Computer Vision

# Mid-Project Submission

## Reflection Removal using Ghosting Cues Github link

Anoushka Vyas(20171057), Adhithya Arun(20171066), Meher Shashwat Nigam(20171062)

### Aim

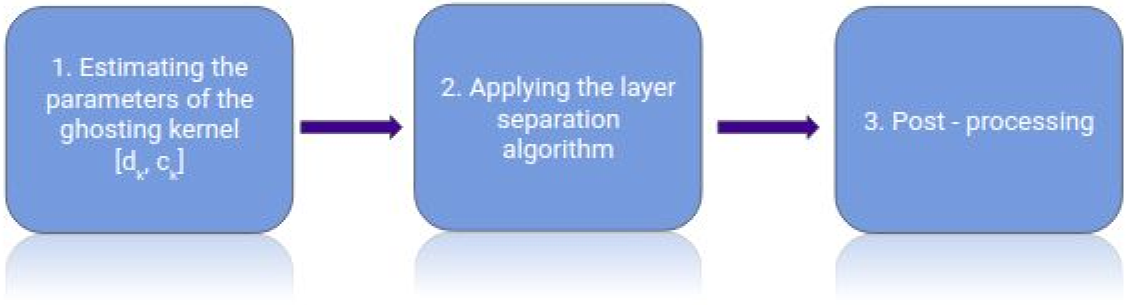
Images taken through glass windows often contain undesirable reflection artifacts which ruin the image. In this project, the original image is considered to be composed of a reflection layer (undesirable) and a transmission layer (desirable).

$$I = T + R \otimes k + n$$

where I is Original Image, T is Transmission layer, R is Reflection layer, k is two-pulse kernel, n is additive Gaussian noise.

The original image is modeled as a mixture of these layers and the desirable image component is recovered after removing the undesired reflection layer.

### Workflow


1. Estimating the parameters of the ghosting kernel [$d_k$, $c_k$]
2. Applying the layer separation algorithm
3. Post - processing

## Implementation done : Estimation of Kernel Parameters dx,dy,c

The ghosting convolution kernel k, is parameterized by a spatial shift vector, $d_k$ and an attenuation factor, $c_k$ between the primary reflection and secondary reflection.

The spatial shift vector is estimated using the autocorrelation map of the laplacian of the input image. The shifted copies of the reflection layer create local maximum on the autocorrelation map. After some processing, the largest local maxima is considered as the spatial shift vector.

The attenuation factor is calculated using the spatial shift vector. Interest points are detected from the input image using Haris Corner detector. A 5x5 normalized contrast patch was extracted from each region of a corner feature. Patches that have a strong correlation with patches at spatial offset dk  are assumed to be due to either of the reflection layers. Attenuation between a pair of matching patches is calculated as :

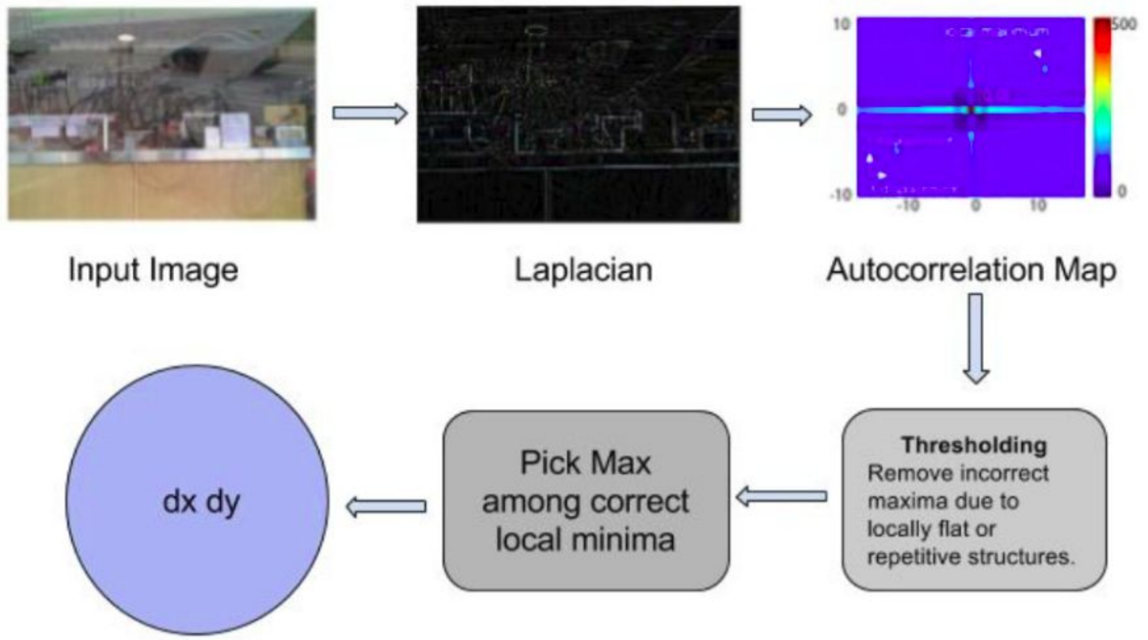
$$a_{ij} = \sqrt{\frac{var[p_i]}{var[p_j]}}$$

And $\boldsymbol{c_k}$ is given as

$$c_k = \frac{1}{Z}\sum_{ij} w_{ij} a_{ij} \quad \text{where } w_{ij} = e^{-\frac{\| p_i - p_j \|^2}{2\theta^2}}$$

Here, ,$p_i$ and $p_j$ are patches and Θ=0.2.

## Estimating Kernel Parameters (spatial offset)



Input Image

Laplacian

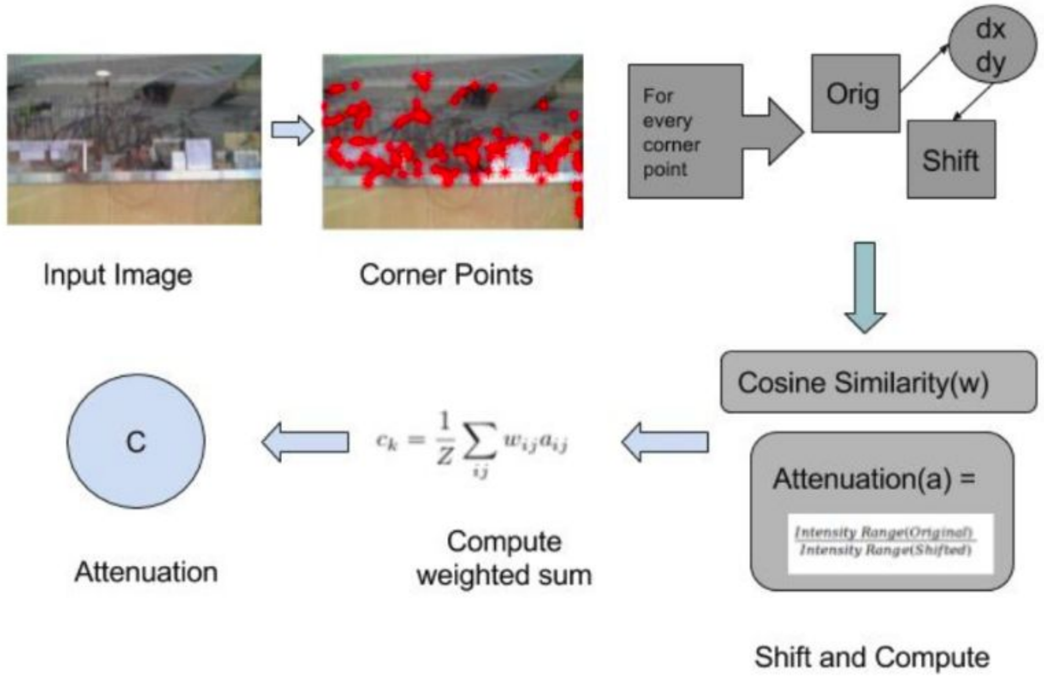
Autocorrelation Map

**Thresholding**
Remove incorrect maxima due to locally flat or repetitive structures.

Pick Max among correct local minima

dx dy

## Estimating Kernel Parameters (atten factor)



Input Image

Corner Points

For every corner point

Orig

dx dy

Shift

Cosine Similarity(w)

Attenuation(a) = $\frac{\textit{Intensity Range(Original)}}{\textit{Intensity Range(Shifted)}}$

Shift and Compute

$c_k = \frac{1}{Z}\sum_{ij} w_{ij}a_{ij}$

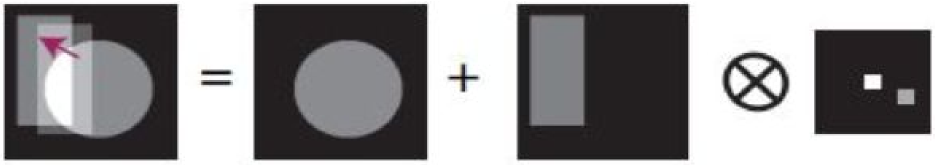
Compute weighted sum

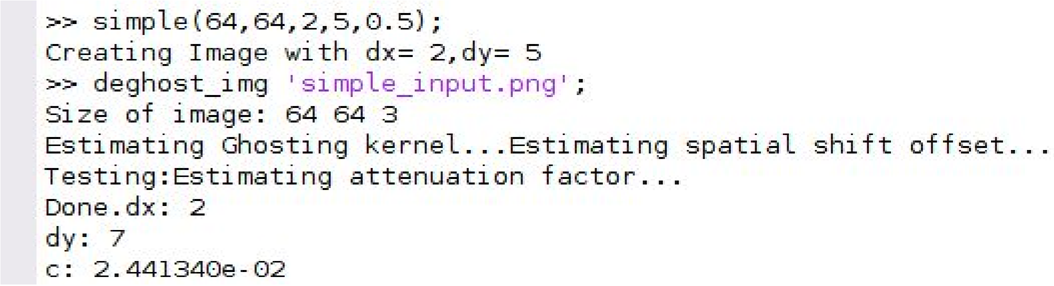
C

Attenuation

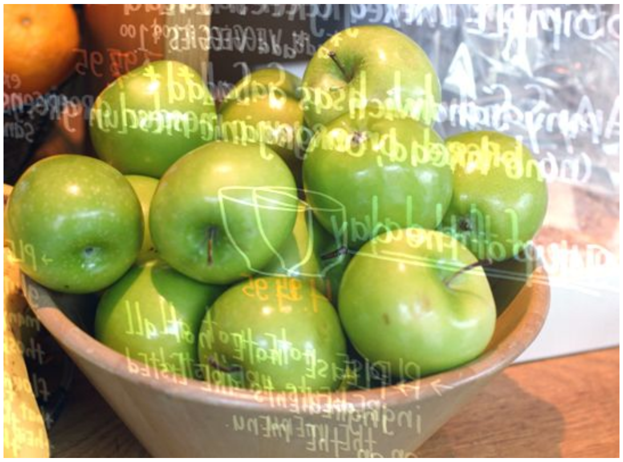
## Results on test images

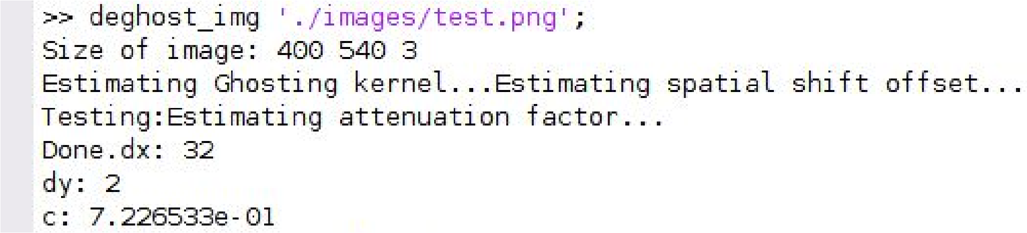
$$I = T + R \otimes k$$





```
>> simple(64,64,2,5,0.5);
Creating Image with dx= 2,dy= 5
>> deghost_img 'simple_input.png';
Size of image: 64 64 3
Estimating Ghosting kernel...Estimating spatial shift offset...
Testing:Estimating attenuation factor...
Done.dx: 2
dy: 7
c: 2.441340e-02
```





```
>> deghost_img './images/test.png';
Size of image: 400 540 3
Estimating Ghosting kernel...Estimating spatial shift offset...
Testing:Estimating attenuation factor...
Done.dx: 32
dy: 2
c: 7.226533e-01
```