Single Image Layer Separation using Relative Smoothness

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

Abstract

This paper addresses extracting two layers from an image where one layer is smoother than the other. This problem arises most notably in intrinsic image decomposition and reflection interference removal. Layer decomposition from a single-image is inherently ill-posed and solutions require additional constraints to be enforced. We introduce a novel strategy that regularizes the gradients of the two layers such that one has a long tail distribution and the other a short tail distribution. While imposing the long tail distribution is a common practice, our introduction of the short tail distribution on the second layer is unique. We formulate our problem in a probabilistic framework and describe an optimization scheme to solve this regularization with only a few iterations. We apply our approach to the intrinsic image and reflection removal problems and demonstrate high quality layer separation on par with other techniques but being significantly faster than prevailing methods.

1. Introduction

This paper addresses the problem of layer separation from a single-image with application to 1) intrinsic image decomposition and 2) single image reflection interference removal using focus. Both of these problems take the form:

$$I = L_1 + L_2, \tag{1}$$

where I is the observed image and L_1 and L_2 are the combined layers. For example, the intrinsic image model [3] assumes that an image scene is the product of a scene's reflectance and illumination at each pixel, expressed as $I = RL$, where R is the reflective property or albedo at each pixel and L is the illumination falling on this pixel. Intrinsic image decomposition's aim is to estimate R and L given an input I. This can be reformulated into the form in Eqn. 1 by taking the log, *i.e.* $\log(I) = \log(R) + \log(L)$. Reflection interference arises when a photo of a scene is taken behind a glass window. This can be expressed as a linear combination of a reflection layer L_R and and the desired background scene L_B, as $I = L_B + L_R$. We use a slightly modified version based on Schechner *et al.*'s [14] proposition of using focus such that the desired layer is more in focus while the reflection is blurred. This can be expressed as: $I = L_B + L_R * h$, where the reflection layer is convolved with the depth of field kernel h modelled as a Gaussian blur.

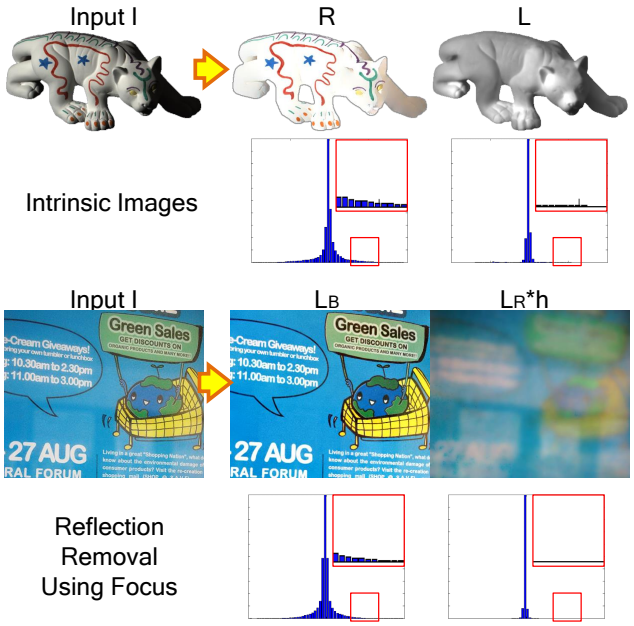

Fig. 1. This figure shows the two problems our method is applied to: intrinsic image decomposition and single image reflection removal using focus. The corresponding gradient histograms of each layer is shown below. In both of these problems one layer has fewer large gradients than the other layer. Note that the layers' intensity has been boosted to improve visualization.

Layer separation is inherently challenging as it attempts to obtain two unknowns (L_1 and L_2) from a single input I. To make the problem tractable prior knowledge on the solutions must be imposed. For example, prevailing methods for intrinsic image decomposition (see [6, 8, 18, 20]) typically adapt the idea that natural images have piecewise constant reflectance while the illumination is smoothly varying. This means that the illumination layer L is smoother than the reflectance R.

A successful approach in single-image reflection separa-

tion [11] employs a strategy that imposes a gradient sparsity prior on the recovered layers. The gradient sparsity prior (also called the natural image prior) has been shown to be successful in other ill-posed problems where multiple solutions are possible (*e.g*. image deblurring [5]). The basic idea is to require the image gradient histogram to have a long-tail distribution.

Our work is inspired by the success of imposing a gradient sparsity constraint on the images, however, in our problem, the two layers' gradients do not have the same distributions. Instead, one of the layers is assumed to be smooth, *i.e*. illumination L and the defocused reflection $L_R * h$, and therefore should have very little large gradient. Figure 1 shows an example. This means we need an additional constraint on the smooth layer.

Contribution We propose a novel method to solve the layer separation problem by building two likelihoods for each layer from the gradient histograms in which one layer is smoother than the other. To get the desired layer separation, the necessary objective function is formulated. An efficient scheme is described to optimize the objective function which is non-convex and has an inequality constraint. Our method provides high-quality results on megapixel images in a matter of seconds. This is much faster than existing intrinsic image and reflection separation methods.

The remainder of this paper is organized as follows. Section 2 provides more details of related methods in our targeted applications; Section 3 overviews our approach; Section 4 provides experimental comparisons with prior approaches. A discussion and summary concludes the paper in Section 5.

2. Related Work

2.1. Intrinsic Image Decomposition

One of the earliest work addressing intrinsic image decomposition was the Retinex algorithm [10] that employed simple heuristics that assumed strong edge gradients belonged to reflectance changes in the image. Other intrinsic image decomposition methods using multiple images [24] or using user markup [4] have been proposed and shown to produce good results. For automatic single image intrinsic image estimation, many later works [8, 17, 20] followed the idea of the Retinex algorithm and focused on separating reflectance and illumination edges. These methods are referred to as edge-based methods, and according to a recent survey [8], the color version Retinex algorithm [8] was still a top performing method. The authors of [8] also created a ground-truth dataset (the MIT dataset) for intrinsic images containing 16 real objects.

More recent approaches in intrinsic images took the advantage of the recent progress in probabilistic models and optimizations methods. Unlike edge-based methods which rely on local information, these new methods use the idea that there is a sparse set of reflectance value present in the scene, which is usually referred to as global sparsity prior [2, 6, 15, 18]. Note that this sparsity is not applied on the gradient histogram, but directly on the allowable reflectance values. These approaches achieve excellent results on the MIT dataset.

Since our method does not make the use of a prior on the reflectance values directly, we categorize using our method as edge-based. We show, however, that our method can achieve much better performance than other edge-based methods. Our results are close (sometimes even better in a few cases) to those obtained by the state-of-art methods that use sophisticated models and inference while being significantly faster.

2.2. Reflection Removal

Most reflection removal algorithms rely on multiple input images. These methods require taking a set of images with different mixing of layers, *e.g*. rotating a polarized lens [9, 16], using flash/no-flash image pairs [1], capturing a lone video sequence [13], or changing view points [12, 19, 21]. From these multiple inputs various schemes are used to recover the desired background layer. Single-image reflection removal is much harder. One of the successful approaches is by Levin and Weiss [11] and relies on user markup to denote gradients that belong to background and reflection. They also proposed an optimization framework that imposes the gradient sparsity. Their method produces compelling results when the user provides proper markup.

The method by Schechner *et al*. [14] discussed in Section 1 also requires two input images, specifically one where the reflection was in focus and one where the background was in focus. We show that using our method we can obtain a high-quality separation of reflection with only a single image focused on the background. Moreover, we do not need to explicitly estimate the blur point-spread-function h as done in [14].

3. Our Approach

3.1. Model

Inspired by the gradient sparsity prior used in [11], we introduce our priors on the two layers' gradients. Suppose L_2 is smoother than L_1, then large gradients are more likely to belong to L_1. We encode this into two probabilities as:

$$\begin{aligned} P_1(x) &= \frac{1}{z}\max\{e^{-\frac{x^2}{\sigma_1^2}}, \epsilon\}, \\ P_2(x) &= \frac{1}{2\pi\sigma_2^2}e^{-\frac{x^2}{\sigma_2^2}}, \end{aligned} \tag{2}$$

where x is the gradient value, z is a normalization factor, σ_1 and σ_2 are both small values making two narrow Gaussians

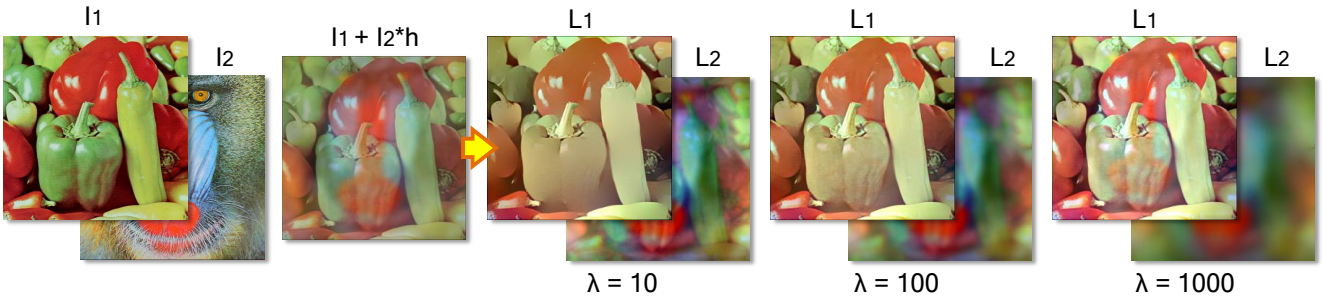

Fig. 2. This figure shows the effect of different λ setting on the final separation results on a synthesized case. Larger λ leads to smoother L_2 and therefore controls the detail transfer in the layer separation. When λ was small ($\lambda = 10$), L_1 lost many details which incorrectly appeared in L_2. When λ was large ($\lambda = 1000$), L_2 became over-smooth and the part of the detail appeared back in L_1. Setting $\lambda = 100$ is an appropriate choice as it gives the most pleasing result.

which drop very fast. However by using the max operator with ϵ in P_1 we explicitly add a tail to prevent the probability from getting close to zero.

In order to solve the layer separation problem, we adapt a probabilistic model to seek the most likely explanation of the input image using the probabilities of the two layers defined in Eqn. 2. In essence, we are maximizing the joint probability $P(L_1, L_2)$. This can be achieved by minimizing the negative log probabilities. Taking the negative log to the probabilities in Eqn. 2, we obtained:

$$\begin{aligned} -\log P_1(x) &\propto \min\{\frac{x^2}{\sigma_1^2(-\log \epsilon)}, 1\} + C_1, \\ -\log P_2(x) &\propto \tfrac{x^2}{\sigma_2^2} + C_2. \end{aligned} \tag{3}$$

Here, C_1 and C_2 are constants that we can drop later. While $-\log P_2(x)$ is in L_2 form, $-\log P_1(x)$ is in truncated L_2 form which we further simplify as $\rho(x) = \min\{x^2/k, 1\}$. The term k is still a small number fixed as a constant 10^{-4} in our method. The function ρ is similar to the sparse penalty used in [25]. With the assumption that the two layers are independent (*i.e.* $P(L_1, L_2) = P(L_1) \cdot P(L_2)$) and the derivative filter output are independent (*i.e.* $P(L_t) = \prod_i P_t(f_j * L)_i, t \in \{1, 2\}$), minimizing $-\log P(L_1, L_2)$ becomes:

$$\min_{L_1, L_2} \sum_{i,j} \left(\rho(L_1 * f_j)_i + \lambda (L_2 * f_j)_i^2 \right), \tag{4}$$

where i is the pixel index, f_j denotes different derivative filters. We used two directional first order derivative filters and a second order Laplacian filter, namely $f_1 = [-1\ 1], f_2 = [-1\ 1]^T, f_3 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$, and for simplicity we write $F_i^j L = (L * f_j)_i$ in the rest of the paper. From the experiments we found using first order derivative filters for L_1 and a second order Laplacian filter for L_2 produced good results. The first order derivative filter helps to recover the significant edges in L_1, while the Laplacian filter encodes smooth variations in L_2. We integrate the weight between the two terms and the multiplier $\frac{1}{\sigma_2^2}$ together as one parameter λ which controls the smoothness of the output L_2. The effect of different λ setting is shown in Figure 2.

Our probabilities are defined on the gradients and to recover meaningful layers we have to bound the solution range *i.e.* $(L_1)_i \in [lb_i, ub_i]$. The ranges are set according to the application which will be discussed in Section 4. Moreover, we can substitute L_2 with $I - L_1$ into the objective, making the final objective function on parameter L_1 as:

$$\begin{aligned} \min_{L_1} \sum_i \left(\sum_{j=1,2} \rho(F_i^j L_1) + \lambda (F_i^3 L_1 - F_i^3 I)^2 \right) \\ \text{s.t.} \quad lb_i \le (L_1)_i \le ub_i. \end{aligned} \tag{5}$$

3.2. Optimization

Our objective function is non-convex due to the non-convex $\rho(x)$ component. There is also an inequality constraint. Such problems require care when optimizing. We employ a two stage approach. First, we use the half-quadratic separation scheme [7, 22] to solve the non-convex problem without the inequality constraint and at the end of each iteration we perform a normalization step to force the solution to fall within the constrained range.

Using the half-quadratic method, auxiliary variables g_i^j are introduced at each pixel that allow us to move the $F_i^j L_1$ term outside the $\rho(\cdot)$ function, giving a new cost function:

$$\min_{L_1, g^j} \sum_i \left(\sum_{j=1,2} (\beta (F_i^j L_1 - g_i^j)^2 + \rho(g_i^j)) + \lambda (F_i^3 L_1 - F_i^3 I)^2 \right), \tag{6}$$

where β is a weight that we will increase during the optimization (in our implementation , starting from 10 or 20 and multiplied by $\eta = 2$ each time). As β gets larger the solution gets closer to that of Eqn. 5. Minimizing Eqn. 6 for a fixed β can be performed by alternating between computing L_1 and updating of g^j. The computation of L_1 and g^j-updates are described in the following paragraphs.

Update g^j Keeping L_1 fixed, the closed-form solution at each pixel is found to minimize Eqn. 6 w.r.t. g^j as:

$$g_i^j = \begin{cases} F_i^j L_1, & (F_i^j L_1)^2 > \frac{1}{\beta} \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

This simple thresholding rule holds when $\beta < \frac{1}{k}$.

Compute L_1 With g^j fixed, the function of Eqn. 6 w.r.t. L_1 is quadratic. Assuming circular boundary conditions, we can apply a 2D FFT $\mathcal{F}$ which diagonalizes the convolution matrices F^j's, allowing us to find the optimal L_1 directly:

$$\begin{aligned} &L_1 = \mathcal{F}^{-1}(A), \\ &A = \frac{\beta \sum_j (\mathcal{F}(F^j)^\star \mathcal{F}(g^j)) + \lambda \mathcal{F}(F^3)^\star \mathcal{F}(F^3) \mathcal{F}(I)}{\beta \sum_j (\mathcal{F}(F^j)^\star \mathcal{F}(F^j)) + \lambda \mathcal{F}(F^3)^\star \mathcal{F}(F^3) + \tau} \end{aligned} , \quad (8)$$

where $\star$ is the complex conjugate, the parameter τ added to the denominator is a small number necessary to increase the stability of our algorithm ($\tau = 10^{-16}$ in our implementation). The multiplication and division are both performed element-wise. Solving Eqn. 8 requires only two FFT for g^1 and g^2 and one IFFT at each iteration since the other terms can all be precomputed.

Normalize L_1 After getting L_1, we perform a normalization step to bring the solution to a meaningful range. This step is important since the solution to Eqn. 6 is not unique and is related by a global constant. Therefore the goal of the normalization step is to make the solution fall in the range $[lb_i, ub_i]$. To find a suitable constant t we try to minimize the following objective function

$$\min_t \sum_i m_i((L_1)_i + t - lb_i)^2 + \sum_i n_i((L_1)_i + t - ub_i)^2, \quad (9)$$

where m_i, n_i are indicative functions such that m_i is equal to 1 only when $(L_1)_i + t < lb_i$, and n_i is equal to 1 only when $(L_1)_i + t > ub_i$, otherwise they all equal to 0. From this, L_1 is updated to $L_1 + t$. Simple gradient descent can be used for this step. After this, a few values may still fall outside the interval $[lb_i, ub_i]$. These values are clipped to lb_i or ub_i. We summarize the whole process in Algorithm 1. In all of our experiments, the optimization converges very quickly (within 5 iterations) and produces high-quality results. Convergence is empirically demonstrated in Section 4.

4. Experimental Results

Our experiments are done on a PC with Intel I7 CPU (3.4GHz) and 8GB RAM. The implementation is done using Matlab without any GPU acceleration. Demo code can be downloaded at the author's webpage [1].

[1] http://www.comp.nus.edu.sg/ liyu1988/

Algorithm 1 Layer Separation using Relative Smoothness

Input: input image I; smoothness weight λ; initial β_0; iterations number $i_{\max}$; increasing rate η;

Initialization: $L_1 \leftarrow I$; $\beta \leftarrow \beta_0$; $i \leftarrow 0$.

while $i < i_{\max}$ **do**
  update g_i^j using Eqn. 7;
  compute L_1 using Eqn. 8;
  normalize L_1 using Eqn. 9;
  $\beta = \eta * \beta, i++$;
end while

$L_2 = I - L_1$;

Output: The estimation of two layers L_1 and L_2;

4.1. Intrinsic Image Decomposition

We denote $\log(I)$, $\log(R)$, and $\log(L)$ as $\hat{I}$, $\hat{R}$, $\hat{L}$ respectively in the equations. In our implementation, the original images are normalized to $[1/256, 1]$. Therefore, after log, $\hat{R}$ should fall in the range $[\hat{I}, 0]$. Using our method, the objective function becomes:

$$\begin{aligned} \min_{\hat{R}} \sum_i \Big(\sum_{j=1,2} \rho(F_i^j \hat{R}) + \lambda (F_i^3 \hat{R} - F_i^3 \hat{I})^2 \Big) \\ \text{s.t.} \quad \hat{I}_i \le \hat{R}_i \le 0. \end{aligned} \quad (10)$$

If we set the smoothness weight λ to zero and just run our whole process once, meaning only threshold the gradient once to get the g^j and then recover $\hat{R}$, our method acts just like the Retinex algorithm [8]. We denote this configuration as Retinex (Ours). Our implementation of the Retinex algorithm can achieve better performance than the original one described in [8]. Therefore we report the Retinex result using our implementation.

4.1.1 Evaluation on the MIT dataset

We have tested our algorithm on the MIT intrinsic image dataset [8].

Fast convergence We show here that our optimization framework can converge to a good solution very fast (no more than 5 iterations needed). We plot the energy values at each iteration for one intrinsic decomposition example in Figure 3. At each iteration we measure the error between our layer estimation at the current state w.r.t. the ground truth data using Local Mean Square Error (LMSE) [8]. The curve is also plotted in Figure 3 to show that our method can converge to high quality results quickly.

Comparison with previous methods We have compared the performance of our method with several representative intrinsic image estimation methods and reported the running time per image as well as the LMSE on the MIT dataset in Table 1. Tappen *et al.*'s method [20] is an edge-based method that learns a classifier to distinguish reflectance

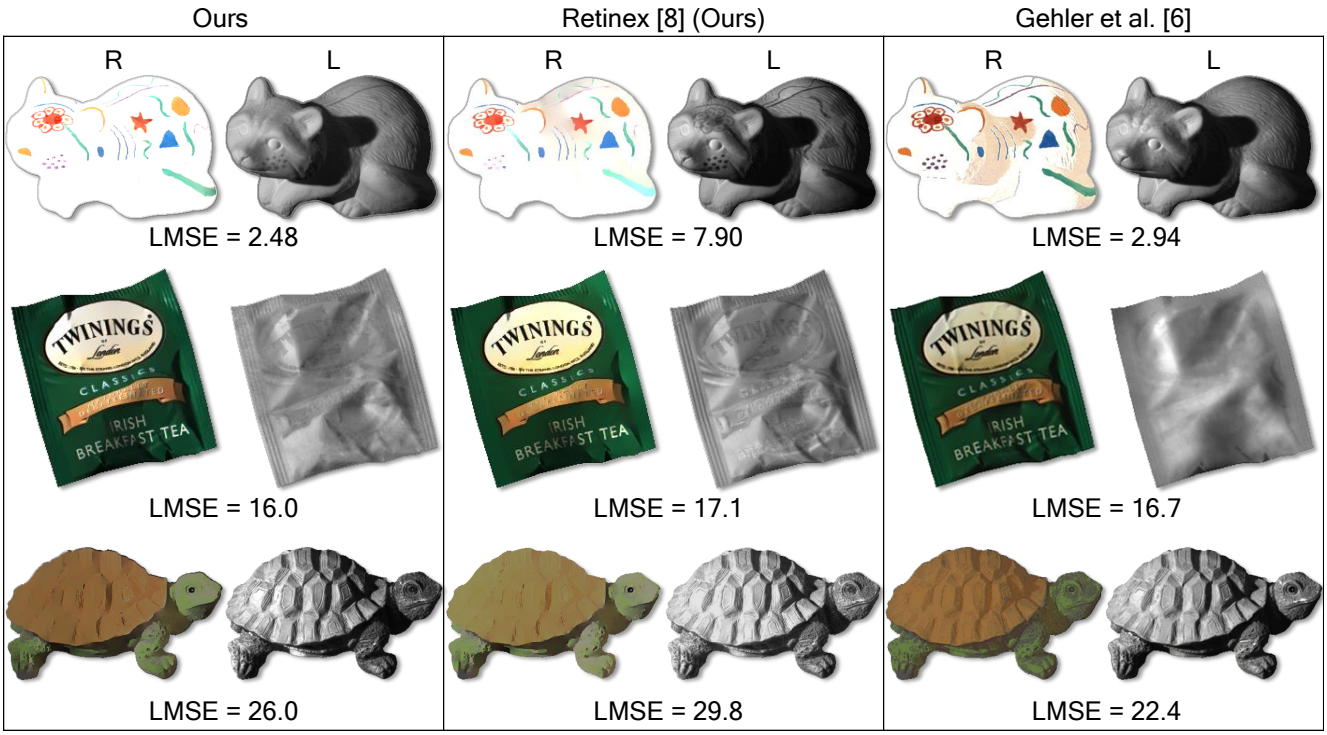

Fig. 4. This figure shows the decomposition results by Retinex [8], the method in [6] and our approach on three images from the MIT intrinsic dataset. LMSE errors shown below are in 10^{-3}.

edges and illumination edges. Methods in [2, 6, 18] use the global sparsity prior and the framework in [2] uses more constraints to solve the shape from shading problem jointly with intrinsic images. These three methods are generally considered as state-of-art in terms of the performance on the MIT dataset. Note that for methods [6] and [2], we cannot get the LMSE as small as in the original paper. We report results provided by the authors that are considered to be their best performance.

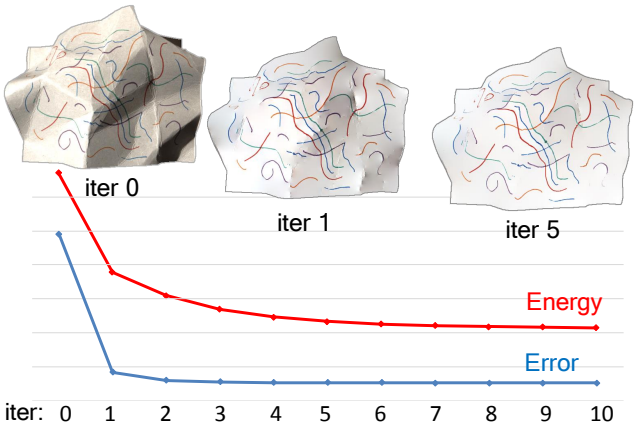

Fig. 3. This figure illustrate the convergence of our algorithm. The red line and the blue line denote the energy defined by our objective function and the error between current estimation and ground truth using LMSE measurement respectively. Note that the scales of the energy and the error are different. We put them together here for illustration. The estimated reflectance of some of the steps are also plotted above.

Table 1. Quantitative Comparison with Previous Methods

Method	**Runtime**	**LMSE**
Tappen *et al*. 2005 [20]	>200 s	0.0347
Shen & Yeo 2011 [18]	>300 s	0.0204
Gehler *et al*. 2011 [6]	>600 s	0.0131*
Barron & Malik 2012 [2]	>200 s	0.0133*
Retinex [8] (Ours)	<1 s	0.0217
Ours	1–3s	0.0149

As can be seen, our optimization with Matlab implementation is efficient compared with others due to the FFT acceleration. Even without using the global sparsity prior, our method can achieve high quality performance close to specially designed methods for intrinsic image (*e.g*. [2, 6]).

We also show three example results in Figure 4 and compare with the Retinex method [8] and the best over-all performance method [6]. Our method gives visually better results than Retinex [8] since our results shows clearer edges and no bleeding artifacts. For the *raccoon* and *teabag* cases, our results are even better than [6]. The results of [6] has more regions with incorrect separations of the two layers, *e.g*. illumination components remaining in *raccoon's*

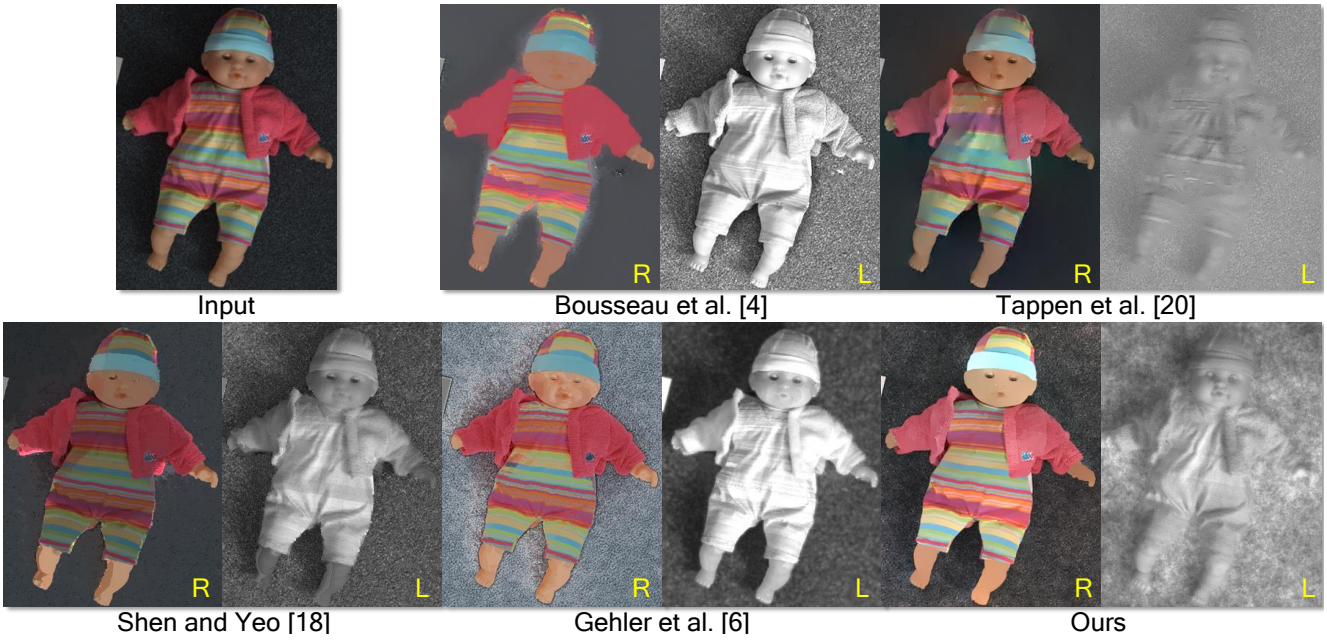

Fig. 5. Comparison of decomposition results on a photo with the user-assisted approach [4] and other representative automatic approaches of [20, 18, 6]. All illumination images are shown in gray scale.

reflectance and illumination details near the border of the *teabag* appears in the reflectance image.

4.1.2 Comparison on Real Input

We have also tested our method on the input image used in previous work in [4]. The method in [4] is a user-assisted one that can generate more piece-wise constant reflectance with user's labelling of regions sharing same reflectance or same illumination. However, their local 2D subspace model would fail on high contrast region (*e.g.* the border of the doll), resulting in artifacts in the reflectance image. Other three methods [6, 18, 20] more or less mixed the texture on the cloth into the illumination map. Our method shows arguably the the best reflectance and illumination decomposition results, considering the piece-wise flat reflectance, clear edges and texture information.

4.2. Single-Image Reflection Removal with Defocus Blur

For the reflection removal problem, the estimated background value $(L_B)_i$ should fall in the range $[0, I_i]$, giving the objective function:

$$\min_{L_B} \sum_i \left(\sum_{j=1,2} \rho(F_i^j L_B) + \lambda (F_i^3 L_B - F_i^3 I)^2 \right) \quad \text{s.t.} \quad 0 \le (L_B)_i \le I_i. \tag{11}$$

4.2.1 Results on Synthetic Data

Based on the mixing process $I = L_B + L_R * h$, we have synthesised layer mixing data. A 2D Gaussian of standard deviation five is used as the defocus blur kernel h in our synthesis. The input mixing images as well as the final

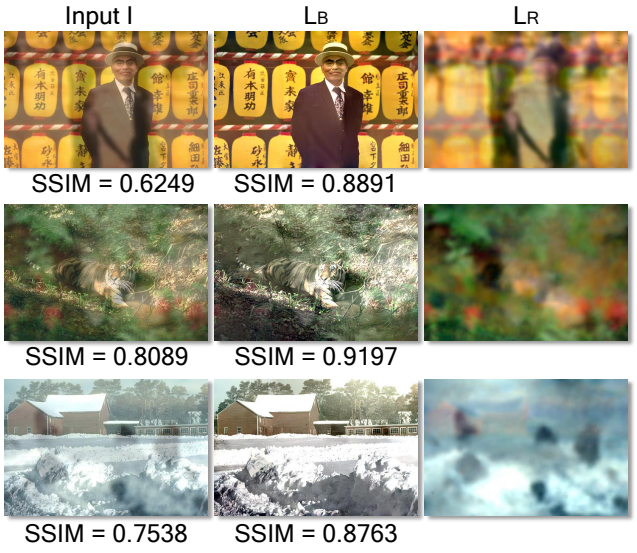

Fig. 6. Three reflection removal examples on synthesized data. The corresponding SSIM with regard to the ground truth background layer are also listed below for quantitatively showing the effectiveness of our separation. Note that we just write the recovered reflection layer as L_R.

Fig. 7. Two examples of reflection removal results of our method and prior single image approach in [11]. Our method provides visually clearer separation results. But in the top case, a small part of the background is smooth (pointed out by yellow arrow) which breaks our assumption, leading to incorrect separation at that small region (How to correct such cases is shown in Figure 8).

separation results are shown in Figure 6. To quantitatively assess our algorithm, we have computed the the Structural Similarity Index (SSIM) [23] as the quality measure of the recovered background layers.

As can be seen, after separation on the synthesized images using our method, the SSIM is increased by at least 0.1 compared with the original mixed image; visually, the background layer is much clearer after separation.

4.2.2 Results on Real World Data

We have tested our method on reflection separation on real world cases and compared ours with the Levin and Weiss's user-assisted method in [11]. For the results produced by [11], large amount of user-markup is provided. However, at some locations, the background edges and reflection edges intersect, making it hard for the user to label the gradients, especially because the reflection layer has defocus blur. Our method can generate clearer separation of the background and the reflection layer than that of [11]. It is worth noting that the method in [11] is time consuming. Manually providing sufficient labelling can be challenging. In addition, this method solves the non-convex optimization using Iterative Reweighed Least Square that takes several minutes. Our method is automatic and requires less than two seconds to produce the results. However, the top image in Figure 7 does reveal a limitation in our work. In particular, the specular highlight (pointed by the yellow arrow) on the ball pattern of the book cover is falsely categorized to reflection layer. This is due to the fact that the highlight is a smooth pattern which violates our assumption that the background layer is sharper than reflection.

5. Discussion and Conclusion

We have presented a method to automatically extract two layers from one image where one layer is smoother than the other. Our approach works by building two likelihoods for each layer from gradient histograms, that models this relative smoothness. In order to solve the layer separation problem, the necessary objective function that finds the most

likely explanation of the two layers is proposed. We also derived an efficient scheme to optimize the objective function which is non-convex and has an inequality constraint. We have tested our method on two layer separation problems of intrinsic image decomposition and reflection removal using defocus blur. Our method provides high-quality results in a manner that is significantly faster than prior work.

One challenging issue is that if our assumption that the two layer have different smoothness is violated, our methods will fail to correctly separate the layers. An example was shown in Section 4. If this happens, user intervention may be used to help. For example, we can simply have the user denote which layer a particular region should belong to as shown in Figure 8.

In the future, we would like to explore other layer separation problems that may benefit from our method.

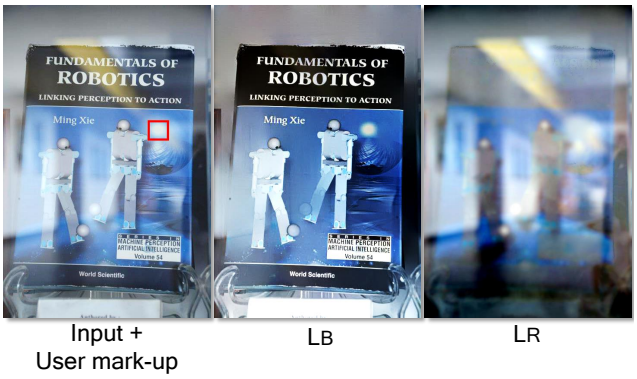

Fig. 8. This is the previous example from Figure 7 where part of the image is incorrectly separated. We show here that a simple user interaction (*e.g.* drawing a red rectangle indicating the region belongs to background) can help solve the problem.

Acknowledgement

This work was supported by the 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] J. T. Barron and J. Malik. Color constancy, intrinsic images, and shape estimation. In *ECCV*, 2012.

[3] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. In *Computer Vision Systems*, 1978.

[4] A. Bousseau, S. Paris, and F. Durand. User-assisted intrinsic images. *ToG*, 28(5):130:1–130:10, 2009.

[5] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ToG*, 25(3):787–794, 2006.

[6] P. V. Gehler, C. Rother, M. Kiefel, L. L. Zhang, and B. Schölkopf. Recovering intrinsic images with a global sparsity prior on reflectance. In *NIPS*, 2011.

[7] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *TIP*, 4(7):932–946, 1995.

[8] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*, 2009.

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

[10] E. H. Land and J. J. McCann. Lightness and retinex theory. *JOSA*, 61(1):1–11, 1971.

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

[12] Y. Li and M. S. Brown. Exploiting reflection change for automatic reflection removal. In *ICCV*, 2013.

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

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

[15] M. Serra, O. Penacchio, R. Benavente, and M. Vanrell. Names and shades of color for intrinsic image estimation. In *CVPR*, 2012.

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

[17] L. Shen, P. Tan, and S. Lin. Intrinsic image decomposition with non-local texture cues. In *CVPR*, 2008.

[18] L. Shen and C. Yeo. Intrinsic images decomposition using a local and global sparse representation of reflectance. In *CVPR*, 2011.

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

[20] M. F. Tappen, W. T. Freeman, and E. H. Adelson. Recovering intrinsic images from a single image. *TPAMI*, 27(9):1459–1472, 2005.

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

[22] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *TIP*, 13(4):600–612, 2004.

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

[25] L. Xu, S. Zheng, and J. Jia. Unnatural L_0 sparse representation for natural image deblurring. In *CVPR*, 2013.