Supplemental Materials for Reflection Removal using Ghosting Cues

YiChang Shih Dilip Krishnan Frédo Durand William T. Freeman
MIT CSAIL

1. Window Reflection Images Sampled from the Internet

To understand how often ghosting occurs "in the wild", we analyze images returned by Google's Image Search. We used the keywords "window reflection", "window photography reflection", "window photography reflection problem", and "reflections on/via/in windows". After removing irrelevant results such as cartoon images and water reflections, we were left with 197 images. These are shown in Figure 1. We observed that around 96 of these images exhibit significant ghosting (49%). Figure 2 shows sampled images with various levels of ghosting. The original links of these images are in `urls.txt`.

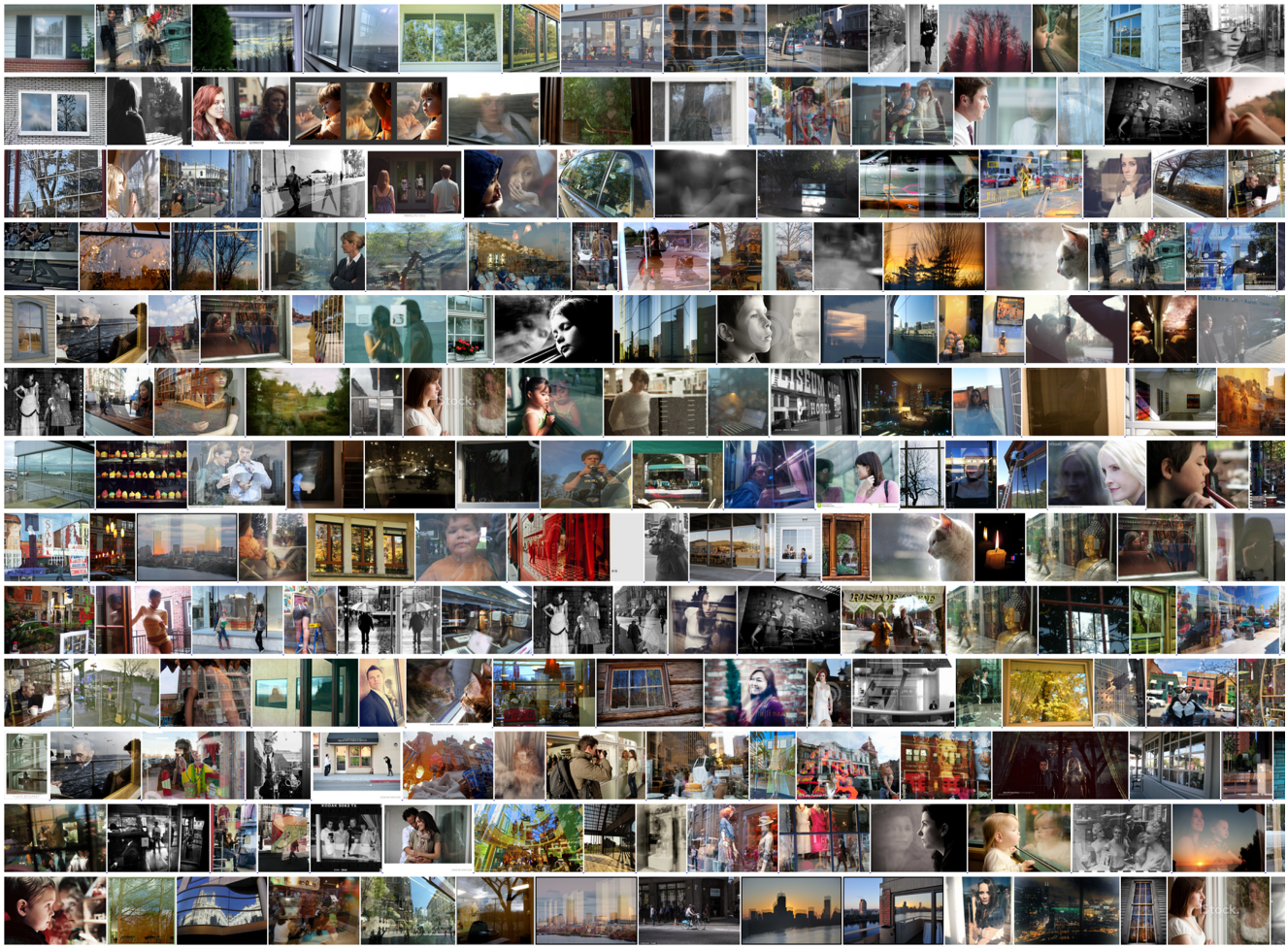
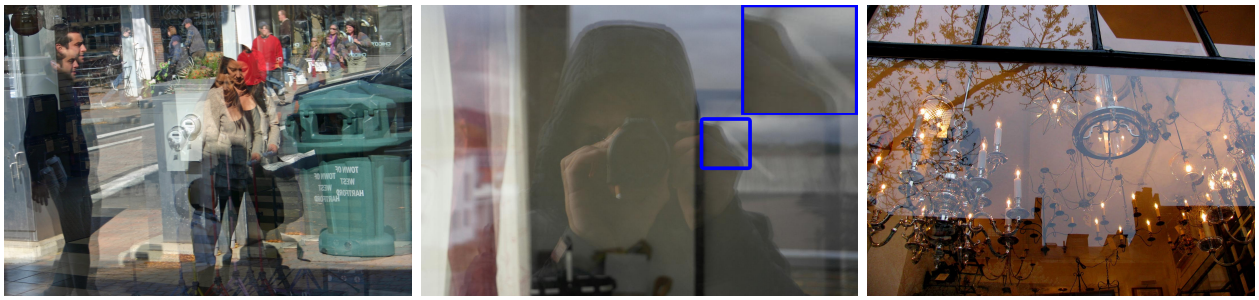


Figure 1: *Images from Google's image search. See text for details.*



(a) *Strong Ghosting* **(b)** *Mild Ghosting* **(c)** *Imperceptible Ghosting*

Figure 2: *Close-up of some images from Figure 1.*

2. Comparisons on Image Priors

Figure 3 compares the separations using various image priors on a synthetic input. Qualitatively, our prior (GMM prior + non-negativity) achieves the best result among other priors on the recovered transmission as well as reflection in Figure 4, and very close to the ground truth. The gradient error maps in Figure 5 show that our prior recovers long edges better than other priors.

3. Initialization with Alternating Direction Method of Multipliers (ADMM) in Equation 7

Here we explain the initialization in Equation 7 in the paper. Our goal is to solve the following equation:

$$\arg\min_{T,R} \frac{1}{\sigma^2}\|I - T - R \otimes k\|_2^2 + \|\nabla T\|_1 + \|\nabla R\|_1 + \|\nabla^2 T\|_1 + \|\nabla^2 R\|_1 \tag{1}$$

We use the ADMM message-passing algorithm described in Bento *et al*. [1]. The ADMM method is well suited to solving cost functions which are sums of decomposable functions. ADMM considers each term in the cost function as a sub-problem, and alternates between optimizing within sub-problems and reaching a consensus between variables shared across sub-problems. For the above cost function, we give the solution for each sub-problem in the following (please see [1] for details on message-passing ADMM and terminology):

Reconstruction sub-problems We solve the following equation given kernel parameters c and $\mathbf{d}_k$:

$$\arg\min_{T_i,R_i,R_j} \frac{1}{\sigma^2}\|I_i - T_i - R_i - cR_j\|_2^2 + \frac{\rho}{2}\left(\|T_i - u_T^i\|_2^2 + \|R_i - u_R^i\|_2^2 + \|R_j - u_R^j\|_2^2\right) \tag{2}$$

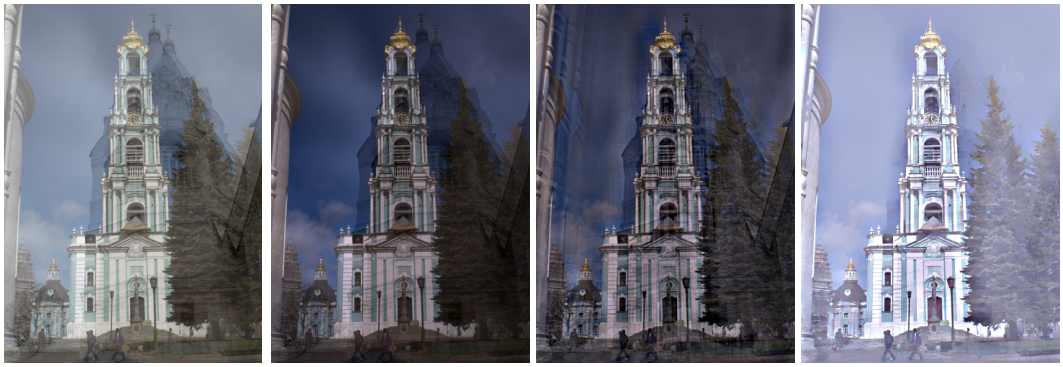
where I_i is the input image value at i^{th} pixel, similar to T_i and R_i. j is the ghosting pixel corresponding to i, *i.e.* $j = i + \mathbf{d}_k$. u_T and u_R are the consensus variables of T and R, see details in [1]. Given the ADMM parameter ρ, the above is a standard quadratic optimization with three variables T_i, R_i, R_j. The solution can be derived by having the derivatives of these variables to zeros.

Gradient sub-problems We express ∇T by $T_i - T_j$, where i and j are neighboring pixels at four directions, then we solve the following discretized gradient sub-problem for each (i, j) pair:

$$\arg\min_{T_i,T_j} \|T_i - T_j\|_1 + \frac{\rho}{2}\left(\|T_i - u_T^i\|_2^2 + \|T_j - u_T^j\|_2^2\right) \tag{3}$$

The above L_1 optimization can be analytically solved by L_1 shrinkage. For clarity, we substitute T_i by $\frac{v_i+v_j}{2}$, and T_j by $\frac{v_i-v_j}{2}$, then the above equation is re-written as below:

$$\arg\min_{v_i,v_j} \|v_j\|_1 + \frac{\rho}{2}\left(\|\frac{v_i + v_j}{2} - u_T^i\|_2^2 + \|\frac{v_i - v_j}{2} - u_T^j\|_2^2\right) \tag{4}$$

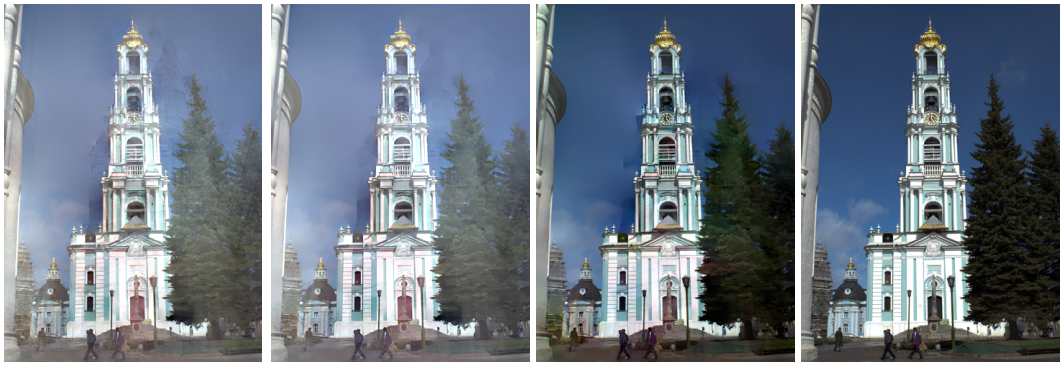


PSNR: 7.04 dB
SSIM: 0.4012
(a) Synthetic input

7.02 dB
0.4310
(b) Without ghosting cues

7.13 dB
0.4271
(c) L_2 on gradients

8.35 dB
0.4493
(d) L_1 on gradients



13.22 dB
0.5267
(e) Sparsity-inducing filters

14.01 dB
0.5499
(f) GMM patch prior

26.76 dB
0.9083
(g) Non-negativity
(Our method)

-
-
(h) Ground truth

Figure 3: *Transmission layers recovered with different image priors: (a) The separation is very poor without ghosting cues; (b) L_2 prior on gradients ($\|\nabla T\|_2$ and $\|\nabla R\|_2$), (c) L_1 prior on gradients ($\|\nabla T\|_1$ and $\|\nabla R\|_1$); (c) sparsity-inducing filter set that includes gradients and Laplacians; (d) GMM patch priors [2], and (f) GMM + non-negativity constraints. (f) achieves the best result among others, and is close to the ground truth (h). Corresponding reflection layers are in Figure 4.*

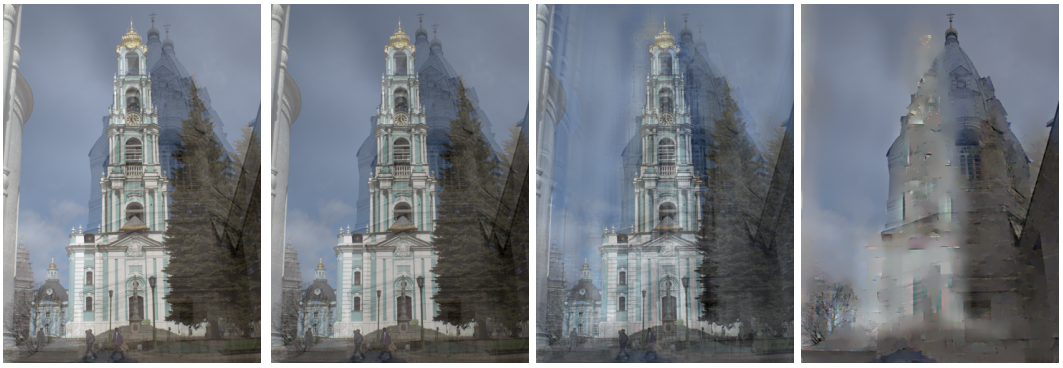
The solution is given by having the derivatives of v_i and v_j to be zeros. Here we give the solution:

$$\begin{aligned} v_i &= \frac{T_i + T_j}{2} \\ v_j &= \text{sign}(\frac{T_i - T_j}{2}) \max(|\frac{T_i - T_j}{2}| - \frac{1}{\rho}, 0); \end{aligned} \tag{5}$$

The $\|\nabla R\|_1$ sub-problems are solved in the similar way.

Laplacian sub-problems Similar to gradient problems, we solve the following equation for Laplacian sub-problems:

$$\arg\min_{T_i,T_j,T_k} \|T_i + T_k - 2T_j\|_1 + \frac{\rho}{2}\left(\|T_i - u_T^i\|_2^2 + \|T_j - u_T^j\|_2^2 + \|T_k - u_T^k\|_2^2\right) \tag{6}$$



PSNR: 8.04 dB
SSIM: 0.4132
(a) Synthetic input

8.22 dB
0.4740
(b) Without ghosting cues

8.43 dB
0.5217
(c) L_2 on gradients

18.03 dB
0.6279
(d) L_1 on gradients

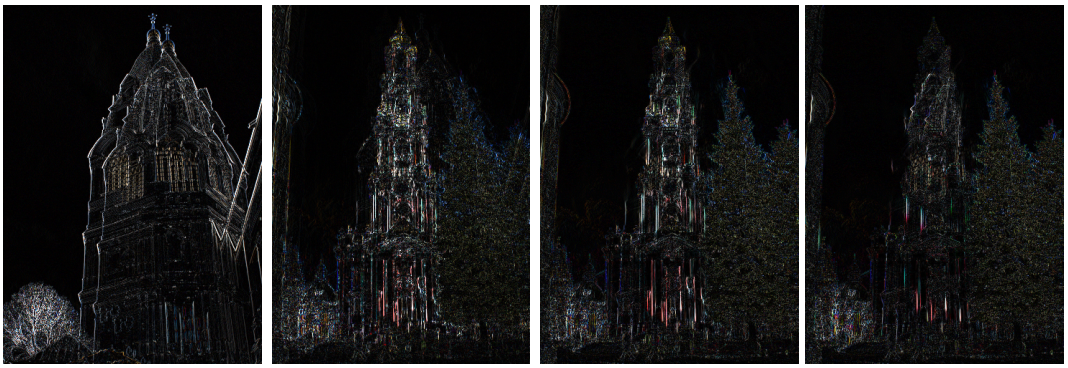
13.52 dB
0.5530
(e) Sparsity-inducing filters

13.39 dB
0.5400
(f) GMM patch prior

29.48 dB
0.9157
(g) Non-negativity
(Our method)

-
-
(h) Ground truth

Figure 4: *Reflection layers recovered with different image priors. Our choice at (g) (GMM + non-negativity) achieves the best result among other priors.*



(a) Synthetic input

(b) Sparsity-inducing filters

(c) GMM patch prior

(d) Non-negativity (ours)

Figure 5: *We show gradient errors of the transmission layers in Figure 3 against the ground truth. Our method (d) achieves minimal errors among other methods.*

where i, j, k are three neighboring pixels in horizontal and vertical directions. The above equation can be solved after variable substitutions, similar to the gradient problems. Here we give the solutions at the below:

$$\begin{aligned} \lambda &\equiv \frac{1}{2}\left(u_T^i - 2u_T^j + u_T^k\right) \\ \gamma &\equiv \text{sign}(\lambda)\max(|\lambda| - \frac{1}{\rho}, 0) \\ T_i &= \frac{1}{6}\left(5u_T^i + 2u_T^j - u_T^k\right) + \frac{1}{3}\gamma \\ T_j &= \frac{1}{3}\left(u_T^i + u_T^j + u_T^k\right) - \frac{2}{3}\gamma \\ T_k &= \frac{1}{6}\left(-u_T^i + 2u_T^j + 5u_T^k\right) + \frac{1}{3}\gamma \end{aligned} \tag{7}$$

4. Results on Real-world Inputs

Figure 10 shows all the reflection layers of the results in our main paper. In general, transmission layers T are visually better than reflection layers R. This asymmetry comes from the usage of ghosting cues in image formation model ($I = T + R \otimes k$). We show both indoor and outdoor scenes, and different times of a day.

5. Ground Truth Experiments

We conducted ground truth experiments using setup illustrated in Figure 6. The two posters in Figure 6 are used for transmitted and reflected objects. We then open the window to take the ground truth data. Figure 7 shows that our results are visually close to the ground truth.

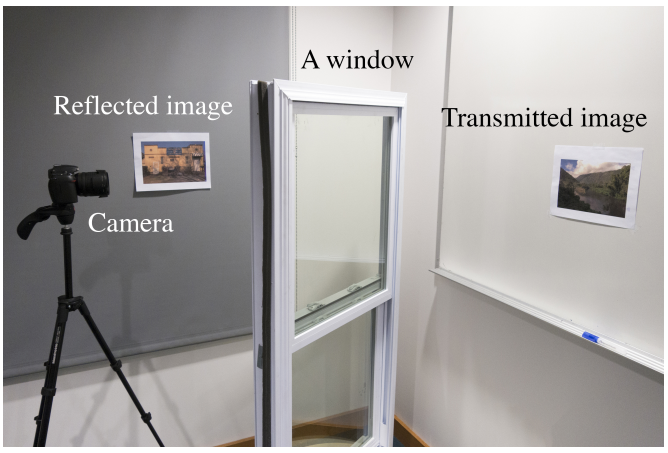
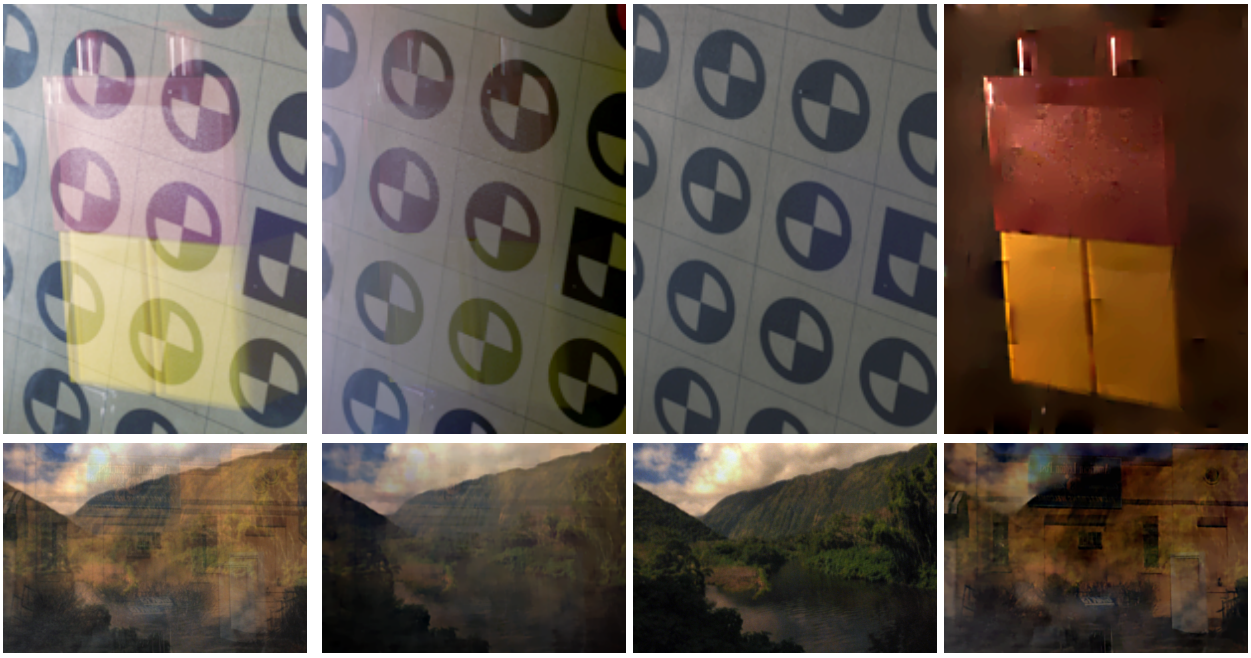


Figure 6: *We take the ground truth image by opening the window. Results are shown in Figure 7.*



Input | Our transmission | Ground truth transmission | Our reflection

Figure 7: *We show ground truth experiments on an in-lab setup. The ground truth transmissions are obtained by removing the glass. Our results are visually close to the ground truth.*

6. Applications

Automatic deghosting Figure 8 shows an application of our algorithm for auto-deghosting. In product photography, the product is often placed on a reflective surface for aesthetic reasons. An example of the resulting image is shown in Figure 8(a). We use our method to decompose the input into the transmission and the reflection layers, and then additively remix them to create the ghosting-free result, shown in Figure 8(b).

(a) *Input* **(b)** *Automatic deghosted*

Figure 8: *An application of automatic de-ghosting using our algorithm.*

7. Measuring Two-pulse Ghosting Kernels

We show that ghosting effects can be well modeled by a two-pulse kernel in Figure 9. Using the measurement setup in Figure 9, we shine the light from a circular iris, and capture the reflection on the glass. The response shown in the inset clearly shows two circles, which are separated by some distance, and one is dimmer than the other. The brighter one is reflected from the surface closer to the camera, and the dimmer one is from the outer surface.

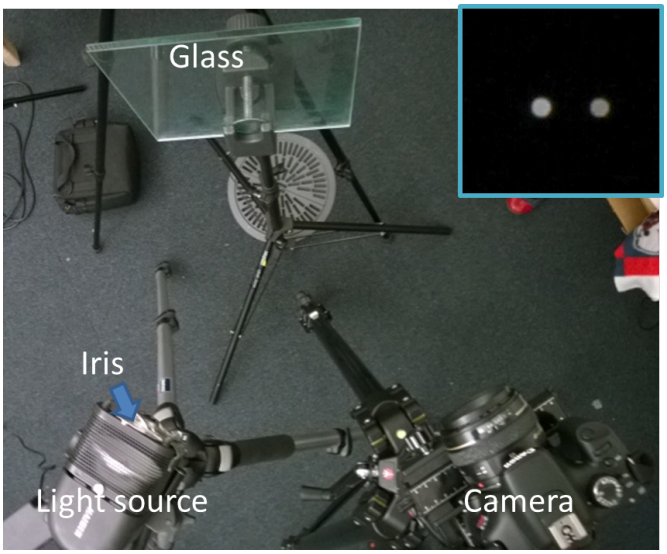
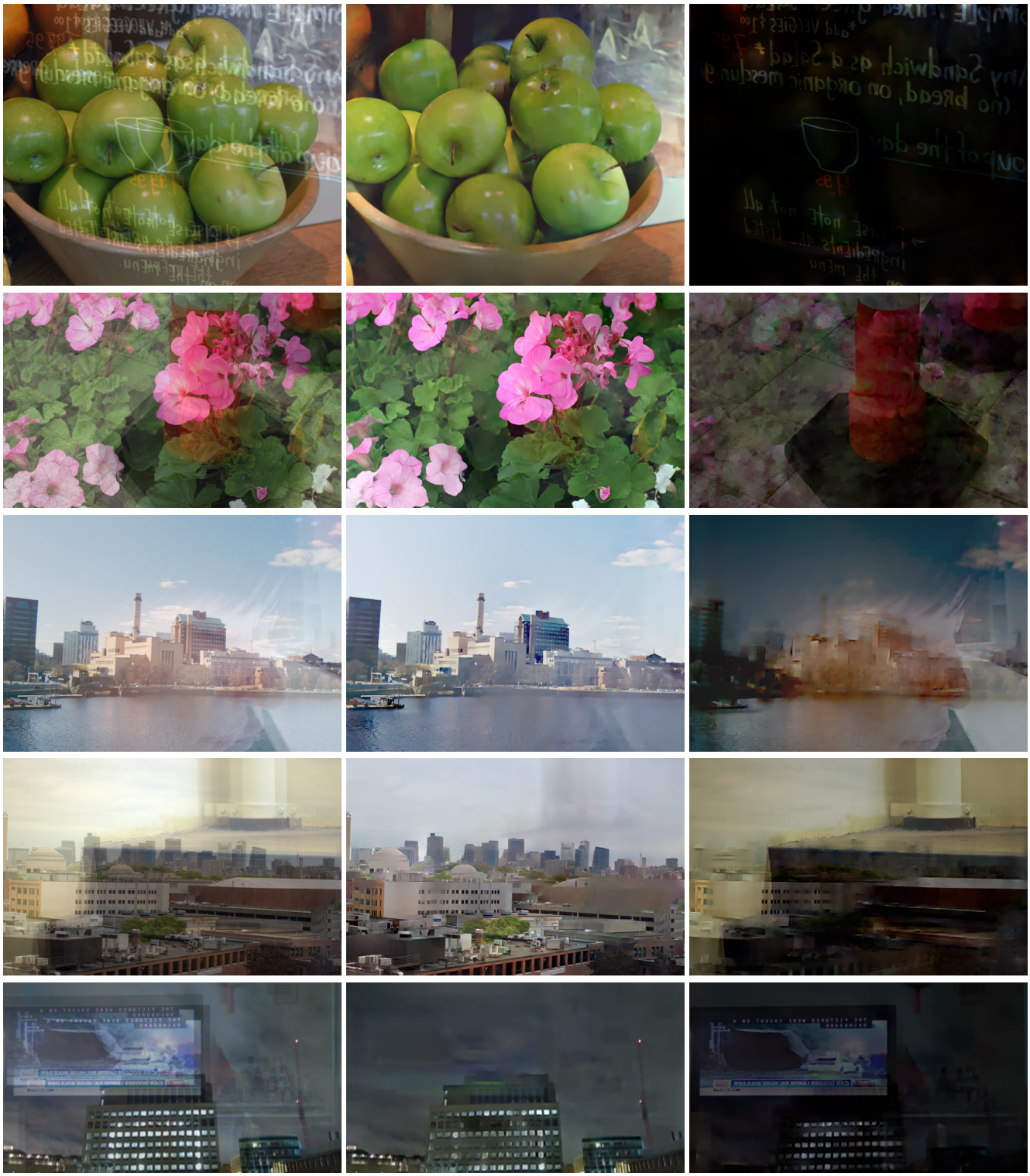


Figure 9: *We show that ghosting effects can be well modeled by a two-pulse kernel through the above experiment. We shine a light through a circular iris, and capture the reflection from the glass. The captured image shown in the inset shows two circles, which are separated and one is dimmer than the other one.*

References

[1] J. Bento, N. Derbinsky, J. Alonso-Mora, and J. S. Yedidia. A message-passing algorithm for multi-agent trajectory planning. In *NIPS*, pages 521–529, 2013. 2

[2] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *IEEE International Conference on Computer Vision*, pages 479–486, 2011. 3



Input (I_{in}) Recovered transmission (T) Recovered reflection (R)

Figure 10: *Our results on real-world inputs. The transmissions from the top to the bottom rows: apples,flowers, river side, a city skyline, and a building at night. The corresponding reflections are texts, a traffic cone, a face profile facing right side, a water pipe, and a television screen.*