Computer Vision

Project Report

Reflection Removal using Ghosting Cues

Project implementation: Github link

TA Mentor : Pranay Gupta

Team Members

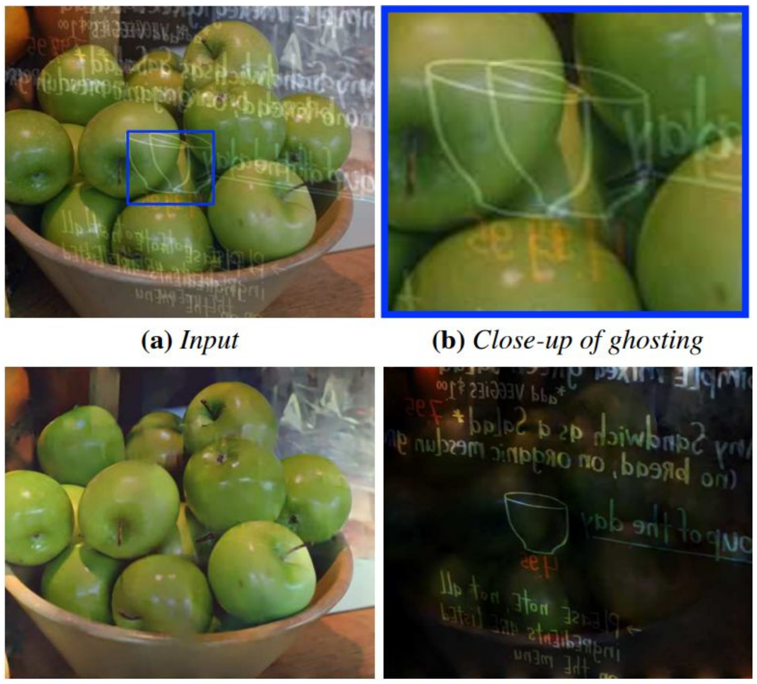
- Anoushka Vyas (20171057)
- Adhithya Arun (20171066)
- Meher Shashwat Nigam (20171062)

Introduction

When taking a photograph through a glass window pane, undesirable reflections of objects on the same side of the glass often ruin the picture. One may try to use different techniques to minimize these reflections, however, this isn't always practical as it involves adjustments that need to be made to the camera or the window pane. This raises the need for post-processing of the captured image to extract the desired image by removing the reflection artifacts.

Objective

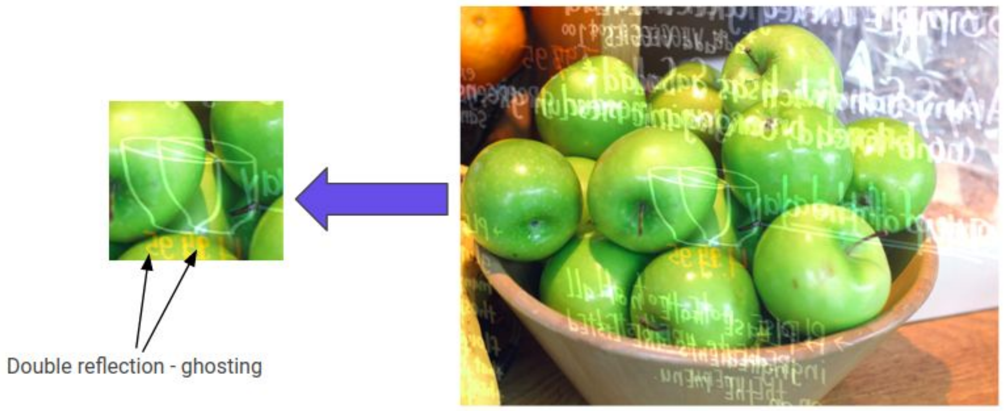
In this project, the original image is considered to be composed of a **reflection layer (undesirable) and a transmission layer (desirable)**. We wish to separate the transmission layer and the reflection layer.



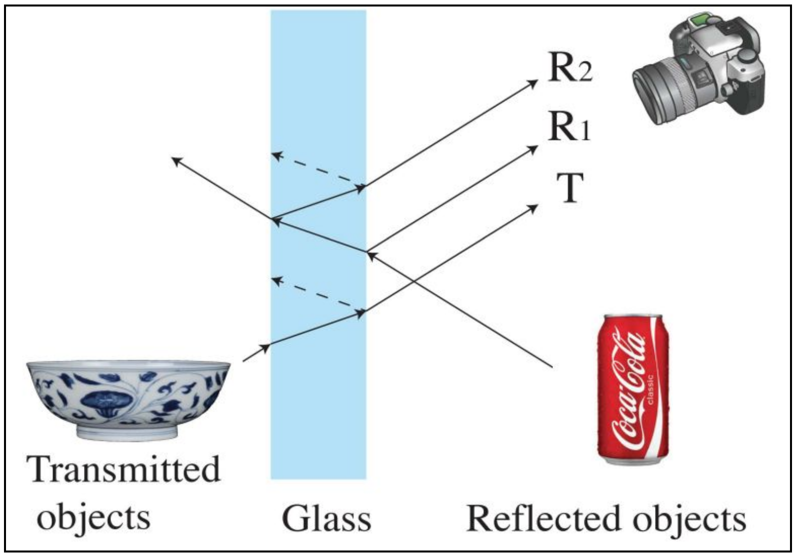
(a) *Input* **(b)** *Close-up of ghosting*

(c) *Recovered Transmission* **(d)** *Recovered Reflection*

Heuristic Applied : Ghosting Cues

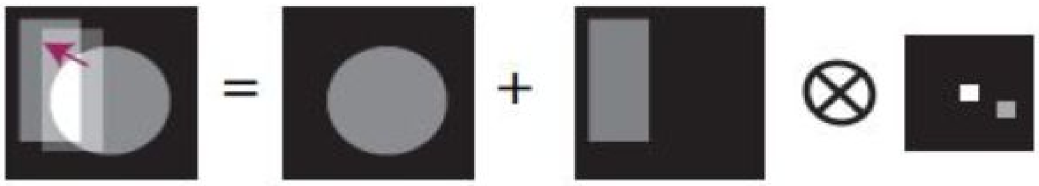


Single and double pane windows produce double reflections, a primary and secondary reflection, of objects on the same side of the camera such that the latter is a shifted and attenuated version of the former. **This can be used to break the symmetry between the transmission and reflection layers to get cues for separation.** For double pane windows, each pane reflects the shifted and attenuated versions of the objects on the same side of the glass as the camera. For single pane windows, the two surfaces of the pane are used as **ghosting cues.**



Problem formulation

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



I = T + R ⊗ k + n

Where, ***I*** *is the original image*

T *is the transmission layer*

R *is the reflection layer*

k *is a 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. Ghosting kernel, k is parameterized by the spatial offset $\mathbf{d_k}$ and the attenuation factor $\mathbf{c_k}$**.**

Assumptions

- 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**

Natural Images

However, the problem becomes challenging due to its ill posed nature, because the transmission and reflection layers in an image appear with the same statistical properties in natural images and hence are very difficult to separate.

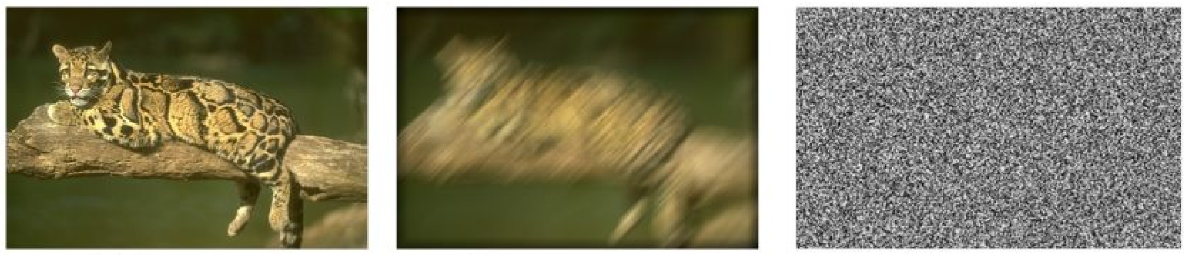

(a) A likely image (b) A less likely image (c) An unlikely image

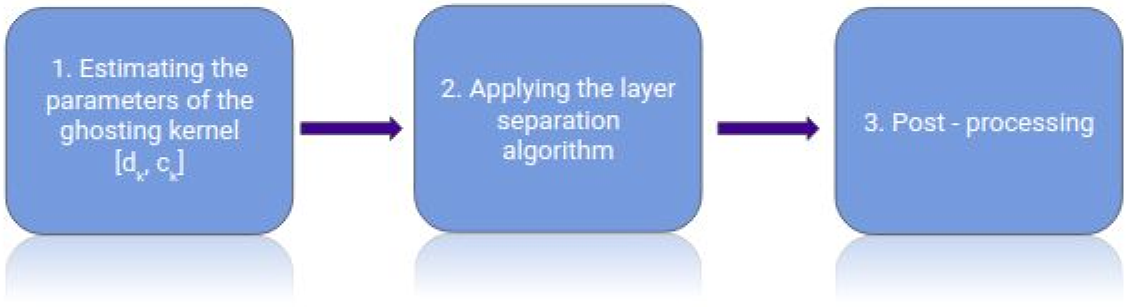
Figure 1.1: An ideal statistical model of natural images would give high likelihood values to images of a natural source, and low likelihood values to other images.

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."**

Workflow

The algorithm for separating the transmission and reflection layers involves three steps:

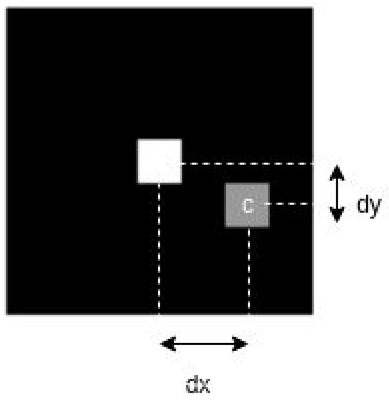
1. Kernel Estimation (Computation of k, parameterized by c_k and d_k)
2. Layer Separation Algorithm (Minimization Problem)
3. Post Processing


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

Implementation

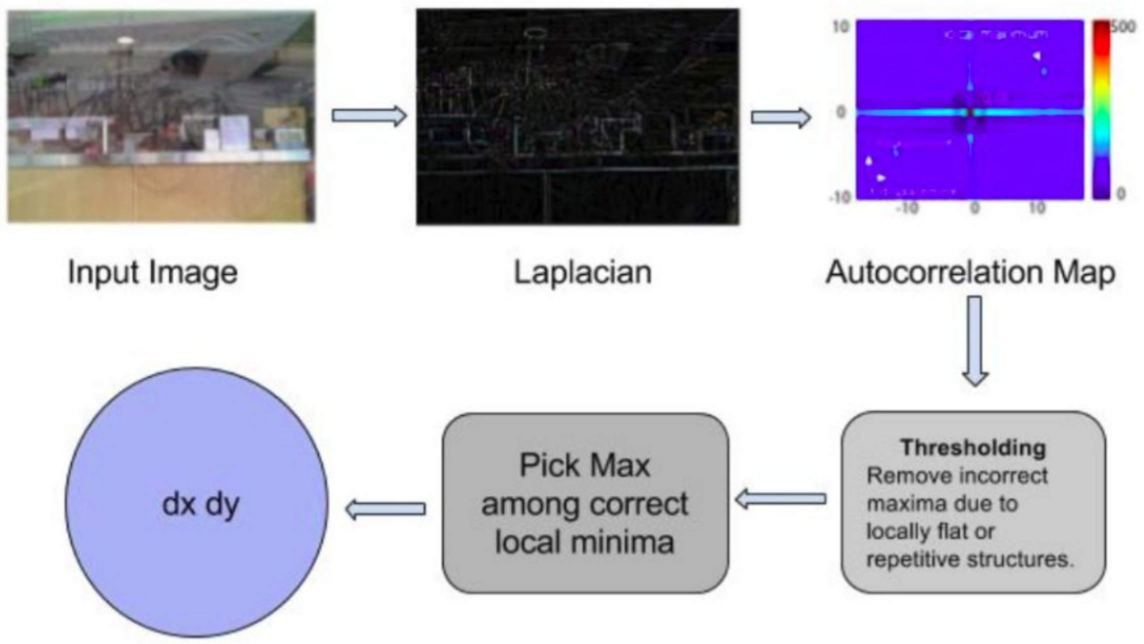
1. Kernel Estimation

The ghosting convolution kernel k, is parameterized by a spatial shift vector, $\mathbf{d_k(d_x,d_y)}$ and an attenuation factor $\mathbf{c_k}$ between the primary reflection and secondary reflection.



Spatial offset (d_k) calculation:

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. We discard incorrect maxima which appear due to locally flat or repetitive structures, or when found near the origin. We then pick the largest local maxima as the spatial shift vector.


Input Image
Laplacian
Autocorrelation Map
Thresholding
Remove incorrect maxima due to locally flat or repetitive structures.
Pick Max among correct local minima
dx dy

Attenuation factor (c_k) calculation:

The attenuation factor is calculated using the spatial shift vector. **Interest points are detected from the input image using Haris Corner detector**.

A 5x5 mean subtracted patch was extracted from each region of a corner feature. Patches that have a strong correlation with patches at spatial offset $\mathbf{d_k}$ are assumed to be due to either of the reflection layers.

Attenuation between a pair of matching patches is calculated as:

a_{ij} = (max(p_i)-min(p_i))/ (max(p_i)-min(p_i))

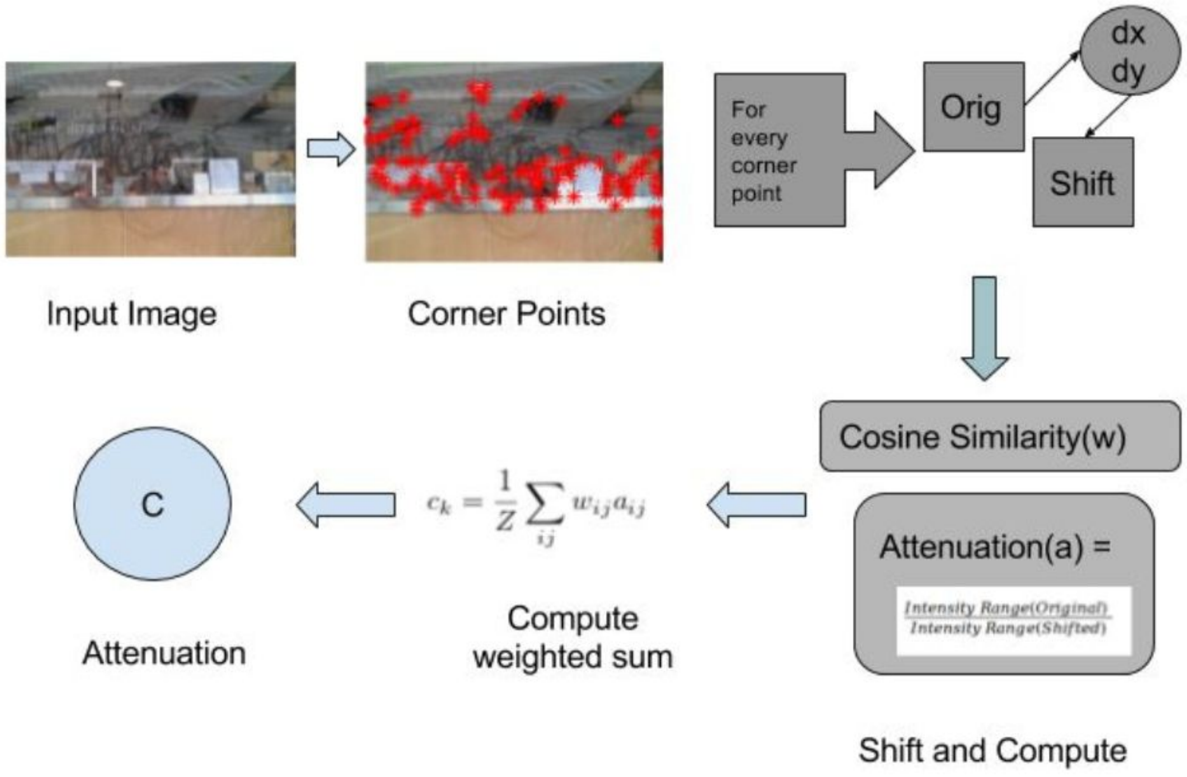
Where, $\boldsymbol{p_i}$ is the ith patch , $\boldsymbol{p_j}$ is the jth patch (with d_k shift)

NOTE: Here that (i, j) are chosen such that $a_{ij} < 0$.

Then, **c$_k$** is given as

$$c_k = \frac{1}{Z}\sum_{ij} w_{ij} a_{ij} \quad \text{,where } w_{ij} = e^{-\frac{\| sij \|^2}{2\sigma^2}}$$, $s_{ij} = (p_i . p_j)/|p_i||p_j|$ (cosine similarity)

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



2. Layer Separation Algorithm

Given a ghosting kernel k, a loss expression can be constructed for reconstruction using T and R:

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

Additional priors are required to regularize this optimization problem. The research paper applies a patch-based prior based on Gaussian Mixture Models (GMM).

We use a pre-trained zero-mean Gaussian Mixture Model with 200 mixture components and a patch size of 8x8 from the paper "***From learning models of natural image patches to whole image restoration" by D. Zoran and Y. Weiss"***.

Patch likelihoods to Whole image restoration:

We wish to find a reconstructed image in which every patch is likely under a 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.

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 ith patch from the image (in vectorized form) out of all overlapping patches, while $\log_p(P_i x)$ is the likelihood of the ith patch under the prior p.

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:

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

Gaussian Mixture Prior:

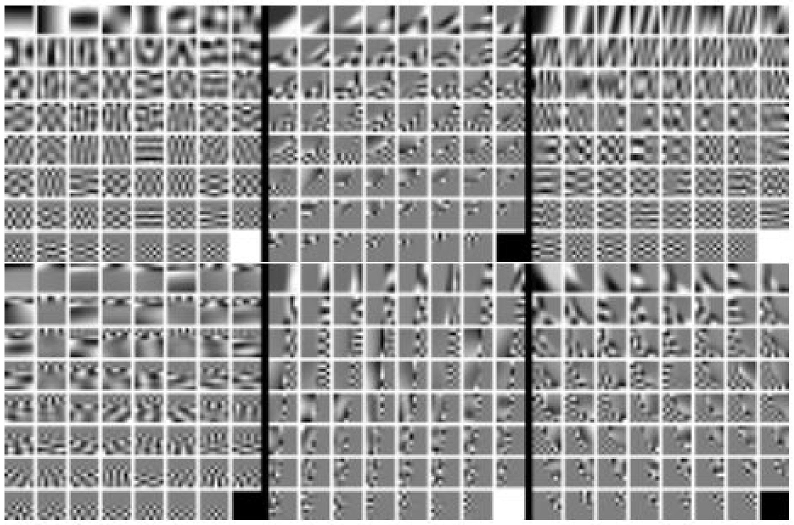
The GMM prior is trained on a set of 2 x 10^6 natural image patches. It consists of 200 mixture components and a patch size of 8x8. The components are zero-mean 64 dimensional distributions.

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 components and μ_k and Σ_k are the corresponding mean and covariance matrix.

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.



This depicts the eigenvectors of the 6 randomly selected mixture components from the learned model. Note that these have rich structures - while some resemble **PCA eigenvectors**, some depict forms of **occlusions**, **modeling texture boundaries** and **edges**. These are very different from the filters usually learned by sparse coding and similar models. It would seem that these structures contribute much to the expressive power of the model.

Thus combining EPLL and GMM prior the regularizer minimizes the following cost function:

$$\boxed{-\sum_i \log(\mathrm{GMM}(P_i T)) - \sum_i \log(\mathrm{GMM}(P_i R))}$$

<u>**Optimization:**</u>

The final combined cost function would be:

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

which is a non-convex optimization problem due to the GMM prior. Auxiliary variables are introduced to perform the optimization. Equation with auxiliary variables :

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

Increasing values of β is used. Alternating minimization is performed for the auxiliary variables, and T & R, i.e., $\{z_T^i\}$ and $\{z_R^i\}$ are fixed while solving for T and R and vice versa. To solve the quadratic sub-problems, L-BFGS-B is used to handle the box constraints. To solve for auxiliary variables while keeping T and R constant, approximate MAP estimation procedure is used, the steps are (all of this is taken from the GMM prior paper):

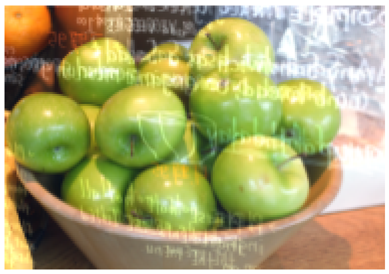
- Given noisy patch y we calculate the conditional mixing weights $\pi_k' = P(k|y)$.
- We choose the component which has the highest conditional mixing weight $k_{max} = \max_k \pi_k'$.
- The MAP estimate x' is then a Wiener filter solution for the k_{max}-th component:

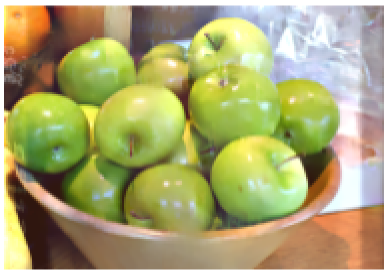
$$\hat{\mathbf{x}} = \left(\mathbf{\Sigma}_{k_{max}} + \sigma^2\mathbf{I}\right)^{-1}\left(\mathbf{\Sigma}_{k_{max}}\mathbf{y} + \sigma^2\mathbf{I}\mu_{k_{max}}\right)$$

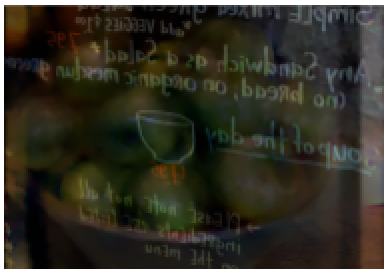
Since initialization is vital to get a better minima, we initialize the GMM prior with sparsity inducing based model, with a convex L_1 prior loss, for which optimization is done using ADMM with sparsity inducing filters like gradients and laplacian.

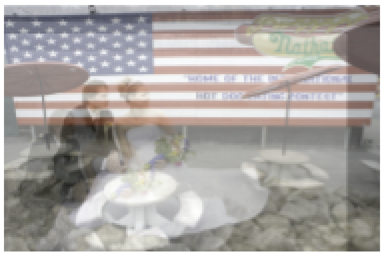
Results on test images

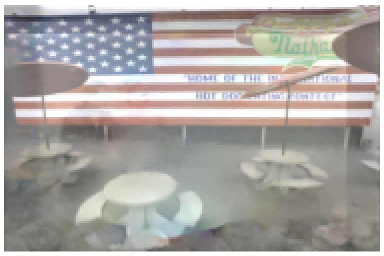
Successful examples:

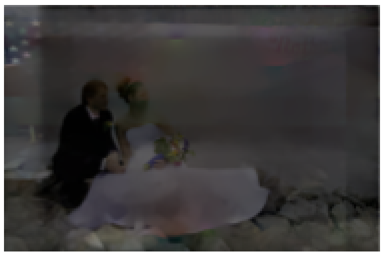

Original Image

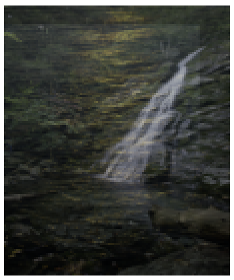

Transmitted Layer

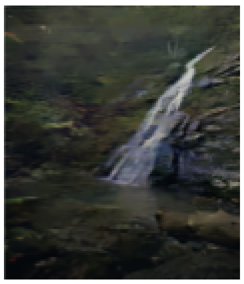

Reflection Layer

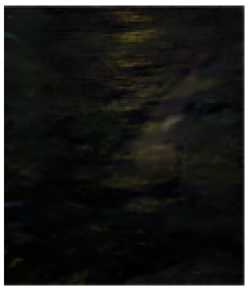

Original Image


Transmitted Layer

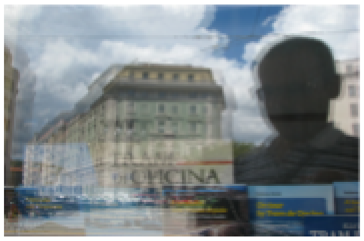

Reflection Layer

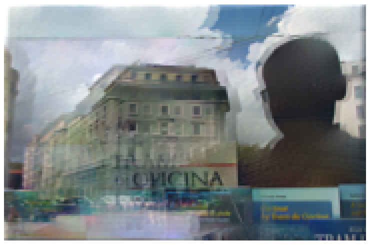

Original Image

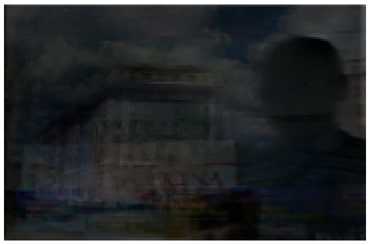

Transmitted Layer


Reflection Layer

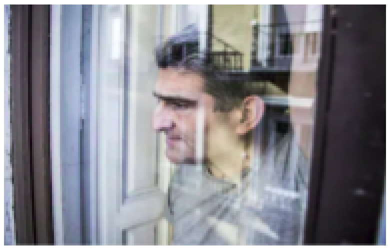
Failed Examples:

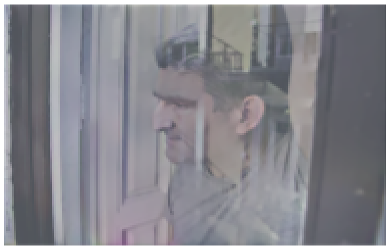

Original Image

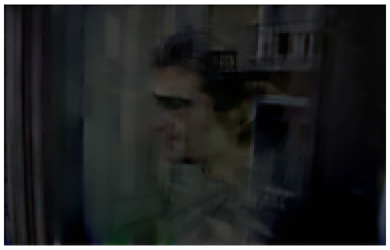

Transmitted Layer


Reflection Layer

Note: Large depth variations in the reflected layer.


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

- 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

References

- Natural Images: Natural Image Statistics for Human and Computer Vision
- GMM priors for image restoration : From Learning Models of Natural Image Patches to Whole Image Restoration
- For explanation of non negativity constraint on T and R: Layer extraction from multiple images containing reflections and transparency 1 Introduction
- Half quadratic regularisation: Nonlinear image recovery with half-quadratic regularization - Image Processing, IEEE Transactions on
- L-BFGS: L-BFGS-B { FORTRAN SUBROUTINES FOR LARGE-SCALE BOUND CONSTRAINED OPTIMIZATION
- To understand initialisation of GMM model using ADAM: Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers