Digital Image Processing Project

Reflection Removal by Ghost Effect from A Single Image

Team: **Photo**

Aniketh Reddimi	2018102014
Ashuthosh Bharadwaj	2019112003
Eshan Gupta	2019102044
Saravanan Senthil	2019101016

Team Mentor: Fiza Hussain

Link to repo: https://github.com/Digital-Image-Processing-IIITH/dip-project-photo

INDEX

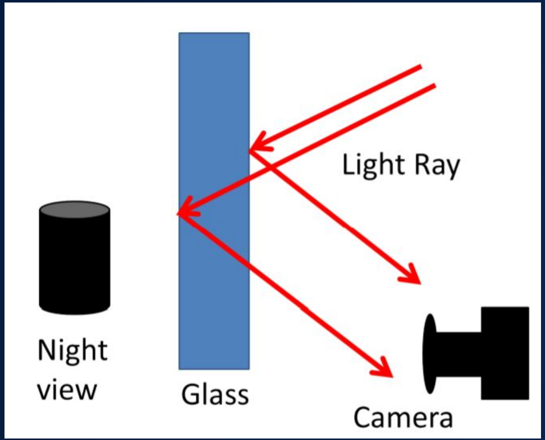
1. Paper Discussion
- Framework, Model
- Step #1 Shift Determination
- Step #2 Gradient Separation
- Step #3 Image Reconstruction

2. Our Implementations
- Codes
- Results and Demonstration

3. Bonus + GUI
- Fast Reflection Suppression
- GUI

1. Discussing the paper

Framework

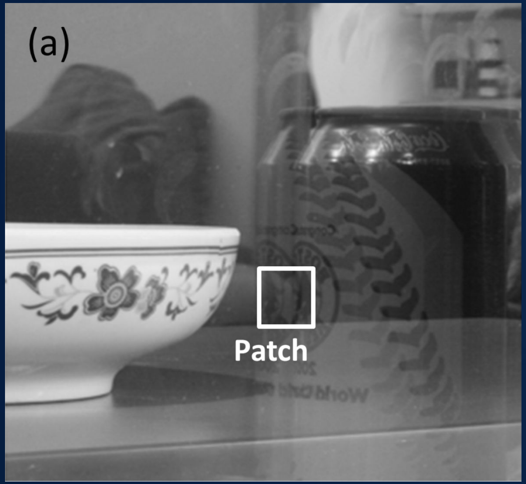


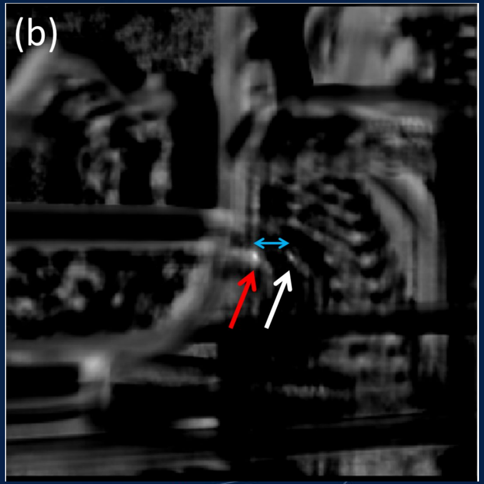
$$I(x,y) = I_s(x,y) + I_r(x,y)$$

$$I(\vec{x}) = I_s(\vec{x}) + I_r(\vec{x}) + \beta I_r(\vec{x} - \vec{d})$$

$\beta \approx 0.5$
[2] Griffith's Electrodynamics

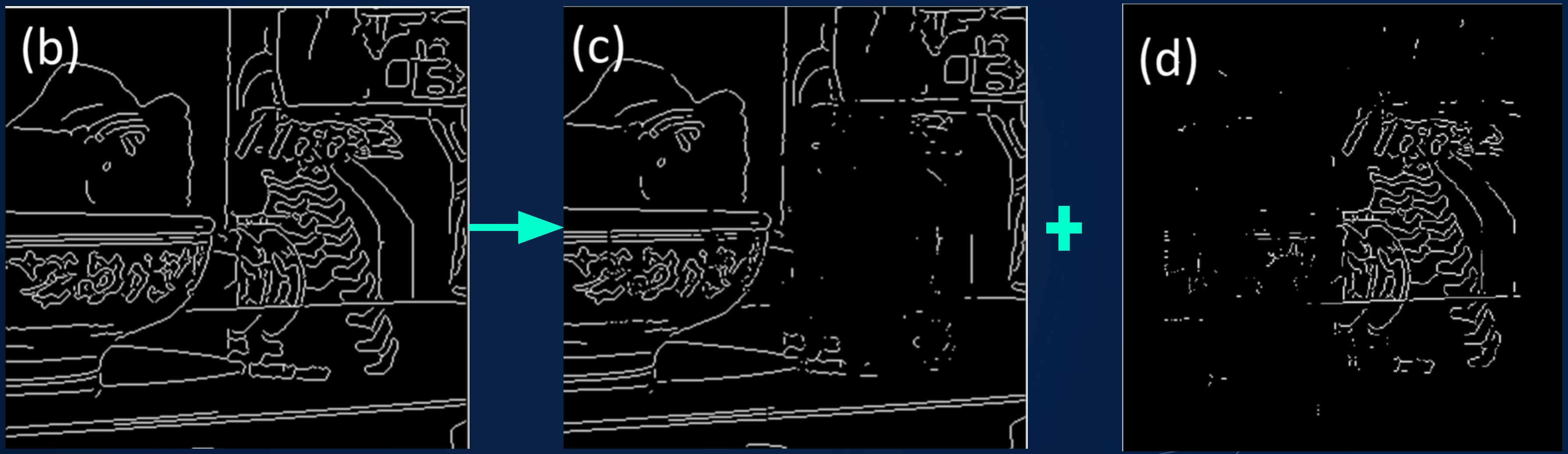
Part #1: Shift Determination





$\vec{d}$

Part #2: Gradient Separation



Part #2: Gradient Separation

If $S(\vec{d})$ or $S(-\vec{d})$ is a local maximum, (i, j) is assigned to be the gradient of the reflection image . Else (i, j) is assigned to be the gradient of the scene image

```
d <- Shift from step #1
(i,j) <- where(edge)
template <- image[i-10:i+10, j-10:j+10]
R <- Template Matching(Edge Image, template)

if
R[(i,j) + d] or R[(i,j) - d] is local maxima
then
Reflection Edge[(i,j)] <- 1
```

Part #3: Image Reconstruction

3.3. Image Reconstruction

After gradient separation, only the scene image's gradients are preserved.The task of image reconstruction is to reconstruct the scene image by scene gradients. Because we heavily modified the gradient field, so the field is no longer conservative. To reconstruct the image, we use the deconvolution technique in [7].

[7] Y. Weiss. Deriving intrinsic images from image sequences. 2001.

Part #3: Image Reconstruction

Deriving intrinsic images from image sequences

Yair Weiss
Computer Science Division
UC Berkeley
Berkeley, CA 94720-1776
yweiss@cs.berkeley.edu

Framework:

$$I(x, y) = L(x, y)R(x, y)$$

Taking $log(\cdot)$

$$i(x, y) = l(x, y) + r(x, y)$$

This looks a lot like our framework for Reflection removal, essentially separating 2 layers of an image using some prior assumptions.

Part #3: Image Reconstruction

$$\boxed{i(x,y) = l(x,y) + r(x,y)}$$

$$\boxed{f * i(x,y) = f * (l(x,y)) + r(x,y))}$$

Given that $f * l(x,y) \approx \quad \sim L(\alpha, Z)$, we can use a ML estimator to arrive at the best possible $\hat{r}(x,y)$.

Part #3: Image Reconstruction

$$f_n \star \hat{r} = \hat{r}_n \tag{5}$$

It can be shown that the psuedo-inverse solution is given by:

$$\hat{r} = g \star \left(\sum_n f_n^r \star \hat{r}_n \right) \tag{6}$$

with f_n^r the reversed filter of f_n: $f_n(x,y) = f_n^r(-x,-y)$ and g a solution to:

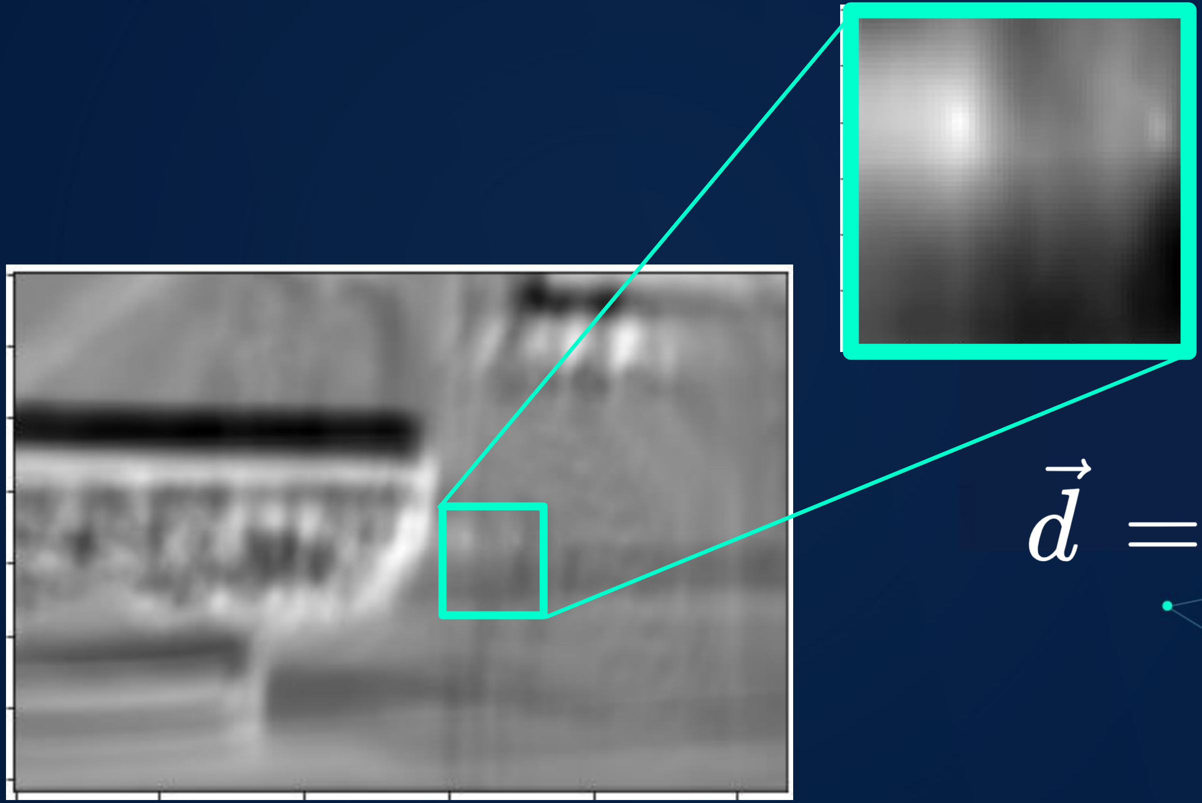
$$g \star \left(\sum_n f_n^r \star f_n \right) = \delta \tag{7}$$

Function = Edge = thresholded dervivative
Step 1 : Find which kernel f(x,y) will act as the thresholded derivative when convolved with an image, ie: f(x,y) * image = edge image (figure out for canny, or we will roll with simple derivative operator and scrap canny)

Step 2 : perform the following:
f1 = f(-x,-y)
h = fft(f1 * f)
g = ifft(ones(image shape) / h)
Output = g*f1*(edge image)

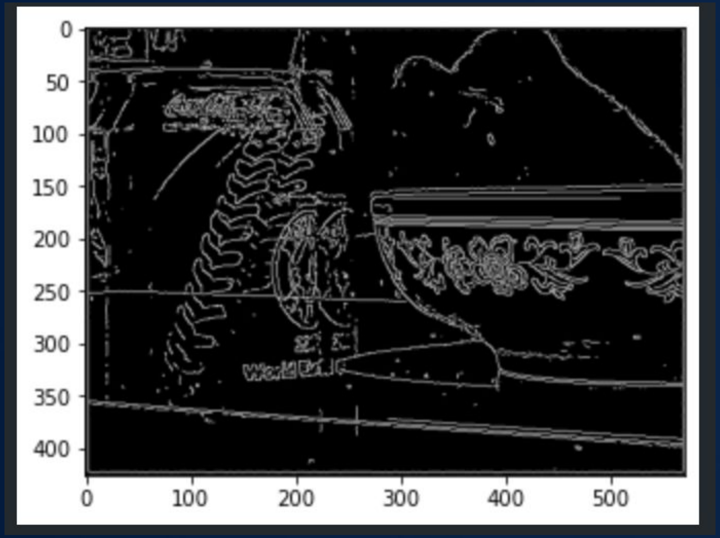
2. Our implementation

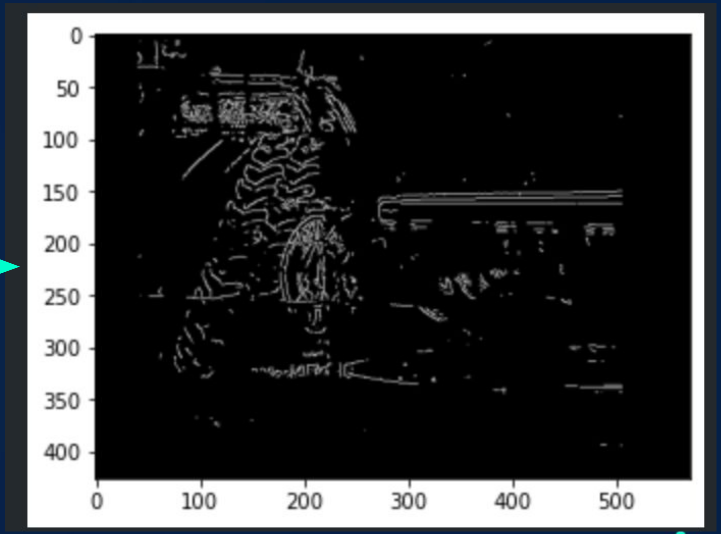
Part #1: Shift Determination



$$\vec{d} = \begin{bmatrix} 0 \\ 35 \end{bmatrix} \text{ Pixels}$$

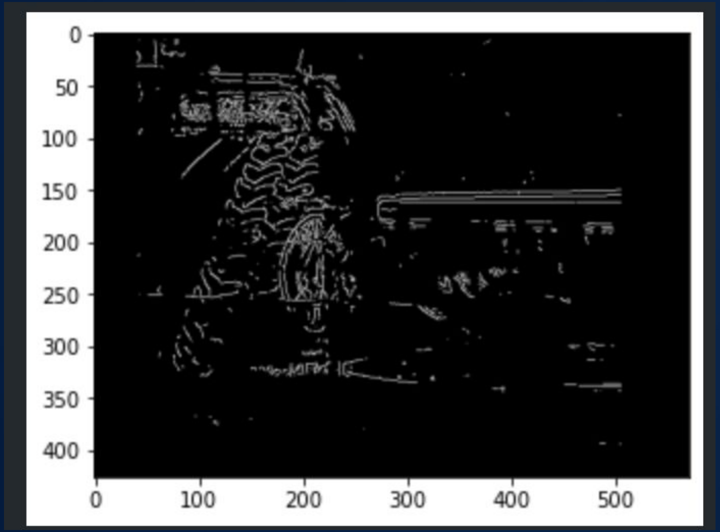
Part #2: Gradient Separation



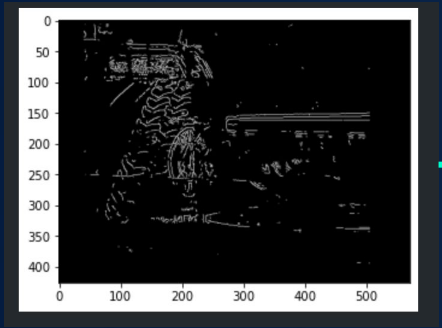


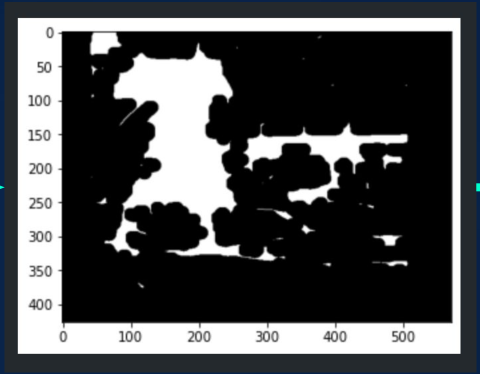
Part #2: Gradient Separation

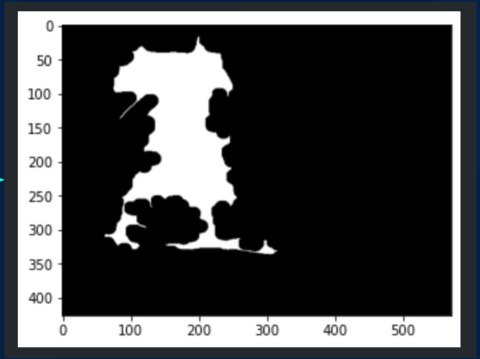
Issue with gradient separation with this method



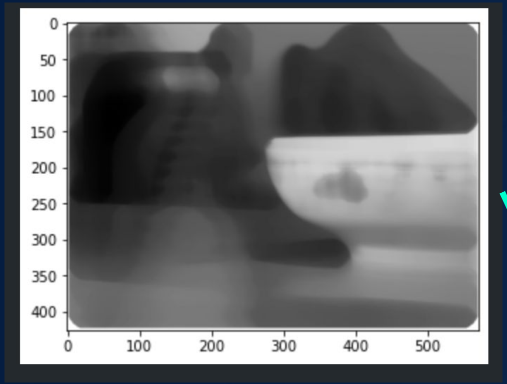
Part #3: Image Reconstruction

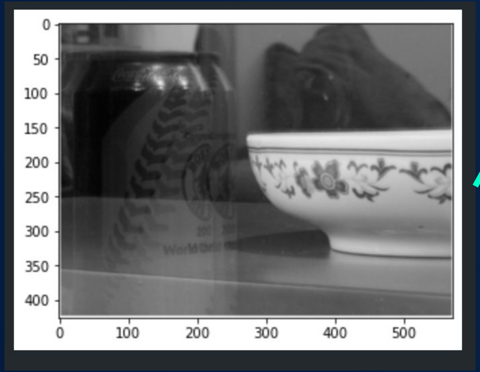


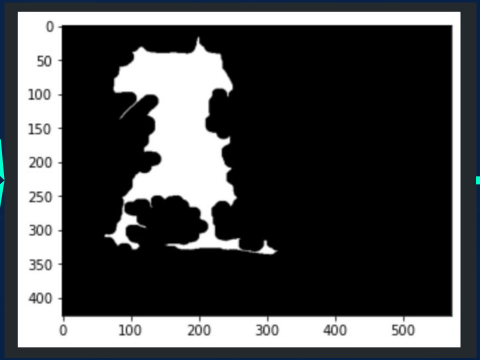




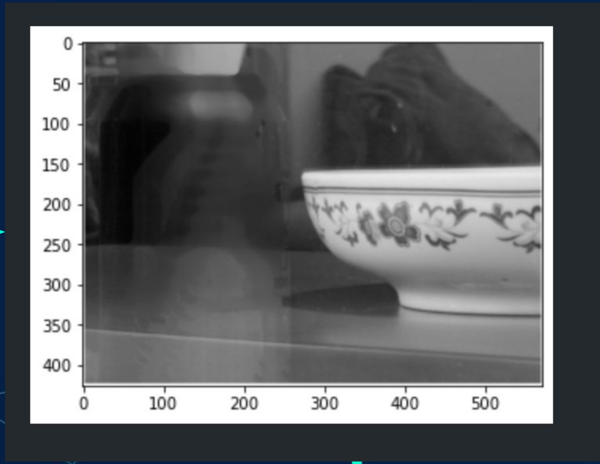
Part #3: Image Reconstruction



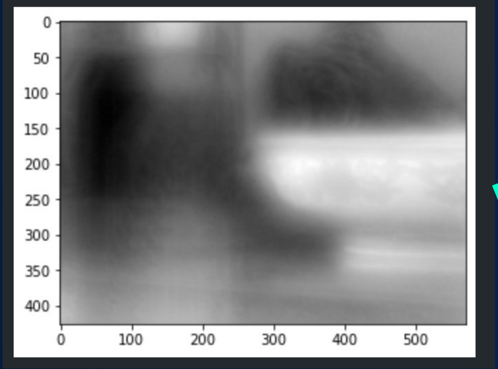


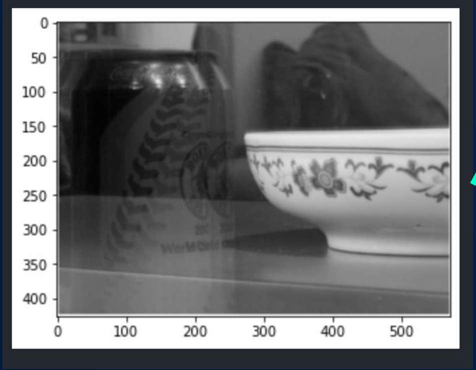


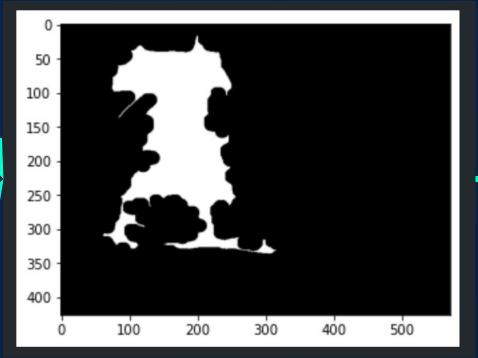
Median



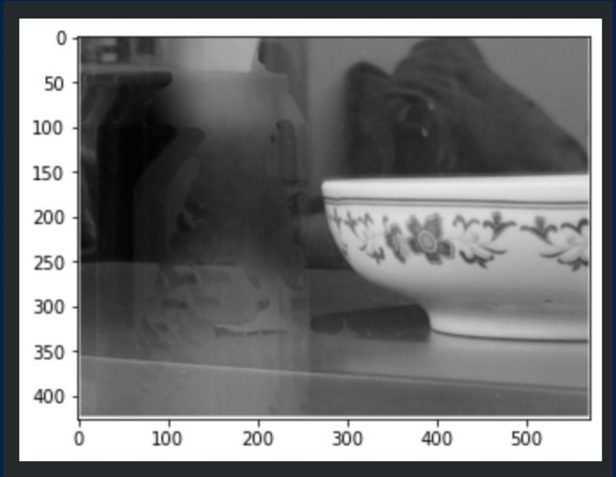
Part #3: Image Reconstruction







Fast Inverse Reflection Suppression



Bonus paper:

Fast Single Image Reflection Suppression via Convex Optimization

Main premise of the paper:

- Quite simple, and well explained in the paper, but was not trivial to implement.
- Assumption:
  - Image is made of reflected and transmission components $Y = T + R$
  - Edges of reflected images are weaker than transmission layer.
- Model of image with these assumptions: $Y = \omega T + (1 - \omega)(\kappa * R)$
- With these assumptions we can deduce that low energy edges belong to the reflection and high energy edges would correspond to the Transmission. This is reflected in the objective function

$$\min_t \frac{1}{2} ||L(T) - div(\delta_h(\nabla Y))||_2^2 + \epsilon ||T - Y||_2^2$$

Main premise of the paper:

- In order to solve the optimization problem posed earlier, we take its gradient as such:

$$\nabla_T = L(L(T) - div(\delta_h(\nabla Y))) + \epsilon(T - Y)$$

- Which on Assuming derivative is zero at the minima, we get

$$(L^2 + \epsilon)T = \epsilon Y + L(div(\delta_h(\nabla Y)))$$

- Which the paper proposes to solve with DCT and iDCT as it is similar to a 2D poisson distribution.

$$T_{m,n} = \mathcal{F}_c^{-1}\left(\frac{[\mathcal{F}_c(\mathbf{P})]_{m,n}}{K_{m,n}^2 + \varepsilon}\right)$$

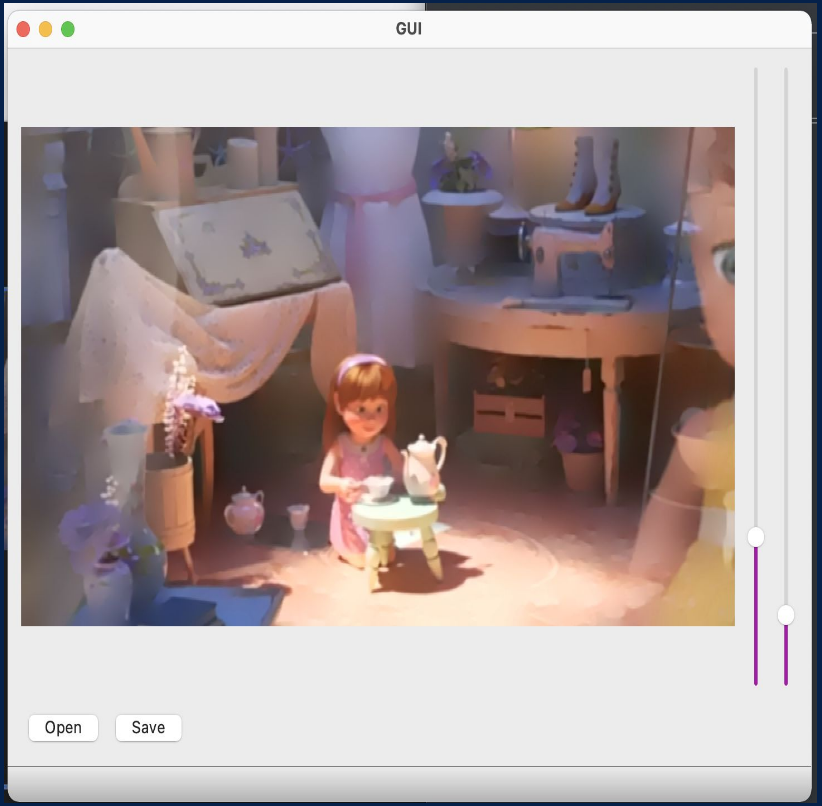
$$K_{m,n} = 2(\cos(\frac{\pi m}{M}) + \cos(\frac{\pi n}{N}) - 2)$$

Results

Results

Results

GUI with PuQT5



Using FSR for reconstruction in the original paper

Key observations on the input to ghosting effect:

- The REFLECTION is in focus, not the TRANSMISSION.
- This is the complement of what the FSR paper assumes.
- The delta function is thus changed to reflect this.

Grad < h = 0

Grad > h = 0

Individual Contributions

Saravanan Senthil: Shift determination, Fast reflection suppression, reconstruction (through a derivative of FRS), GUI.

Ashuthosh Bharadwaj: Template matching & Gradient Separation, Shift determination.

Eshan Gupta: Analysing the equations and testing the pseudo codes

Aniketh Reddimi: Implementing algorithms and cleaning up useless code

Thank you

:)