Reflection Removal by Ghost Effect from A Single Image

Yichang Shih
CSAIL, Massachusetts Institutes of Technology

Abstract

When we take an image through a window, the glass reflects the light rays from the photographer side. Hence, the obtained photo is a linear superposition of two images: a scene image beyond the window plus a reflection image before the window. Removing reflection image is traditionally regarded as a heavily ill-posed problem, because there are infinite possible decompositions for reflection / scene image pairs.

However, we observe that if there is a non-zero angle between the camera optical axis and the glass normal, the reflection image appears a "ghost effect" due to the thickness of the glass. This ghost effect can be used as a cue to distinguish the reflection image from the scene image. In this paper, we propose a reflection removal method using this ghost effect. In particular, we separate edges of the scene image from edges of the reflection image, and then reconstruct the scene image from its gradients. The result shows that our method can remove most reflection image, and preserve the scene image.

1. Introduction

In many situations, we need to take an image through a transparent glass or a window. For instance, taking an antique photo in the museum, a product photo during a window shopping, and a night view photo at the top floor of a skycraper. When taking these photos, the glass reflects light rays from photographer side. Hence, the obtained image is mixed by scene image and reflection image. In general. there are two types of reflection: (1) reflection caused by flash light from a camera and (2) caused by scene light from the user side. Agrawal has studied the reflection caused by flash light [1], while in our paper, we address the issue that reflection is caused by the scene light instead of flash light, as shown in Fig.1.

It is desirable to remove the reflection image from the obtained image. However, the problem is heavily ill-posed. Traditionally, the obtained image $I(x, y)$ can be represented

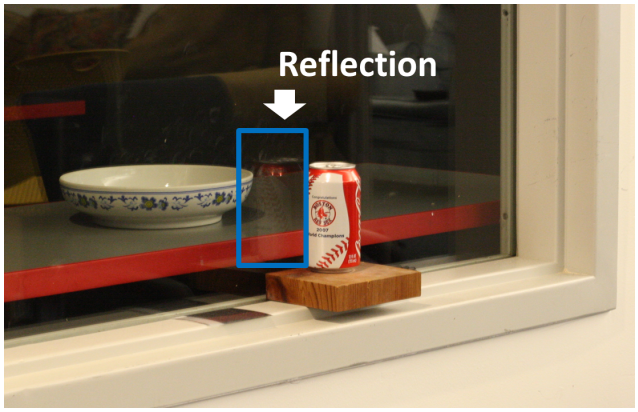


Figure 1. An example of reflection image caused by scene objects at the photographer side.

by

$$I(x, y) = I_s(x, y) + I_r(x, y) \quad (1)$$

where $I_s(x, y)$ is the scene image and $I_r(x, y)$ is the reflection image. Reflection removal is the task that given input image $I(x, y)$, solve for scene image $I_s(x, y)$. Apparently, without the knowledge of $I_r(x, y)$ or $I_s(x, y)$, there are infinite possible valid combination, and it is unlikely to distinguish $I_s(x, y)$ and $I_r(x, y)$.

However, we observed that, as shown in Fig.2, when there is an non-zero angle between the camera optical axis and the glass normal, camera will see two duplicates of reflection image. This is because each side of the glass generates a reflection image, and the difference between the two duplicates is only up to a scale in image intensity. In this paper, we call this effect ghost effect. Figures 3 shows an example of ghost effect. Ghost effect is obvious when (1) the glass is thick enough (typically around more than 0.5 cm) and (2) angle between camera optical axis and glass normal is around 45 degrees, and (3) the reflected object is close to the glass (typically around 20 cm).

Ghost effect can be a good criteria to distinguish the reflection image and the scene image: reflection image has

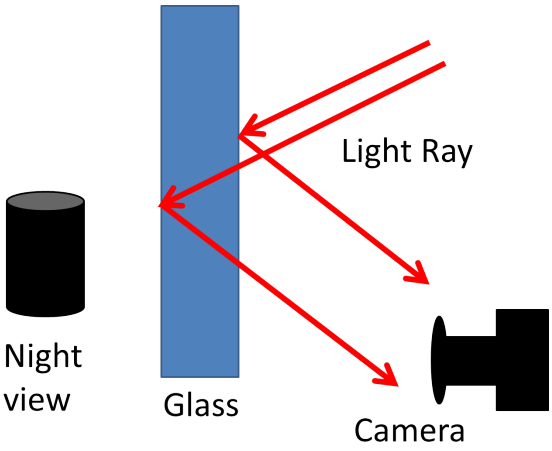


Figure 2. When there is a non-zero angle between camera optical center and glass normal, the camera will record a duplicate image of reflection image. We call this duplication "ghost effect".

a duplicate at a fixed distance, while the scene image does not have such duplicates. For human being, it is easy to distinguish reflection and desired image by ghost effect, but for computer, the two images are still mixed together and hard to be separated. Fortunately, we observed that even the images are mixed together, the edges of the two image are usually have less overlap. Hence, instead of decomposing the two images directly, we separate the gradient of the two images, and then reconstruct the scene image from scene gradient.

We should emphasize that using ghost effect for reflection removal does not lose generality. In most cases, the glasses are thick enough to produce ghost effect, and the photographer can choose the angle between camera and glass normal to make ghost effect more obvious for reflection removal.

1.1. Related Work

Reflection Removal by Polarized Lens For most professional photographers, they reduce reflection by putting a polarized lens in front of the camera lens. According to Maxwell equations, reflected light has a certain amount of polarized light, and polarized lens can eliminate some of the reflection. By rotating polarized lens and taking multiple images, the obtained images are the scene image plus reflection images of different intensity levels. Hence, the scene image can be extracted from these multiple images [6]. While these methods are useful to professional photographers, more than one image are required. In our work, we focus on reflection removal from a single image, without any additional hardware (ex: polarized lens).



Figure 3. The coke can is the reflection image, while the plate and the desk are desired image. Coke can has ghost effect, while plate and desk does not. The image is taken at 45 degree between the camera and the glass normal

Reflection Removal from a Single Image Levin has proposed a simple criteria that the two decomposed image $I_s(x, y)$ and $I_r(x, y)$ has minimum number of corners. [4]. With this intuitive criteria, they can identify which combination of $I_s(x, y)$ and $I_r(x, y)$ is the valid one, and so remove the reflection image. Their method is neat and has pretty good result, but requires a large data base. Besides, while their criteria is heuristic, our criteria is based on physics: ghost effect is a physical phenomenon because of the thickness of the glass.

Because the main challenge is to distinguish desired image and reflection image, Levin proposed a image separation by annotation from a user [3]. Instead of depending on users, our method in general automatically annotate the image by using ghost effect.

Gradient Domain Image Editing In out work, we first separate the gradient of the scene image from the reflection image, and then select the gradient of the scene image to reconstruct the scene image. The reconstruction is gradient domain image editing [5]. In our work, because we heavily modify gradient domain, so the image field is far away from conservative field, and so [5]'s method does not work well. Instead, we use [7]'s trick for reconstruction, which give us a visually good result.

2. Image Model for Ghost Effect

In this section, we discuss ghost effect generation model. In general, the obtained image $I(\vec{x})$ can be represented by

$$I(\vec{x}) = I_s(\vec{x}) + \sum_i I_{r,i}(\vec{x}) + \beta I_{r,i}(\vec{x} - \vec{d_i}); \qquad (2)$$

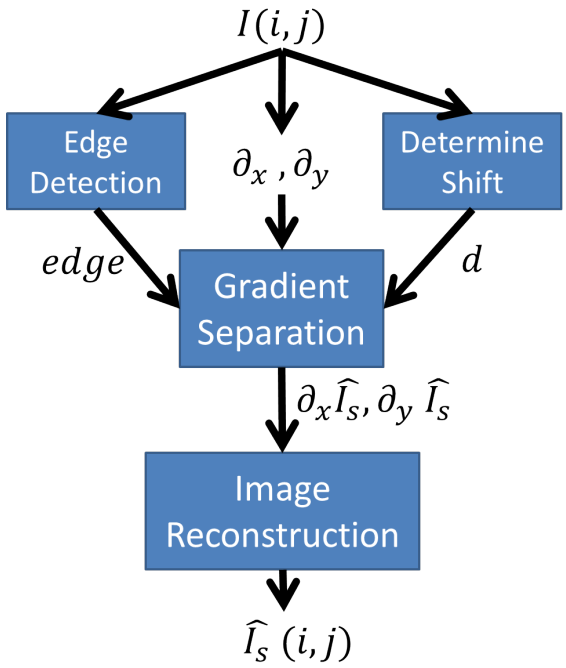


Figure 4. Algorithm over view of our proposed reflection removal.

where $I_s(\vec{x})$ is the scene image, $I_{r,i}(\vec{x})$ is the reflection image of the i^{th} object, β is the intensity attenuation caused by the glass, and d_i the shift of the reflection image of the i^{th} object. The shift is caused by ghost effect.

For simplicity, now we consider the case only one single reflection object in the image, i.e. Eq. 2 becomes

$$I(\vec{x}) = I_s(\vec{x}) + I_r(\vec{x}) + \beta I_r(\vec{x} - \vec{d}) \tag{3}$$

Figure 3 shows an example of the above equation (coke can is the only reflection object in the image).

β, the attenuation coefficient, is a function of the glass' property and the glass' geometry. If the glass absorbs more energy when light rays are passing through, then β is lower. β also depends on material property of air and glass. A complete derivation of β can be found in [2]. Typically, β is about 0.5.

d is the distance between two duplicate reflection images, and is a function depending on relative geometric relation between the reflection object, the glass, and the camera. Typically, if the angle between camera and glass normal is greater than 15 degrees, the ghost effect is obvious enough for our purpose.

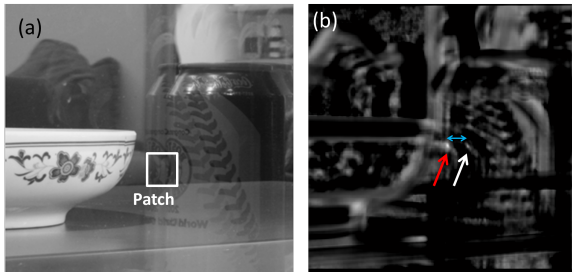


Figure 5. Determine shift amount of ghost effect. (a) The obtained image from camera. White square: the patch (b) Template matching result. Red arrow indicates the global maximum, which is the location of the patch itself. White arrow indicates a local maximum, which is the location of the duplicate caused by ghost effect. The distance between these two maximum indicates the shift amount.

3. Reflection Removal

Reflection removal from a single image is equivalent to solve for $I_r(\vec{x})$ and $I_s(\vec{x})$ in Eq. 3 given $I(\vec{x})$. Even with the knowledge given by ghost effect, directly solving for Eq. 3 still seems impossible. Our first try is to treat $I_s(\vec{x})$ as a noise, and solve $I_r(\vec{x})$ by deconvolving $I(\vec{x})$ using kernel $\delta(\vec{x}) + \beta\delta(\vec{x} - \vec{d})$. The deconvolution can remove ghost effect, but the reflection image and the scene image are still mixed together (i.e. the output result looks like $I_{out}(\vec{x}) = I_s(\vec{x}) + I_r(\vec{x})$.) This method does not give us a good result because $I_s(\vec{x})$ is too far away from noise.

In image domain, the reflection image and the scene image are mixed together, so we cannot separate the two images by assigning each pixel to one of the two images. Instead, we focus on gradient domain. We observe that the edges of reflection image and desired image does not have much overlap, so we are able to assign the gradient of each pixel to one of the two images. In our work, we collect the gradient of the scene image by what we call gradient separation technique, and then reconstruct the scene image accordingly.

3.1. Shift Amount Determination

Before separating gradient, we need to determine the shift amount $\vec{d}$ between the two reflection duplicates. In our work, we design a user assisted system to determine the shift amount. A user first cropped a patch of reflection image as a template, and do template matching on the original image. The ghost effect causes a second local maximum on template matching result, and so we can determine the shift amount $\vec{d}$, as shown in Fig. 5.

3.2. Gradient Separation

Given 2 gradient inputs $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$, the task of gradient separation is to determine each gradient belongs to either the reflection image or the scene image. We utilize ghost effect in this part. For each pixel (i_0, j_0) , we cropped a patch P by a window function center (i_0, j_0) i.e. $P = W(i - i_0, j - j_0)I(i, j)$. Typically, the window size is 20 pixels. Then we do template matching between the patch and the original image. If (i_0, j_0) corresponds to the reflection image, then there is a duplicate at $(i_0, j_0) + \vec{d}$ or $(i_0, j_0) - \vec{d}$, and so there is a local maximum on template matching result S at the corresponding position. Otherwise, if P is corresponding to an edge of desired image, then it is unlikely there is a local maximum on S at $(i_0, j_0) + \vec{d}$ or $(i_0, j_0) - \vec{d}$.

In particular, we first translate $S(i, j)$ so that the cropped patch is at the origin, i.e. we denote the translated result by $S'(i, j)$, and so $S'(0, 0) = 1$. Then

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

Running template matching on each pixel costs a huge amount of calculation, and it is unnecessary to run on every pixel. Instead, we only run the pixels of the important gradient. In particular, we first use an edge detection to label strong gradient, and then only separate these important gradients. In other words, we discard weak edges. A user can setup lower threshold for edge detection to preserve more gradients.

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].

3.4. Algorithm Overview

Figure 4 shows the algorithm flow chart. Given input image (obtained by a camera), we first determine the shift amount of ghost effect and the important gradient by edge detection. Then, according to ghost effect, we separate the gradient of the reflection image from the gradient of the desired image. Then we discard the gradient of the reflection image, and then reconstruct the desired image by the scene gradient.

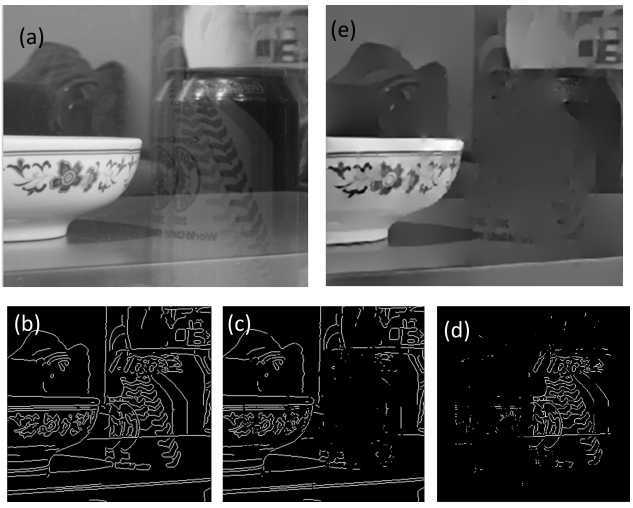


Figure 6. A result from our method. (a) Input image. The coke can is the reflection image. (b) Edge map of the input image. (c,d) gradient separation result: (c) edge of the scene image (d) edge of the reflection image. (e) Reconstructed desired image. The coke can is removed, while the plate is preserved.

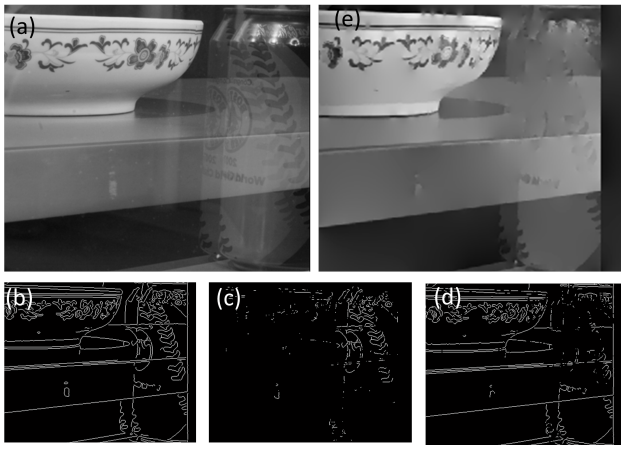


Figure 7. One more result from our method. (a) Input image. The coke can is the reflection image. (b) Edge map of the input image. (c,d) gradient separation result: (c) edge of the scene image (d) edge of the reflection image. (e) Reconstructed desired image. The coke can is removed, while the desk is preserved.

4. Result

Figure 6 shows the result of each step in the algorithm. In Fig. 6, the shift amount between 2 coke can image is (22,1). We use canny algorithm to determine the edge map, using threshold = 0.1. Because canny algorithm only preserve important edges, so the weak edge, corresponding to weak gradient, are discarded. Hence, the reconstructed image looks smoother.

Figure 7 shows one more example.

5. Discussion

Our method can remove most reflection image, while preserve the desired image. By choosing different thresholds of edge detection algorithm, a user can choose the computational task amount and details to be preserved.

One major limitation on our method is the obtained image must have strong ghost effect. When a user is perpendicular to the glass (i.e the angle between camera and glass normal is zeros), then d is zero, and there is no ghost effect, and so our method fails. Besides, if the reflection object is too far away from the glass, or the glass is too thin, or the glass absorbs a lot of energy when light rays are passing through (β is too small), then the ghost effect is less obvious.

The key step in our method is gradient separation. However, when the reflection image is too complex, and has too much overlap, our method cannot work well. Besides, we determine a gradient to be the reflection image or the scene image by checking if there is a duplicate at the distance d. However, if the scene image is highly textured, and itself has a lot of duplicates with period d, then our method will regard scene gradient as a reflection gradient, and so fails.

6. Conclusion

We propose a reflection removal technique from a single image by using ghost effect. Instead of using heuristic criteria for choosing the valid composition of reflection image and scene image, our criteria is physics-based, and so more reliable.

References

[1] A. Agrawal, R. Raskar, S. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. In *ACM SIGGRAPH 2005 Papers*, pages 828–835. ACM, 2005.

[2] D. Griffiths and R. College. *Introduction to electrodynamics*, volume 3. prentice Hall New Jersey;, 1999.

[3] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. *Computer Vision-ECCV 2004*, pages 602–613, 2004.

[4] A. Levin, A. Zomet, and Y. Weiss. Separating reflections from a single image using local features. 2004.

[5] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 313–318. ACM, 2003.

[6] Y. Schechner, J. Shamir, and N. Kiryati. Polarization-based decorrelation of transparent layers: The inclination angle of an invisible surface. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 814–819. IEEE, 1999.

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