Approach to Retrieve Reflected Edges from an Image

Suresha M and Madhusudhan S
Department of Computer Science, Kuvempu University, India
srit_suresh@yahoo.com, madhu.hsd@gmail.com

Abstract. **C**apturing a photograph through semi reflecting surfaces such as glass window, the captured image contains combination of transmitted objects and reflected objects. Objects behind the glass are called as transmitted objects and reflection of objects at other side of glass falls on the glass surface are called as reflected objects. Separation of reflected objects and transmitted objects from an image is wide scoped area in computer vision research. By considering reflected object edges are less significant comparing to transmitted object edges. In this paper, proposed a method for separating reflected objects from an image using projection tensors and image smoothing algorithms.

Keywords: Projection Tensor, Reflected objects, Transmitted objects

1. Introduction

A photograph of a scene has reflecting or semi reflecting surfaces produces reflections of surrounding objects which are not directly visible in a photo. In general, reflected objects are not clear visible. Humans can easily decompose information in reflection layer but it is a challenging task in computer vision. In this paper addressed the problem of separation of reflection layer from an image. On the basis of literature review, the image *I* is a linear combination of background *B* and reflection *R*. Mathematically image can be modelled as,

$$I = B + R \tag{1}$$

2. Related Work

Many attempts have been made to decompose reflection from an image. Reflection removal methods are broadly categorised into two groups: Reflection removal using multiple image inputs and reflection removal using single image input. In the first category, reflection part is decomposed from an image using set of input image sequence. Reflection removal using multiple inputs is further categorised into four subcategories: Usage of image sequences or video [2,4,5,8,9,11,12], usage of flash in camera [13], usage of polarisers [6,14,15,16,17], usage of focus in camera [21]. Even though these methods work well but have practical constraints to use everyone. Li and Brown [2] proposed a method to separate reflection and background by estimating the difference of constant background object and varying reflected objects from different angle of view, this method takes substantial amount of computation time. Han et al. [4] used low-rank matrix completion in gradient domain for reflection decomposition. Szeliski et al. [8] used constrained least squares to separate reflections. Guo et al. [9] proposed a method based on correlation between transmission layers with multiple images. Simon et al.[11] exploits spatio-temporal coherence of reflection to separate reflections. In the second category, Levin et al. [20] proposed a method to separate reflections based on local feature of image. Levin and Weiss [1] provide user interactions to decompose the reflection. Fan et al. [10] separate reflection using deep learning. Shih et al. [24] used ghosting effect of reflection to separate reflections from background.

3. Proposed Method

On inspection images containing background object edges as well as reflected object edges from glass surface, in many cases reflected edges are less significant and background edges are more significant. Based on this criterion image *I* in Eq. (1) can be modelled as

$$I = B' + R' \tag{2}$$

where, *B'* is more significant background edges and *R'* is less significant reflected edges.

Here proposed a method to segment glass reflected edges from an input image and flow diagram is shown in Fig.1.

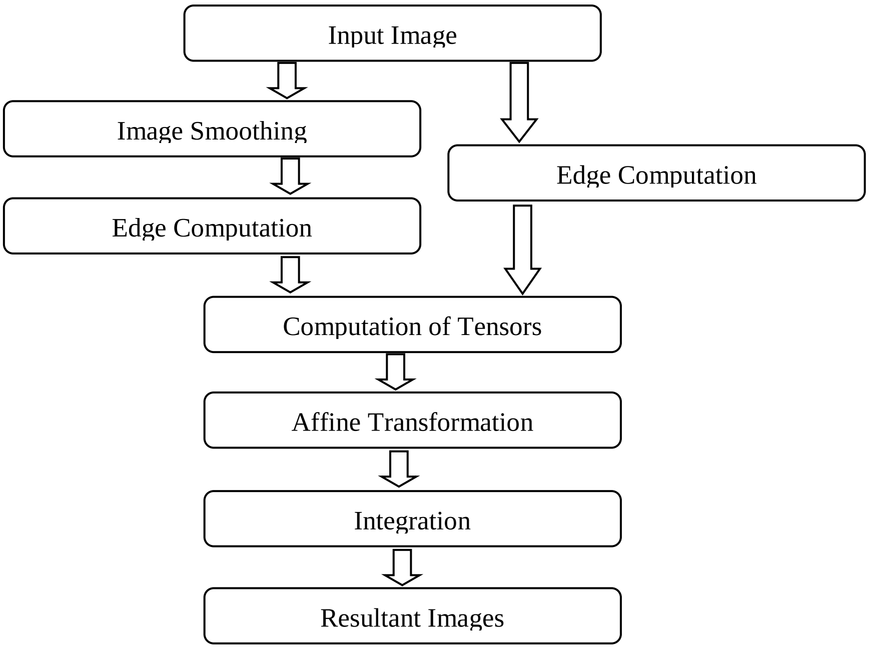


Fig. 1: Architecture of the proposed method

After reading an input image, it is converted from RGB to YUV color space using Eq. (3).

$$\begin{aligned}&Y = 0.299R + 0.587G + 0.114B\\&U = -0.147R - 0.289G - 0.436B\\&V = 0.615R – 0.515G – 0.100B\end{aligned}\qquad(3)$$

To smooth input image or suppress less significant edges applied Gaussian filters Eq. (4) and convert the resultant image from RGB to YUV color space. Apply padding for integration operation. Compute tensor matrix (Γ) using Eq. (5) for gradients ∇I and $\nabla I'$ of input image *I* and the processed image *I'*.

$$I'=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}\qquad(4)$$

Where, σ is standard deviation.

$$\Gamma=(\nabla I\,\nabla I^T)*k_\sigma\qquad(5)$$

Where, *k* is Gaussian kernel with variance *σ*.
Calculate the diffusion tensors *D* by using Eq. 6 proposed by Weickert [23]

$$D=\begin{bmatrix}D_{11} & D_{12}\\ D_{12} & D_{22}\end{bmatrix}=\begin{bmatrix}u_1 & u_2\end{bmatrix}\begin{bmatrix}\mu_1 & 0\\ 0 & \mu_2\end{bmatrix}\begin{bmatrix}u_1^T\\ u_2^T\end{bmatrix}\qquad(6)$$

Where, u_1, u_2 are eigen-vectors and μ_1 and μ_2 are eigen-values of tensor matrix Γ.
According to Aubert et al. [25], the tensor matrix Γ can be represented as:

$$\Gamma=\begin{bmatrix}X_1 & X_2\end{bmatrix}\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}\begin{bmatrix}X_1^T\\ X_2^T\end{bmatrix}\qquad(7)$$

Where, X_1 and X_2 are the eigen-vectors, λ_1 and λ_2 are corresponding eigen-values.
Compute projection tensor *D'* using Eq. 6 of tensor matrix Γ.

$$D'=[X_1\ X_2]\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}X_1^T\\ X_2^T\end{bmatrix}\qquad(8)$$

Apply affine transformation to image gradient ∇I using *D'* of Eq.8 to remove local edges and integration operation to reconstruct the resultant images. Remove padding and convert resultant images from YUV to RGB color space. The resultant is decomposition of most significant edged image (background) and less significant edged image (reflection). Algorithm of the proposed method is shown in Fig. 2.

```
Step 1: Reading Input Image viz., $I$.
Step 2: Convert an image from RGB to YUV color space.
Step 3: Apply Gaussian filter to input image and obtain smoothed image viz., $I_G$.
Step 4: Convert smoothed image $I_G$ from RGB  to YUV color space.
Step 5: Apply padding to $I$ and $I_G$.
Step 6: Calculate gradients of $I_G$.
Step 7: Calculate tensors T1= {t1,t2,..} for $I_G$.
Step 8: Find eigen values and eigen vectors for $I_G$.
Step 9: Calculate tensors T2= {t1,t2,...} for $I$.
Step 10: Find eigen values and eigen vectors for $I$.
Step 11: Calculate projection tensors $D$.
Step 12: Apply affine transformation using projection tensors $D$ for each channel.
Step 13: Integration operation.
Step 14: Remove padding on resultant images.
Step 15: Convert resultant images from YUV to RGB color space.
Step 16: Post processing of resultant images.
```

Fig. 2: Algorithm of the proposed method

4. Experimental Results

In this section, the experiments were conducted on system configured with Intel i7® PC (3.4GHz CPU, 8 GB RAM). Compared obtained results with Li and Brown [3], Levin and Weiss [1] methods which uses single image as input and Li and Brown [2] method which uses multiple images as input. The results shown in Fig. 3 and Fig. 4 are regenerated by using source codes from the authors' websites. In Levin and Weiss [1] approach provided user interactions to select edges belongs to background or reflection in an input image and it takes substantial amount of computational time to process the results and for users it is difficult to judge the edges belongs to background or foreground in complex input images.

For quantitative evaluations, evaluated the performance of results in terms of structural similarity (SSIM) Wang et al. [22]. SSIM is a measure to assess similarity between two images, which measures image similarity through a combination of luminance, contrast and structure. SSIM is defined as follows:

$$\mathrm{SSIM(x,\ y)} = \frac{(2\mu_{\mathrm{x}}\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x{}^2 + \mu_y{}^2 + c_1)(\sigma_x{}^2 + \sigma_y{}^2 + c_2)} \tag{9}$$

Where μ_x and μ_y are means, σ_x and σ_y are standard deviations, $c_1 = (0.01\ MAX)^2$ and $c_2 = (0.03\ MAX)^2$.

PSNR is commonly used to measure the quality of reconstructed images. It is defined by the mean squared error (MSE) as follows:

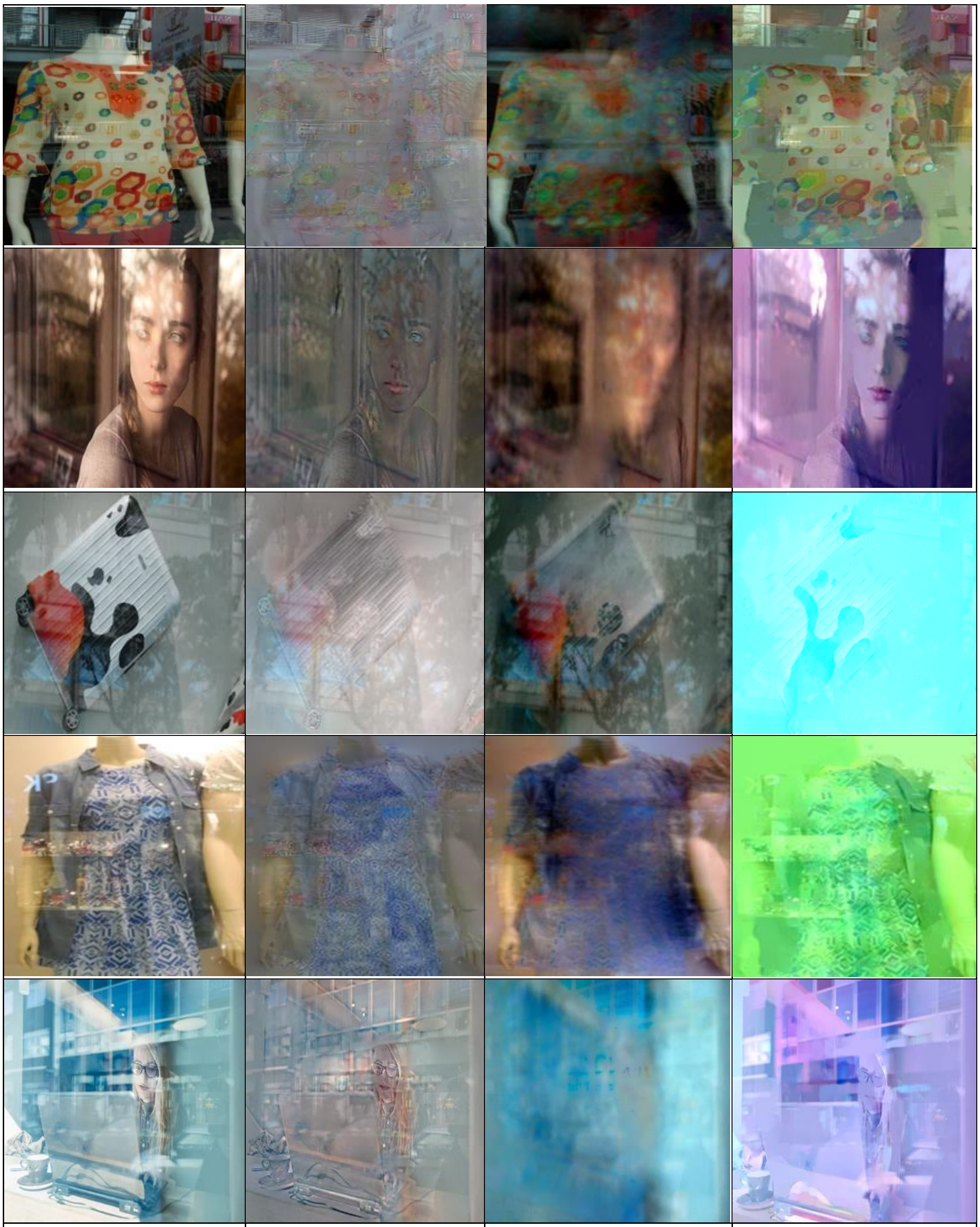
$$PSNR = 20 \log\left(\frac{MAX}{\sqrt{MSE}}\right) \tag{10}$$

Where, *MAX* is the dynamic range of the image.

Fig. 4 shows two example results of Li and Brown [2] which takes multiple images as inputs and the proposed method. We conducted experiment on all twelve case images of Li and Brown [2] provided in the authors website, with the proposed method and Li and Brown [2,3] and Table 1 shows comparison among them. SSIM, PSNR and Computational time are parameters. Due to non availability of ground truth reflection images we considered the reflection result of Li and Brown

[2] method as reference for computing PSNR and SSIM values. By analysing table 1, in terms of PSNR, SSIM and Computation time is evident for performance of the proposed method.

Experimentation conducted with ground truth reflected images of SIR dataset Wan R et al. [21] and computed SSIM values are shows that proposed method has limitations to process synthetic images given in the dataset and this is treated as future scope of our research.



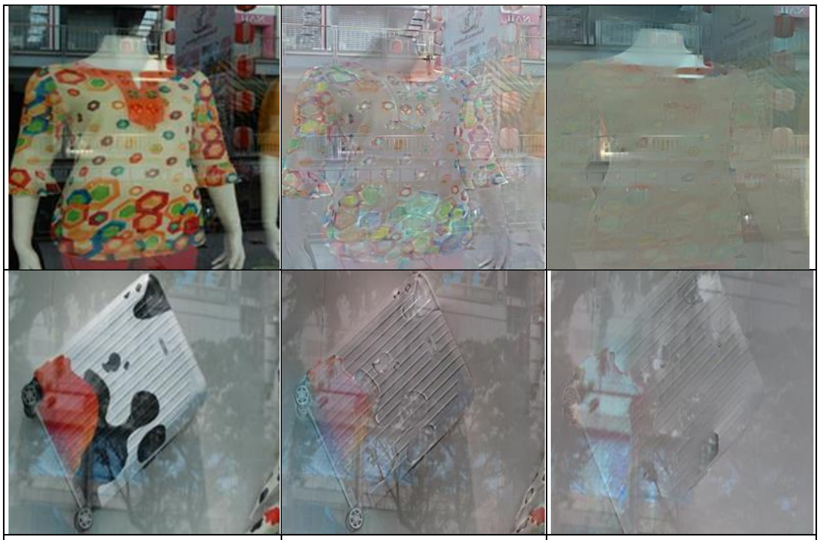
(a) Input Image

(b) Proposed Method

(c) Li & Brown [3]

(d) Levin & Weiss [1]

Fig. 3: Visual comparison of the proposed method with [3] and [1]. (a) Input image, (b) Proposed method, (c) Li and Brown [3] (d) Levin & Weiss [1]



(a) Input Image

(b) Proposed Method

(c) Li & Brown [2]

Fig. 4: Visual comparison between the proposed method with [2].(a) Input images, (b) the proposed method (c) Li & Brown [2].

Sl No.	SSIM of proposed method	SSIM of [3]	PSNR of proposed method	PSNR [3]	CPU time of proposed method	CPU time of [2]
Case 1	0.3821	0.5177	27.05db	24.74db	20.48 sec	119.62 sec
Case 2	0.4395	0.2498	26.27db	24.09db	15.10 sec	91.19 sec
Case 3	0.3933	0.4048	25.79db	24.09db	16.70 sec	52.87 sec
Case 4	0.5161	0.4228	34.48db	24.13db	17.11 sec	108.98 sec
Case 5	0.4491	0.3333	40.00db	24.09db	18.04 sec	119.08 sec
Case 6	0.5074	0.3207	30.09db	24.09db	8.03 sec	28.97 sec
Case 7	0.4740	0.3458	49.81db	24.09db	16.44 sec	104.70 sec
Case 8	0.3869	0.5010	43.04db	24.09db	13.88 sec	61.97 sec
Case 9	0.6679	0.5278	24.82db	24.37db	15.80 sec	97.97 sec
Case 10	0.5709	0.5111	25.94db	24.09db	16.45 sec	79.94 sec
Case 11	0.4174	0.3371	29.89db	24.09db	17.44 sec	112.61 sec
Case 12	0.6347	0.5902	27.93db	24.10db	15.56 sec	75.25 sec

Table 1: Comparison of the proposed method with Li & Brown [2 and 3] in terms of SSIM, PSNR and Computational time.

5. Conclusion

In this work a new method to separate reflection edges from an image has been developed. Proposed new image model that image is a combination of most significant background edges and less significant reflected edges. Exploited tensor matrices for input image and resultant smoothed image after Gaussian filter, computed projection tensor then applied affine transformation to extract less significant reflected edged image. Compared the proposed method with other state of the art reflection separation methods and it shows fair performance of our method. Future scope of this work is to segment reflection edges from synthetic images.

Acknowledgement

Kuvempu university has been extended the financial support to carry out this work.

References

1. Levin A, Weiss Y (2007) User assisted separation of reflections from a single image using a sparsity prior. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 9.

2. Li Y, Brown MS (2013) Exploiting reflection change for automatic reflection removal. Proceedings of the IEEE International Conference on Computer Vision, pp. 2432-2439.
3. Li Y, Brown MS (2014) Single image layer separation using relative smoothness. In: Proceedings of IEEE CVPR, 2014, pp. 2752-2759. doi:10.1109/CVPR.2014.346
4. Han BJ, Sim JY (2017) Reflection removal using low-rank matrix completion. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
5. Han BJ, Sim JY (2018) Glass reflection removal using co-saliency-based image alignment and low-rank matrix completion in gradient domain. IEEE Transactions on Image Processing, vol.27, no.10, pp. 4873-4888.
6. Farid H, Adelson EH (1999) Separating reflections and lighting using independent component analysis. Computer Vision and Pattern Recognition, vol. 1, pp. 262-267.
7. Sarel B, Irani M (2004) Separating transparent layers through layer information exchange.Computer Vision-ECCV 2004, pp. 328-341.
8. Szeliski R, Avidhan S,Anandan P (2000) Layer extraction from multiple images containing reflections and transparency. Computer Vision and Pattern Recognition, vol. 1, pp. 246-253.
9. Guo X,Cao X,Ma Y (2014) Robust separation of reflection from multiple images. IEEE Conference on Computer Vision and Pattern Recognition, pp. 2187-2194.
10. Fan Q, Yang J,Hua G, Chen B, Wipf D (2017) A generic deep architecture for single image reflection removal and image smoothing. arXiv:1708.03474, 2017.
11. Simon C, Park IK (2015) Reflection removal for in-vehicle black box videos. IEEE Conference on Computer Vision and Pattern Recognition, pp. 4231-4239.
12. Yang J, Li H, Dai Y, Tan RT (2016) Robust opticle flow estimation of double-layer images under transparency or reflection. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1410-1419.
13. Agrawal A, Raskar R, Nayar SK, Li Y (2005) Removing photography artifacts using gradient projection and flash exposure sampling. In: ACM Transactions on Graphics,vol. 24, no.3, 828-835. doi: 10.1145/1186822.1073269
14. Nayar SK, Fang XS,Boult T (1997) Separation of reflection components using color and polarization. International Journal of Computer Vision, vol. 21, no. 3, pp. 163-186.
15. Schechner YY, Shamir J, Kiryati N (2000) Polarization and statistical analysis of scenes containing a semireflector.JOSA A, vol. 17, no.2, pp. 276-284.
16. Diamant Y, Schechner YY (2008) Overcoming visual reverberations.In:Computer Vision and Pattern Recognition, pp. 1-8.
17. Kong N, Tai YW, Shin SY (2013) A physically-based approach to reflection separation: from physical modelling to constrained optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 209-221.
18. Shih YC, Krishnan D, Durand F, Freeman WT (2015) Reflection removal using ghosting cues. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3193-3201.
19. Schechner YY, Kiryati N, Basri B (2000) Separation of transparent layers using focus.International Journal of Computer Vision, vol. 39, no. 1, pp. 25-39.
20. Levin A, Zomet A,Weiss Y (2004) Separating reflections from a single image using local features. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1.
21. Wan R, Shi B, Duan LY, Tan AH, Kot AC (2017) Benchmarking Single-image Reflection Removal Algorithms.In: International Conference on Computer Vision (ICCV).
22. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2011) Image quality assessment: from error visibility to structural similarity. In: IEEE Transactions on Image Processing, vol. 20, no. 8, pp. 2378-2386.
23. Weickert J (1997) A review of nonlinear diffusion filtering. In: Scale Space Theory in Computer Vision, Lecture Notes in Comp. Science (Springer, Berlin)
24. Shih YC, Krishnan D, Durand F, Freeman WT (2015) Reflection removal using ghosting cues. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3193-3201.
25. Aubert G, Kornprobst P (2002) Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations, volume 147 of Applied Mathematical Sciences, Springer- Verlag.