Robust Separation of Reflection from Multiple Images

Xiaojie Guo[1], Xiaochun Cao[1] and Yi Ma[2]
[1]State Key Laboratory Of Information Security, IIE, Chinese Academy of Sciences, Beijing, 100093, China
[2]School of Information Science and Technology, ShanghaiTech University, Shanghai, 200031, China
xj.max.guo@gmail.com caoxiaochun@iie.ac.cn mayi@shanghaitech.edu.cn

Abstract

When one records a video/image sequence through a transparent medium (e.g. glass), the image is often a superposition of a transmitted layer (scene behind the medium) and a reflected layer. Recovering the two layers from such images seems to be a highly ill-posed problem since the number of unknowns to recover is twice as many as the given measurements. In this paper, we propose a robust method to separate these two layers from multiple images, which exploits the correlation of the transmitted layer across multiple images, and the sparsity and independence of the gradient fields of the two layers. A novel Augmented Lagrangian Multiplier based algorithm is designed to efficiently and effectively solve the decomposition problem. The experimental results on both simulated and real data demonstrate the superior performance of the proposed method over the state of the arts, in terms of accuracy and simplicity.

(a) (b) (c)

Figure 1. (a) Sample images with reflection, in which the regions of interest are bounded by blue windows. (b) and (c) are the corresponding reflected and transmitted layers recovered by our method, respectively.

1. Introduction

As mobile imaging devices become more and more popular, we see more consumer videos or image sequences taken under less controlled conditions. Very often people are shooting a video through a transparent medium such as glass. For instance, one might take a video of a busy street through the window of his office; or we may take images of a glass-framed painting. In such cases, the images will contain both the scene transmitted through the medium and some reflection. For the purpose of image enhancement, it is often desirable to be able to separate the transmitted component and the reflected one. Figure 1 shows one such example: Two sample images of a glass-framed picture taken by a mobile phone (a), its reflection (b) and transmitted component (c) (the picture behind the glass) recovered by our method.

Mathematically, we can model the captured superimposed image $\boldsymbol{f}$ as a linear combination of two components: $\boldsymbol{f} = \boldsymbol{t} + \boldsymbol{r}$, where $\boldsymbol{t}$ and $\boldsymbol{r}$ represent the transmitted scene and the reflection, respectively. The goal of this paper is to recover the two layers from a sequence of images. However, from this model, we see that the number of unknowns to be recovered is twice as many as that of the given measurements, which indicates that the problem is severely ill-posed. Even with multiple observed images, the problem remains under-determined. Therefore, to make the problem well-posed, we need to impose additional priors on the desired solution for $\boldsymbol{t}$ and $\boldsymbol{r}$.

Reflection has always been an annoying nuisance for high-quality imaging that professional or computational photographers try to reduce or remove. [3] and [15] propose to separate the reflection using two images captured by rotating the polarizing lens with different angles, and then find an optimal way to linearly combine the two images. More polarization filter based techniques can be found in the literature [14, 8]. [1] develops a similar method to reduce the reflection effect through employing a flash and no-flash image pair. Although these approaches can effectively reduce reflection, they require the photographer have professional photographing skills and tools, which limits the applicabil-

ity of such methods to typical consumers.

Levin *et al.* attempt to release the professional requirement on photographers [10, 11], and develop a user-assisted system [9] to recover the transmitted and reflected layers from a single image, in which users need to interactively label part of the gradients as belonging to one of the two layers. Alternatively, with the assistance of a user, [20] introduces an expectation-maximization algorithm with a hidden Markov model to accomplish the task of layer decomposition from a single image. Although manually dealing with a few images is acceptable, such methods become impractical if one has to deal with many images or a video sequence. Automatic methods are more desired in such cases.

Given a sequence of images with reflection, the relative motion between the two components can be exploited to decompose the two layers. A variety of schemes have been proposed to extract multiple motions from image sequences, like [7, 6, 19]. However, they mostly care only about recovering the motions, not restoring the two components. [17] and [18] make use of the relative motion to further restore the layers. But, they require sufficient variation in the motion and their performance may significantly degrade when that condition is violated. More recently, Béery and Yeredor [2] propose to decompose superimposed images of two shifting layers. [4] introduces a fast algorithm named sparse blind separation with spatial shifts to achieve the goal. But both are under the assumption that the motions are only uniform translations, and thus are not applicable to general cases. [5] gives a more general approach to blindly separating superimposed images using image statistics, the results of which are promising. However, the main limitation of this method is that it requires a considerably large amount of memory to process. The model of RASL [13] fits that of the reflection separation task from the perspective of component decomposition. Since it barely considers the relationship and characteristics of the two components, the visual quality of the recovered layers is not guaranteed. [16] recovers the two components relying on two layer stereo matching and multiple depth estimating. As the reflection usually occupies only a small fraction of the image and has very low intensity, feature matching and motion estimation for the reflected layer are highly likely to be inaccurate, if not impossible, for most practical sequences (like the one shown in Figure 1).

Contributions. In this paper, we show how to decompose the transmitted and reflected layers for a sequence of images by exploiting some strong structural priors in both the transmitted layer and the reflected one. More specifically, our framework will harness three structural priors in a unified fashion: 1) the *correlation* of the transmitted layer across different image frames, 2) the *sparsity* of the gradient fields of the two layers, and 3) the *independence* between the gradient fields of the transmitted and the reflected layers. Additionally, in an image sequence, the superimposed region can be scaled, rotated, or deformed throughout the sequence, *e.g.* Figure 1. Our method will automatically seek an optimal alignment of the region of interest in all images. The only interaction required from the user is to specify the region of interest in the very first image frame, *e.g.* the blue window in the upper image of Figure 1 (a), and the rest will be computed automatically. We propose an efficient algorithm based on augmented Lagrangian multiplier and alternating direction minimization method to solve the associated optimization problem. We conduct extensive experiments to verify the effectiveness of our method in comparison with the state of the art.

2. Our Method

2.1. Problem Formulation

Recall that the superimposed image (area) is a linear combination of the transmitted layer and the reflection. For an image sequence, the superimposed area in every image frame satisfies $\forall i \in [1, ..., n], \boldsymbol{f}_i = \boldsymbol{t}_i + \boldsymbol{r}_i$. If we collect each frame as a column of a matrix, the above relationship can be rewritten in a matrix form as $\boldsymbol{F} = \boldsymbol{T} + \boldsymbol{R}$, where the columns of $\boldsymbol{F} \in \mathbb{R}^{m \times n}$ are the vectorized images and m is the number of pixels. $\boldsymbol{T}$ and $\boldsymbol{R}$ represent the transmitted and the reflected components, respectively.

Structural priors of the solution. The multiple images of the transmitted layer $\boldsymbol{t}_i$ are strongly correlated. As a result, we here introduce *the correlation prior: if the transmitted area is well aligned in the multiple frames, the rank of the matrix $\boldsymbol{T}$ is low, ideally* 1. It is well known that natural images are largely piecewise smooth and their gradient fields are typically sparse. We call this *the edge-sparse prior: the responses of both two layers, $\sum_{j=1}^{J} \|\boldsymbol{d}_j * \boldsymbol{t}_i\|_0$ and $\sum_{j=1}^{J} \|\boldsymbol{d}_j * \boldsymbol{r}_i\|_0$, to derivative-like filters ($\boldsymbol{d}_1$, $\boldsymbol{d}_2$,...,$\boldsymbol{d}_J$) are sparse.* $\|\cdot\|_0$ denotes the ℓ^0 norm, and $*$ is the operator of convolution. In this work, we only employ the filters in horizontal direction $\boldsymbol{d}_1$ and in vertical direction $\boldsymbol{d}_2$. For brevity, we define $\|\boldsymbol{DT}\|_0 \equiv \sum_{i=1}^{n} \sum_{j=1}^{2} \|\boldsymbol{d}_j * \boldsymbol{T}_i\|_0$ and $\|\boldsymbol{DR}\|_0 \equiv \sum_{i=1}^{n} \sum_{j=1}^{2} \|\boldsymbol{d}_j * \boldsymbol{R}_i\|_0$. In addition, gradient fields of the two layers should be statistically uncorrelated. Thus, *the independence prior: the two layers' responses to derivative filters are (approximately) independent.* Furthermore, we observe that the fraction of reflection is usually much smaller and sparser than that of the transmitted layer. Of course, as real images, both the transmitted and the reflected components have to have non-negative values.

In real cases, the region of interest is distorted differently in different frames. We assume that the targeting region lies on a (nearly) planar surface in the scene. Then there exist

2D homographs, say $(\tau_1, \tau_2,\ldots,\tau_n)$, transforming the misaligned regions to well aligned $\boldsymbol{f}_1 \circ \tau_1, \boldsymbol{f}_2 \circ \tau_2,\ldots, \boldsymbol{f}_n \circ \tau_n$. Based on the priors and the constraints stated above, the desired decomposition $(\boldsymbol{T}, \boldsymbol{R})$ should minimize the following objective:

$$\begin{aligned} &\min \ \ \operatorname{rank}(\boldsymbol{T}) + \lambda_1\|\boldsymbol{M}\|_1 + \lambda_2\|\boldsymbol{N}\|_F^2 + \lambda_3\|\boldsymbol{DT}\|_0 + \\ &\lambda_4\|\boldsymbol{DR}\|_0 + \lambda_5\|\boldsymbol{DT}\odot\boldsymbol{DR}\|_0 + \lambda_6\|\boldsymbol{\Omega} - \boldsymbol{DT} - \boldsymbol{DR}\|_F^2, \\ &\text{s.t.} \ \ \boldsymbol{F}\circ\Gamma = \boldsymbol{T} + \boldsymbol{M}; \boldsymbol{M} = \boldsymbol{R} + \boldsymbol{N}; \boldsymbol{T} \succeq 0; \ \ \boldsymbol{R} \succeq 0; \end{aligned} \tag{1}$$

where $\|\cdot\|_1$ and $\|\cdot\|_F$ stand for the ℓ^1 norm and the Frobenius norm respectively, $\odot$ means element-wise multiplication, and $\boldsymbol{\Omega} \doteq \boldsymbol{DF}$ which can be computed beforehand. λ_1, λ_2, λ_3, λ_4 , λ_5 and λ_6 are the coefficients controlling the weights of different terms. Γ consists of all the transformations, say $[\tau_1, \tau_2, \ldots, \tau_n]$. Please notice that we utilize $\boldsymbol{M}$ to represent the residual between the observation and the transmitted component, which can be split into the reflection $\boldsymbol{R}$ and a noise term $\boldsymbol{N}$.

In the above objective function (1), the first term enforces the (aligned) transmitted regions to be highly correlated. The residual $\boldsymbol{M}$ should be (approximately) sparse in its spatial support as reflection $\boldsymbol{R}$ is typically sparse. The third term penalizes the Gaussian noise. The fourth and fifth terms essentially enforce the recovered two layers to have sparse gradient fields; and the remaining two terms enforce they are independent of each other. Note that the non-negative properties of the two layers are enforced as hard constraints in the above formulation.

2.2. Optimization

As we have seen in (1), it has combined all aforementioned priors and constraints for decomposing the superimposed images in a unified optimization framework. However, it is extremely difficult to directly minimize (1). There are two main difficulties: 1) the non-convexity of the rank function and the ℓ^0 norm; and 2) the nonlinearity of the constraint $\boldsymbol{F}\circ\Gamma = \boldsymbol{T} + \boldsymbol{M}$ due to domain transformations by Γ. To overcome these obstacles, we will use convex surrogates for all the non-convex low rank and sparsity promoting terms. To deal with the nonlinear constraints, we will linearize them with respect to a current estimate and solve the nonlinear problem iteratively.

Specifically, through convex relaxation, we may replace the rank function and the ℓ^0 norm with the nuclear norm $\|\cdot\|_*$ and the ℓ^1 norm, respectively. As for the alignment constraint, we linearize with respect to the transformation and obtain: $\boldsymbol{F}\circ\Gamma + \sum_{i=1}^n \boldsymbol{J}_i\Delta\Gamma\epsilon_i\epsilon_i^T = \boldsymbol{T} + \boldsymbol{M}$, where $\boldsymbol{J}_i$ is the Jacobian of the i^{th} region with respect to the transformation parameters τ_i, and $\{\epsilon_i\}$ is the standard basis for $\mathbb{R}^n$. The linearization effectively approximates the original constraints around the current estimate when the transformations change infinitesimally. With the convex relaxation and the linearization, the (linearized) optimization problem can be rewritten as:

$$\begin{cases} \min \|\boldsymbol{T}\|_* + \lambda_1\|\boldsymbol{M}\|_1 + \lambda_2\|\boldsymbol{N}\|_F^2 + \lambda_3\|\boldsymbol{DT}\|_1 + \\ \lambda_4\|\boldsymbol{DR}\|_1 + \lambda_5\|\boldsymbol{DT}\odot\boldsymbol{DR}\|_1 + \lambda_6\|\boldsymbol{\Omega} - \boldsymbol{DT} - \boldsymbol{DR}\|_F^2, \\ \quad \text{s.t.} \ \ \boldsymbol{F}\circ\Gamma + \sum_{i=1}^n \boldsymbol{J}_i\Delta\Gamma\epsilon_i\epsilon_i^T = \boldsymbol{T} + \boldsymbol{M}; \\ \qquad \boldsymbol{M} = \boldsymbol{R} + \boldsymbol{N}; \boldsymbol{T} \succeq 0; \ \ \boldsymbol{R} \succeq 0. \end{cases} \tag{2}$$

With a proper initialization of Γ, we solve (2) in an iterative fashion so as to converge to a (local) optimal solution for the original problem.

The Augmented Lagrange Multiplier (ALM) with Alternating Direction Minimizing (ADM) strategy [12] has proven to be an efficient and effective solver of problems like (2) (the inner loop). To adopt ALM-ADM to our problem, we need to make our objective function separable. Thus we introduce three auxiliary variables, *i.e.* $\boldsymbol{L}$, $\boldsymbol{K}$ and $\boldsymbol{Q}$, to replace $\boldsymbol{T}$, $\boldsymbol{DT}$, and $\boldsymbol{DR}$ in the objective function, respectively. Accordingly, $\boldsymbol{L} = \boldsymbol{T}$, $\boldsymbol{K} = \boldsymbol{DT}$ and $\boldsymbol{Q} = \boldsymbol{DR}$ act as the additional constraints. The augmented Lagrangian function of (2) is given by:

$$\begin{cases} \mathcal{L}_{\boldsymbol{T}\succeq 0;\boldsymbol{R}\succeq 0;}(\boldsymbol{L}, \boldsymbol{M}, \boldsymbol{N}, \boldsymbol{K}, \boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{R}, \Delta\Gamma) \\ = \|\boldsymbol{L}\|_* + \lambda_1\|\boldsymbol{M}\|_1 + \lambda_2\|\boldsymbol{N}\|_F^2 + \lambda_3\|\boldsymbol{K}\|_1 \\ \quad + \lambda_4\|\boldsymbol{Q}\|_1 + \lambda_5\|\boldsymbol{K}\odot\boldsymbol{Q}\|_1 + \lambda_6\|\boldsymbol{\Omega} - \boldsymbol{K} - \boldsymbol{Q}\|_F^2 \\ \quad + \Phi(\boldsymbol{Z}_1, \boldsymbol{F}\circ\Gamma + \sum_{i=1}^n \boldsymbol{J}_i\Delta\Gamma\epsilon_i\epsilon_i^T - \boldsymbol{T} - \boldsymbol{M}) \\ \quad + \Phi(\boldsymbol{Z}_2, \boldsymbol{M} - \boldsymbol{R} - \boldsymbol{N}) + \Phi(\boldsymbol{Z}_3, \boldsymbol{L} - \boldsymbol{T}) \\ \quad + \Phi(\boldsymbol{Z}_4, \boldsymbol{K} - \boldsymbol{DT}) + \Phi(\boldsymbol{Z}_5, \boldsymbol{Q} - \boldsymbol{DR}), \end{cases}$$

with the definition $\Phi(\boldsymbol{Z}, \boldsymbol{C}) \equiv \frac{\mu}{2}\|\boldsymbol{C}\|_F^2 + \langle \boldsymbol{Z}, \boldsymbol{C}\rangle$, where $\langle\cdot,\cdot\rangle$ represents matrix inner product and μ is a positive penalty scalar. $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$, $\boldsymbol{Z}_3$, $\boldsymbol{Z}_4$ and $\boldsymbol{Z}_5$ are the Lagrangian multipliers. Besides the Lagrangian multipliers, there are eight variables to solve. The solver iteratively updates one variable at a time by fixing the others. Fortunately, each step has a simple closed-form solution, and hence can be computed efficiently. For brevity, we denote $\boldsymbol{P}^t \equiv \boldsymbol{F}\circ\Gamma + \sum_{i=1}^n \boldsymbol{J}_i\Delta\Gamma^t\epsilon_i\epsilon_i^T$. Below, the solutions of the subproblems are provided:

$$\boldsymbol{L}\textbf{-subproblem: } \boldsymbol{L}^{t+1} = \operatorname*{argmin}_{\boldsymbol{L}} \|\boldsymbol{L}\|_* + \Phi(\boldsymbol{Z}_3^t, \boldsymbol{L} - \boldsymbol{T}^t) = \boldsymbol{U}\mathcal{S}_{\frac{1}{\mu^t}}[\boldsymbol{\Sigma}]\boldsymbol{V}^T, \tag{3}$$

where $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the Singular Value Decomposition (SVD) of $\left(\boldsymbol{T}^t - \frac{\boldsymbol{Z}_3^t}{\mu^t}\right)$. $\{\mu^t\}$ is a monotonically increasing positive sequence, and $\mathcal{S}_{\varepsilon>0}[\cdot]$ represents the shrinkage operator, the definition of which on scalars is: $\mathcal{S}_\varepsilon[x] = \operatorname{sgn}(x)\max(|x| - \varepsilon, 0)$. The extension of the shrinkage operator to vectors and

matrices is simply applied element-wise.

$$\begin{aligned}&\boldsymbol{M}\textbf{-subproblem: } \boldsymbol{M}^{t+1} = \underset{\boldsymbol{M}}{\operatorname{argmin}}\, \lambda_1\|\boldsymbol{M}\|_1 \\ &+ \Phi(\boldsymbol{Z}_1^t, \boldsymbol{P}^t - \boldsymbol{T}^t - \boldsymbol{M}) + \Phi(\boldsymbol{Z}_2^t, \boldsymbol{M} - \boldsymbol{R}^t - \boldsymbol{N}^t) \\ &= \mathcal{S}_{\frac{2\lambda_1}{\mu^t}}\Big[\boldsymbol{P}^t - \boldsymbol{T}^t + \boldsymbol{R}^t + \boldsymbol{N}^t + \frac{\boldsymbol{Z}_1^t - \boldsymbol{Z}_2^t}{\mu^t}\Big].\end{aligned} \tag{4}$$

$$\begin{aligned}&\boldsymbol{N}\textbf{-subproblem: } \boldsymbol{N}^{t+1} = \\ &\underset{\boldsymbol{N}}{\operatorname{argmin}}\, \lambda_2\|\boldsymbol{N}\|_F^2 + \Phi(\boldsymbol{Z}_2^t, \boldsymbol{M}^{t+1} - \boldsymbol{R}^t - \boldsymbol{N}) \\ &= \frac{\boldsymbol{Z}_2^t + \mu^t(\boldsymbol{M}^{t+1} - \boldsymbol{R}^t)}{2\lambda_2 + \mu^t}.\end{aligned} \tag{5}$$

$$\begin{aligned}&\boldsymbol{K}\textbf{-subproblem: } \boldsymbol{K}^{t+1} = \\ &\underset{\boldsymbol{K}}{\operatorname{argmin}}\, \lambda_3\|\boldsymbol{K}\|_1 + \lambda_5\|\boldsymbol{K} \odot \boldsymbol{Q}^t\|_1 \\ &+ \lambda_6\|\boldsymbol{\Omega} - \boldsymbol{Q}^t - \boldsymbol{K}\|_F^2 + \Phi(\boldsymbol{Z}_4^t, \boldsymbol{K} - \boldsymbol{D}\boldsymbol{T}^t) \\ &= \hat{\mathcal{S}}_{\frac{\lambda_3\mathbf{1} + \lambda_5|\boldsymbol{Q}^t|}{2\lambda_6 + \mu^t}}\Big[\frac{\lambda_6(\boldsymbol{\Omega} - \boldsymbol{Q}^t) + \mu^t\boldsymbol{D}\boldsymbol{T}^t/2 - \boldsymbol{Z}_4^t/2}{\lambda_6 + \mu^t/2}\Big],\end{aligned} \tag{6}$$

where $\hat{\mathcal{S}}_{\boldsymbol{W}}[\boldsymbol{X}]$ performs the shrinkage on the elements of $\boldsymbol{X}$ with thresholds given by corresponding entries of $\boldsymbol{W}$.

$$\begin{aligned}&\boldsymbol{Q}\textbf{-subproblem: } \boldsymbol{Q}^{t+1} = \\ &\underset{\boldsymbol{Q}}{\operatorname{argmin}}\, \lambda_4\|\boldsymbol{Q}\|_1 + \lambda_5\|\boldsymbol{K}^{t+1} \odot \boldsymbol{Q}\|_1 \\ &+ \lambda_6\|\boldsymbol{\Omega} - \boldsymbol{Q} - \boldsymbol{K}^{t+1}\|_F^2 + \Phi(\boldsymbol{Z}_5^t, \boldsymbol{Q} - \boldsymbol{D}\boldsymbol{R}^t) \\ &= \hat{\mathcal{S}}_{\frac{\lambda_4\mathbf{1} + \lambda_5|\boldsymbol{K}^{t+1}|}{2\lambda_6 + \mu^t}}\Big[\frac{\lambda_6(\boldsymbol{\Omega} - \boldsymbol{K}^{t+1}) + \mu^t\boldsymbol{D}\boldsymbol{R}^t/2 - \boldsymbol{Z}_5^t/2}{\lambda_6 + \mu^t/2}\Big].\end{aligned} \tag{7}$$

$$\begin{aligned}&\boldsymbol{T}\textbf{-subproblem: } \boldsymbol{T}^{t+1} = \\ &\underset{\boldsymbol{T}}{\operatorname{argmin}}\, \Phi(\boldsymbol{Z}_1^t, \boldsymbol{P}^t - \boldsymbol{M}^{t+1} - \boldsymbol{T}) \\ &+ \Phi(\boldsymbol{Z}_3^t, \boldsymbol{L}^{t+1} - \boldsymbol{T}) + \Phi(\boldsymbol{Z}_4^t, \boldsymbol{K}^{t+1} - \boldsymbol{D}\boldsymbol{T}).\end{aligned}$$

By assuming circular boundary conditions, we can apply 2D FFT on the $\boldsymbol{T}$-subproblem, which enables us to compute the solution fast. So, for each $\boldsymbol{T}_i\ \forall i \in [1, ..., n]$, we have

$$\boldsymbol{T}_i^{t+1} = \mathcal{F}^{-1}\Big(\mathcal{F}\big(\mathcal{R}(\boldsymbol{O}_i)\big)/\big(\overline{\mathcal{F}(\boldsymbol{D})} \odot \mathcal{F}(\boldsymbol{D}) + \mathbf{2}\big)\Big), \tag{8}$$

where $\boldsymbol{O} \equiv \boldsymbol{P}^t - \boldsymbol{M}^{t+1} + \boldsymbol{L}^{t+1} + \frac{\boldsymbol{Z}_1^t + \boldsymbol{Z}_3^t}{\mu^t} + \boldsymbol{D}^T(\boldsymbol{K}^{t+1} + \frac{\boldsymbol{Z}_4^t}{\mu^t})$. $\mathcal{R}(\cdot)$ is to reshape the vectorized 2D information back to its 2D form. $\mathcal{F}(\cdot)$ is the 2D FFT operator, while $\mathcal{F}^{-1}(\cdot)$ and $\overline{\mathcal{F}(\cdot)}$ stand for the 2D inverse FFT and the complex conjugate of $\mathcal{F}(\cdot)$, respectively. The division is conducted component-wise.

$$\begin{aligned}&\boldsymbol{R}\textbf{-subproblem: } \boldsymbol{R}^{t+1} = \underset{\boldsymbol{R}}{\operatorname{argmin}} \\ &\Phi(\boldsymbol{Z}_2^t, \boldsymbol{M}^{t+1} - \boldsymbol{N}^{t+1} - \boldsymbol{R}) + \Phi(\boldsymbol{Z}_5^t, \boldsymbol{Q}^{t+1} - \boldsymbol{D}\boldsymbol{R}).\end{aligned}$$

Algorithm 1: SID: Superimposed Image Decomposition

Input: $\lambda_1 > 0, \lambda_2 > 0, \lambda_3 > 0, \lambda_4 > 0, \lambda_5 > 0, \lambda_6 > 0$. The observation $\boldsymbol{F}$, and the initial transformation Γ.

while *not converged* **do**

  $\boldsymbol{L}^0 = \boldsymbol{M}^0 = \boldsymbol{N}^0 = \boldsymbol{T}^0 = \boldsymbol{R}^0 = \boldsymbol{Z}_1^0 = \boldsymbol{Z}_2^0 = \boldsymbol{Z}_3^0 = \mathbf{0} \in \mathbb{R}^{m\times n}$, $\Delta\Gamma^t = \mathbf{0}$, $t = 0$, $\mu^0 > 0$, $\rho > 1$, $\boldsymbol{K}^0 = \boldsymbol{Q}^0 = \boldsymbol{Z}_4^0 = \boldsymbol{Z}_5^0 = \mathbf{0} \in \mathbb{R}^{2m\times n}$. Compute the warped areas $\boldsymbol{F} \circ \Gamma$ and their Jacobians $\boldsymbol{J}_i = \frac{\partial}{\partial \tau_i}\boldsymbol{F}_i \circ \tau_i$.

  while *not converged* **do**

    Update $\boldsymbol{L}^{t+1}$ via Eq. (3);
    Update $\boldsymbol{M}^{t+1}$ via Eq. (4);
    Update $\boldsymbol{N}^{t+1}$ via Eq. (5);
    Update $\boldsymbol{K}^{t+1}$ via Eq. (6);
    Update $\boldsymbol{Q}^{t+1}$ via Eq. (7);
    for i *from* 1 *to* n **do**
      Update $\boldsymbol{T}_i^{t+1}$ via Eq. (8);
      Update $\boldsymbol{R}_i^{t+1}$ via Eq. (9);
    end
    $\boldsymbol{T}^{t+1}(\boldsymbol{T}^{t+1} < 0) = 0; \boldsymbol{R}^{t+1}(\boldsymbol{R}^{t+1} < 0) = 0$;
    Update $\Delta\Gamma^{t+1}$ via Eq. (10);
    Update the multipliers via Eq. (11);
    $\mu^{t+1} = \mu^t\rho$; $t = t + 1$;
  end
  $\Gamma = \Gamma + \Delta\Gamma^t$;
end

Output: Optimal solution ($\boldsymbol{T}^* = \boldsymbol{T}^t$, $\boldsymbol{R}^* = \boldsymbol{R}^t$).

Similar to (8), for each $\boldsymbol{R}_i\ \forall i \in [1, ..., n]$, $\boldsymbol{R}_i^{t+1} =$

$$\mathcal{F}^{-1}\Big(\mathcal{F}\big(\mathcal{R}(\boldsymbol{E}_i)\big)/\big(\overline{\mathcal{F}(\boldsymbol{D})} \odot \mathcal{F}(\boldsymbol{D}) + \mathbf{1}\big)\Big), \tag{9}$$

with $\boldsymbol{E} \equiv \boldsymbol{M}^{t+1} - \boldsymbol{N}^{t+1} + \frac{\boldsymbol{Z}_2^t}{\mu^t} + \boldsymbol{D}^T(\boldsymbol{Q}^{t+1} + \frac{\boldsymbol{Z}_5^{t+1}}{\mu^t})$. To guarantee $\boldsymbol{T}$ and $\boldsymbol{R}$ to be non-negative, we set the negative elements in them to be zero after (8) and (9), respectively.

$$\begin{aligned}&\Delta\Gamma\textbf{-subproblem: } \Delta\Gamma^{t+1} = \\ &\underset{\Delta\Gamma}{\operatorname{argmin}}\, \Phi(\boldsymbol{Z}_1^t, \boldsymbol{F} \circ \Gamma + \sum_{i=1}^{n} \boldsymbol{J}_i\Delta\Gamma\epsilon_i\epsilon_i^T - \boldsymbol{T}^{t+1} - \boldsymbol{M}^{t+1}) \\ &= \sum_{i=1}^{n} \boldsymbol{J}_i^\dagger(\boldsymbol{T}^{t+1} + \boldsymbol{M}^{t+1} - \boldsymbol{F} \circ \Gamma - \frac{\boldsymbol{Z}_1^t}{\mu^t})\epsilon_i\epsilon_i^T,\end{aligned} \tag{10}$$

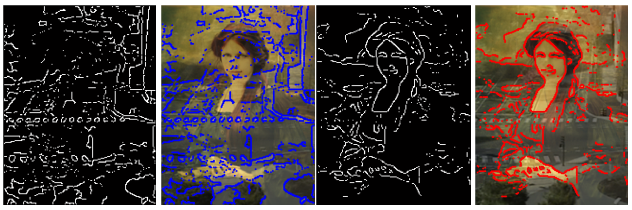
where $\boldsymbol{J}^\dagger$ denotes the Moore-Penrose pseudoinverse of $\boldsymbol{J}$.

Besides, there are still five multipliers to update, which are simply given by:

$$\begin{aligned}\boldsymbol{Z}_1^{t+1} &= \boldsymbol{Z}_1^t + \mu^t(\boldsymbol{P}^{t+1} - \boldsymbol{T}^{t+1} - \boldsymbol{M}^{t+1}); \\ \boldsymbol{Z}_2^{t+1} &= \boldsymbol{Z}_2^t + \mu^t(\boldsymbol{M}^{t+1} - \boldsymbol{R}^{t+1} - \boldsymbol{N}^{t+1}); \\ \boldsymbol{Z}_3^{t+1} &= \boldsymbol{Z}_3^t + \mu^t(\boldsymbol{L}^{t+1} - \boldsymbol{T}^{t+1}); \\ \boldsymbol{Z}_4^{t+1} &= \boldsymbol{Z}_4^t + \mu^t(\boldsymbol{K}^{t+1} - \boldsymbol{D}\boldsymbol{T}^{t+1}); \\ \boldsymbol{Z}_5^{t+1} &= \boldsymbol{Z}_5^t + \mu^t(\boldsymbol{Q}^{t+1} - \boldsymbol{D}\boldsymbol{R}^{t+1}).\end{aligned} \tag{11}$$

(a) (b) (c) (d)

Figure 2. The images used to synthesize the simulated data. (a) is an image used as the transmitted layer $\boldsymbol{t}$. (b), (c) and (d) are the reflected layer $\boldsymbol{r}$ for different frames, respectively.



(a) (b) (c) (d)

Figure 3. Illustration of input to the algorithm [9]. (a) and (c) are the gradients of the reflected and the transmitted layers, respectively. (b) and (d) are the marked results on the synthesized image according to (a) and (c), respectively.

For clarity, the entire algorithm of solving the problem (1) is summarized in Algorithm 1. The outer loop of Algorithm 1 terminates when the change of objective function value between two neighboring iterations is sufficiently small or the maximal number of iterations is reached. The inner loop is stopped when $\|\boldsymbol{P}^{t+1} - \boldsymbol{T}^{t+1} - \boldsymbol{M}^{t+1}\|_F \leq \delta\|\boldsymbol{F} \circ \Gamma\|_F$ with $\delta = 10^{-6}$ or the maximal number of inner iterations is reached. To give a reasonable initialization of the transformation Γ, we employ feature matching with RANSAC. The only user intervention is to specify the targeting region in one image.

3. Experiments

In this section, we verify the efficacy of our method on both simulated and real data, and demonstrate the advantages of the proposed algorithm compared to the state of the arts including SIUA [9], SPBS-M [5] and RASL [13][1]. The Matlab codes of these methods can be downloaded from the authors' websites, the parameters of which are all set as default. Unless otherwise stated, the parameters of Algorithm 1 (referred to as SID) are fixed throughout the experiments empirically: $\lambda_1 = 0.3w$, $\lambda_2 = 50w$, $\lambda_3 = 1w$, $\lambda_4 = 5w$, $\lambda_5 = 50w$ and $\lambda_6 = 50w$ with $w = \frac{1}{\sqrt{m}}$. For color images, we apply the algorithms to each of the R, G and B channel, then concatenate the three as the final results. The experiments are conducted in Matlab on a PC running Windows 7 32bit operating system with Intel Core i7 3.4 GHz CPU and 4.0 GB RAM.

We first synthesize a sequence including 15 superimposed images (resolution: 215×162), the picture shown in Figure 2 (a) is used as the transmitted layer $\boldsymbol{t}$ while the others in Figure 2 as reflections $\boldsymbol{r}_i$. The superimposition takes the form of $\boldsymbol{f}_i = 0.6\boldsymbol{t} + 0.4\boldsymbol{r}_i$. Please note that, in the simulation, we focus on the decomposition performance of the compared algorithms. The results are compared in Figure 4. The top row is SIUA [9] *vs.* SID. Since SIUA requires the interaction from the users, to guarantee that our implementation is correct, we compute the Canny edges on both the superimposed images (the leftmost image in the top row of Figure 4) and the transmitted layer (Figure 2 (a)), instead of labeling the gradients manually. With the Canny edges, say $\boldsymbol{c}_s$ from the mixture and $\boldsymbol{c}_t$ from the ground truth transmitted layer, the gradients belonging to the transmitted layer are finally given by the intersection between $\boldsymbol{c}_s$ and $\boldsymbol{c}_t$ as shown in Figure 3 (c) and (d). The intersection preserves the gradients from the transmitted as well as eliminates the ones overlapping with those of the reflection. While the difference set between $\boldsymbol{c}_s$ and $\boldsymbol{c}_t$ is used to indicate the reflection as shown in Figure 3 (a) and (b). Although, as reported by the authors of [5], their method can be applied to multiple images, the algorithm is, in practice, not able to handle more than three images with the original resolu-

(a) (b) (c) (d) (f)

Figure 4. Visual comparison of the decomposed results. (a) The superimposed images. (b) The recovered reflections by the competitors. (c) The reflected layers obtained by our method. (d) The transmitted layers recovered by the competitors. (f) The transmitted layers by our method. **Top row:** SIUA [9] *vs.* SID. **Middle row:** SPBS-M [5] *vs.* SID. **Bottom row:** RASL [13] *vs.* SID.

[1] Since the code of [16] is not available when this paper is prepared, we do not compare with [16].

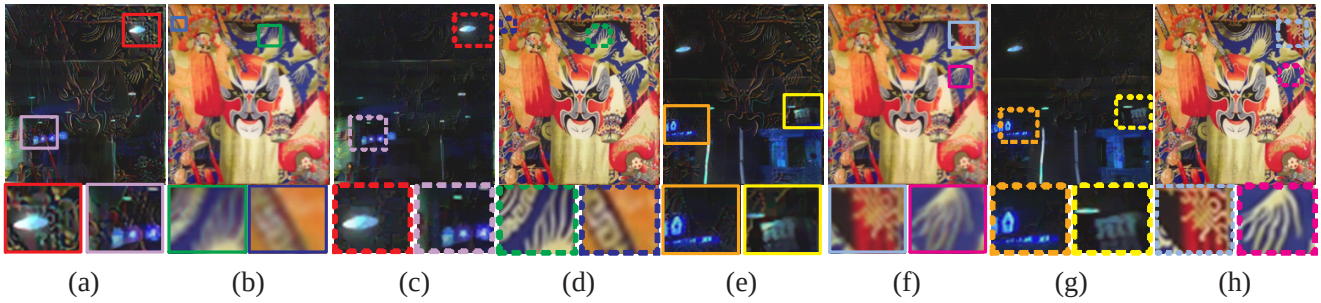


Figure 6. The benefit of alignment from Algorithm 1. (a) and (b) are the decomposed layers, *i.e.* the reflected and the transmitted, of one superimposed image after 1^{st} iteration. (c) and (d) are the final results corresponding to (a) and (b). (e) and (f) give the decomposition on another superimposed image after 1^{st} iteration, while (g) and (h) are the final converged results.

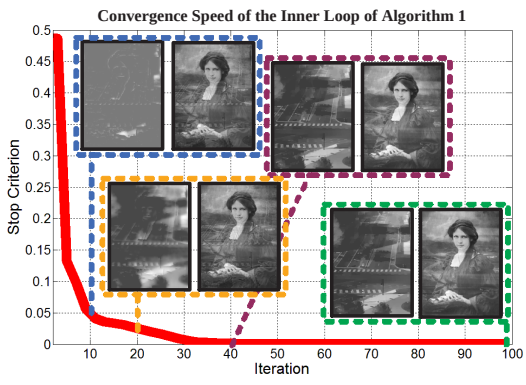


Figure 5. The convergence speed of the inner loop of Algorithm 1.

tion due to the requirement of large memory. Thus, to make the comparison as fair as possible, we alternatively downsample the images to resolution 179×135, 6 of which (the most can be handled together on our PC) are used as the input to [5]. Moreover, for [5], we use the gray-scale images of the mixtures to estimate the layer motion parameters without using the color information, which is crucial for [5] to further compute the layer gradients and reconstruct the source layers[2]. Then R, G and B channels are separately reconstructed using the same parameters to avoid the inconsistency of motion (please see the middle row of Figure 4). The bottom row of Figure 4 shows the comparison between RASL [13] and SID.

Both RASL and SID can simultaneously deal with multiple images, *e.g.* all the 15 images in this simulation. As can be seen from the results, SID significantly outperforms the others in terms of quality of the recovered images. The difference is better viewed in electronic version with room-in. The results of SIUA and SPBS-M have the problem of color consistency and ghosting effect. RASL achieves better performance than SIUA and SPBS-M, but there is ghosting effect in both the recovered transmitted and reflected layers. In terms of speed, SIUA spends about $13s$ for only one image without considering the gradient labeling time by the users. For SPBS-M, it takes almost $536s$ for 6 downsampled images, while RASL and SID cost about $4s$ and $30s$ for 15 images, respectively.

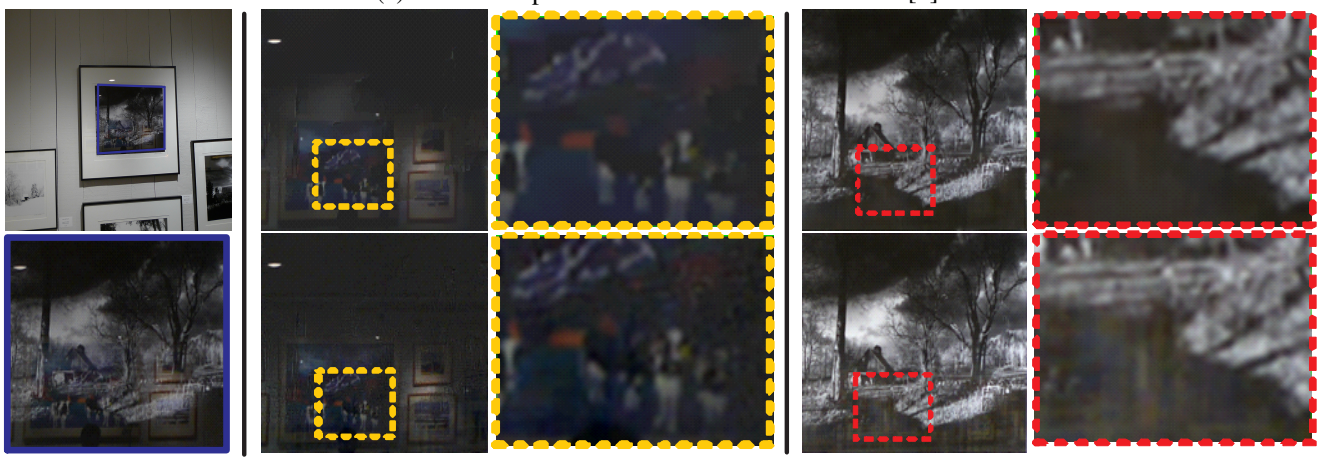
Figure 5 displays the convergence speed of the inner loop of Algorithm 1, without loss of generality, on the R channel of the synthesized sequence, in which the stop criterion sharply drops to the level of 10^{-4} with about $40 - 50$ iterations. We also show four pairs of the decomposed layers at 10, 20, 40 and 100 iterations. We see that the results at 40 iterations is very close to those at 100.

For experiments below, we apply SID to real data. For real images, the targeting region usually appears with various poses in different images, therefore the alignment of the regions across different images needs to be taken into account. Figure 6 provides the evidence about the benefit of the alignment, in which (a) and (b) ((e) and (f)) are the decomposed reflection and transmission components for one superimposed image after 1^{st} iteration of the outer loop of Algorithm 1, while (c) and (d) ((g) and (h)) are the final results. It is easy to see that as the alignment gets better, so does the separation, as indicated by the zoomed-in patches in Figure 6. More comparisons are given in Figure 7. The proposed SID algorithm obtains the most visually pleasing results not only for the transmitted layer but also the reflected layer among all the methods.

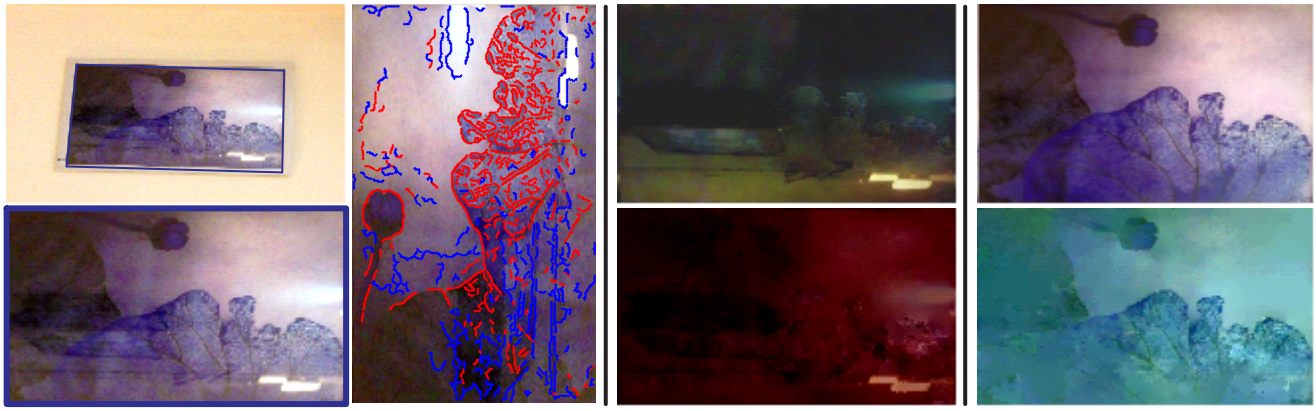
Finally, we show a failure case in Figure 8, in which the original superimposed image is blurred. The second picture of Figure 8 is the rectified region of interest. Notice that the recovered reflection is largely correct but still with some ghosting effect, as shown in the third image of Figure 8. Nevertheless, the transmitted layer has blur reduced and is of relatively good quality thanks to the correlation from multiple images (the rightmost image shown in Figure 8).

[2]As the motions on different channels for the same layers should be consistent, if we separately estimate the model parameters for the R, G, B channels, the estimated motion parameters can be different, which lead to color consistency issue and ghosting.

(a) Visual comparison between SID and SPBS-M [5].



(b) Visual comparison between SID and RASL [13].



(c) Visual comparison between SID and SIUA [9].

Figure 7. **Left:** the original images are displayed on the upper rows, the targeting superimposed regions are marked by blue windows, and the rectified versions of the regions are given below the originals. **Middle:** the decomposed reflection by our method SID (upper) and the competitor (lower), the highlighted patches are zoomed in and shown to the right. **Right:** the recovered transmitted layer by our method SID (upper) and the competitor (lower), the highlighted patches are zoomed in and shown to the right. Specially for (c), we give the mark-up for SIUA [9].

Figure 8. A failure case with motion blurred input images. **Left:** the original superimposed image. **Middle Left:** the rectified region of interest. **Middle Right:** the recovered reflection. **Right:** the recovered transmitted layer.

4. Conclusion

Reflection separation from superimposed images is an interesting, yet severely ill-posed problem. To overcome its difficulty, this paper has shown how to harness three prior structures of decomposed layers, including the correlation, the sparsity, and the independence priors, to make the problem well-defined and feasible to solve. We have formulated the problem in a unified optimization framework and proposed an efficient algorithm to find the optimal solution. The experimental results, compared to the state of the arts, have demonstrated the clear advantages of the proposed method in terms of speed, accuracy, and simplicity. In addition, both the transmitted and the reflected layers are recovered with high quality by our method, which can be used for many advanced image/video processing, rendering, or manipulation tasks.

Acknowledgment

This work was supported by National Natural Science Foundation of China (No.61332012), National Hightech R&D Program of China (2014BAK11B03), and 100 Talents Programme of The Chinese Academy of Sciences.

References

[1] A. Agrawal, R. Raskar, S. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Trans. Graphics*, 23(3):828–835, 2005. 1

[2] E. Béery and A. Yeredor. Blind separation of superimposed shifted images using parameterized joint diagonalization. *IEEE TIP*, 17(3):340–353, 2008. 2

[3] H. Farid and E. Adelson. Separating reflections and lighting using idependent components analysis. In *CVPR*, pages 262–267, 1999. 1

[4] K. Gai, Z. Shi, and C. Zhang. Blindly separating mixtures of multiple layers with spatial shifts. In *CVPR*, pages 1–8, 2008. 2

[5] K. Gai, Z. Shi, and C. Zhang. Blind separation of superimposed moving images using image statistics. *IEEE TPAMI*, 34(1):19–32, 2012. 2, 5, 6, 7

[6] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *IJCV*, 12(1):5–16, 1994. 2

[7] A. Jepson and M. Black. Mixture models for optical flow computation. In *CVPR*, pages 760–761, 1993. 2

[8] N. Kong, Y. Tai, and S. Shin. A physically-based approach to reflection separation. In *CVPR*, pages 9–16, 2012. 1

[9] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. *IEEE TPAMI*, 29(9):1647–1654, 2007. 2, 5, 7

[10] A. Levin, A. Zomet, and Y. Weiss. Learning to perceive transparency from the statistics of natural scenes. In *NIPS*, pages 1271–1278, 2002. 2

[11] A. Levin, A. Zomet, and Y. Weiss. Separating reflections from a single image using local features. In *CVPR*, pages 306–313, 2004. 2

[12] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical Report UILU-ENG-09-2215, UIUC Technical Report, 2009. 3

[13] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE TPAMI*, 34(11):2233–2246, 2012. 2, 5, 6, 7

[14] B. Sarel and M. Irani. Separating transparent layers through layer informaion exchange. In *ECCV*, pages 328–341, 2004. 1

[15] Y. Schechner, J. Shamir, and N. Kiryati. Polarization-based decorrelation of transparent layers: the inclination angle of an invisible surface. In *ICCV*, pages 814–819, 1999. 1

[16] S. Sinha, J. Kopf, M. Goesele, D. Scharstein, and R. Szeliski. Image-based rendering for scenes with reflections. *ACM Trans. Graphics*, 31(4):100:1–100:10, 2012. 2, 5

[17] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *CVPR*, pages 246–253, 2000. 2

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

[19] Y. Weiss and E. Adelson. A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. In *CVPR*, pages 321–326, 1996. 2

[20] S. Yeung, T. Wu, and C. Tang. Extracting smooth and transparent layers from a single image. In *CVPR*, pages 1–7, 2008. 2