We use dark channel method mentioned in paper to further increase performance of our smoke detection algorism. The pixels in a image often have at least one color RGB channel with very low intensity, in most smoke region, the intensity of dark pixels in a channel are mainly because of air light. So that we can use dark pixels to detected pixels with low transmissions, and those pixels can be marked as smoke pixels. The well known formation of a smoke image is as follow:

where I is observed intensity, J is scene radiance, A is global atmospheric light, and t is refined transmission which describes how much light transfer to camera. As mentioned above, in a nature image, at least one of the color channel has very low intensity so we can get dark channel by calculating the minimum intensity in three channels. We can define the equation as follow:

And we define our getDarkChannel function as follow:

:

After apply this function to original image, we can get a dark channel image where low intensity pixels (smoke pixels and smoke likely pixels) is shows brighter color and other pixels shows dark color as the figure below.

Original image: dark channel:



The low intensity of pixels is mainly caused by colorful object, shadows or surfaces. A smoke region has neither of characteristics above, so it always brighter than non-smoke region due to the additive air light.If we simplify the smoke image equation, we will get the equation of transmission as follow:

In order to calculate the transmission we need to calculate the global atmospheric light first, we store each pixels’ position and size in to a list then sort to get the brightest 0.1% pixels, we look back into original image with same pixels position and return the highest value as atmospheric. The function getAtomsLight design as below:

define getAtomsLight(image, dark channel)

list = dark channel pixels

list = sorted list \* 0.1

for pixels in list:

search atmospheric light in original image with same position

return highest value

The final step is to calculate transmission using the equation above, we design the function as below:

Define transmission(image, A)

T = 1- omega \* dark channel

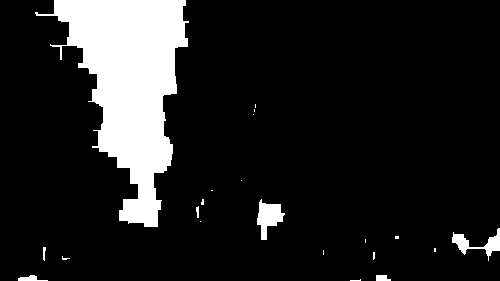
For every T< 0.1 :

T = 0.1

# T = guided\_filter(T, image)

We get the transmission directly from equation or apply guided filter to smooth with original image for a better accuracy but a slower run time. We can compare each pixels’ transmission with our threshold and mark pixels as smoke pixel if transmission is smaller than threshold. The below result is using threshold = 0.4:

Original image: smoke detection without guided filter:



smoke detection with guided filter:

