Image Reflection Removal

Using Ghosting Cues

Anoushka Vyas	20171057	ECE
Adhithya Arun	20171066	CSE
Meher Shashwat Nigam	20171062	CSE

TA Mentor Pranay Gupta

Repository [Link]

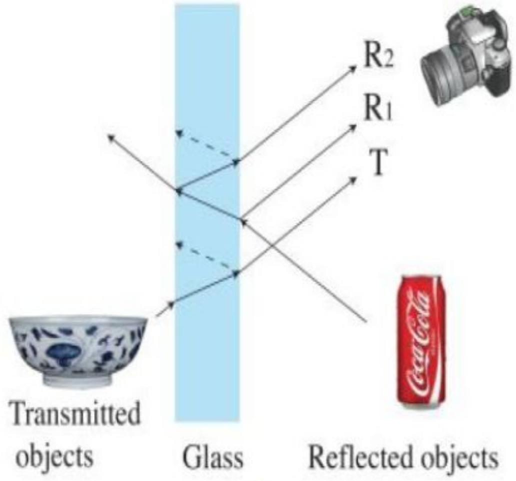
Paper : http://people.csail.mit.edu/yichangshih/mywebsite/reflection.pdf | CVPR 2015

Introduction

When taking a picture through a window pane, undesirable reflections of objects often cause a hindrance to the captured image.

One may try to use different techniques to minimize these reflections but this isn't always possible.

This raises the **need for post-processing** to remove reflection artifacts.

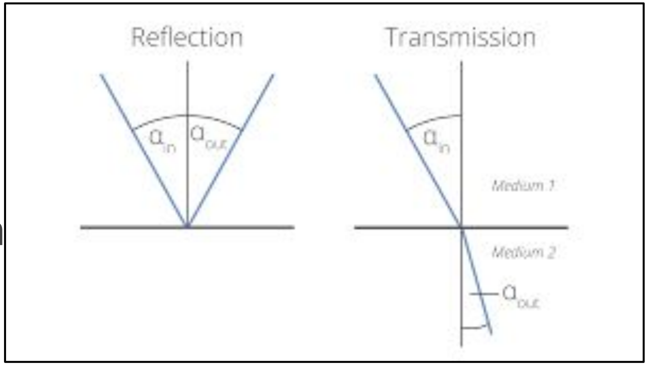


Challenge

Traditional imaging model formulation :

$$I=T+R$$

As both **T and R are natural images** and appear with the same statistical properties, separating them is an ill-posed problem.



What are natural images?

Given some image I with dimensions (w, h, c), there are several possible images within this image space (all random combinations). **Natural image prior is a subset of those images that look “natural” and not like random noise.**
"Natural images" is essentially used as shorthand for "images which have a rich local covariance structure."

(a) A likely image (b) A less likely image (c) An unlikely image

Figure 1.1: An ideal statistical model of natural images would give high likelihood values to images of a natural source, and low likelihood values to other images.

Key Idea Used

Break the symmetry between transmission and reflection layers using the concept of ‘Ghosting cues’.

What’s Ghosting ?

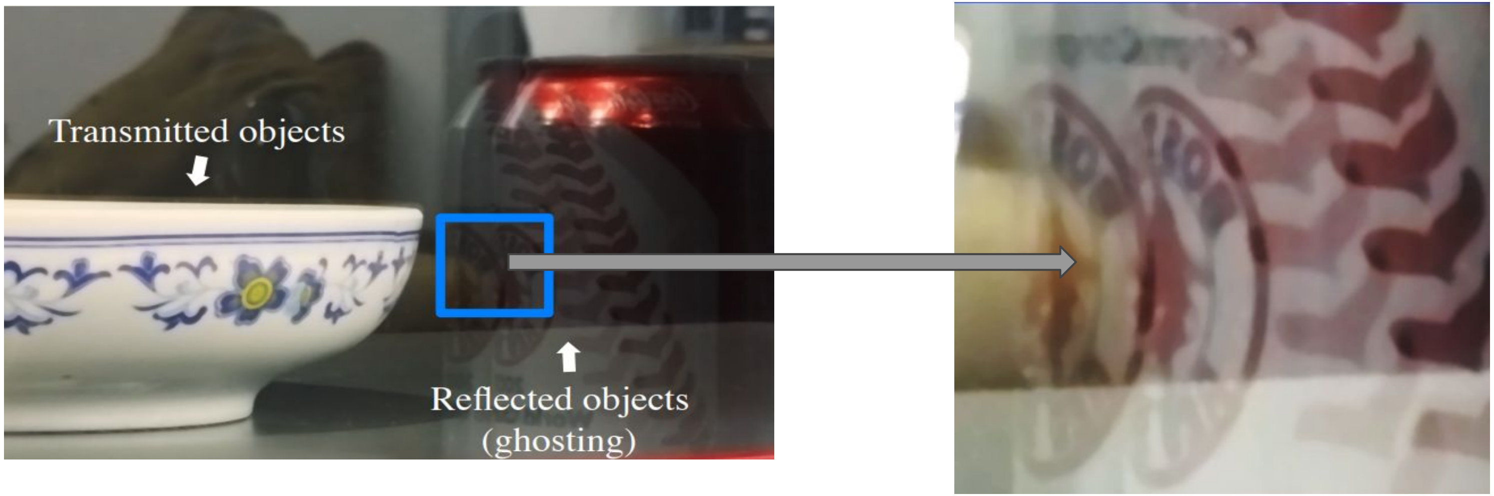
It’s the appearance of a secondary reflected image on the captured image, like when window reflections often appear multiple times.

Double reflection - ghosting

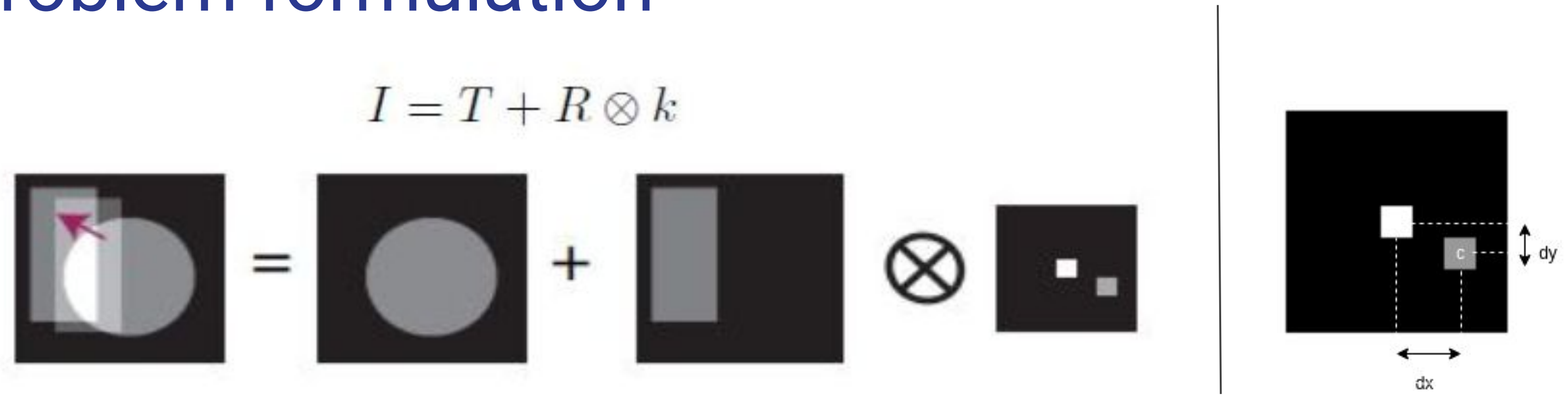
Heuristic applied

Ghosting cues exploit asymmetry between the layers which arise from **shifted double reflections** of the reflected scene off the glass surface.

In **double-pane windows**, each pane reflects shifted and attenuated versions of objects on the same side of the glass as the camera. For **single-pane windows**, ghosting cues arise from shifted reflections on the two surfaces of the glass pane.



Problem formulation

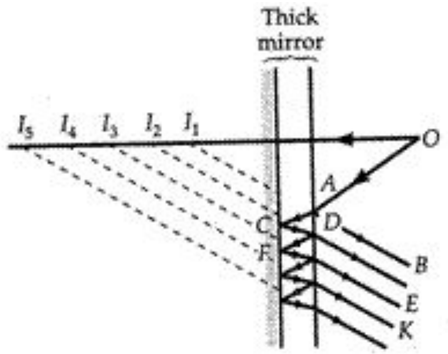


I = T + R*k + n ; where I is Original Image, T is Transmission layer, R is Reflection layer, k is two-pulse ghosting kernel, n is additive Gaussian noise.

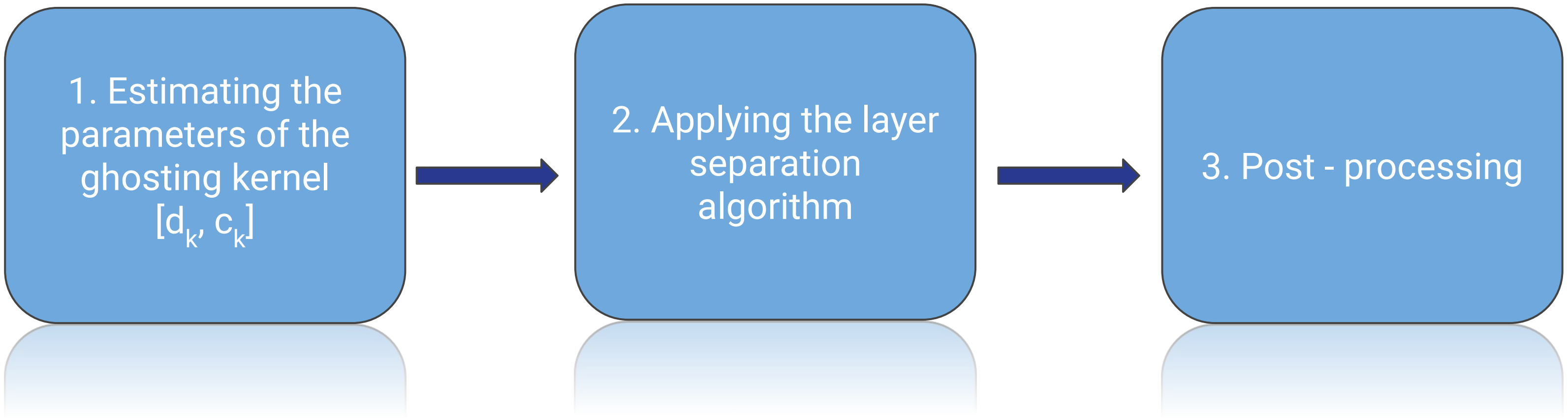
The original image is modeled as a mixture of these layers and the desirable image component is recovered after removing the undesired reflection layer. Ghosting kernel, k is parameterized by the spatial offset $\mathbf{d_k(d_x,d_y)}$ and the attenuation factor $\mathbf{c_k}$.

Assumptions while modelling ghosting

- We assume that the **spatial shift and relative attenuation between R1 and R2 is spatially invariant.**
- We **ignored higher order reflections** as they carry minimal energy
- In this case we **have not considered ghosting in the transmitted layer**



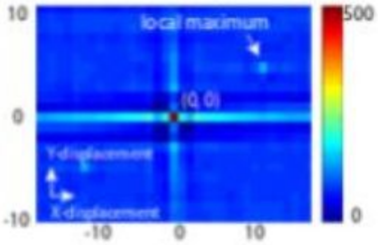
Solution Overview



Step 1:Estimating Kernel Parameters (Spatial offset: dx,dy)

Input Image

Laplacian



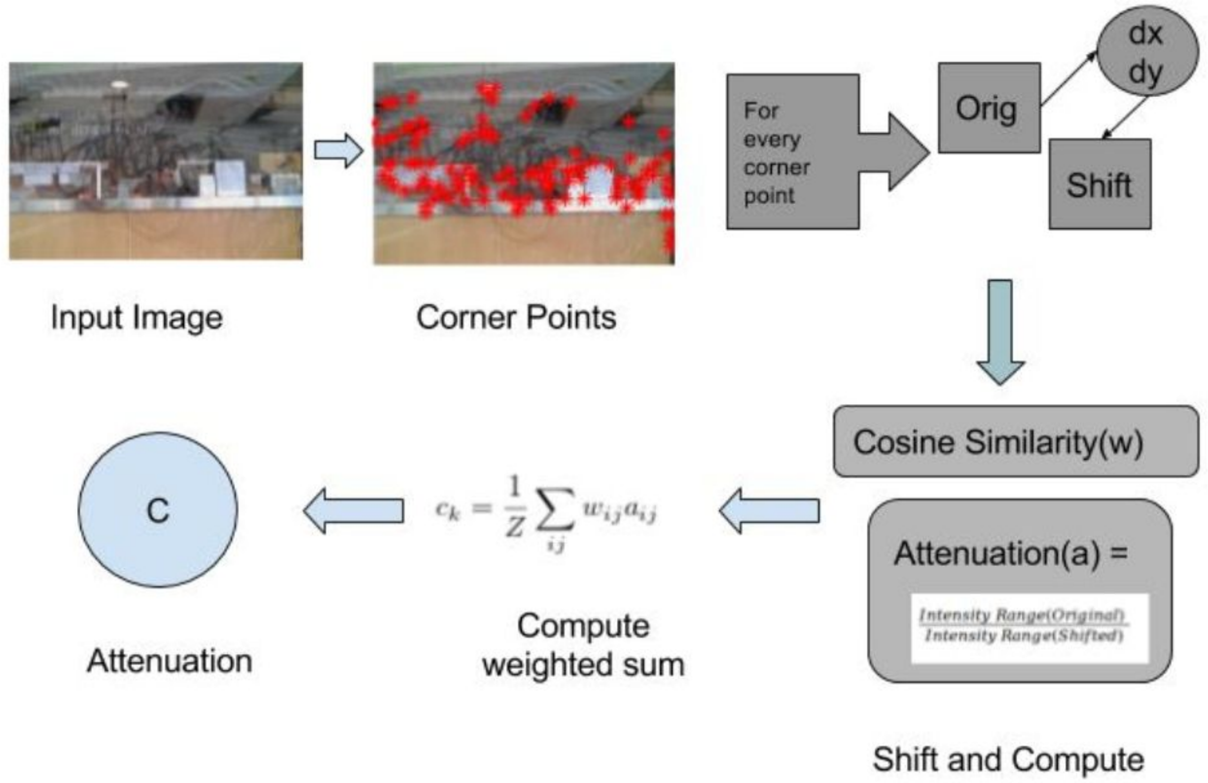
Autocorrelation Map

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

Pick Max among correct local minima

dx dy

Step1:Estimating Kernel Parameters (Attenuation factor: c)



Step 2: Layer Separation Algorithm

Given a ghosting kernel *k*, reconstruction loss using T and R can be expressed as :

$$\boxed{L(T,R) = \frac{1}{\sigma^2}\|I - T - R \otimes k\|_2^2}$$

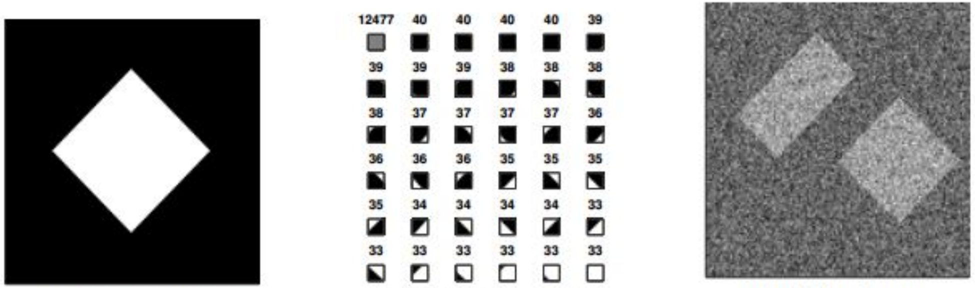
To regularize this optimization problem, the research paper applies a patch-based prior using a pre-trained Gaussian Mixture Models (GMM). The prior captures the covariance structure and pixel dependencies over a specified patch size of 8.

The final combined cost function :

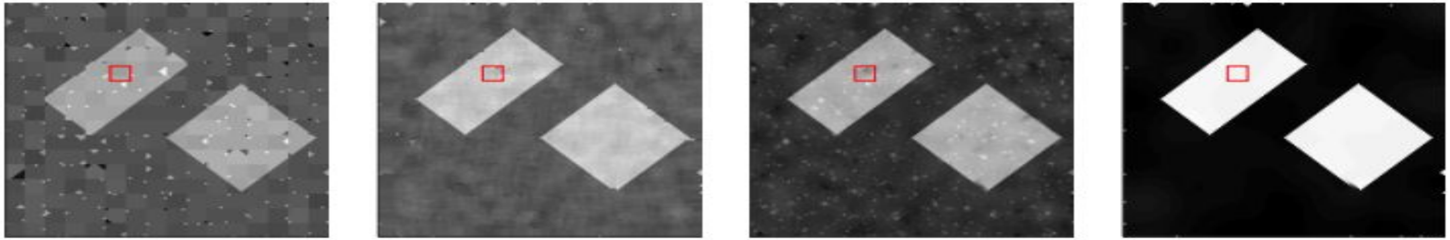
$$\boxed{\begin{aligned}\min_{T,R} \quad & \tfrac{1}{\sigma^2}\|I - T - R \otimes k\|_2^2 - \textstyle\sum_i \log(\mathrm{GMM}(P_i T)) \\ & - \textstyle\sum_i \log(\mathrm{GMM}(P_i R)), \ \text{s.t. } 0 \le T, R \le 1\end{aligned}}$$

Patch likelihoods to Whole image restoration

We wish to find a reconstructed image in which every patch is likely under our prior while keeping the reconstructed image still close to the corrupted image: maximizing **the Expected Patch Log Likelihood (EPLL)** subject to constraining it to be close to the corrupted image.



(a) Training Image (b) Prior Learned (c) Noisy Image



(d) Non Overlapping (e) Center Pixel (f) Averaged Overlapping (g) Our Method

Expected Patch Log Likelihood - EPLL

Given an image x (in vectorized form) we define the EPLL under prior p as:

$$EPLL_p(\mathbf{x}) = \sum_i \log p(\mathbf{P}_i \mathbf{x})$$

Where P_i is a matrix which extracts the i^{th} patch from the image (in vectorized form) out of all overlapping patches.

Assuming a patch location in the image is chosen uniformly at random, EPLL is the expected log likelihood of a patch in the image (up to a multiplication by 1/N).

Now, assume we are given a corrupted image y, and a model of image corruption of the form $||Ax-y||^2$. The cost to minimize under prior p is:

$$f_p(\mathbf{x}|\mathbf{y}) = \frac{\lambda}{2}||\mathbf{A}\mathbf{x} - \mathbf{y}||^2 - EPLL_p(\mathbf{x})$$

Gaussian Mixture Prior

A finite Gaussian mixture model over the pixels of natural image patches is learnt. Learning is performed using the **Expectation Maximization algorithm (EM)**. The calculation of log likelihood of a patch is:

$$\log p(\mathbf{x}) = \log \left(\sum_{k=1}^{K} \pi_k N(\mathbf{x}|\mu_k, \Sigma_k) \right)$$

Where π_k are the mixing weights for each of the mixture component and μ_k and Σ_k are the corresponding mean and covariance matrix.

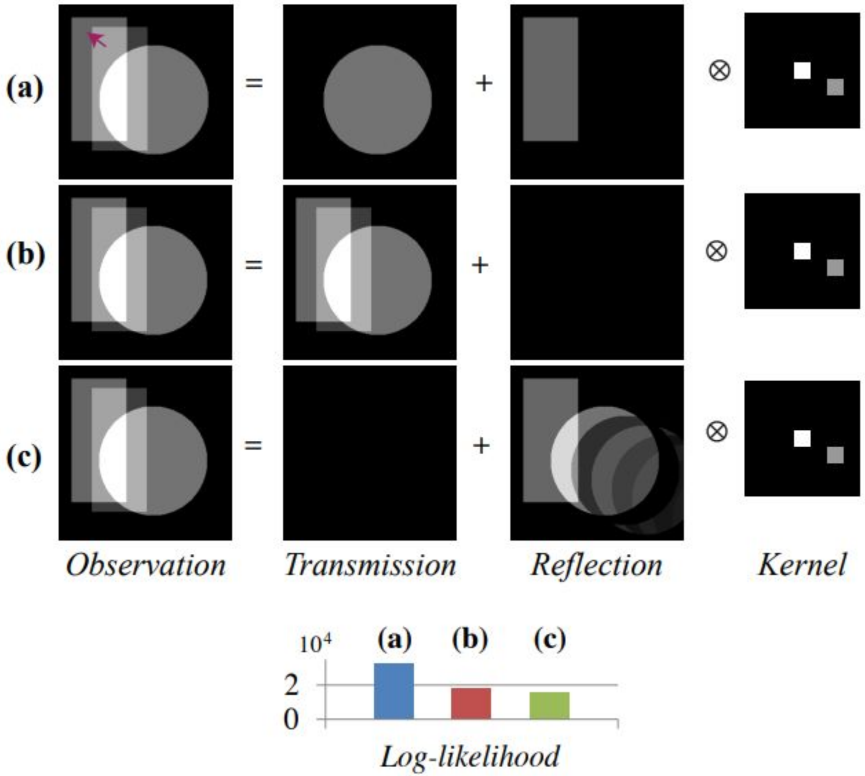
The model is learned from a set of 2×10^6 patches. The model is with learned 200 mixture components with zero means and full covariance matrices.

An example - a comparison

A synthetic example with a circle as the transmission layer and a rectangle as the reflection layer.

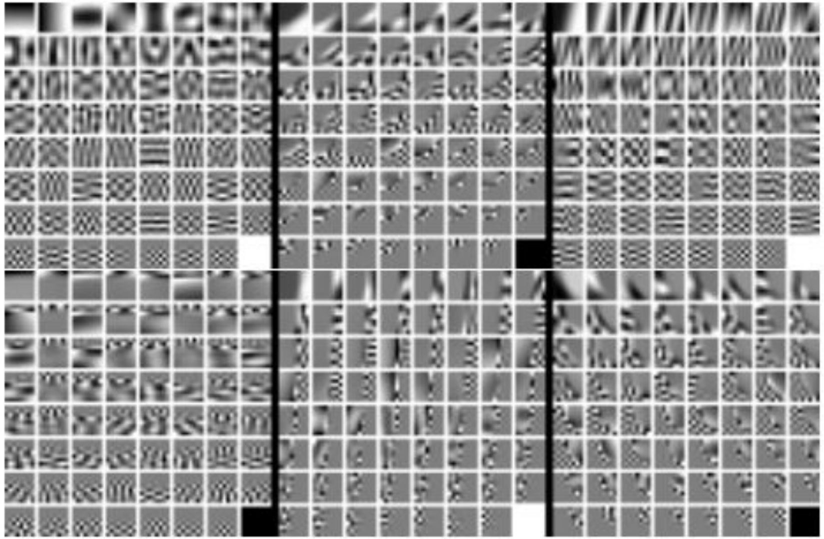
We compare the log likelihoods of the various possible decompositions under a GMM Model.

The log likelihood of the **a** is the highest, (implying it's most "natural") which is the ground truth.



Why GMM priors work better?

Every sample x is well approximated by the top m eigenvectors of the covariance matrix of the mixture component that it belongs to.
If we consider the set of all m eigenvectors of all mixtures as a "dictionary" then every sample is approximated by a sparse combination of these dictionary elements.



These eigenvectors have rich structures- PCA, edges, depict occlusions, texture boundaries, etc.

Step 2: Layer Separation Algorithm (Optimization)

The cost function is non-convex due to the use of the GMM prior. We use an optimization scheme based on half-quadratic regularization. We introduce auxiliary variables z_i^T and z_i^R for each patch P_iT and P_iR, respectively. We then optimize the the following cost function :

$$\begin{aligned}
&\min_{T,R,z_T,z_R} \frac{1}{\sigma^2}\|I - T - R \otimes k\|_2^2 \\
&\quad + \frac{\beta}{2}\sum_i \left(\|P_iT - z_T^i\|^2 + \|P_iR - z_R^i\|^2\right) \\
&\quad - \sum_i \log(\mathrm{GMM}(z_T^i)) - \sum_i \log(\mathrm{GMM}(z_R^i)) \\
&\text{s.t. } 0 \le T, R \le 1
\end{aligned}$$

Step 2: Layer Separation Algorithm (Minimization)

To solve the above auxiliary problem, we use increasing values of β in successive iterations.

- For each fixed value of β, we perform **alternating minimization**
- We first **fix z_T^i and z_R^i and solve for T and R**. We solve this sub-problem using extended L-BFGS
- Next, we **fix T and R, and update z_T^i** for each patch i
  For this, we first select the component with the largest likelihood in the GMM model, and then perform Wiener filtering using only that component. An analogous update is used for z_R^i.

L-BFGS-B is a limited memory algorithm for solving large nonlinear optimization problems subject to simple bounds on the variables.

Approximate MAP estimation procedure

1. Given noisy patch y we calculate the conditional mixing weights $\pi_k' = P(k|y)$.

2. We choose the component which has the highest conditional mixing weight

$$k_{max} = \max_k \pi_k' .$$

3. The MAP estimate x' is then a Wiener filter solution for the k_{max}-th component:

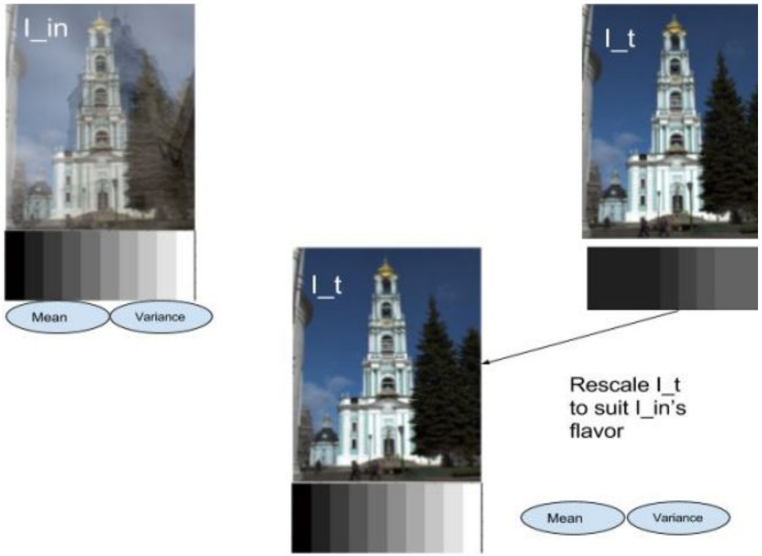
$$\hat{\mathbf{x}} = \left(\mathbf{\Sigma}_{k_{max}} + \sigma^2\mathbf{I}\right)^{-1}\left(\mathbf{\Sigma}_{k_{max}}\mathbf{y} + \sigma^2\mathbf{I}\mu_{k_{max}}\right)$$

Step 3: Post Processing

Using the mean and variance of the original image, we adjust the contrast of the transmitted layer to preserve the flavour of the original image.

$$sig = (\sigma_{I\_in} / \sigma_{I\_t})$$

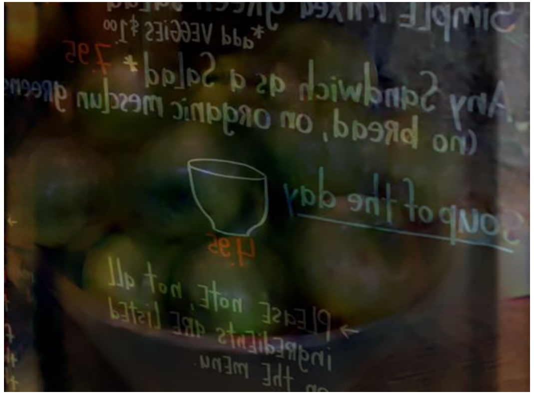
$$I\_t = sig * (I\_t - \mu_{I\_t} + \mu_{I\_in})$$



Results

Original Image

Transmitted Layer



Reflection Layer

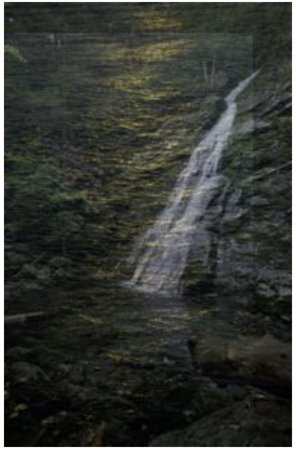
Results

Original Image

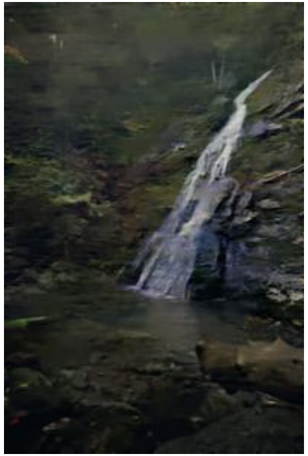
Transmitted Layer

Reflection Layer

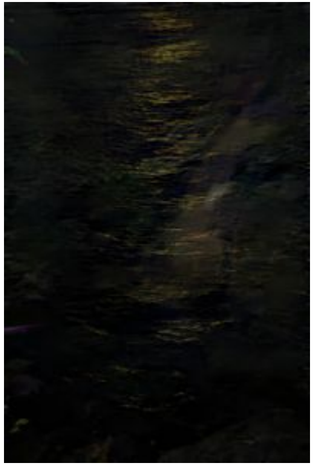
Results



Original Image



Transmitted Layer



Reflection Layer

Failure Case

Original Image

Transmitted Layer

Reflection Layer

Note : Large depth variations in the the reflected layer.

Failure Case

Original Image

Transmitted Layer

Reflection Layer

Note : Thin single pane glass reflection does not give rise to ghosting in the reflection layer

Limitations

- Requires thick glass windows, and large angles between camera viewing angle and glass surface for sufficient ghosting.
- Sensitive to strong repetitive textures in the transmission layer, as that can be mistaken to be ghosting.
- We assume spatially-invariant ghosting
  - the reflection layer does not have large depth variations
  - the angle between camera and glass normal is not too oblique

Applications of the work

- Image classification on the recovered transmissions
- Automated de-ghosting for product photography
- Automated driver assistance systems with dashboard cameras for object detection
- Obstruction removal using appropriate image prior and exploitable cues

THANK YOU