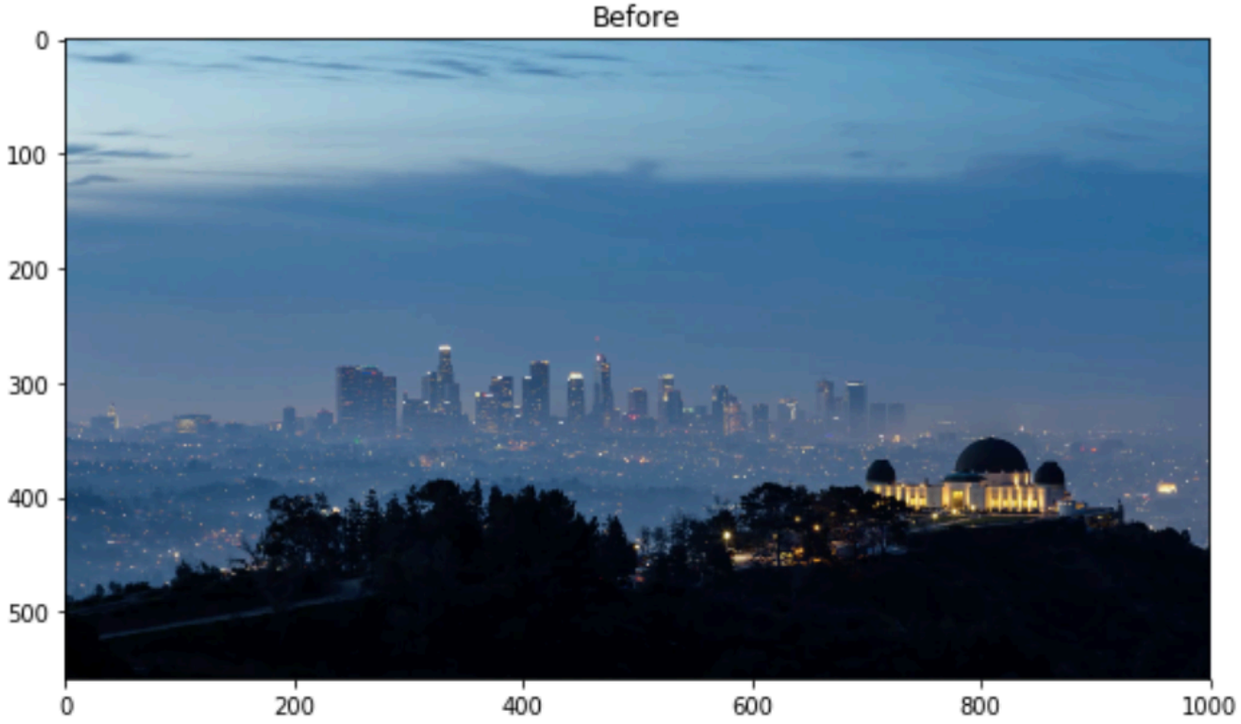
$$\frac{I^c(x)}{A^c} = \frac{J^c(x)}{A^c}t(x) + 1 - t(x)$$

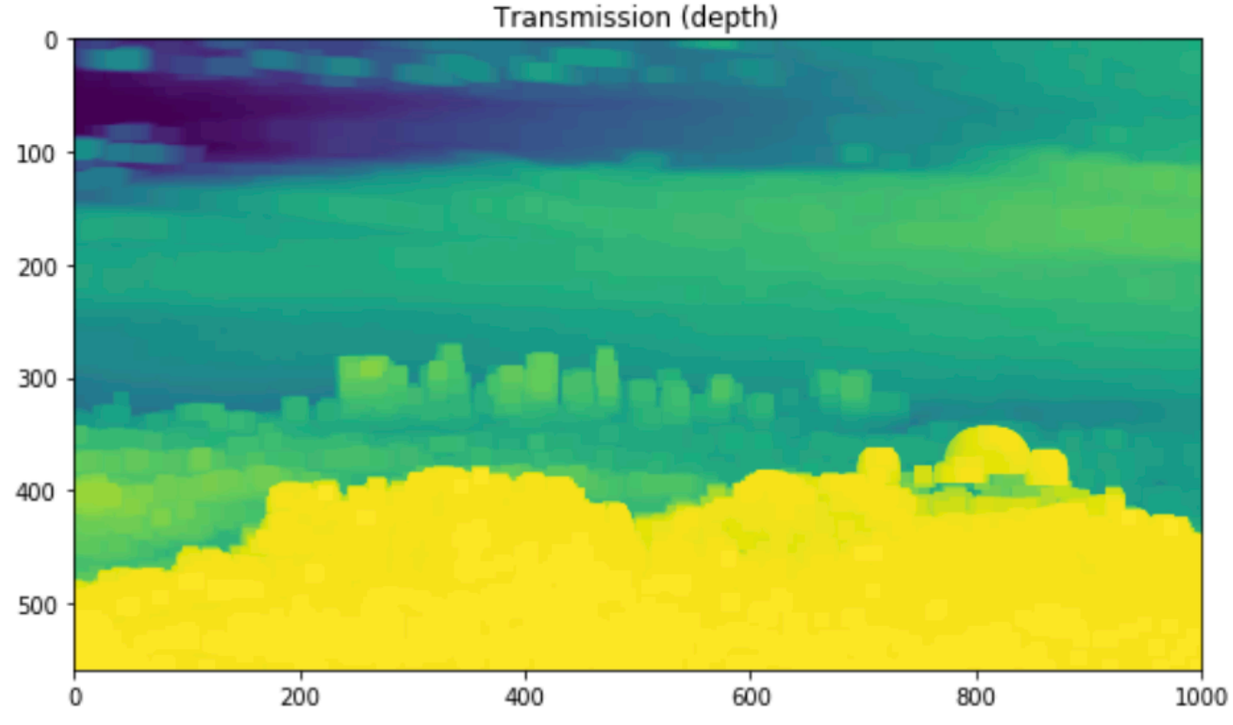
$$\min_{y \in \Omega(x)} \left(\min_{c} \frac{I^c(y)}{A^c}\right) = \bar{t}(x) \min_{y \in \Omega(x)} \left(\min_{c \in \{R,G,B\}} \frac{J^c(y)}{A^c}\right) + 1 - \bar{t}(x)$$

$$\boxed{\bar{t}(x) = 1 - \omega \min_{y \in \Omega(x)} \left(\min_{c} \frac{I^c(x)}{A^c}\right)}$$

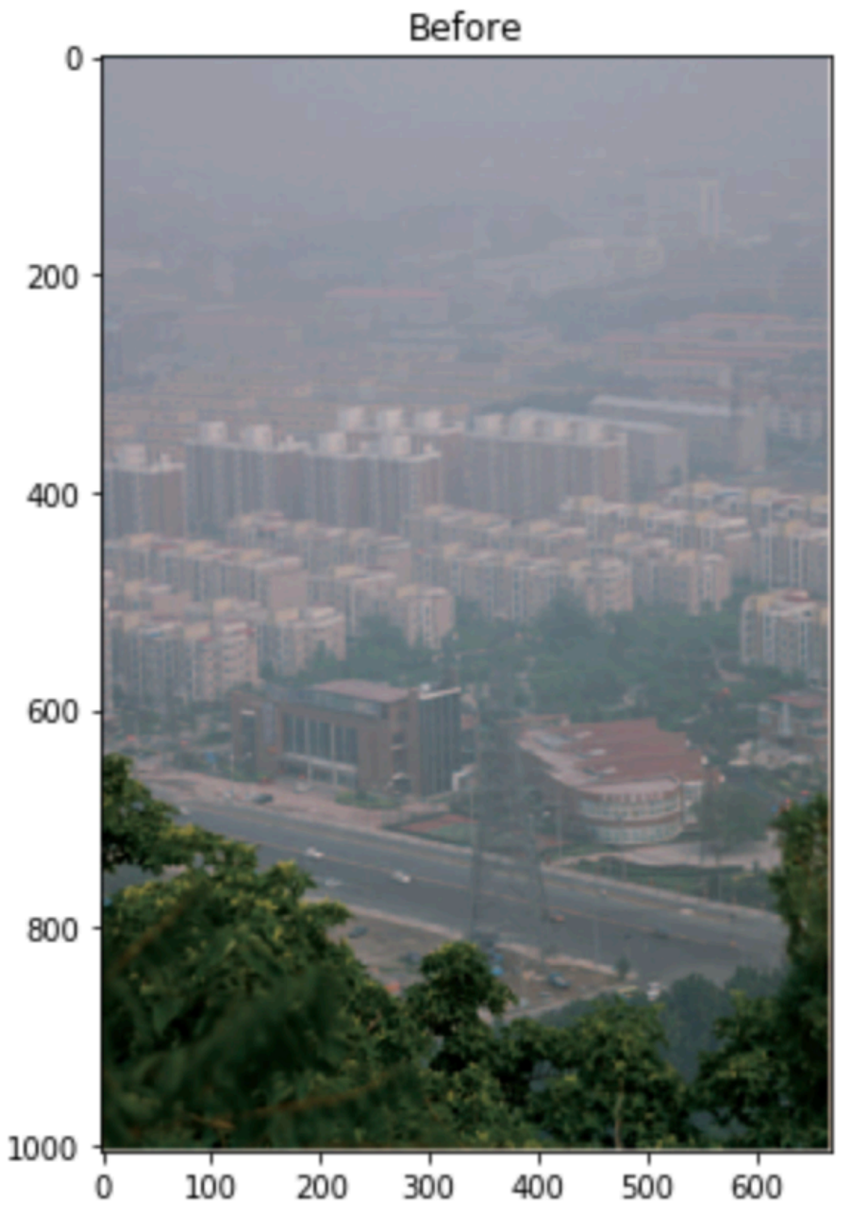
Transmission

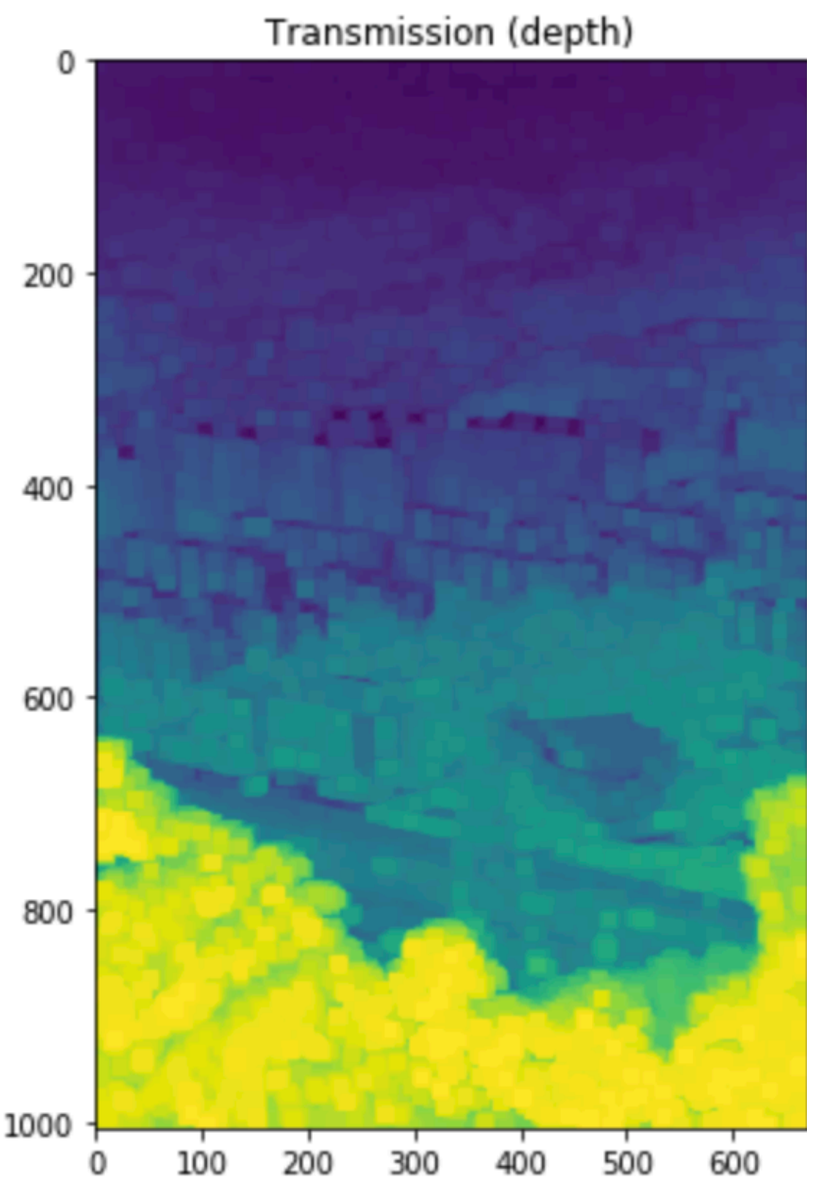
$$t(x) = e^{-\beta d(x)}$$





Transmission



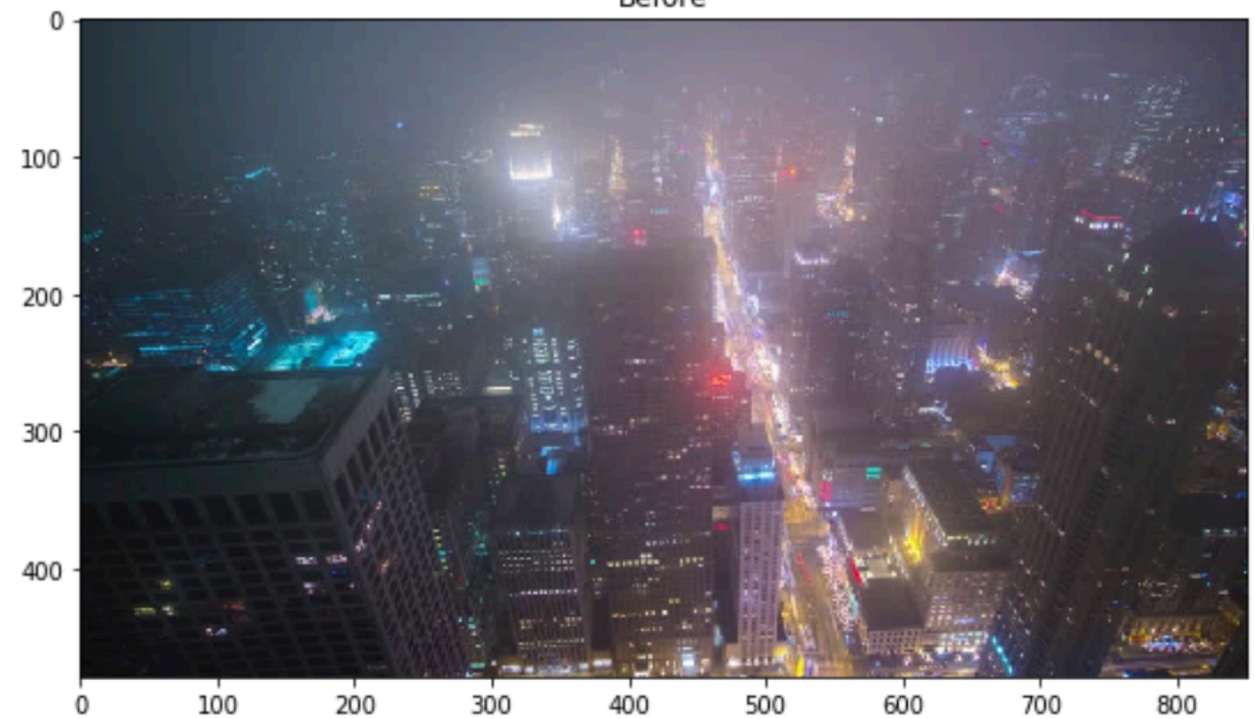


Scene Radiance

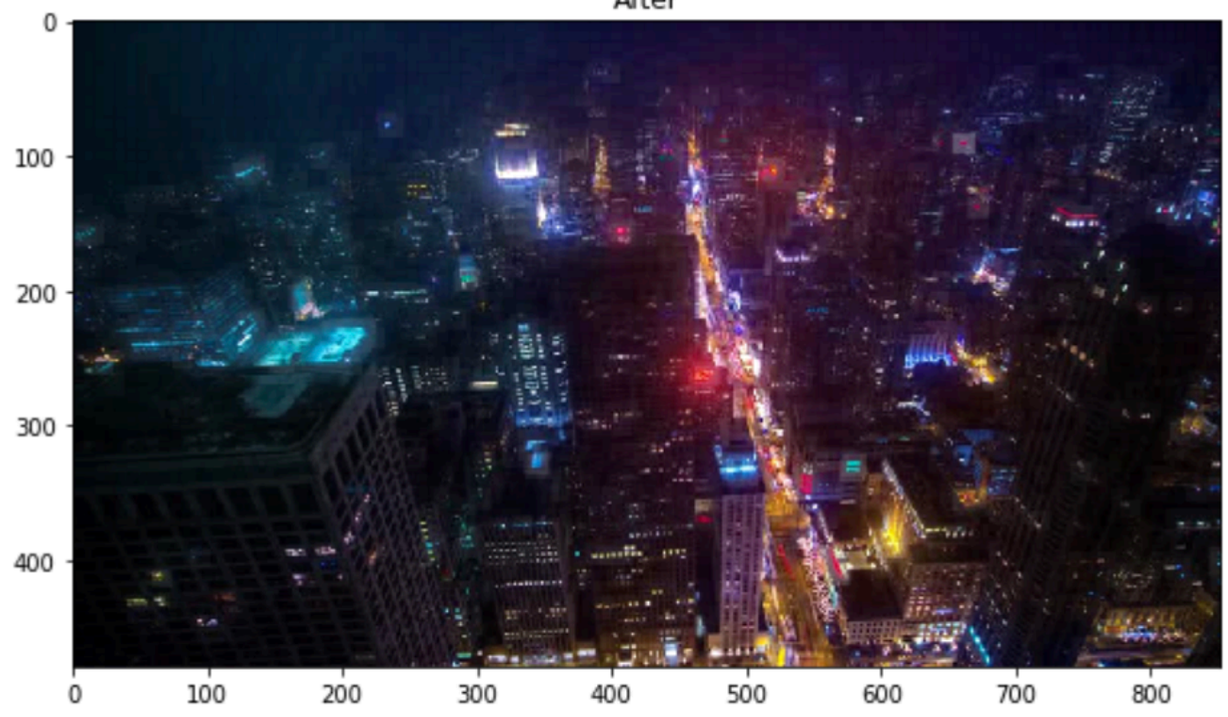
$$I(x) = J(x)t(x) + A(1 - t(x))$$

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

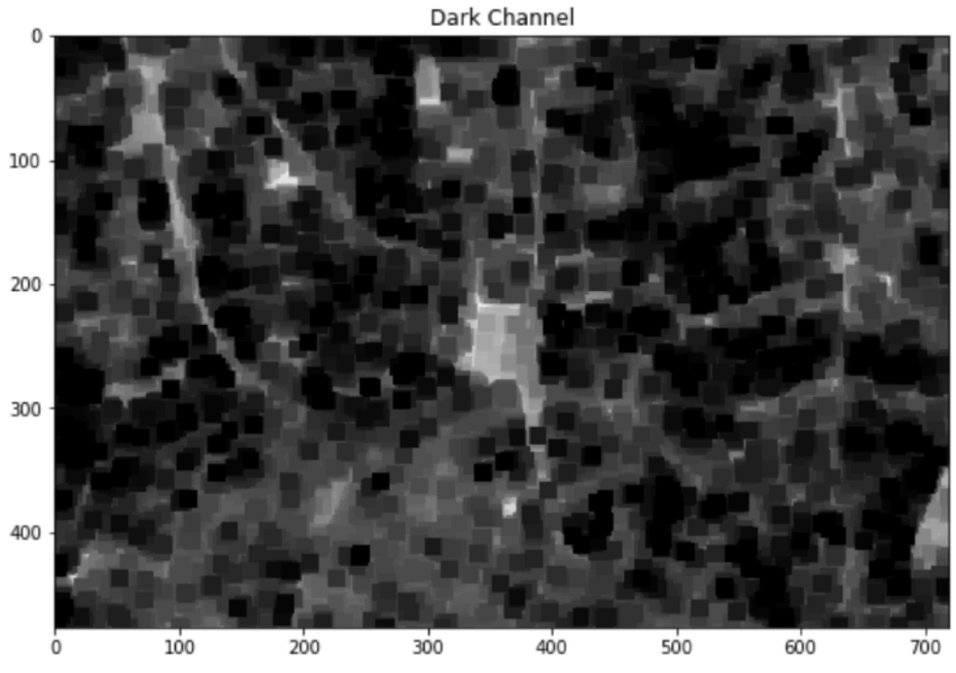
Before

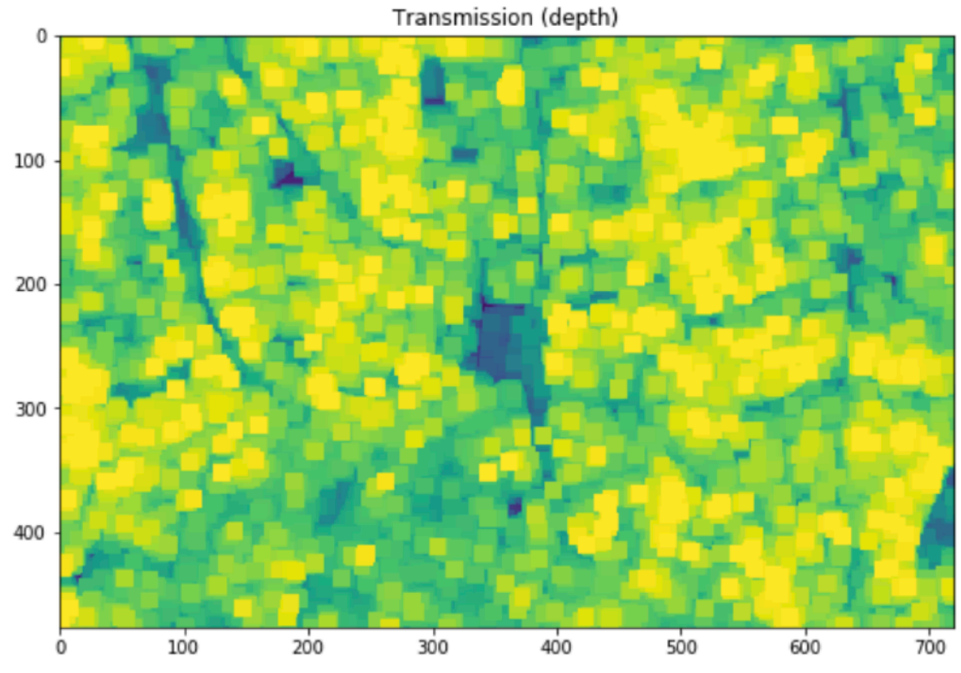


After

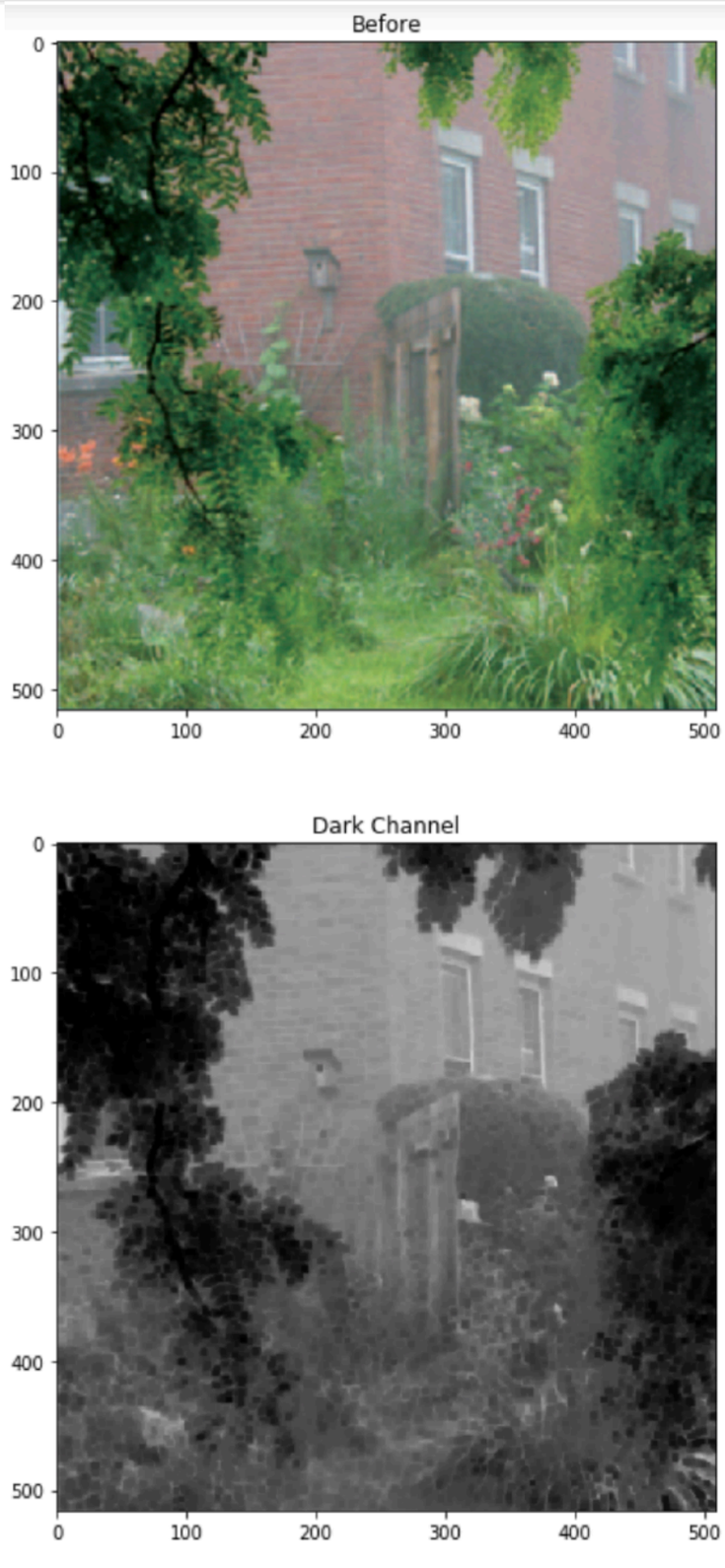


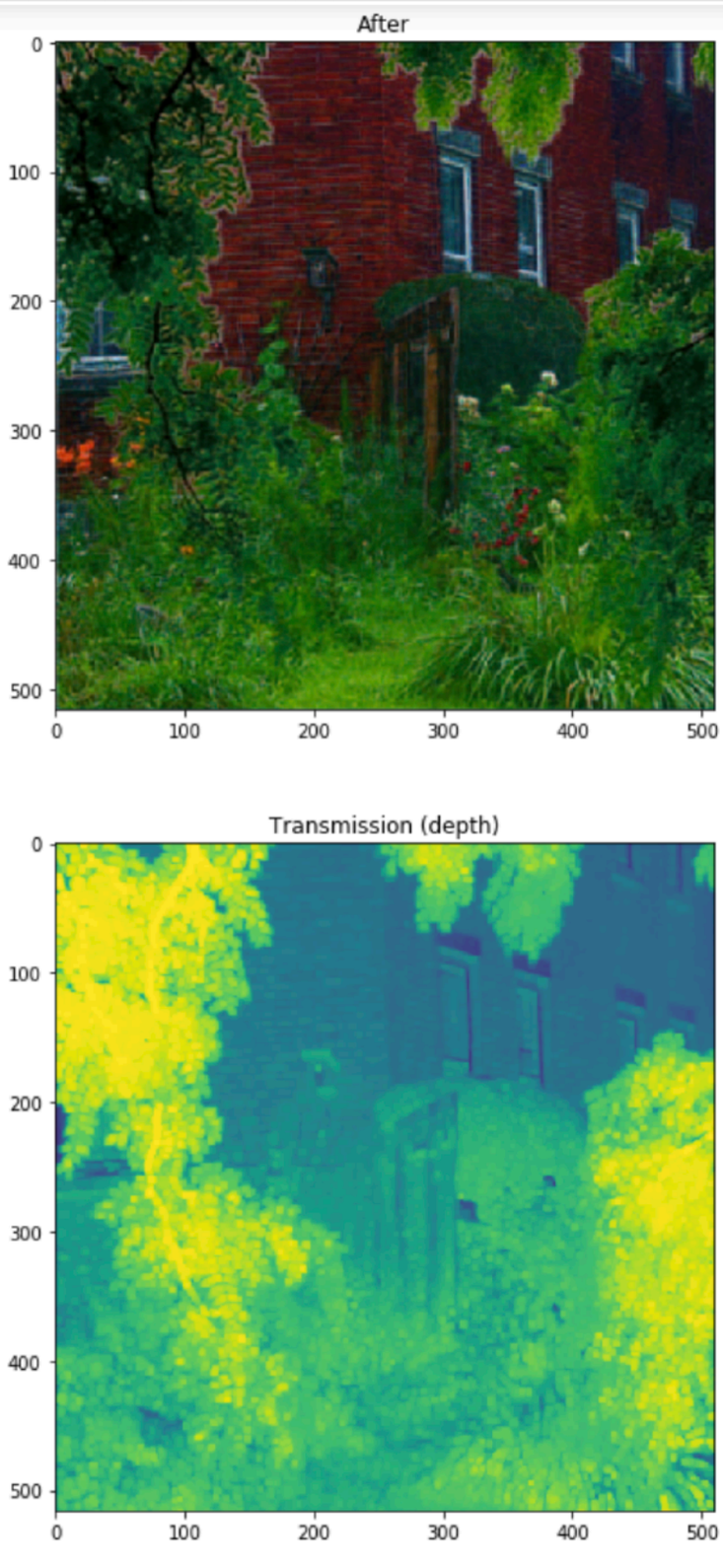
Scene Radiance



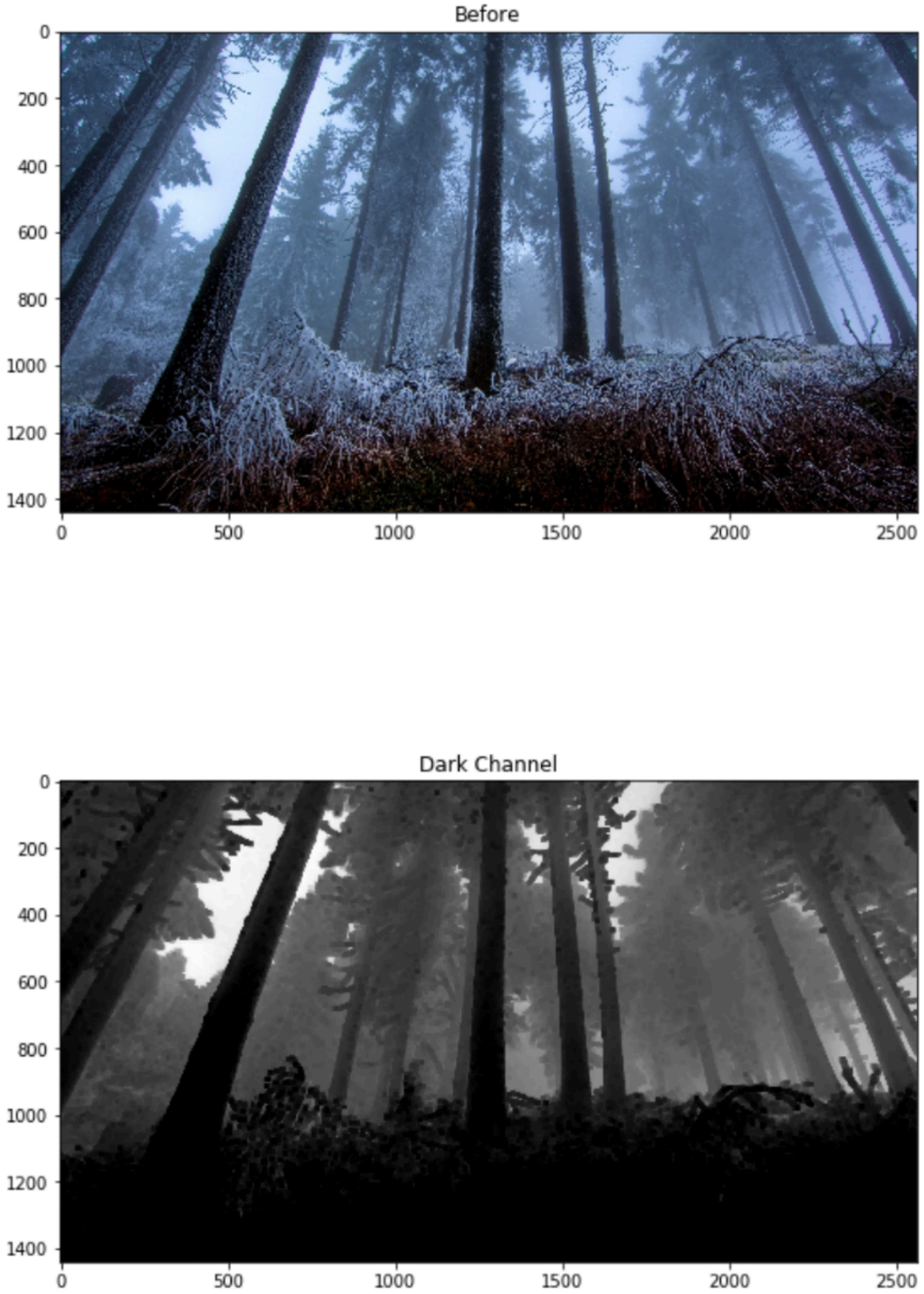


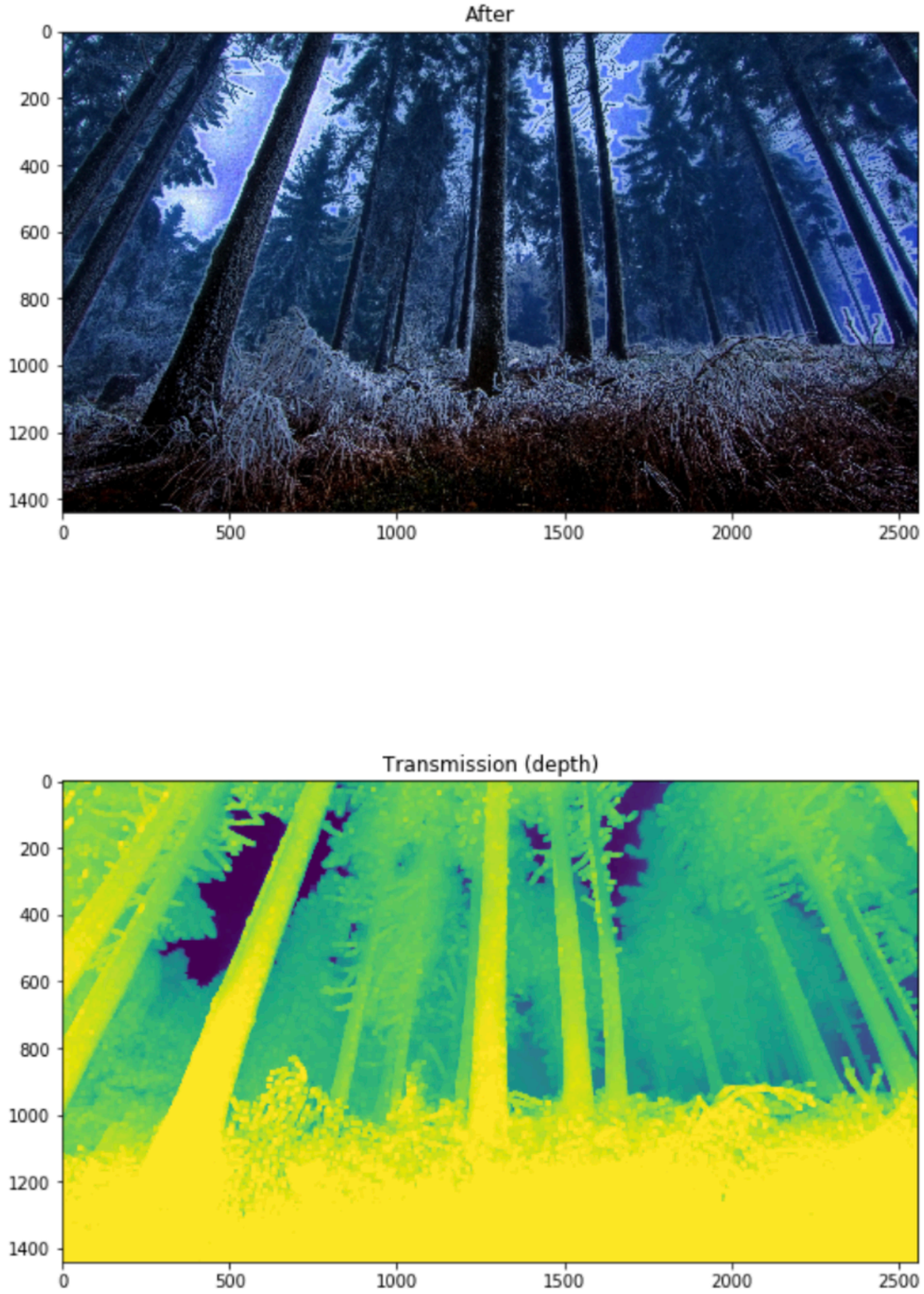
Scene Radiance





Scene Radiance

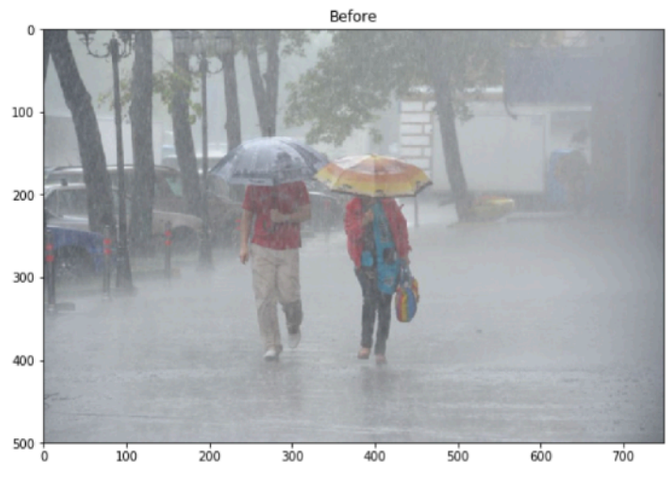


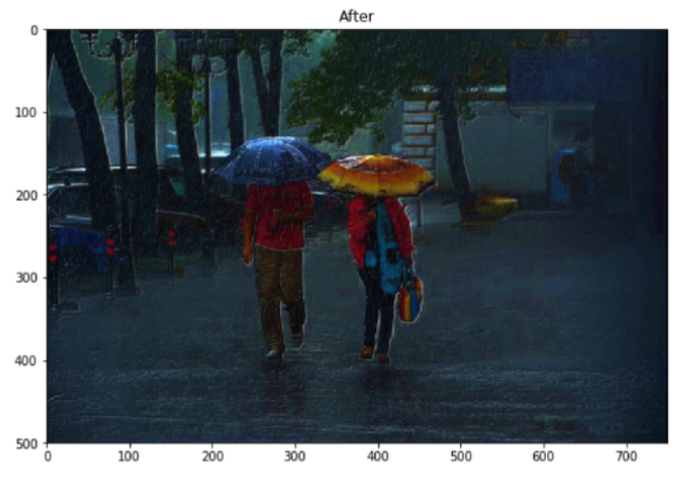


Patchsize

```
%%time
img5 = Img_Haze2remove("5.jpg", patch = 3)
img5.go()
img5.show()
```

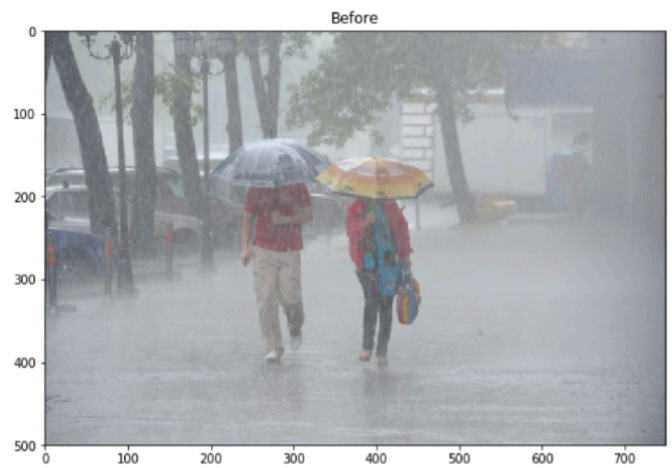
```
CPU times: user 3.76 s, sys: 106 ms, total: 3.87 s
Wall time: 3.01 s
```






```
%%time
img5 = Img_Haze2remove("5.jpg", patch = 7)
img5.go()
img5.show()
```

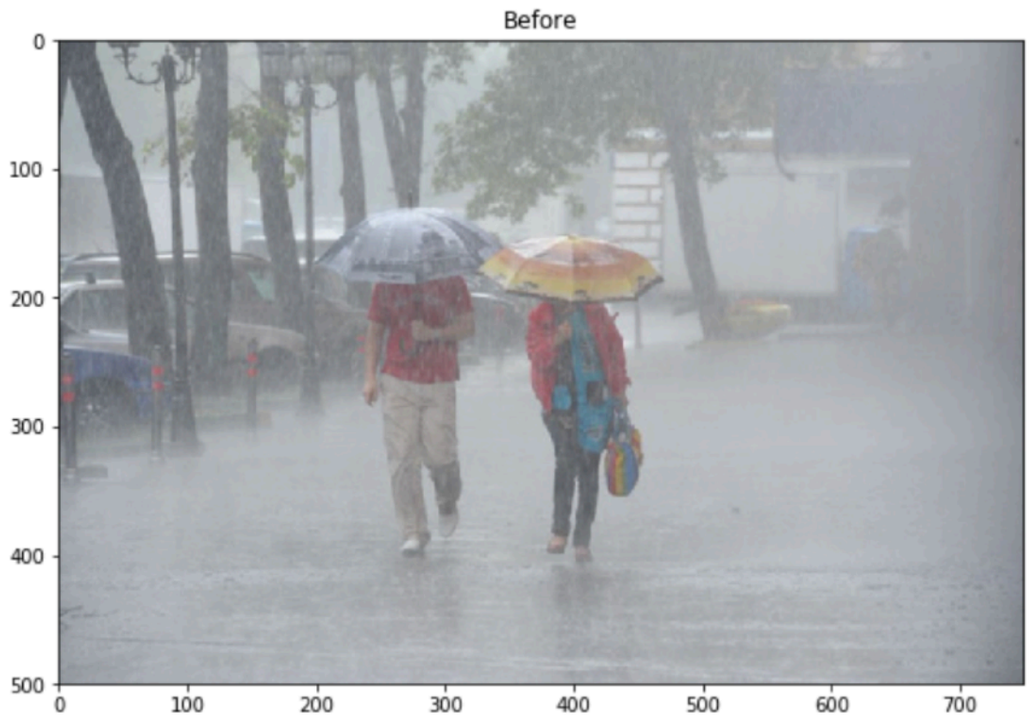
```
CPU times: user 3.87 s, sys: 150 ms, total: 4.02 s
Wall time: 3.11 s
```



Patchsize

```
%%time
img5 = Img_Haze2remove("5.jpg", patch = 15)
img5.go()
img5.show()
```

```
CPU times: user 3.74 s, sys: 110 ms, total: 3.85 s
Wall time: 2.97 s
```

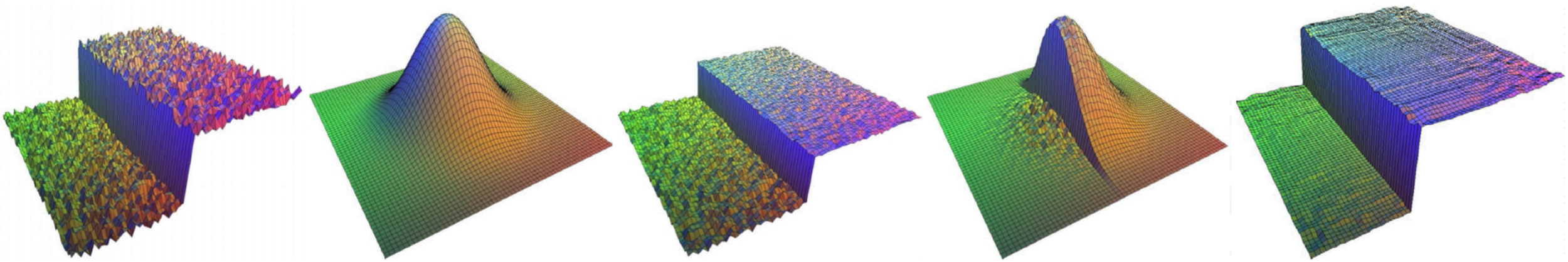


Soft Matting

$$E(t) = t^T L t + \lambda (t - \bar{t})^T (t - \bar{t})$$

$$(L + \lambda U) t = \lambda \bar{t}$$

Bilateral Filter



input | spatial kernel f | influence g in the intensity domain for the central pixel | weight $f \times g$ for the central pixel | output

http://cs.brown.edu/courses/csci1290/labs/lab_bilateral/index.html

Soft Matting

(a) (b) (c) (d) (e)

https://ieeexplore.ieee.org/abstract/document/5567108

Experimental code

Code: https://github.com/bulatuseinov/Haze2Remove