Exploiting Reflection Change for Automatic Reflection Removal

Yu Li Michael S. Brown
School of Computing, National University of Singapore
liyu@nus.edu.sg | brown@comp.nus.edu.sg

Abstract

This paper introduces an automatic method for removing reflection interference when imaging a scene behind a glass surface. Our approach exploits the subtle changes in the reflection with respect to the background in a small set of images taken at slightly different view points. Key to this idea is the use of SIFT-flow to align the images such that a pixel-wise comparison can be made across the input set. Gradients with variation across the image set are assumed to belong to the reflected scenes while constant gradients are assumed to belong to the desired background scene. By correctly labelling gradients belonging to reflection or background, the background scene can be separated from the reflection interference. Unlike previous approaches that exploit motion, our approach does not make any assumptions regarding the background or reflected scenes' geometry, nor requires the reflection to be static. This makes our approach practical for use in casual imaging scenarios. Our approach is straight forward and produces good results compared with existing methods.

1. Introduction and Related Work

There are situations when a scene must be imaged behind a pane of glass. This is common when "window shopping" where one takes a photograph of an object behind a window. This is not a conducive setup for imaging as the glass will produce an unwanted layer of reflection in the resulting image. This problem can be treated as one of layer separation [7, 8], where the captured image I is a linear combination of a reflection layer I_R and the desired background scene, I_B, as follows:

$$I = I_R + I_B. \quad (1)$$

The goal of reflection removal is to separate I_B and I_R from an input image I as shown in Figure 1.

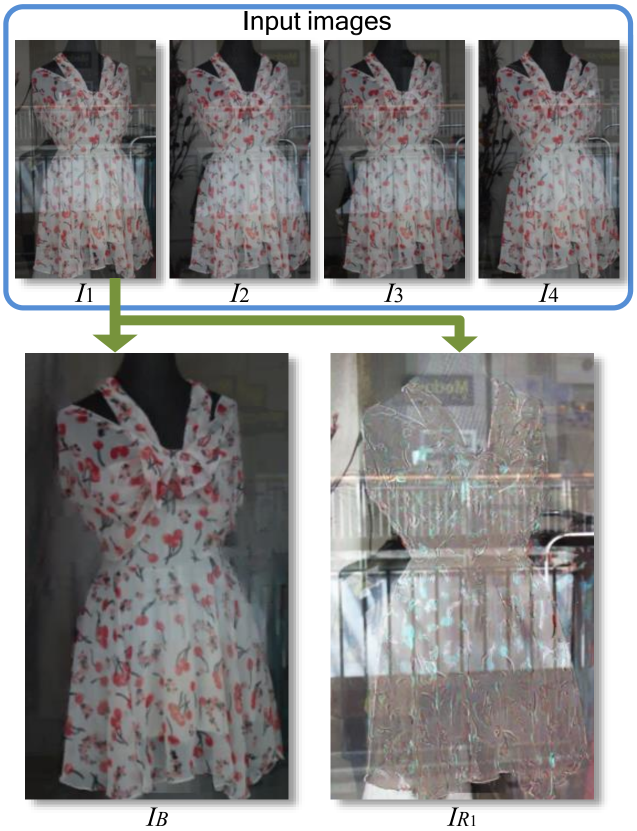

Fig. 1. Example of our approach separating the background (I_B) and reflection (I_R) layers of one of the input images. Note that the reflection layer's contrast has been boosted to improve visualization.

This problem is ill-posed, as it requires extracting two layers from one image. To make the problem tractable additional information, either supplied from the user or from multiple images, is required. For example, Levin and Weiss [7, 8] proposed a method where a user labelled image gradients as belonging to either background or reflection. Combing the markup with an optimization that imposed a sparsity prior on the separated images, their method produced compelling results. The only drawback was the need for user intervention. An automatic method was proposed by Levin *et al.* [9] that found the most likely decomposition which minimized the total number of edges and corners in the recovered image using a database of natural images. As with example-based methods, the results were reliant on the similarity of the examples in the database.

Another common strategy is to use multiple images. Some methods assume a fixed camera that is able to capture a set of images with different mixing of the layers through various means, e.g. rotating a polarized lens [3, 6, 12, 16, 17], changing focus [15], or applying a flash [1]. While these approaches demonstrate good results, the ability of controlling focal change, polarization, and flash may not always be possible. Sarel and Irani [13, 14] proposed video based methods that work by assuming the two layers, reflection and background, to be statistically uncorrelated. These methods can handle complex geometry in the reflection layer, but require a long image sequence such that the reflection layer has significant changes in order for a median-based approach [21] to extract the intrinsic image from the sequence as the initial guess for one of the layers.

Techniques closer to ours exploit motion between the layers present in multiple images. In particular, when the background is captured from different points of view, the background and the reflection layers undergo different motions due to their different distance to the transparent layer. One issue with changing viewpoint is handling alignment among the images. Szeliski *et al*. [19] proposed a method that could simultaneously recover the two layers by assuming they were both static scenes and related by parametric transformations (i.e. homographies). Gai *et al*. [4, 5] proposed a similar approach that aligned the images in the gradient domain using gradient sparsity, again assuming static scenes. Tsin *et al*. [20] relaxed the planar scene constraint in [19] and used dense stereo correspondence with stereo matching configuration which limits the camera motion to unidirectional parallel motion. These approaches produce good results, but the constraint on scene geometry and assumed motion of the camera limit the type of scenes that can be processed.

Our Contribution Our proposed method builds on the single-image approach by Levin and Weiss [8], but removes the need for user markup by examining the relative motion in a small set (*e.g*. 3-5) of images to automatically label gradients as either reflection or background. This is done by first aligning the images using SIFT-flow and then examining the variation in the gradients over the image set. Gradients with more variation are assumed to be from reflection while constant gradients are assumed to be from the desired background. While a simple idea, this approach does not impose any restrictions on the scene or reflection geometry. This allows a more practical imaging setup that is suitable for handheld cameras.

The remainder of this paper is organized as follows. Section 2 overviews our approach; section 3 compares our results with prior methods on several examples; the paper is concluded in section 4.

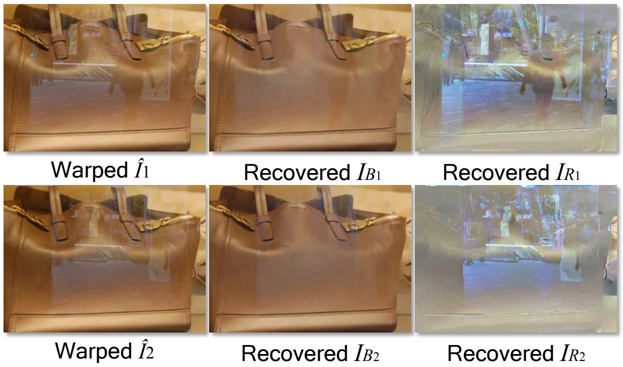

Fig. 2. This figure shows the separated layers of the first two input images. The layers illustrate that the background image I_B has little variation while the reflection layers, I_{R_i}, have notable variation due to the viewpoint change.

2. Reflection Removal Method

2.1. Imaging Assumption and Procedure

The input of our approach is a small set of k images taken of the scene from slightly varying view points. We assume the background dominates in the mixture image and the images are related by a warping, such that the background is registered and the reflection layer is changing. This relationship can be expressed as:

$$I_i = w_i(I_{R_i} + I_B), \tag{2}$$

where I_i is the i-th mixture image, $\{w_i\}$, $i = 1, \ldots, k$ are warping functions caused by the camera viewpoint change with respect to a reference image (in our case I_1). Assuming we can estimate the inverse warps, w_i^{-1}, where w_1^{-1} is the identity, we get the following relationship:

$$w_i^{-1}(I_i) = I_{R_i} + I_B. \tag{3}$$

Even though I_B appears static in the mixture image, the problem is still ill-posed given we have more unknowns than the number of input images. However, the presence of a static I_B in the image set makes it possible to identify gradient edges of the background layer I_B and edges of the changing reflection layers I_{R_i}. More specifically, edges in I_B are assumed to appear every time in the image set while the edges in the reflection layer I_{R_i} are assumed to vary across the set. This reflection-change effect can be seen in Figure 2. This means edges can be labelled based on the frequency of a gradient appearing at a particular pixel across the aligned input images. After labelling edges as either background or reflection, we can reconstruct the two layers using an optimization that imposes the sparsity prior on the separated layers as done by [7, 8]. Figure 3 shows the processing pipeline of our approach. Each step is described in the following sections.

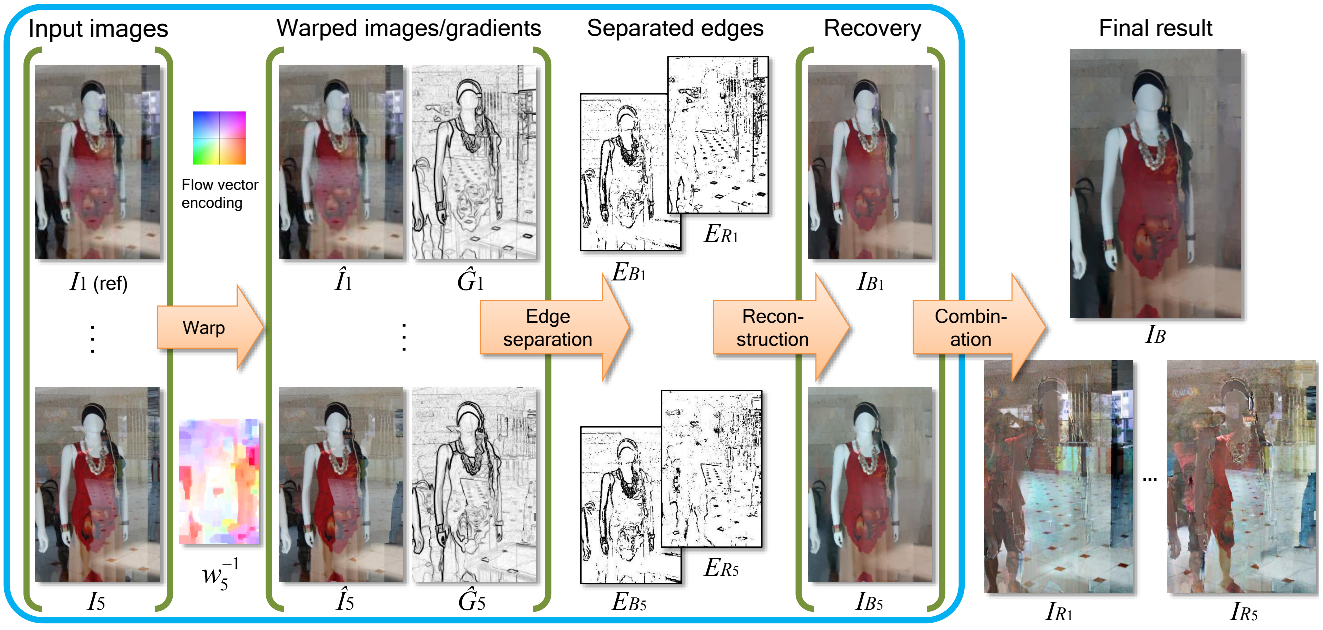

Fig. 3. This figure shows the pipeline of our approach: 1) warping functions are estimated to align the inputs to a reference view; 2) the edges are labelled as either background or foreground based on gradient frequency; 3) a reconstruction step is used to separate the two layers; 4) all recovered background layers are combined together to get the final recovered background.

2.2. Warping

Our approach begins by estimating warping functions, w_i^{-1}, to register the input to the reference image. Previous approaches estimated these warps using global parametric motion (*e.g.* homographies [4, 5, 19]), however, the planarity constraint often leads to regions in the image with misalignments when the scene is not planar.

Traditional dense correspondence method like optical flow is another option. However, even with our assumption that the background should be more prominent than the reflection layer, optical flow methods (e.g. [2, 18]) that are based on image intensity gave poor performance due to the reflection interference. This led us to try SIFT-flow [10] that is based on more robust image features. SIFT-flow [10] proved to work surprisingly well on our input sequences and provide a dense warp suitable to bring the images into alignment even under moderate interference of reflection. Empirical demonstration of the effectiveness of SIFT-flow in this task as well as the comparison with optical flow are shown in our supplemental materials.

Our implementation fixes I_1 as the reference, then uses SIFT-flow to estimate the inverse-warping functions $\{w_i^{-1}\}$, $i = 2, \ldots, k$ for each of the input images $I_2, \ldots, I_k$ against I_1. We also compute the gradient magnitudes G_i of the each input image and then warp the images I_i as well as the gradient magnitudes G_i using the same inverse-warping function w_i^{-1}, denoting the warped images and gradient magnitudes as $\hat{I}_i$ and $\hat{G}_i$.

2.3. Edge separation

Our approach first identifies salient edges using a simple threshold on the gradient magnitudes in $\hat{G}_i$. The resulting binary edge map is denoted as E_i. After edge detection, the edges need to be separated as either background or foreground in each aligned image $\hat{I}_i$. As previously discussed, the edges of the background layer should appear frequently across all the warped images while the edges of the reflection layer would only have sparse presence. To examine the sparsity of the edge occurrence, we use the following measurement:

$$\Phi(\mathbf{y}) = \frac{\|\mathbf{y}\|_2^2}{\|\mathbf{y}\|_1^2}, \tag{4}$$

where $\mathbf{y}$ is a vector containing the gradient magnitudes at a given pixel location. Since all elements in $\mathbf{y}$ are non-negative, we can rewrite equation 4 as $\Phi(\mathbf{y}) = \sum_{i=1}^{k} y_i^2 / (\sum_{i=1}^{k} y_i)^2$. This measurement can be considered as a L_1 normalized L_2 norm. It measures the sparsity of the vector which achieves its maximum value of 1 when only one non-zero item exists and achieve its minimum value of $\frac{1}{k}$ when all items are non-zero and have identical values (*i.e.* $y_1 = y_2 = \ldots = y_k > 0$). This measurement is used to assign two probabilities to each edge pixel as belonging to either background or reflection.

We estimate the reflection edge probability by examining the edge occurrence, as follows:

$$P_{R_i}(\mathbf{x}) = s\left(\frac{\sum_{i=1}^{k}\hat{G}_i(\mathbf{x})^2}{(\sum_{i=1}^{k}\hat{G}_i(\mathbf{x}))^2} - \frac{1}{k}\right), \tag{5}$$

where, $\hat{G}_i(\mathbf{x})$ is the gradient magnitude at pixel $\mathbf{x}$ of $\hat{I}_i$. We subtract $\frac{1}{k}$ to move the smallest value close to zero. The sparsity measurement is further stretched by a sigmoid function $s(t) = (1 + e^{-(t-0.05)/0.05})^{-1}$ to facilitate the separation. The background edge probability is then estimated by:

$$P_{B_i}(\mathbf{x}) = s\left(-\left(\frac{\sum_{i=1}^{k}\hat{G}_i(\mathbf{x})^2}{(\sum_{i=1}^{k}\hat{G}_i(\mathbf{x}))^2} - \frac{1}{k}\right)\right), \tag{6}$$

where $P_{B_i}(\mathbf{x}) + P_{R_i}(\mathbf{x}) = 1$. These probabilities are defined only at the pixels that are edges in the image. We consider only edge pixels with relatively high probability in either the background edge probability map or reflection edge probability map. The final edge separation is performed by thresholding the two probability maps as:

$$E_{B_i/R_i}(\mathbf{x}) = \begin{cases} 1, & E_i(\mathbf{x}) = 1 \text{ and } P_{B_i/R_i}(\mathbf{x}) > 0.6 \\ 0, & \text{otherwise} \end{cases}$$

Figure 4 shows the edge separation procedure.

2.4. Layer Reconstruction

With the separated edges of the background and the reflection, we can reconstruct the two layers. Levin and Weiss [7, 8] showed that the long tailed distribution of gradients in natural scenes is an effective prior in this problem. This kind of distributions is well modelled by a Laplacian or hyper-Laplacian distribution ($P(t) \propto e^{-|t|^p/s}$, $p = 1$ for Laplacian and $p < 1$ for hyper-Laplacian). In our work, we use Laplacian approximation since the L_1 norm converges quickly with good results. For each image $\hat{I}_i$, we try to maximize the probability $P(I_{B_i}, I_{R_i})$ in order to separate the two layers and this is equivalent to minimizing the cost $-\log P(I_{B_i}, I_{R_i})$. Following the same deduction in [7], with the independent assumption of the two layers (*i.e.* $P(I_{B_i}, I_{R_i}) = P(I_{B_i}) \cdot P(I_{R_i})$), the objective function becomes:

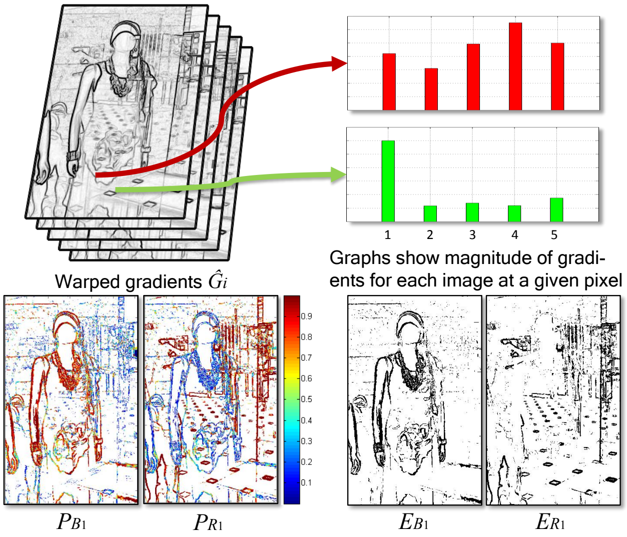
1
2
3
4
5
Graphs show magnitude of gradients for each image at a given pixel
Warped gradients $\hat{G}_i$
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
P_{B1}
P_{R1}
E_{B1}
E_{R1}

Fig. 4. Edge separation illustration: 1) shows the all $\hat{G}$ gradient maps – in this case we have five input images; 2) plots the gradient values at two position across the five images - top plot is a pixel on a background edge, bottom plot is a pixel on a reflection edge; 3) shows the probability map estimated for each layer; 4) Final edge separation after thresholding the probability maps.

$$\begin{aligned} J(I_{B_i}) = &\sum_{\mathbf{x},n} |(I_{B_i} * f_n)(\mathbf{x})| + |((\hat{I}_i - I_{B_i}) * f_n)(\mathbf{x})| \\ &+ \lambda \sum_{\mathbf{x},n} E_{B_i}(\mathbf{x})|((\hat{I}_i - I_{B_i}) * f_n)(\mathbf{x})| \\ &+ \lambda \sum_{\mathbf{x},n} E_{R_i}(\mathbf{x})|(I_{B_i} * f_n)(\mathbf{x})|, \end{aligned} \tag{7}$$

where f_n denotes the derivative filters and $*$ is the 2D convolution operator. For f_n, we use two orientations and two degrees (first order and second order) derivative filters. While the first term in the objective function keeps the gradients of the two layer as sparse as possible, the last two terms force the gradients of I_{B_i} at edges positions in E_{B_i} to agree with the gradients of input image $\hat{I}_i$ and gradients of I_{R_i} at edge positions in E_{R_i} agree with the gradients of $\hat{I}_i$. This equation can be further rewritten in the form of $J = \|Au - b\|_1$ and be minimized efficiently using iterative reweighted least square [11].

2.5. Combining the Results

Our approach processes each image in the input set independently. Due to the reflective glass surface, some of the images may contain saturated regions from specular highlights. When saturation occurs, we can not fully recover the structure in these saturated regions because the information about the two layers are lost.

In addition, sometimes the edges of the reflection in some regions are too weak to be correctly distinguished. This can lead to local regions in the background where the reflection is still present. These erroneous regions are often in different places in each input image due to changes in the reflection. In such cases, it is reasonable to assume that the minimum value across all recovered background layers may be a proper approximation of the true background. As such, the last step of our method is to take the minimum of the pixel value of all reconstructed background images as the final recovered background, as follows:

$$I_B(\mathbf{x}) = min_i\, I_{B_i}(\mathbf{x}). \tag{8}$$

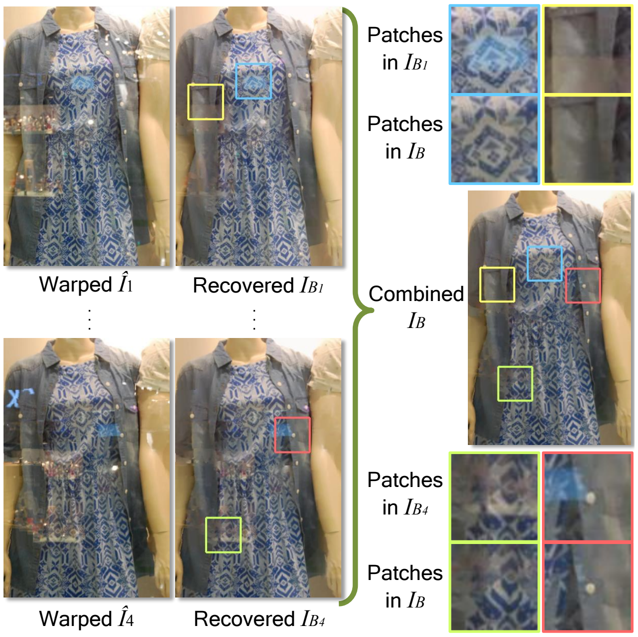

Fig. 5. This figure shows our combination procedure. The recovered background on each single image is good at first glance but may have reflection remaining in local regions. A simple minimum operator combining all recovered images gives a better result in these regions. The comparison can be seen in the zoomed-in regions.

Based on this, the reflection layer of each input image can be computed by $I_{R_i} = \hat{I}_i - I_B$. The effectiveness of this combination procedure is illustrated in Figure 5.

3. Results

In this section, we present the experimental results of our proposed method. Additional results and test cases can be found in the accompanying supplemental materials. The experiments were conducted on an Intel i7® PC (3.4GHz CPU, 8.0GB RAM). The code was implemented in Matlab. We use the SIFT-Flow implementation provided by the authors [1]. Matlab code and images used in our paper can be downloaded at the author's webpage [2]. The entire procedure outlined in Figure 3 takes approximately five minutes for a 500×400 image sequence containing up to five images. All the data shown are real scene captured under various lighting conditions (*e.g*. indoor, outdoor). Input sequences range from three to five images.

Figure 6 shows two examples of our edge separation results and final reconstructed background layers and reflection layers. Our method provides a clear separation of the edges of the two layers which is crucial in the reconstruction step. Figure 9 shows more reflection removal results of our method.

We also compare our methods with those in [8] and [5]. For the method in [8], we use the source code [3] of the author to generate the results. The comparisons between our and [8] are not entirely fair since [8] uses single image to generate the result, while we have the advantage of the entire set. For the results produced by [8], the reference view was used as input. The required user-markup is also provided. For the method in [5], we set the layer number to be one, and estimate the motions of the background layer using their method. In the reconstruction phase, we set the remaining reflection layer in k input mixture images as k different layers, each only appearing once in one mixture.

Figure 8 shows the results of two examples. Our results are arguably the best. The results of [8] still exhibited some edges from different layers even with the elaborate user mark-ups. This may be fixed by going back to further refine the user markup. But in the heavily overlapping edge regions, it is challenging for users to indicate the edges. If the edges are not clearly indicated the results tend to be over smoothed in one layer. For the method of [5], since it uses global transformations to align images, local misalignment effects often appear in the final recovered background image. Also, their approach uses all the input image into the optimization to recover the layers. This may lead to the result that has edges from different reflection layers of different images mixed and appear as ghosting effect in the recovered background image. For heavily saturated regions, none of the two previous methods can give visually plausible results like ours.

4. Discussion and Conclusion

We have presented a method to automatically remove reflectance interference due to a glass surface. Our approach works by capturing a set of images of a scene from slightly varying view points. The images are then aligned and edges are labelled as belonging to either background or reflectance. This alignment was enabled by SIFT-flow, whose robustness to the reflection interference enabled our method. When using SIFT-flow, we assume that the background layer will be the most prominent and will provide sufficient SIFT features for matching. While we found this to work well in practice, images with very strong reflectance can produce poor alignment as SIFT-flow may attempt to align to the foreground which is changing. This will cause problems in the subsequent layer separation. Figure 7 shows such a case. While these failures can often be handled by cropping the image or simple user input (see supplemental material), it is a notable issue.

Another challenging issue is when the background scene

[1] http://people.csail.mit.edu/celiu/SIFTflow/SIFTflow.zip
[2] http://www.comp.nus.edu.sg/ liyu1988/
[3] http://www.wisdom.weizmann.ac.il/ levina/papers/reflections.zip

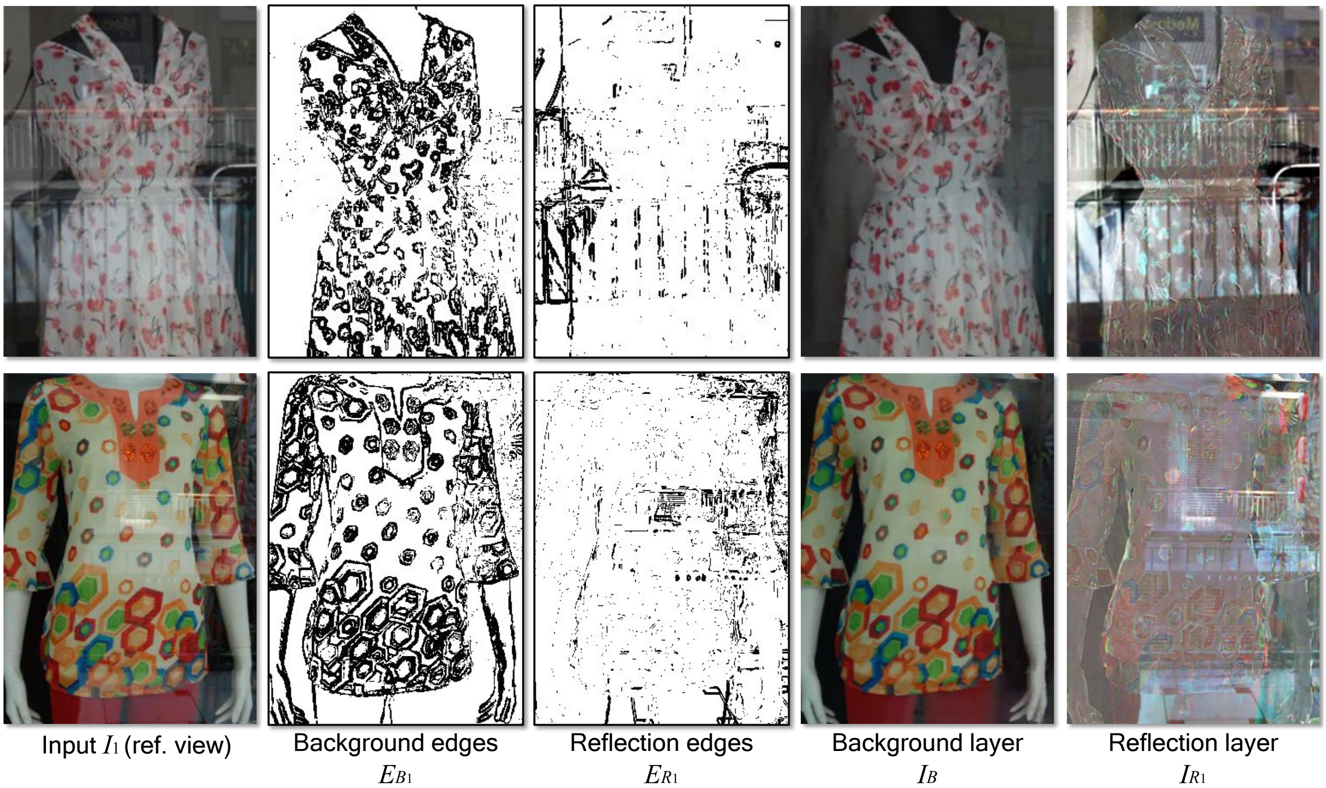
Input I_1 (ref. view)
Background edges E_{B1}
Reflection edges E_{R1}
Background layer I_B
Reflection layer I_{R1}

Fig. 6. Example of edge separation results and recovered background and foreground layer using our method

has large homogeneous regions. In such cases there are no edges to be labelled as background. This makes subsequent separation challenging, especially when the reflection interference in these regions is weak but still visually noticeable. While this problem is not unique to our approach, it is an issue to consider. We also found that by combining all the background results of the input images we can overcome local regions with high saturation. While a simple idea, this combination strategy can be incorporated into other techniques to improve their results. Lastly, we believe reflection removal is an application that would be welcomed on many mobile devices, however, the current processing time is still too long for real world use. Exploring ways to speed up the processing pipeline is an area of interest for future work.

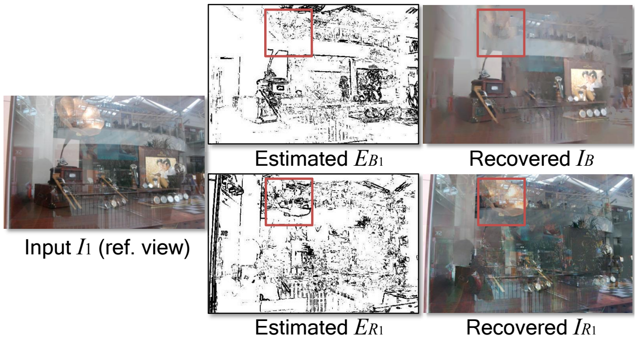

Fig. 7. A failure case of our approach due to dominant reflection against the background in some regions (*i.e*. the upper part of the phonograph). This will cause unsatisfactory alignment of the background in the warping procedure which further lead to our edge separation and final reconstruction failure as can be seen in the figure.

Acknowledgement

This work was supported by Singapore A*STAR PSF grant 11212100.

References

[1] A. K. Agrawal, R. Raskar, S. K. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ToG*, 24(3):828–835, 2005.

[2] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *IJCV*, 61(3):211–231, 2005.

[3] H. Farid and E. H. Adelson. Separating reflections from images by use of independent component analysis. *JOSA A*, 16(9):2136–2145, 1999.

[4] K. Gai, Z. Shi, and C. Zhang. Blindly separating mixtures of multiple layers with spatial shifts. In *CVPR*, 2008.

[5] K. Gai, Z. Shi, and C. Zhang. Blind separation of superimposed moving images using image statistics. *TPAMI*, 34(1):19–32, 2012.

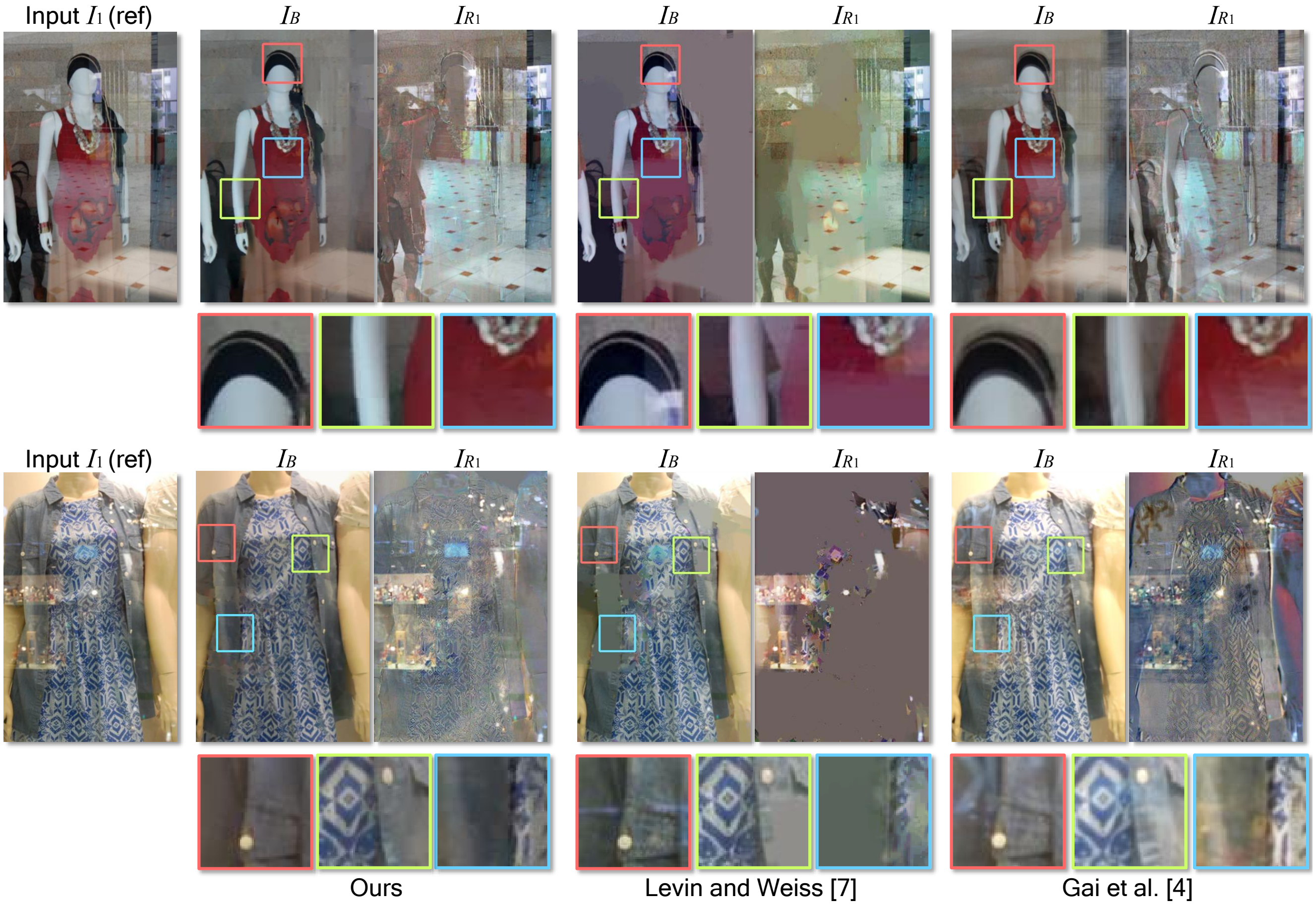

Fig. 8. Two example of reflection removal results of our method and those in [8] and [5] (user markup for [8] provided in the supplemental material). Our method provides more visual pleasing results. The results of [8] still exhibited remaining edges from reflection and tended to over smooth some local regions. The results of [5] suffered misalignment due to their global transformation alignment which results in ghosting effect of different layers in the final recovered background image. For the reflection, our results can give very complete and clear recovery of the reflection layer.

[6] N. Kong, Y.-W. Tai, and S. Y. Shin. A physically-based approach to reflection separation. In *CVPR*, 2012.

[7] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. In *ECCV*, 2004.

[8] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. *TPAMI*, 29(9):1647–1654, 2007.

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

[10] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *TPAMI*, 33(5):978–994, 2011.

[11] P. Meer. Robust techniques for computer vision. *Emerging Topics in Computer Vision*, 2004.

[12] N. Ohnishi, K. Kumaki, T. Yamamura, and T. Tanaka. Separating real and virtual objects from their overlapping images. In *ECCV*, 1996.

[13] B. Sarel and M. Irani. Separating transparent layers through layer information exchange. In *ECCV*, 2004.

[14] B. Sarel and M. Irani. Separating transparent layers of repetitive dynamic behaviors. In *ICCV*, 2005.

[15] Y. Y. Schechner, N. Kiryati, and R. Basri. Separation of transparent layers using focus. *IJCV*, 39(1):25–39, 2000.

[16] Y. Y. Shechner, J. Shamir, and N. Kiryati. Polarization-based decorrelation of transparent layers: The inclination angle of an invisible surface. In *ICCV*, 1999.

[17] Y. Y. Shechner, J. Shamir, and N. Kiryati. Polarization and statistical analysis of scenes containing a semireflector. *JOSA A*, 17(2):276–284, 2000.

[18] D. Sun, S.Roth, and M. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010.

[19] R. Szeliski, S. Avidan, and P. Anandan. Layer Extraction from Multiple Images Containing Reflections and Transparency. In *CVPR*, 2000.

[20] Y. Tsin, S. B. Kang, and R. Szeliski. Stereo matching with linear superposition of layers. *TPAMI*, 28(2):290–301, 2006.

[21] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV*, 2001.

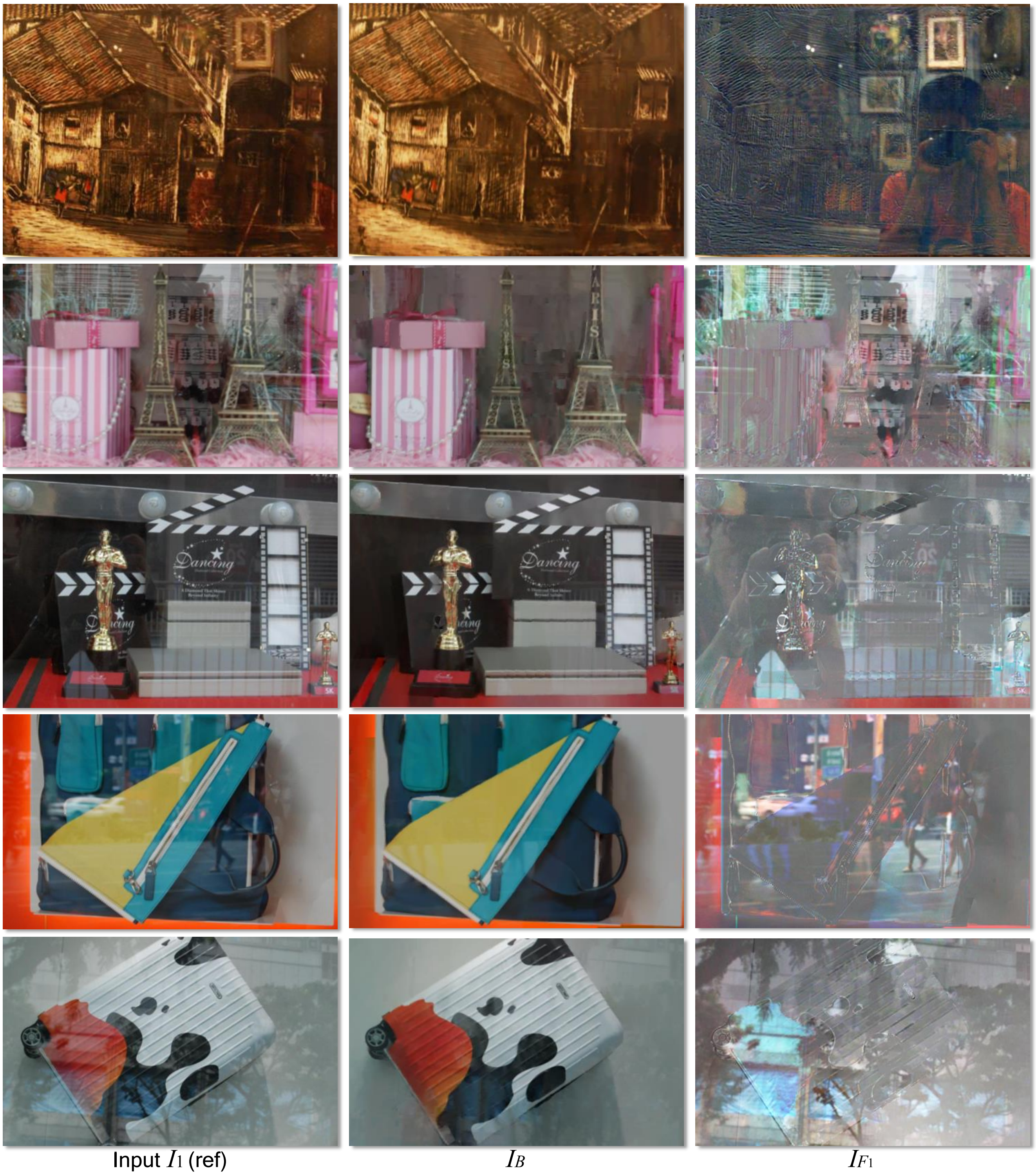

Fig. 9. More results of reflection removal using our method in varying scenes (*e.g*. art museum, street shop, *etc*.).