

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



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 Harris 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 d_k are assumed to be due to either of the reflection layers. Attenuation between a pair of matching patches is calculated as :

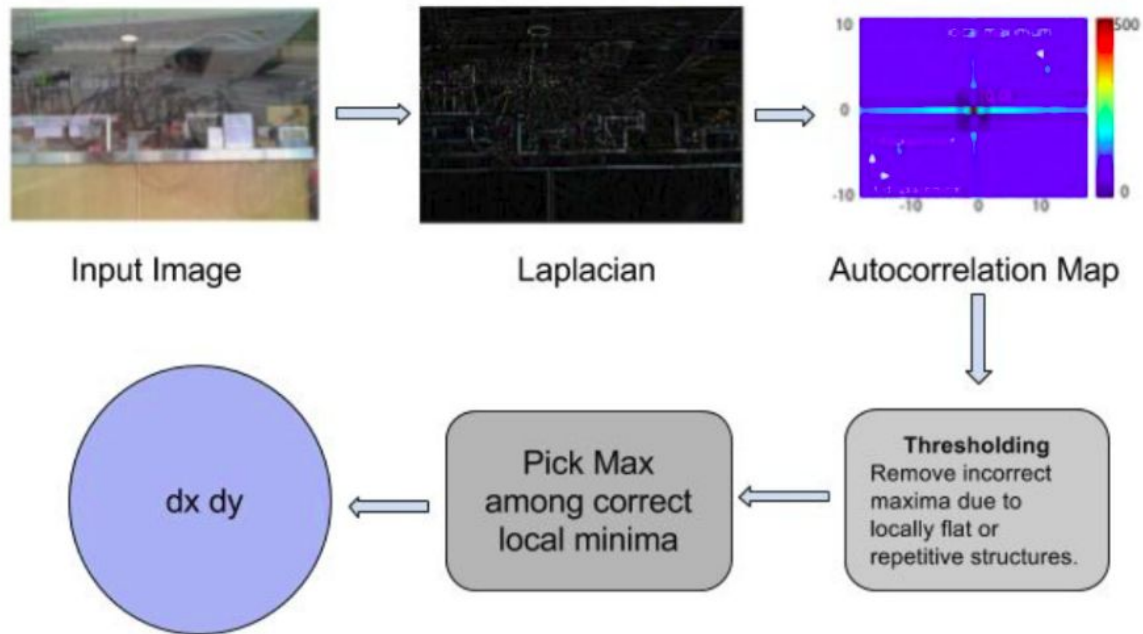
$$a_{ij} = \sqrt{\frac{\text{var}[p_i]}{\text{var}[p_j]}}$$

And c_k is given as

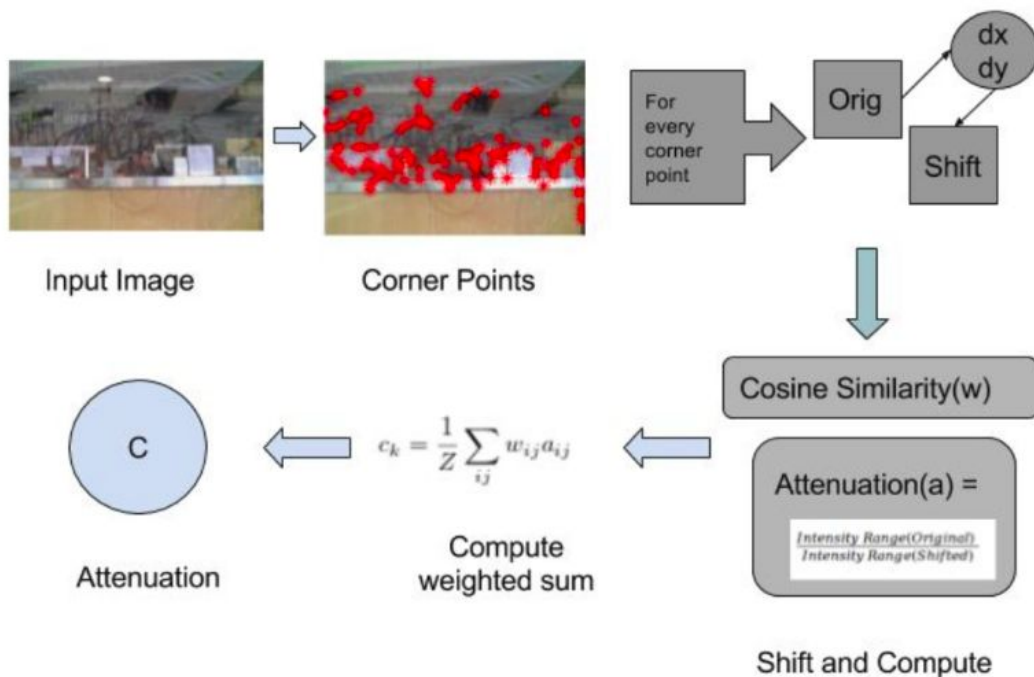
$$c_k = \frac{1}{Z} \sum_{ij} w_{ij} a_{ij} \quad \text{where } w_{ij} = e^{-\frac{\|p_i - p_j\|^2}{2\theta^2}}$$

Here, p_i and p_j are patches and $\theta=0.2$.

Estimating Kernel Parameters (spatial offset)



Estimating Kernel Parameters (atten factor)



Results on test images

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



```
>> simple(64,64,4,6,0.7)
Creating Image with dx= 4,dy= 6
>> deghost_img 'simple_input.png'
Size of image: 64 64 3
Estimating Ghosting kernel...Estimating spatial shift offset.
Estimating attenuation factor...
Done.dx: 2
dy: 8
c: 6.951574e-01
```



```
>> deghost_img 'images/test.png'
Size of image: 400 540 3
Estimating Ghosting kernel...Estimating spatial shift offset...
Estimating attenuation factor...
Done.dx: 32
dy: 2
c: 9.334746e-01
```