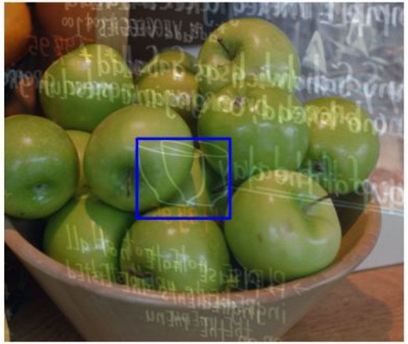
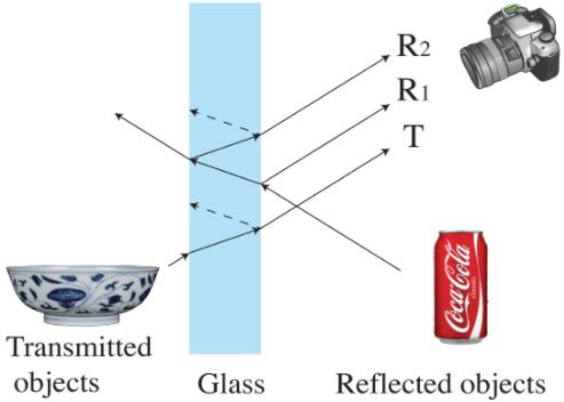
Reflection Removal using Ghosting Cues

Team : Framed

Problem Statement Overview

The primary goal of the project is to implement an algorithm that performs post-processing on the images to remove reflection artifacts. The important property that is leveraged here is the ghosting cues that arise from double shifted reflections of the reflected scene off the glass surface.





How do these cues occur?

Many a times images that are taken through glasses surfaces have some unwanted artifacts in the final image. These arise due to the reflections of the scene on the same side of the glass as the camera.

Visual Examples



Input Image = Transmission Layer + Reflection Layer $\otimes$ K



Estimating K

In order to remove the ghosting effects of the reflections, a ghosting kernel is used.

This kernel has two variables c_k *and* d_k. The result of convolving the input image with the kernel ***K*** is given by the formula :

- **d_k** represents the spatial shift : it depends on glass thickness, camera focal length etc.
- **c_k** is the attenuation factor that is affected by wave optics.

$$Y_i = X_i + c_k X_{i-d_k}$$

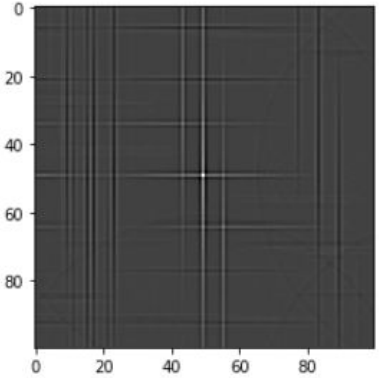
Estimating D_k :

We first compute the 2D - autocorrelation map of the *laplacian* of the input image.

Autocorrelation refers to how similar one image is with respect to another, in terms of its structure. This can be obtained by sliding one image over the other and thus computing the similarity values.

The maxima obtained on this map would be an indication of a possible candidate for the spatial shift factor $\boldsymbol{d_k}$.

Since the map for an image is obtained with itself, ∃ two values for the spatial shift where the maxima occurs.



Estimating C_k

- We first use Harris Corner detector to get the set of interest points from the image (in this case, corners).
- Around each of these interest points, we extract a 5x5 patch. With respect to this patch, we extract another patch at an offset of $\boldsymbol{d_k}$. If there is a strong correlation between these two patches, we assume that the edges are due to the reflection layer. We denote the patches as $\boldsymbol{p_i}$ and $\boldsymbol{p_j}$
- We then calculate $\boldsymbol{a_{ij}}$ as:

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

- $\boldsymbol{W_{ij}}$ as

$$e^{-\frac{\| p_i - p_j \|^2}{2\theta^2}}$$

- For all $\boldsymbol{a_{ij}}$ <1, we calculate $\boldsymbol{c_k}$. as:

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

Gaussian Mixture Model

- We model the layer separation as an Optimization problem which consists of minimizing the Reconstruction Cost and Maximizing the Patch Priors of the Reflection and Transmission layers subject to the non negativity constraints. GMM Priors are used to regularize the inference.

- GMM framework is based on the assumption that the accumulation of similar patches in a neighborhood are derived from a multivariate Gaussian probability distribution with a specific covariance and mean.

- GMM performs well for image restoration as compared to other methods like MVG, ICA and PCA.

Test Images (Synthetic DataSet)

- We need to generate synthetic data to verify that we are able to obtain correct de-ghosted outputs.

- The paper assumes that each ghosted image is generated as:

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

- Here ***k*** is generated by assuming some $\boldsymbol{c_k}$ and $\mathbf{d_k}$ values. The goal is that our code should give matching $\boldsymbol{c_k}$ and $\mathbf{d_k}$ values when run on these images.

Current Progress

- ❖ Estimation of $\boldsymbol{C_k}$ and $\boldsymbol{D_k}$
- ❖ Understanding the Gaussian Mixture Model
- ❖ Preparation of synthetic dataset

Note: So far we have coded in python. The idea is to get k using Python, then switch to Matlab.

Future Plan & Tasks

- Currently we have decoupled the computation of $\boldsymbol{C_k}$ and $\boldsymbol{D_k}$. Hence we will work on integrating the two in the next tasks.

- Applying the algorithm to images and testing. Further verifying with synthetic dataset.

- Optimisation of the algorithm.