Handling Outliers in Non-Blind Image Deconvolution

Sunghyun Cho[1] sodomau@postech.ac.kr

Jue Wang[2] juewang@adobe.com

Seungyong Lee[1,2] leesy@postech.ac.kr

[1]POSTECH [2]Adobe Systems

Abstract

Non-blind deconvolution is a key component in image deblurring systems. Previous deconvolution methods assume a linear blur model where the blurred image is generated by a linear convolution of the latent image and the blur kernel. This assumption often does not hold in practice due to various types of outliers in the imaging process. Without proper outlier handling, previous methods may generate results with severe ringing artifacts even when the kernel is estimated accurately. In this paper we analyze a few common types of outliers that cause previous methods to fail, such as pixel saturation and non-Gaussian noise. We propose a novel blur model that explicitly takes these outliers into account, and build a robust non-blind deconvolution method upon it, which can effectively reduce the visual artifacts caused by outliers. The effectiveness of our method is demonstrated by experimental results on both synthetic and real-world examples.

1. Introduction

Single image deblurring has gained considerable attention in recent years. Given a blurry input image b, existing deblurring approaches model the image degradation as:

$$b = k * l + n, \tag{1}$$

where k is the blur kernel, l refers to the underlying sharp latent image, n is noise, and $*$ is the convolution operator. The task for blind image deconvolution is to infer both k and l from a single input b, which is severely ill-posed. Blind deconvolution approaches intensively use non-blind deconvolution, the process of estimating the latent image l given the blurred image b and the estimated blur kernel k, while inferring k and for generating the final output l [7, 5, 16, 12, 4, 10]. This makes non-blind deconvolution a key component in the deblurring pipeline.

Various non-blind deconvolution approaches have been proposed in the literature, ranging from classic Wiener filter to modern optimization approaches with image priors. However, in practice these methods often produce severe ringing artifacts even when the blur kernel is known or well estimated (Fig. 1). We argue that this is mainly because the linear blur model in Eq. (1) does not consider non-linear outliers that often exist in real imaging process. For instance, one common outlier is saturated pixels. When we shoot a low lighting scene with a long exposure time, a few bright spots in the scene will be saturated (Fig. 1a). The intensities of these pixels can no longer be modeled using Eq. (1) since a non-linear clipping function has been applied at the corresponding sensor locations. Other types of outliers include dead pixels of sensors, hot pixels, non-linear in-camera processing, and so on. These outliers are common in real imaging systems, but are ignored in previous deconvolution methods.

We develop an algorithm that explicitly handles outliers in the deconvolution process. We first analyze how various types of outliers violate the linear blur assumption and consequently cause severe ringing artifacts to the result image. We then propose a new deconvolution method which contains an explicit component for outlier modeling. In our approach, we classify image pixels into two main categories: inlier pixels which satisfy the linear blur model and can be well-recovered using traditional deconvolution, and outlier pixels which cannot be explained by the linear model. We employ an Expectation-Maximization (EM) method to iteratively refine the outlier classification and the latent image.

To evaluate the effectiveness of the proposed method, we compare our approach with the state-of-the-art deconvolution methods on both synthetic and real-world examples. The results show that by explicitly detecting and handling outliers in the deconvolution process, severe ringing artifacts which are common to previous methods can be effectively reduced in our approach.

2. Related Work

Non-blind deconvolution has been extensively studied in image processing and computer vision fields. Classical methods include Wiener filtering, Kalman filtering, constrained least squares filtering and the Richardson-Lucy algorithm [14, 13]. We refer the readers to the comprehensive survey [1] for more details of these methods.

In order to suppress ringing artifacts and restore image features effectively, various image priors and regularization schemes have been proposed for deconvolution, such as total variation (TV) regularization [15], sparse image prior [11], and natural image statistics [16]. In previous work, these image priors are usually combined with an L^2-norm based data fidelity term, which is derived from a Gaussian noise model. This combination leads to high quality results when the observed image contains only a small amount of noise that can be well approximated by a Gaussian distribution. To handle impulsive noise such as salt-and-pepper noise, Bar et al. [2] proposed a data fidelity term based on an L^1-norm, which can be derived from the assumption of a Laplacian distribution for noise. However, since these priors and data fidelity terms are derived from specific noise models, they cannot effectively suppress artifacts caused by other types of noise and outliers. In contrast, our method restores image details and suppresses ringing artifacts better than these approaches, as we will demonstrate later.

For handling saturated pixels in non-blind deconvolution, Harmeling et al. [9] proposed a method that detects saturated pixels by thresholding input blurry images, and masks out them from the deconvolution process. While the method recovers saturated regions better than previous deconvolution methods, using a single threshold to detect saturated pixels in the input image is erroneous, as there is no guidance on how to find the optimal threshold value. In contrast, our method does not involve a threshold, and is more reliable and accurate as demonstrated later.

To suppress ringing artifacts, Yuan et al. [18] proposed a coarse-to-fine progressive deconvolution approach, where bilateral regularization is iteratively applied at each scale for restoring sharp edges while avoiding ringing. This method thus tries to *implicitly* handle outliers by directly suppressing artifacts. We show that ringing artifacts can be more efficiently removed by *explicitly* modeling the outliers in the deconvolution process. To the best of our knowledge, we are the first to systematically model outliers for non-blind deconvolution.

3. Outlier Analysis

In this section we analyze how various types of outliers violate the linear blur model and cause artifacts in previous approaches. All deconvolution results in this section were generated using the sparse prior based method proposed by Levin et al. [11].

Saturated/Clipped pixels As camera sensors have limited dynamic ranges, pixels receiving more photons than the maximum capacity will be saturated and the corresponding pixel intensities will be clipped. This is a common scenario when shooting a night image with long exposure, where the majority of the scene is dark but there are some bright lights. The non-linear clipping violates the linear blur model, leading to ringing artifacts around these spots. To demonstrate this, we synthetically blurred an HDR image using a known kernel, and then clipped pixel values that are larger than a threshold, as shown in Fig. 1a. Directly applying deconvolution produces severe ringing artifacts shown in Fig. 1b. Note that in this case the blur kernel is accurate and no noise is added. For comparison, we applied the same deconvolution method without clipping the input pixel values, which yields a high quality result in Fig. 1c. It clearly suggests that saturated pixels can be a main source of ringing artifacts. Besides saturation, parts of an image, such as shadow areas, could be underexposed and intensities in those regions could be clipped to black. Such clipping also breaks the linear blur model, and may cause ringing artifacts too.

Non-Gaussian noise An image can be degraded by noise from various sources, and they are typically non-Gaussian [2]. In addition, the camera sensor may contain defective pixels causing bright or dark spots in an image [6]. For low quality inputs, there may be complicated compression artifacts. Previous methods often assume a specific type of noise, thus is not robust enough to handle all cases. Fig. 1d shows a synthetic example where uniform noise was added to the input image. Severe artifacts can be seen in the deconvolution result (Fig. 1e), as the deconvolution method assumes a Gaussian noise model.

Nonlinear camera response curve Digital cameras all have built-in image processing units, and some of the processing steps are highly nonlinear. For instance, a camera usually applies a non-linear response curve to map the scene radiance to pixel intensity. This will violate the linear blur model, especially around high contrast edges where pixels on the two sides of an edge are separated far away on the response curve. Fig. 1f shows a synthetic example of applying a non-linear response curve to a blurred raw image, and Fig. 1g shows the deconvolution result, where artifacts are obvious around strong edges. Fig. 1h shows the deconvolution result without applying the response curve, which is artifact-free.

4. Deconvolution with Outlier Handling

Our goal is to develop a robust deconvolution algorithm which can perform reliably well when outliers present. Among different types of outliers, nonlinear in-camera processing can be avoided by using raw camera output, or reduced by first applying an inverse response curve obtained from camera calibration [8] to the input image. However, other outliers are extremely hard to remove using image processing techniques. Our key observation is that although these outliers vary in nature, their impacts to the deconvolution process are common: First, they cause the linear blur model to fail. Second, they cause the noise in Eq. (1) to be non-Gaussian. We thus propose a deconvolution algorithm

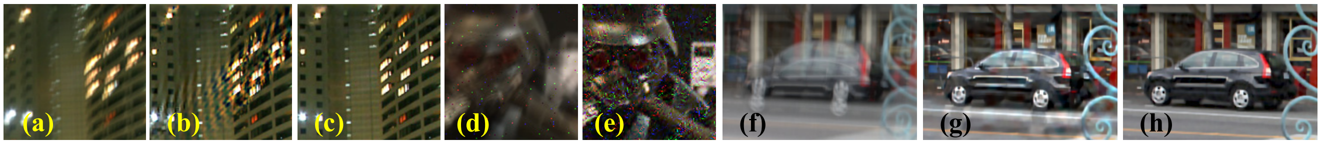

Figure 1. Illustrations of deconvolution outliers. (a) Input blurred HDR image with high intensity value clipping. (b) Deconvolution result of (a). (c) Deconvolution result of (a) without applying intensity clipping. (d) Input blurred image with added uniform noise. (e) Deconvolution result of (d). (f) Input blurred image with non-linear response curve applied. (g) Deconvolution result of (f). (h) Deconvolution result of (f) without applying the response curve.

that specifically handles these two violations. For simplicity we will derive our algorithm under the assumption that the blur kernel is spatially invariant. In practice we can easily extend it to handle non-uniform blur kernels.

The key idea of our approach is to classify observed pixel intensities into two categories: *inliers* whose formation satisfies Eq. (1); and *outliers* whose formation does not, which include clipped pixels and those from other sources. For classification, we introduce a binary map m such that $m_x = 1$ if the observed intensity b_x is an inlier, and $m_x = 0$ otherwise, where the subscript x is the pixel index. To find the most probable latent image l given the blurred image b with outliers and the blur kernel k, we exclude the outliers from the deconvolution process using the inlier map m. Since we do not know the true value of m, we propose an EM method which alternatingly computes the expectation of m and performs deconvolution using the expectation.

To establish a more accurate blur model including outliers, we assume that a noise-free blurred image is captured by sensors and then the captured intensities are clipped into the dynamic range of a camera. Noise and outliers are added to the clipped blurred image within the dynamic range. This model can be represented as:

$$b = c(k * l) + n, \tag{2}$$

where c is a clipping function. When u is within the dynamic range, $c(u) = u$; otherwise, $c(u)$ returns the maximum or minimum intensity of the dynamic range. We assume that the additive noise n is spatially independent, and it follows a Gaussian distribution only for inliers. For outliers, we assume that the observed intensity at an outlier pixel is completely independent of k and l, and may have an arbitrary value in the dynamic range. Recall that outliers can come from multiple sources that are hard to model accurately, as we discussed in Sec. 3. Therefore we assume a *uniform distribution* for outliers, without superimposing any strong priors. In the following, we formulate our objective function, and derive the detailed process for deconvolution using this blur and noise model.

Mathematically, finding the most probable latent image l can be formulated as a maximum a posteriori (MAP) estimation problem such that:

$$l_{\text{MAP}} = \operatorname*{argmax}_l p(l|k,b). \tag{3}$$

According to the Bayes' theorem:

$$\begin{aligned} l_{\text{MAP}} &= \operatorname*{argmax}_l p(b|k,l)p(l) \\ &= \operatorname*{argmax}_l \sum_{m\in\mathcal{M}} p(b,m|k,l)p(l) \\ &= \operatorname*{argmax}_l \sum_{m\in\mathcal{M}} p(b|m,k,l)p(m|k,l)p(l) \end{aligned} \tag{4}$$

where $\mathcal{M}$ is the space of all possible configurations of m. We define the latent image prior $p(l)$ as

$$p(l) = \exp(-\lambda\phi(l))/Z_p, \tag{5}$$

where Z_p is a normalization constant. Using the sparse prior [11], we set $\phi(l) = \sum_x \left\{|(\nabla^h l)_x|^\alpha + |(\nabla^v l)_x|^\alpha\right\}$, where ∇^h and ∇^v are differential operators along the x and y directions, respectively. We use $\alpha = 0.8$ in our system.

Since we assumed that noise is spatially independent, the likelihood $p(b|m,k,l) = \prod_x p(b_x|m,k,l)$. Then, based on our noise model, we define $p(b_x|m,k,l)$ as:

$$p(b_x|m,k,l) = \begin{cases} \mathcal{N}(b_x|f_x,\sigma) & \text{if } m_x = 1 \\ C & \text{if } m_x = 0 \end{cases} \tag{6}$$

where $f = k * l$, $\mathcal{N}$ is a Gaussian distribution, and σ is the standard deviation. C is a constant defined as the inverse of the width of the dynamic range in the input image.

For the classification prior $p(m|k,l)$, we assume that m is also spatially independent, i.e., $p(m|k,l) = \prod_x p(m_x|k,l) = \prod_x p(m_x|f_x)$. We then define $p(m_x|f_x)$ based on the value of f_x as:

$$p(m_x = 1|f_x) = \begin{cases} P_{\text{in}} & \text{if } f_x \in \text{DR} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where DR is the dynamic range, and $P_{\text{in}} \in [0,1]$ is the probability that b_x is an inlier. For example, by setting $P_{\text{in}} = 0.9$, we assume that 90% of non-clipped observed pixels b_x are inliers. According to our blur model in Eq. (2), when f_x is out of DR, the observed intensity b_x cannot be an inlier. It is either a clipped value or an outlier of another type, thus m_x should be always 0 in this case. Note that we do not limit the dynamic range of l_x in Eq. (2), so f_x can go beyond DR. In our implementation, we use the normalized range, DR $= [0,1]$.

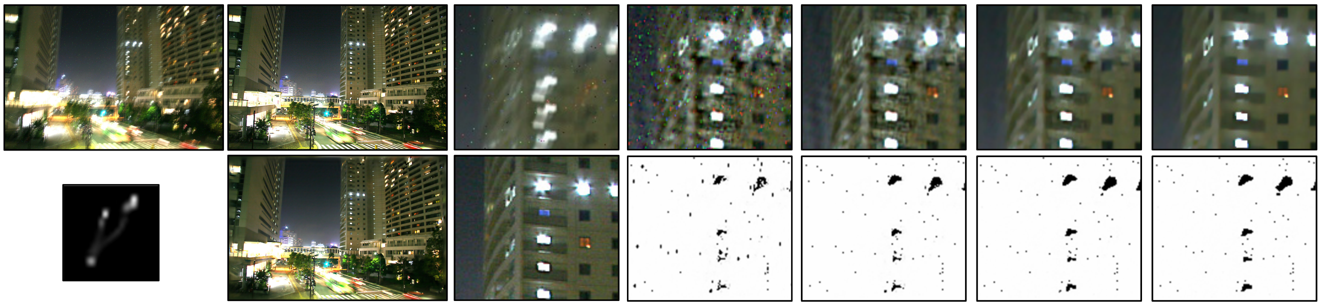

Figure 2. Deconvolution example of our method with intermediate results. The input image is synthetically blurred and high intensity values are clipped. Uniform noise is also added. The right four columns show magnified patches of intermediate estimates of l and the weights w^m computed for the red channel at EM iterations. Due to saturated pixels and noise, the estimate at the first iteration shows severe artifacts, but as iteration goes, artifacts are removed, and details are recovered.

Solving Eq. (4) is challenging, since the large number of possible configurations for m makes marginalizing $p(b, m|k, l)$ intractable. We thus adopt an EM method [3] to solve it. Instead of marginalizing likelihood $p(b, m|k, l)$ with respect to m, our EM method tries to find an optimal l, which maximizes the complete-data posterior $p(l|b, m, k) \propto p(b, m|k, l)p(l)$. As we do not know the true value of m, we cannot use the complete-data likelihood $p(b, m|k, l)$ when maximizing $p(l|b, m, k)$. Our EM method thus evaluates the expectation of $p(b, m|k, l)$ and uses it for finding the optimal l.

The expectation of $p(b, m|k, l)$ and the estimate of l are updated by performing the E and M steps alternatingly. Specifically, the E step finds the posterior distribution $p(m|b, k, l)$ of m using the current estimate l^{o} of l, and then evaluates the expectation $Q(l, l^{\mathrm{o}})$ of the complete-data log likelihood $p(b, m|k, l)$ under $p(m|b, k, l)$. Note that the evaluated $Q(l, l^{\mathrm{o}})$ is not a value, but a functional with respect to l whose parameters have been determined using $p(m|b, k, l)$. The subsequent M step revises the estimate of l by maximizing the complete-data log posterior, which is defined as $Q(l, l^{\mathrm{o}}) + \log p(l)$. In the following, we will derive the details of the two steps.

E step Using the current estimate l^{o} of l, the expectation $Q(l, l^{\mathrm{o}})$ is defined by:

$$\begin{aligned} Q(l, l^{\mathrm{o}}) &= E[\log p(b, m|k, l)] && (8) \\ &= E[\log p(b|m, k, l) + \log p(m|k, l)], && (9) \end{aligned}$$

where E is the expectation under $p(m|b, k, l^{\mathrm{o}})$. We put Eqs. (6) and (7) into Eq. (9), assuming if f_x^o is in or out of DR, so is f_x, respectively. Then, up to a constant, we have:

$$\begin{aligned} Q(l, l^{\mathrm{o}}) &= E\left[\sum_x m_x \log \mathcal{N}(b_x|f_x, \sigma)\right] && (10) \\ &= -\sum_x \frac{E[m_x]}{2\sigma^2}|b_x - f_x|^2 && (11) \end{aligned}$$

where $E[m_x]$ is given by:

$$E[m_x] = p(m_x = 1|b, k, l^{\mathrm{o}}). \tag{12}$$

By the Bayes' theorem, the posterior $p(m_x|b, k, l^{\mathrm{o}})$ is

$$p(m_x|b, k, l^{\mathrm{o}}) = \frac{p(b_x|m_x, k, l^{\mathrm{o}})p(m_x|k, l^{\mathrm{o}})}{p(b_x|k, l^o)}, \tag{13}$$

where

$$p(b_x|k, l^o) = \sum_{m_x=0}^{1} p(b_x|m_x, k, l^{\mathrm{o}})p(m_x|k, l^{\mathrm{o}}). \tag{14}$$

Again, by putting Eqs. (6) and (7) into Eq. (13), we have:

$$E[m_x] = \begin{cases} \dfrac{\mathcal{N}(b_x|f_x^{\mathrm{o}}, \sigma)P_{\text{in}}}{\mathcal{N}(b_x|f_x^{\mathrm{o}}, \sigma)P_{\text{in}} + CP_{\text{out}}} & \text{if } f_x^{\mathrm{o}} \in \text{DR} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where $f^{\mathrm{o}} = k * l^{\mathrm{o}}$ and $P_{\text{out}} = 1 - P_{\text{in}}$. As $E[m_x]$ completely determines $Q(l, l^{\mathrm{o}})$ as a functional of l, $E[m_x]$ is the only value that is actually computed in the E step.

In Eq. (6), $p(b_x|m_x = 0, k, l^{\mathrm{o}})$ always follows a uniform distribution. If f_x^o is out of DR, we could expect b_x to be close to one bound of DR, which can be better represented by a nonuniform distribution. However, in this case, $p(m_x = 1|k, l^{\mathrm{o}})$ becomes zero and $p(b_x|m_x = 0, k, l^{\mathrm{o}})$ has no effect on $E[m_x]$. Therefore, we use a uniform distribution in Eq. (6) for all cases of outlier pixels, including clipped pixels.

M step The M step finds the revised estimate l^{n} such that

$$l^{\mathrm{n}} = \underset{l}{\operatorname{argmax}}\{Q(l, l^{\mathrm{o}}) + \log p(l)\}, \tag{16}$$

which is equivalent to minimizing:

$$\sum_x w_x^m |b_x - (k * l)_x|^2 + \lambda\phi(l), \tag{17}$$

where $w_x^m = E[m_x]/2\sigma^2$. To minimize Eq. (17), we use the iteratively reweighted least squares (IRLS) method used in previous deconvolution approaches [11]. First, we introduce pixelwise weights w^h and w^v such that $w_x^h = |(\nabla^h l)|^{\alpha-2}$ and $w_x^v = |(\nabla^v l)_x|^{\alpha-2}$. Then, Eq. (17) can be approximated by:

$$\sum_x w_x^m |b_x - (k * l)_x|^2 + \lambda\psi(l), \tag{18}$$

where

$$\psi(l) \;=\; \sum_x \left\{ w_x^h |(\nabla^h l)_x|^2 + w_x^v |(\nabla^v l)_x|^2 \right\}. \tag{19}$$

For fixed w^h and w^v, Eq. (18) becomes a quadratic function with respect to l, which can be effectively minimized using the conjugate gradient (CG) method. We thus can minimize Eq. (17) by alternating between updating (w^h, w^v) and minimizing Eq. (18).

The above EM procedure is easy to understand if we consider the meaning and role of $E[m_x]$. If the observed intensity b_x is likely to be an inlier, $E[m_x]$ computed in the E step is close to one, and vise versa. These values are then used as pixel weights in the actual deconvolution process, which is performed in the M step. As a result, only inliers with large weights are used for deconvolution in the M step, while outliers with low weights are excluded. In Sec. 5, we provide more analysis on this outlier handling scheme.

Algorithm Algorithm 1 shows the overall process for our EM-based deconvolution algorithm. For the initial estimate l^{o}, we use the result of the deconvolution using a Gaussian prior, by setting all weights to one. Then, we perform the EM iterations. In the E step, we update weights w^m, w^h and w^v given the currently estimated l. In the M step, we minimize Eq. (18) using the updated weights. We typically used N_iters $= 15$ and $\sigma = 5/255$, and adjusted λ according to the amount of noise in the input image. We consistently used $P_{\text{in}} = 0.9$ for generating all results in the paper. To minimize Eq. (18), we run the CG method with 25 iterations. Fig. 2 shows an example of applying our method to a night city image, and its intermediate results for the latent image l with the associated weights w^m.

5. Analysis on Outlier Handling

In this section we provide more insightful analysis on how our method can recover outlier pixels. We also demonstrate why we need to model clipped pixels explicitly in the classification prior $p(m|k, l)$.

5.1. Recovering missing information

In a blurred image b, due to the scattering nature of blur, a pixel x contains partial information of the original intensities of neighboring pixels. If x is an inlier, $w_x^m = E[m_x]/2\sigma^2$ is non-zero. Then, the observed intensity b_x at x contributes to recovering pixel values around x by minimizing Eq. (18), while the original intensity l_x at x

Algorithm 1 EM Deconvolution for Outlier Handling

```
procedure DECONVOLUTION($b, k$)
    Let $w_x^m$, $w_x^h$ & $w_x^v \leftarrow 1$ for all $x$
    Set $l^{\text{o}}$ by minimizing Eq. (18)
    for iter $\leftarrow 1$, N_iters do
        E step updates $w^m$, $w^h$ and $w^v$ using $l^{\text{o}}$
        M step updates $l^{\text{n}}$ by minimizing Eq. (18)
        $l^{\text{o}} \leftarrow l^{\text{n}}$
    end for
    return $l^{\text{o}}$
end procedure
```

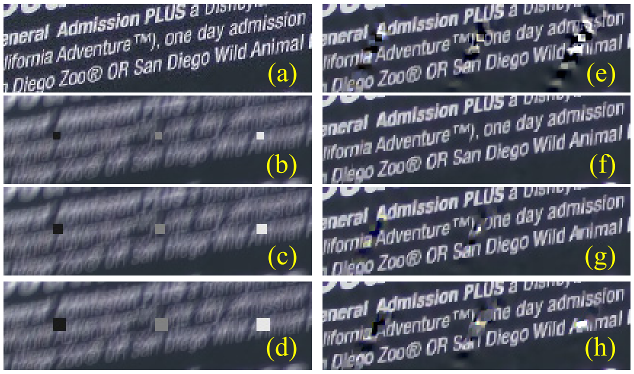

Figure 3. Recovering outlier pixels. (a) ground truth image. (b, c, d) blurred images with outliers of different sizes and intensities. (e) sparse deconvolution result of (b). (f, g, h) our deconvolution results of (b, c, d), respectively.

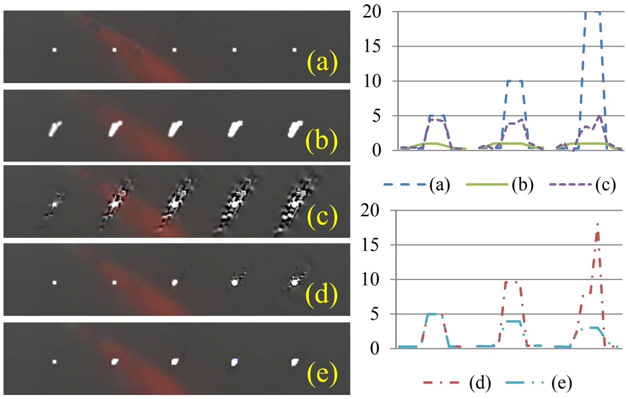

Figure 4. Recovering high intensity pixels. (a) ground truth image with pixels of different high intensities. From left to right, the original intensities of the white dots are 5, 10, 20, 40, and 80. (b) blurred image of (a), where high intensities are clipped. (c) sparse deconvolution result of (b). (d) our deconvolution result with only clipped pixel handling. (e) our result with both clipped pixel handling and outlier handling. Right: pixel intensities of each image along a scan line crossing the left three clipped pixels.

is recovered from its own and neighboring pixel information. If x is an outlier, l_x can still be reconstructed using the information in the neighboring pixels around x, although b_x does not contribute to recovering pixel values because w_x^m is close to zero. As a result, in our method outliers are not just smoothed out, their original intensities may be recovered actually if the neighboring pixels maintain enough amount of information about them.

To verify this observation, we conducted an experiment shown in Fig. 3. We synthetically blurred the original image with a known kernel and added outlier pixels of different intensities and sizes. The deconvolution results show that, although the blurred image is contaminated by outliers, underlying textures can be recovered well for small outliers. For large outliers, neighboring pixels have less information about the true intensities of the outlier pixels. These pixels cannot be accurately recovered in our method, and can only be smoothed out.

Clipped pixels can also be recovered in the same way. In Fig. 4, we synthetically added high intensity pixels with different intensity values into the original image. We then blurred the image with a known kernel, and clipped high intensity pixels. To study the effect of clipped pixel handling separately from non-Gaussian noise handling, we turn off the non-Gaussian noise handling by setting $P_{\text{in}} = 1$. The deconvolution result in Fig. 4d shows that added pixels can be better recovered when their original intensities are lower. As the original intensity goes higher, more neighboring pixels around the added pixel are saturated as well. After intensity clipping, the scattered portion of the original high intensity in the neighboring pixels is also lost, which makes recovering the original intensity to be impossible. Ringing artifacts are introduced in this case.

Can the deconvolution algorithm generate reasonable results even if original high intensity pixels cannot be recovered from their neighbors? The answer is yes. Since the pixels around the high intensity pixels also violate the linear blur model, they will be treated as outliers. By dealing with them properly in the EM algorithm, we can reduce the ringing artifacts, even though we cannot recover the true intensities. Fig. 4e shows that our full deconvolution method can successfully recover a good latent image from the input image.

5.2. Effects of modeling clipped pixels

Since clipped pixels also break the linear blur model, one may wonder if we can drop the dynamic range check in Eq. (7), and let the clipped pixels be handled together with other outliers. This leads to another EM procedure which is similar to the one derived in Sec. 4, where the M step remains the same, and in the E step, $E[m_x]$ is computed as

$$E[m_x] = \frac{\mathcal{N}(b_x|f_x^{\text{o}},\sigma)P_{\text{in}}}{\mathcal{N}(b_x|f_x^{\text{o}},\sigma)P_{\text{in}} + CP_{\text{out}}}. \qquad (20)$$

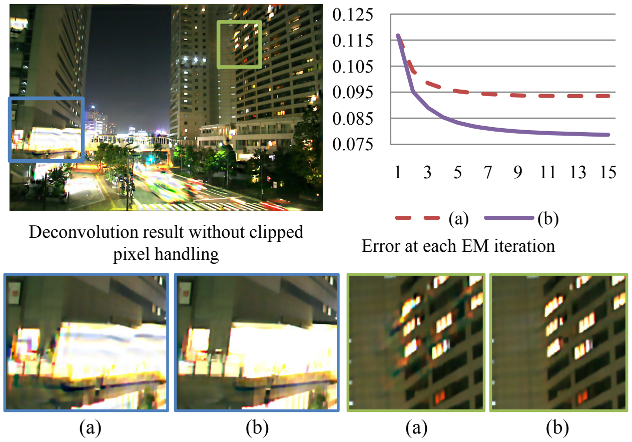

Figure 5. Deconvolution result of the blurred image in Fig. 2 obtained without modeling clipped pixels. Bottom row: magnified patches of (a) the deconvolution result without modeling clipped pixels and (b) the result in Fig. 2. Upper right: plots of root mean squared errors (RMSE) from the ground truth at EM iterations for the methods without and with modeling clipped pixels.

In Eq. (15), non-zero $E[m_x]$ is obtained only when $f_x^{\text{o}} \in$ DR. In contrast, in Eq. (20), $E[m_x]$ is always non-zero regardless of f_x^{o}, even though it can be small for a clipped pixel x. The difference looks small, but it has a significant impact on the performance of the algorithm. By completely excluding the clipped pixels from the optimization process, our method can avoid being distracted by the corrupted information from them, and converge to a better solution quickly.

Fig. 5 shows an example where we apply Eq. (20) in the E step. Due to the proposed outlier handling mechanism, the deconvolution result does not have significant artifacts overall, except some moderate ringing artifacts around saturated regions (Fig. 5a). However, using Eq. (15) in the E step still leads to a better result (Fig. 5b). This is consistent with the plots of error values in Fig. 5, which shows that the algorithm converges to a more accurate result just in a few iterations using Eq. (15). This leads to the conclusion that for images with saturated pixels, which is a common scenario in night photographing, explicitly modeling clipped pixels can help the proposed algorithm achieve better results more efficiently.

6. Results

We implemented our method in Matlab. Our testing environment is a PC running MS Windows 7 64bit version with Intel Core i7 CPU and 12GB RAM, but we did not use the multi-core facility. The average computation time is about six minutes for images of one mega-pixels. Our

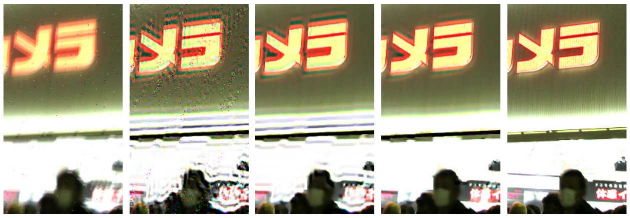

Figure 6. Deconvolution results of a synthetically blurred image. From left to right: blurred input with uniform noise and saturated pixels, sparse deconvolution, L^1-norm based deconvolution, our method, and the ground truth image.

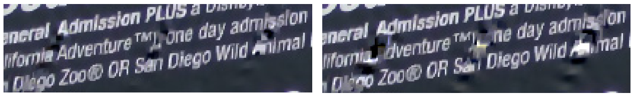

Figure 7. Deconvolution results of (b) and (c) of Fig. 3 using L^1-norm based deconvolution.

project webpage[1] provides high-resolution versions of the images shown in this section and additional examples, as well as Matlab source code of our method.

Fig. 6 shows a synthetic example. The input image was synthetically blurred and high intensity pixels were clipped. Uniform noise was also added. We deconvolved the blurred image using a sparse prior [11], L^1-norm based deconvolution [2], and our method. The L^1-norm based deconvolution method uses an L^1-norm data fidelity term, which is known to be robust to outliers, with a sparse image prior. For comparison, we implemented L^1-norm based deconvolution using an IRLS method. In Fig. 6 we can see the result of sparse deconvolution contains artifacts caused by both uniform noise and clipped pixels. The result of L^1-norm based deconvolution shows no artifacts from uniform noise, but still has artifacts around saturated pixels. In contrast, our result contains no artifacts for both uniform noise and saturated pixels. We also measured peak-signal-to-noise ratios (PSNR) of the results. From the blurred input to our result in Fig. 6 (left to right), PSNR values are 17.15, 16.46, 18.56, and 21.91. Our result achieves a much higher PSNR than the others.

While we added outliers with the size of one pixel to the blurred image in Fig. 6, outliers can be larger in practice. In Fig. 7, we applied L^1-norm based deconvolution to the blurred images in Fig. 3, to compare its performance on outlier handling with our method. The results show that our method suppresses large outliers better than L^1-norm based deconvolution. This is because our blur model better describes outliers than a Laplace distribution, which is the basis of the L^1-norm data fidelity term.

Fig. 8 shows a comparison between the deconvolution method of Yuan et al. [18] and ours. As our model explicitly handles outliers, our results have no visible artifacts around saturated pixels and other outliers. In contrast, the results of Yuan et al.'s method present noticeable artifacts, and less details are recovered in their results.

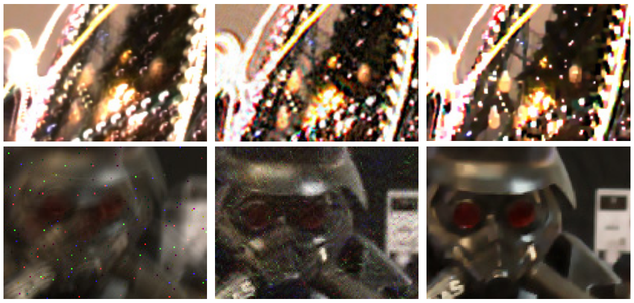

Figure 8. Comparison between Yuan et al.'s method [18] and ours. From left to right: Synthetically blurred inputs with uniform noise and saturated pixels, Yuan et al.'s method, and our method. Our results show less ringing artifacts than Yuan et al.'s results.

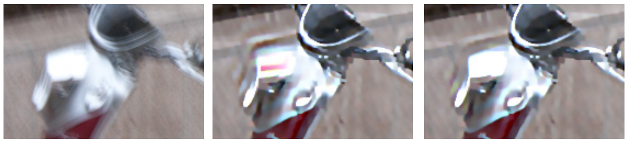

Figure 9. Comparison between Harmeling et al.'s method [9] and ours. From left to right: real blurred input with saturated pixels, Harmeling et al.'s method, and our method. Saturated pixels are better recovered in our result.

Fig. 9 shows a comparison between the deconvolution method of Harmeling et al. [9] and ours. Harmeling et al.'s method detects saturated pixels by thresholding input blurry images, which can be erroneous as there is no guidance on the optimal threshold value. In contrast, our method detects saturation on the latent image without involving a threshold, and better recovers saturated pixels.

Fig. 10 shows deconvolution results of real blurred images. To obtain blur kernels for real images, we used the blind deconvolution method of Cho and Lee [4]. Since outliers may degrade the quality of kernel estimation, we cropped rectangular regions without obvious outliers (e.g., saturated pixels) from input images, and estimated blur kernels using them. We then applied deconvolution methods to the input images using the estimated kernels. The results show that our method handles outliers most effectively. In summary, our experimental results show that our deconvolution method leads to significantly less artifacts, such as ringing around saturated pixels or high-frequency noise, than previous approaches.

7. Conclusion and Future Work

In this paper, we analyzed various types of outliers, which cause severe ringing artifacts in previous deconvolution methods. Based on the outlier analysis, we proposed a robust, EM-based non-blind deconvolution method, which explicitly detects outliers and properly handles them in the

[1] http://cg.postech.ac.kr/research/deconv_outliers

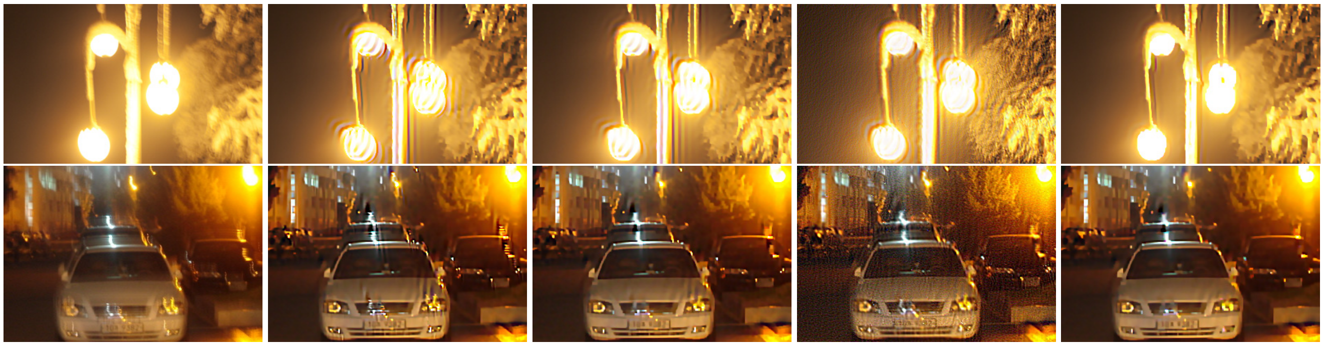

Figure 10. Deconvolution results of real blurred images. From left to right: blurred inputs, sparse deconvolution, L^1-norm based deconvolution, Yuan et al.'s method [18], and our method. Our results show less ringing artifacts than the others.

deconvolution process. We further provided detailed analysis on outlier pixel recovery and clipped pixel handling of our approach. We evaluated our method with both synthetic and real-world blur images, and compared the results with other state-of-the-art methods.

Previous studies have pointed out that real blurs are often non-uniform and spatially varying [12]. Due to the limited space, our method was derived under the assumption that the blur is spatially invariant. However, our method can be easily extended to handle non-uniform blurs. This can be achieved by replacing the convolution operation in our formulation with a non-uniform blurring operation, e.g., the blurring operation using a projective motion chain [17].

Despite the recent progress, blind deconvolution remains as a challenging problem, especially when the blurred image contains outliers. Our approach handles outliers only in the non-blind deconvolution step, not in the kernel estimation step. However, we have discovered that outliers have a significant impact on kernel estimation too. As future work we plan to incorporate outlier modeling to the kernel estimation step, in order to develop a blind deconvolution method that is robust enough against various outliers.

Acknowledgements We thank Lu Yuan for generating the deconvolution results of his method. We also thank Flickr user OiMax (Figs. 2 & 6) and Pusan National University (the bottom row of Fig. 10) for sharing their images. This work was supported by the Brain Korea 21 Project, the Industrial Strategic Technology Development Program of MKE/MCST/KEIT (KI001820, Development of Computational Photography Technologies for Image and Video Contents), and the Basic Science Research Program of MEST/NRF (2010-0019523).

References

[1] M. R. Banham and A. K. Katsaggelos. Digital image restoration. *Signal Processing Magazine, IEEE*, 14(2):24–41, Mar. 1997. 1

[2] L. Bar, N. Sochen, and N. Kiryati. Image deblurring in the presence of impulsive noise. *International Journal of Computer Vision*, 70(3):279–298, December 2006. 2, 7

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 4

[4] S. Cho and S. Lee. Fast motion deblurring. *ACM Trans. Graphics*, 28(5):145:1–145:8, 2009. 1, 7

[5] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *Proc. ICCV 2007*, pages 1–8, 2007. 1

[6] M. A. Covington. *Digital SLR Astrophotography*. Cambridge University Press, 2007. 2

[7] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graphics*, 25(3):787–794, 2006. 1

[8] M. Grossberg and S. Nayar. Modeling the Space of Camera Response Functions. *IEEE Trans. Pattern Analysis Machine Intelligence*, 26(10):1272–1282, Oct 2004. 2

[9] S. Harmeling, S. Sra, M. Hirsch, and B. Schölkopf. Multiframe blind deconvolution, super-resolution, and saturation correction via incremental EM. In *Proc. ICIP 2010*, pages 3313–3316, 2010. 2, 7

[10] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Trans. Graphics*, 29(3):30:1–30:9, 2010. 1

[11] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graphics*, 26(3):70:1–70:9, 2007. 2, 3, 5, 7

[12] A. Levin, Y. Weiss, F. Durand, and W. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proc. CVPR 2009*, pages 1–8, 2009. 1, 8

[13] L. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79(6):745–754, 1974. 1

[14] W. Richardson. Bayesian-based iterative method of image restoration. *J. Opt. Soc. Am.*, 62(1), 1972. 1

[15] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica. D*, 60:259–268, Nov. 1992. 2

[16] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graphics*, 27(3):73:1–73:10, 2008. 1, 2

[17] Y.-W. Tai, P. Tan, and M. S. Brown. Richardson-lucy deblurring for scenes under projective motion path. *IEEE Trans. Pattern Analysis Machine Intelligence*, Accepted. 8

[18] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Progressive interscale and intra-scale non-blind image deconvolution. *ACM Trans. Graphics*, 27(3):74:1–74:10, 2008. 2, 7, 8