Blind deconvolution of images with model discrepancies using maximum a posteriori estimation with heavy-tailed priors

Jan Kotera[a,b] and Filip Šroubek[a]

[a]Institute of Information Theory and Automation, Czech Academy of Sciences, Prague, Czech Republic;
[b]Charles University in Prague, Faculty of Mathematics and Physics, Czech Republic

ABSTRACT

Single image blind deconvolution aims to estimate the unknown blur from a single observed blurred image and recover the original sharp image. Such task is severely ill-posed and typical approaches involve some heuristic or other steps without clear mathematical explanation to arrive at an acceptable solution. We show that a straightforward maximum a posteriori estimation incorporating sparse priors and mechanism to deal with boundary artifacts, combined with an efficient numerical method can produce results which compete with or outperform much more complicated state-of-the-art methods. Our method is naturally extended to deal with overexposure in low-light photography, where linear blurring model is violated.

Keywords: image blind deconvolution, blur estimation, heavy-tailed priors, boundary artifacts, pixel saturation, alternating direction method of multipliers

1. INTRODUCTION

Single channel blind deconvolution amounts to estimating an image u from a single observed image g satisfying a convolutional degradation model

$$g = u * h + n, \tag{1}$$

where h, called point spread function (PSF), is unknown and n is random additive noise. Since we have only one observation (Single-Channel) and no knowledge of the PSF, the problem is extremely ill-posed. One way to tackle this problem is to assume a parametric model of the PSF and search in the space of parameters and not in the full space of PSFs. Chang *et al.* in[1] investigated zero patterns in the Fourier transform or in the cepstrum, and assumed only parametric motion or out-of-focus blurs. More low-level parametric methods for estimating general motion blurs were proposed in.[2–4] Unfortunately real PSFs seldom follow parametric models and this prevents the parametric methods from finding the exact solution.

There has been a considerable effort in the image processing community in the last three decades to find a reliable algorithm for SC blind deconvolution with general (non-parametric) PSFs. First algorithms appeared in telecommunication and signal processing in early 80's.[5] For a long time, the problem seemed too difficult to be solvable for arbitrary blur kernels. Proposed algorithms usually worked only for special cases, such as symmetric PSFs or astronomical images with uniform black background, see.[6]

Over the last few years, SC blind deconvolution based on the Bayesian paradigm experiences a renaissance. In probabilistic point of view, simultaneous recovery of u and h amounts to solving standard MAP (Maximum A Posteriori) estimation

$$P(u,h|g) \propto P(g|u,h)P(u,h) = P(g|u,h)P(u)P(h) \tag{2}$$

where $P(g|u,h) \propto \exp(-\frac{\gamma}{2}\|u*h-g\|^2)$ is the noise distribution (in this case assumed Gaussian) and $P(u)$, $P(h)$ are the prior distributions on the latent image and blur kernel, respectively. The key idea of new algorithms is to address the ill-posedness of blind deconvolution by characterizing the prior $P(u)$ using natural image statistics and by a better choice of estimators.

E-mail: kotera@utia.cas.cz. This work was supported by GA UK under grant 938213.

Digital Photography XI, edited by Nitin Sampat, Radka Tezaur, Dietmar Wüller, Proc. of SPIE-IS&T Electronic Imaging
SPIE Vol. 9404, 94040B · © 2015 SPIE-IS&T · CCC code: 0277-786X/15/$18 · doi: 10.1117/12.2077158

Various image priors were proposed in the literature.[7–14] Unfortunately, mainly due to the fact that blur reduces image variance, most of these priors still favor the blurred image over the sharp one. Levin *et al.* in[11, 15] claim that a proper estimator matters more than the shape of priors. They showed that marginalizing the posterior with respect to the latent image u leads to the correct solution of the PSF h. The marginalized probability $P(h|g)$ can be expressed in a closed form only for simple priors that are, e.g., Gaussian. Otherwise approximation methods such as variational Bayes[16] or the Laplace approximation[17] must be used. A significant breakthrough in this area started with the work of Molina *et al.*[7] and Fergus *et al.*,[8] who applied variational Bayes to approximate the posterior $P(u, h|g)$ by a simpler distribution $q(u, h) = q(u)q(h)$. This methodology was later improved in.[12] Other authors[9, 10, 18, 19] stick to the "good old" alternating MAP approach, but by using ad hoc steps, which often lack rigorous explanation, it converges to the correct solution.

Often silently neglected is the problem of boundary artifacts in blind deconvolution. The information about pixel intensities outside the image support is missing and this produces heavy artifacts (so-called ringing) in deconvolution if not properly treated. A method that reduces boundary artifacts by modifying boundary regions was proposed in.[20] Other techniques, such as,[21] try to solve the problem precisely by considering boundary regions separately from inner regions. Recently a method of masking has been proposed in[22] for deconvolution and extended to blind deconvolution in.[23] The masking method allows very fast implementation in FFT, which renders the method applicable to large scale problems.

The main contribution of our paper is to show that a simple alternating MAP approach without any ad hoc steps results in an efficient blind deconvolution algorithm that outperforms sophisticated state-of-the-art methods. We use image priors $P(u)$ that are more heavy-tailed than Laplace distribution and apply a method of augmented Lagrangian to tackle this non-convex optimization problem. In addition, masking that tackles boundary problems is included in a similar fashion as in.[23] We evaluate experimentally that masking can be used to increase robustness of blind deconvolution to model discrepancies such as overshoots. The resulting algorithm is also very fast and can handle large images and blurs.

In the next section we define the energy function of u and h that we want to minimize. Sec. 3 provides a detailed description of the optimization algorithm and the final experimental section illustrates algorithm's performance.

2. MATHEMATICAL MODEL

Let us assume that the variables in (1) are discrete quantities (vectors) with indexing denoted as u_i or $[u]_i$. Maximization of the posterior $P(u, h|g)$ in (2) is equivalent to minimization of its negative logarithm, also called energy. In pure MAP formulation with the assumption of i.i.d Gaussian noise, we get the following general form of energy to minimize

$$L(u, h) = -\log(P(u, h|g)) + const = \frac{\gamma}{2}\|u * h - g\|_2^2 + Q(u) + R(h) + const,$$

where $Q(u) = -\log P(u)$ and $R(h) = -\log P(h)$ are regularizers corresponding to prior distributions of u and h, respectively. A typical prior on u favors sparsity of coefficients of some linear image transform. The most popular choice is probably the l_1 norm of the image derivatives, either directionally separable $Q(u) = \sum_i |[D_x u]_i| + |[D_y u]_i|$ (this corresponds to the Laplace distribution of image partial derivatives) or isotropic (in terms of image gradient magnitude) $Q(u) = \sum_i \sqrt{[D_x u]_i^2 + [D_y u]_i^2}$, where D_x and D_y are partial derivative operators. The prior on h mainly forces positivity, optionally also promotes sparsity of PSF values, especially in case of motion blurs.

It has been reported (e.g.[11]) that the distribution of gradients of natural images is even more heavy-tailed than Laplace distribution, we therefore use a generalized version of $Q(u)$ defined as

$$Q(u) = \alpha_u \sum_i |[Du]_i|^p ,$$

where α_u is the regularizer weighting factor, corresponding to precision (inverse of variance) of the distribution of derivatives, and $D = [D_x^T, D_y^T]^T$ is the gradient operator. For the blur kernel we use Laplace distribution on the positive kernel values to force sparsity and zero on the negative values. This results in the following regularizer R

$$R(h) = \alpha_h \sum_i \Psi(h_i), \quad \Psi(h_i) = \begin{cases} h_i & h_i \geq 0 \\ +\infty & h_i < 0. \end{cases}$$

α_h is again the regularizer weight.

2.1 Image boundaries and model violation

Boundaries present a well-known problem in image deconvolution tasks. In case of nontrivial blur, the observed image g in the image formation model (1) contains information of a larger part of the photographed scene, denoted u. If we consider the dimensions of the observed image g as $m_1 \times m_2$ and the PSF size as $s_1 \times s_2$, then the number of pixels in u necessary to uniquely determine g by (1) is $n_1 \times n_2$ where $n_i = m_i + s_i - 1$. This is often referred to as "valid" convolution and it is arguably the only realistic model. Another possibility is to assume u with bounded support (black borders), or consider circular convolution, which however treats u as periodic. Neither of these options is therefore applicable to real life photography.

Let us denote $\hat{H}$ the $m_1 m_2 \times n_1 n_2$ matrix which performs valid convolution with h. Equation (1) then reads $g = \hat{H}u + n$. This presents little modification to the theory, but it is significant for the practical numerical solution. Image recovery typically requires repeatedly solving a linear system with matrix containing $\hat{H}^T \hat{H}$, which can be very time consuming for all but very small images. If instead we consider circular convolution, denoted by circulant matrix H, inverting $H^T H$ is very fast because H is diagonalized by 2D Fourier transform. Circular convolution, however, means that blurring "wraps" to the other side of the image on image boundary (i.e. u is considered periodic), which is clearly only a crude approximation of the true blurring process.

It has been almost a standard practice to sacrifice fidelity and use the circular formulation with H for practical reasons, which results in visible boundary ringing. This can be partly remedied e.g. by using `edgetaper` function in MATLAB. A method has been recently reported[22] which exploits the numerical properties of circular convolution and yet keeps the valid convolution in the model. The "trick" is in splitting the valid convolution into circular part and cropping part as in $\hat{H} = MH$, where H is $n_1 n_2 \times n_1 n_2$ circular convolution and M is $m_1 m_2 \times n_1 n_2$ cropping matrix with at most one "1" in each row, all other elements zero. Inverting the Hessian $H^T M^T M H$ is not any easier than inverting $\hat{H}^T \hat{H}$ but by utilizing variable splitting in the minimization, the action of the two operators is separated and they can be inverted individually. We present the details in the next section.

The operator also M gives us the possibility to ignore regions of the input image where the convolutional model (1) does not hold. This can be caused by several reasons in photography – moving objects, variable scene depth, or intensity clipping to name a few. We focus explicitly on intensity clipping, which occurs very often in photographs captured in low-light conditions requiring long exposures, which make the photographs susceptible to blurring. Too bright objects or too long exposure time result in intensities higher then the detection device can capture, these intensities are then clipped (set to maximum allowable value). This clipping, however, occurs *after* the blurring, which is a violation of our linear model (1).

To deal with this problem, we set to zero entries in M which correspond to supposedly clipped pixels in g, i.e. pixels with maximum intensity. These pixels are then effectively removed from the data term and their estimated value is governed by the image prior. We will show in the experimental section how this modification significantly helps the blur estimation in case of intensity clipping.

To summarize, we strive to solve the following problem

$$\min_{u,h} L(u,h), \quad L(u,h) = \frac{\gamma}{2}\|MHu - g\|^2 + \alpha_u \sum_i |[Du]_i|^p + \alpha_h \sum_i \Psi(h_i). \tag{3}$$

Algorithm 1 Alternating estimation of u and h

```
1: Set $h^0 = \delta$, and $j = 0$. Choose $\gamma_0$.
2: repeat
3:     Solve $u^{j+1} = \arg\min_u L_{\gamma_j}(u, h^j)$
4:     Solve $h^{j+1} = \arg\min_h L_{\gamma_j}(u^{j+1}, h)$
5:     Set $\gamma_{j+1}$ according to (7)
6: until stopping criterion is satisfied
7: return $u^j$, $h^j$
```

3. OPTIMIZATION ALGORITHM

In order to numerically find the solution u, h, we alternately minimize the functional L in (3) with respect to either u or h while keeping the other constant, as outlined by the algorithm 1. This allows for easier minimization of the joint data fitting term. To solve the resulting subproblems (lines 3, 4), we use the so-called alternating direction method of multipliers (ADMM). Let us briefly explain the method on a toy example, detailed application for our problem will follow.

Suppose we are solving the following problem

$$\min_u F(A_1 u) + G(A_2 u), \tag{4}$$

in which minimization of F and G terms separately is preferable to simultaneous minimization. ADMM consists of the following steps:

1. Make substitutions $v := A_1 u$, $w = A_2 u$, thus turning the unconstrained problem into equivalent constrained one. We get
$$\min_{u,v,w} F(v) + G(w) \quad \text{s.t. } v = A_1 u,\, w = A_2 u.$$

2. Use augmented Lagrangian method (ALM) (see e.g. [24, Chap. 17]) to solve the resulting constrained problem. ALM is basically a combination of the Lagrange method of multipliers for constrained problems and the penalty method. We add quadratic penalty term for each constraint to the Lagrangian of the constrained problem. In our case we get
$$\min_{u,v,w} F(v) + G(w) + \frac{\beta_1}{2}\|v - A_1 u - a_1\|_2^2 + \frac{\beta_2}{2}\|w - A_2 u - a_2\|_2^2,$$
where a_1, a_2 are proportional to the Lagrange multipliers of the $v = A_1 u$, $w = A_2 u$ constraints, respectively. By "completing the square" we can conveniently place each multiplier and the corresponding constraint into one quadratic term.

3. Solve the problem by alternately minimizing w.r.t u, v and w. The update equations for a_i in each iteration consist of a single gradient descent step, this is dictated by the ALM theory to ensure convergence, see [24, eq. (17.39)]. In our case we get
$$\begin{aligned} a_1^{j+1} &= a_1^j - v + A_1 u, \\ a_2^{j+1} &= a_2^j - w + A_2 u. \end{aligned} \tag{5}$$

If the original problem (4) has a solution, the sequence u^j of estimates from the last step converges to this solution provided some mild conditions are satisfied [25, Thm. 8]. Namely, F, G are closed, proper, convex functions, $\beta_i > 0$, and $[A_1^T, A_2^T]^T$ has full column rank. Application of this process to our problem is explained in detail in the following sections.

3.1 Minimization with respect to u

This section explains solution to the problem on line 3 of Alg. 1. We wish to solve

$$\min_u \frac{\gamma}{2}\|MHu - g\|^2 + \alpha_u \sum_i |[Du]_i|^p \,,$$

where H denotes a (fixed) circular convolution operator constructed from the h estimate from the previous iteration. According to ADMM we introduce substitutions $v = Du$, $w = Hu$, and get the following equivalent problem

$$\min_{u,v,w} \frac{\gamma}{2}\|Mw - g\|^2 + \alpha_u \sum_i |v_i|^p \quad \text{s.t. } v = Du,\, w = Hu.$$

Following the above outlined procedure we apply ALM by adding quadratic penalty terms for constrains. The final functional to minimize is then

$$L_u(u,v,w) = \frac{\gamma}{2}\|Mw - g\|^2 + \alpha_u \sum_i |v_i|^p + \frac{\beta_u}{2}\|v - Du - a_u\|^2 + \frac{\beta_w}{2}\|w - Hu - a_w\|^2,$$

where the new variables a_u, a_w are proportional to the estimates of the Lagrange multipliers of the corresponding constraints. Using ADMM allows us to minimize the data term $\|MHu - g\|^2$ and the regularizer $Q(u)$ separately since they now depend on different variables, which is significantly easier and computationally more efficient than simultaneous minimization.

We now proceed with the minimization of L_u via coordinate descent in the u, v, w “direction” alternately. That is, we compute derivative with respect to one variable while keeping others fixed, find its root and update that variable accordingly, then move on to the next variable and so on for sufficiently many iterations. The whole process is summarized in Alg. 2, let us review and explain the individual steps.

Algorithm 2 u estimation

```
 1: Set $v^0 := 0, w^0 := 0, a_u^0 := 0, a_w^0 := 0$, and $j := 0$
 2: repeat
 3:     Solve $\left(\gamma M^T M + \beta_w I\right) w^{j+1} = \gamma M^T g + \beta_w (Hu^j + a_w^j)$ for $w^{j+1}$
 4:     $v_i^{j+1} := \text{lookup}_{(p,\alpha_u,\beta_u)}([Du^j + a_u^j]_i), \quad \forall i$
 5:     Solve $(\beta_w H^T H + \beta_u D^T D) u^{j+1} = \beta_w H^T (w^{j+1} - a_w^j) + \beta_u D^T (v_u^{j+1} - a_u^j)$ for $u^{j+1}$
 6:     $a_u^{j+1} := a_u^j - v^{j+1} + Du^{j+1}$
 7:     $a_w^{j+1} := a_w^j - w^{j+1} + Hu^{j+1}$
 8:     $j := j + 1$
 9: until stopping criterion is satisfied
10: return $u^j$
```

After differentiating L_u w.r.t. w and setting the derivative to zero, we must solve the linear system on line 3 of Alg. 2 for w. Due to the structure of M, $M^T M$ is diagonal (containing 1s in positions corresponding to interior pixels for which the convolutional model holds and 0s for boundary pixels or pixels in which the model is violated) so exact inversion is straightforward.

Minimization of L_u w.r.t. v on line 4 is trickier. If we disregard terms not depending on v, we get $\min_v \alpha_u \sum_i |v_i|^p + \frac{\beta_u}{2}\|v - Du - a_u\|^2$, where both terms are summations of simpler terms over all image pixels. Minimization can be therefore carried out pixel by pixel independently. Let i be fixed index and let r_i denote $[Du + a_u]_i$, the problem of minimizing L_u w.r.t. v_i can be rewritten as

$$\min_{v_i} \alpha_u |v_i|^p + \frac{\beta_u}{2}\|v_i - r_i\|^2. \tag{6}$$

Solution to this problem for given r_i, i.e. the function $r_i \to \arg\min_{v_i} \alpha_u |v_i|^p + \frac{\beta_u}{2}\|v_i - r_i\|^2$, is generally called *proximal mapping* of the function $t \to \frac{\alpha_u}{\beta_u}|t|^p$. For some p, a closed form solution can be computed. After simple calculation it can be seen that for the common choice of $p = 1$, minimization of (6) results in soft thresholding, i.e. $v_i = \operatorname{sign}(r_i) \max\left(|r_i| - \frac{\alpha_u}{\beta_u}, 0\right)$. Similarly, for the binary penalty $p = 0$ we get hard thresholding with threshold $\sqrt{2\alpha/\beta}$. Closed form solutions exist for a few other special cases of $0 < p < 1$ (which result in low-order polynomial equation), but in general case we must solve the minimization numerically. Since the proximal mapping depends only on the parameters p, α_u, and β_u, which we keep constant during iterations, we can precalculate the mapping function beforehand using any of the 1D minimization methods and use the result in the form of a lookup table. Even if we were changing any of the critical parameters during optimization and the lookup table had to be periodically recalculated, the minimization is so fast that the performance drop would be almost negligible.

To minimize L_u w.r.t. u we again set the corresponding derivative to zero, which leaves us with the linear system on line 5. Because $u \to Hu$ represents circular convolution as discussed in section 2.1, the corresponding matrix H is block-circulant and consequently the whole matrix of the left-hand side is block-circulant, which means that it can be diagonalized by 2D Fourier transform. Thus, the solution u can be computed directly and only at the cost of Fourier transform.

It is worth noting that the application of ADMM allows us to split the difficult inversion of the "valid" convolution matrix $\hat{H}^T\hat{H} = H^T M^T M H$ into two much easier subproblems, each of which can be solved directly.

Update equations for a_u, a_w on lines 6 and 7 are in accordance with the ALM theory as presented in our toy example, eq. (5).

3.2 Minimization with respect to h

Minimization of L with respect to h, line 4 of Alg. 1, is done in similar fashion. The problem at hand is

$$\min_h \frac{\gamma}{2}\|MUh - g\|^2 + \alpha_h \sum_i \Psi(h_i).$$

To separate the minimization of data term and regularizer and to split convolution and cropping operator we introduce the substitutions $v_h = h$ and $w = Uh$, where U is a (fixed) circular convolution operator constructed from u estimate from the previous iteration. Applying ALM as in the previous case yields the functional

$$L_h(h, v_h, w) = \frac{\gamma}{2}\|Mw - g\|^2 + R(v_h) + \alpha_h \sum_i \Psi([v_h]_i) \\ + \frac{\beta_h}{2}\|v_h - h - a_h\|^2 + \frac{\beta_w}{2}\|w - Uh - a_w\|^2,$$

where a_h is again related to ALM and is proportional to the estimate of Lagrange multiplier of the prescribed constraint. This functional can be minimized by the alternating coordinate descent outlined in Alg. 3.

Algorithm 3 h estimation

```
1: Set $v_h^0 = 0, w^0 = 0, a_h^0 = 0, a_w^0 = 0$, and $j = 0$
2: repeat
3:     Solve $\left(\gamma M^T M + \beta_w I\right) w^{j+1} = \gamma M^T g + \beta_w (U h^j + a_w^j)$ for $w^{j+1}$
4:     $[v_h^{j+1}]_i = \max\left([h^j + a_h^j]_i - \frac{\alpha_h}{\beta_h}, 0\right), \quad \forall i$
5:     Solve $(\beta_w U^T U + \beta_h I) h^{j+1} = \beta_w U^T (w^{j+1} - a_w^j) + \beta_h (v_h^{j+1} - a_h^j)$ for $h^{j+1}$
6:     $a_h^{j+1} = a_h^j - v_h^{j+1} + h^{j+1}$
7:     $j := j + 1$
8: until stopping criterion is satisfied
9: return $h^j$
```

The linear system on line 3 is exactly the same as the system on line 3 of u estimation Algorithm 2, solvable by direct inversion of the diagonal matrix.

Minimization of L_h w.r.t. v_h can be done pixel-wise. Let $t_i = [v_h]_i$, $r_i = [h + a_h]_i$, then said minimization amounts to solving

$$\min_{t_i} \alpha_h \Psi(t_i) + \frac{\beta_h}{2}\|t_i - r_i\|^2, \quad \forall i.$$

The solution is proximal mapping of the function $t \to \left\{\frac{\alpha_h}{\beta_h}|t|^p \text{ for } t \geq 0, +\infty \text{ for } t < 0\right\}$ applied to $r_i = [h + a_h]_i$. It can be easily seen that the mapping is soft thresholding with $\frac{\alpha_h}{\beta_h}$ for positive values, zero for negative values, as specified on line 4.

The linear system on line 5, originating from differentiating L_h w.r.t. h, can be again diagonalized by 2D Fourier transform and solved directly.

3.3 Implementation details

To avoid getting trapped in a local minimum, we estimate the PSF in the multiscale fashion. The input image g is downsampled such that the corresponding PSF at this scale is small (typically 5×5), then we upsample such estimated PSF (roughly with factor 2) and use this as the initial point of the next level estimation. This procedure is repeated until the target PSF size is reached.

The no-blur solution is very much favored by blind deconvolution algorithms based on alternating MAP, especially in the first stages of estimation when $u = g$ and $h = \delta$ form an acceptable pair for the data term and priors based on image gradient statistics are too weak to overcome this initial barrier, as pointed out by.[11] The solution is to force creation of strong edges in the initial estimations of u by excessive sparse regularization, then $h = \delta$ will no longer fit the data term and the main features of h will emerge. By decreasing the pressure on regularization during later iterations the estimation of both the PSF and the image becomes progressively more accurate. This technique, somewhat reminiscent of simulated annealing, serves the same purpose as estimating u, h from a shock-filtered or otherwise enhanced input image (e.g.[19]) and fits the Bayesian paradigm better, because such method remains in full accordance with the MAP formulation in (2), while application of shock filtering doesn't.

To this end, we increase the value of the data term weight γ before every switch between h and u estimation, as indicated on line 5 of Algorithm 1. γ increases from the initial value γ_{min}, chosen deliberately small, to the final value γ_{max}, which corresponds to the true noise variance in the input image. We observe that it makes the estimation more stable if we let γ saturate on the γ_{max} value, instead of letting it grow exponentially, as is more common in the literature (e.g.[10]). We use the following update for γ

$$\gamma_j = \frac{\gamma_{max}}{\left(\frac{\gamma_{max}}{\gamma_{min}} - 1\right) \mathrm{e}^{-\kappa j} + 1}, \tag{7}$$

where κ is the rate of increase.

For our experiments we use the following parameters $p = 0.5$, $\gamma_{min} = 2$, $\gamma_{max} = 2^{13}$, $\kappa = 1.25$, $\alpha_u = 2$, $\beta_u = 2^4$, $\beta_w = 2^6$, $\alpha_h = 1$, $\beta_h = 2^{13}$. For the final nonblind deconvolution, we use $p = 1$.

4. EXPERIMENTAL RESULTS

We tested our algorithm on the dataset provided by Levin et al.[11] consisting of four grayscale images and eight PSFs of true motion blur, resulting in 32 test images. The images were taken with a hand-held camera and the PSFs were estimated as the blur of dots placed in the corners of the image region, please refer to[11] for detailed description of the acquisition process. Let us emphasize that the images are degraded by true motion blur (not synthetically) so the measured PSFs are only our best knowledge of the true motion blurs but are not in fact exact ground-truth data, although they will be referred to as such. Figure 1 shows the sharp photographed images and PSFs measured in 8 different experiments. We compare our method to the method of Xu and Jia,[19] which is arguably currently the best performing single-channel blind deconvolution method, and the method of Fergus et al.,[8] which frequently appears in comparisons of blind deconvolution methods. Please note that we used the results of[8] as presented in,[11] so we cannot guarantee that the computation methodology was the same. In our comparison, we focus on the accuracy assessment of the estimated PSF, which we measure by the MSE

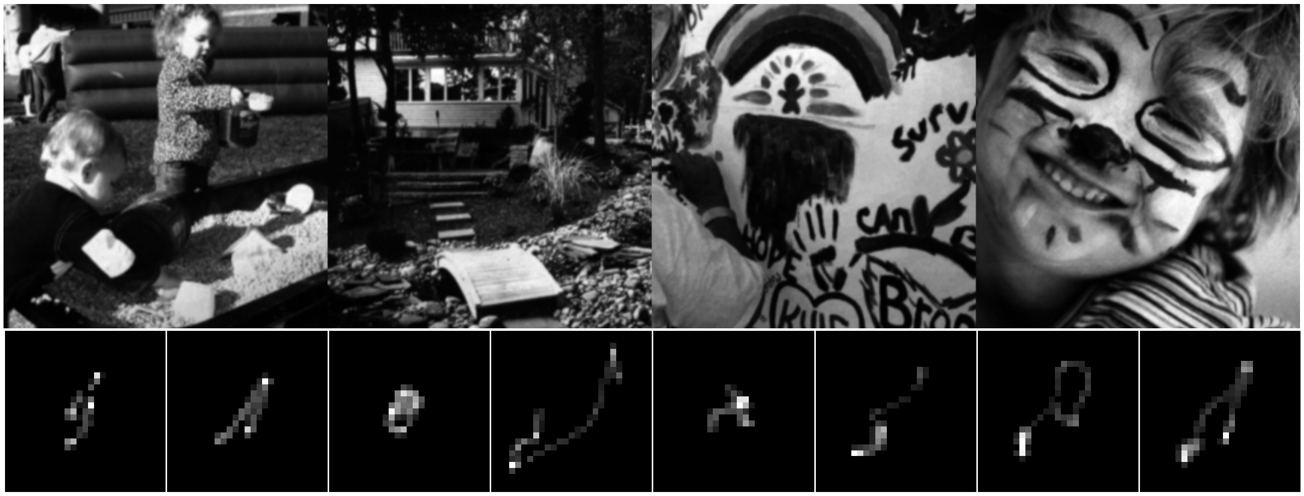


Figure 1. The dataset of.[11] First row contains sharp images, second row measured motion blur PSFs.

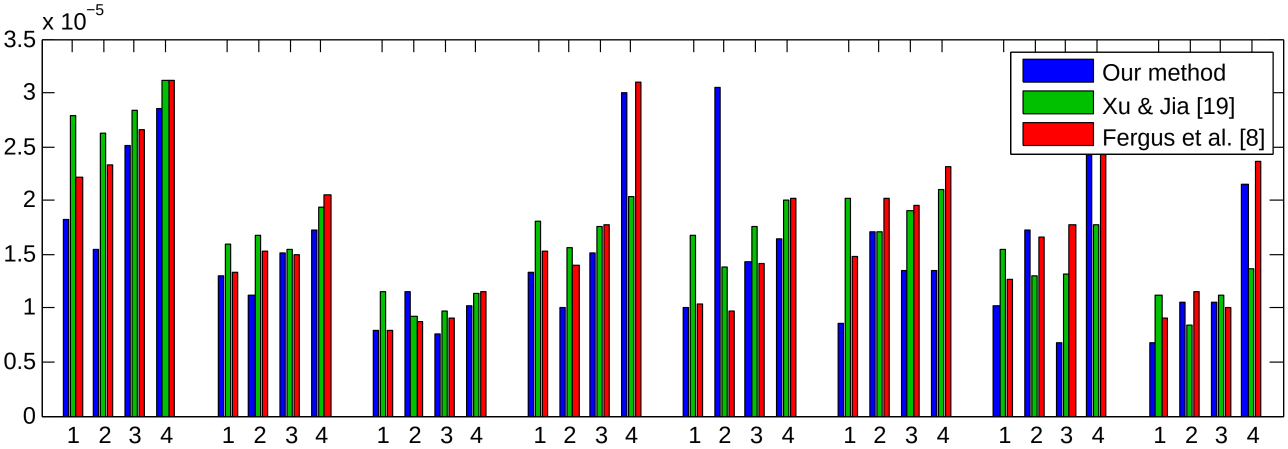


Figure 2. MSE of estimated kernels (low values mean better performance) in the 32 test examples, grouped by PSFs. Numbers on x-axis indicate image index.

of the (registered) estimated PSF to the ground truth. Figure 2 shows the result of kernel estimation measured as MSE from the ground truth kernel. We see that in significant majority of cases (24 of 32) our method is superior. Figure 3 shows the estimated PSFs for the first input image. The remaining 24 estimates look similar. All methods perform very well but it can be seen that our method produces more accurate results.

In our next experiment we focused on how intensity clipping in images with saturated pixels affects blur estimation. In this case we worked with synthetically blurred images to be able to quantitatively measure the relation between the degree of image degradation and accuracy of blur estimation. We first distributed a random pattern of increasing number of overshooted pixels in the input image, which we subsequently blurred and thresholded all the intensities above allowed maximum value (i.e. the same way it happens in normal photography). We then proceeded to estimate the PSF from such image. To decrease the effects of fluctuations in PSF estimation (depending on image used and the particular pattern of saturated dots), we used 4 images (of size 256×256) and 10 random dot patterns for each image, resulting in 40 test cases for each degree of image degradation, which we then averaged.

In order to take clipped pixels into account in the blur estimation, all we need to do is to alter the cropping/masking matrix M from eq. (3). Each row in matrix M corresponds to one pixel in the input image g. We set to zero all values in such rows, where the corresponding pixel in g has maximum intensity and is therefore most likely clipped. It may of course happen that such pixel is in fact perfectly valid, but the probability of

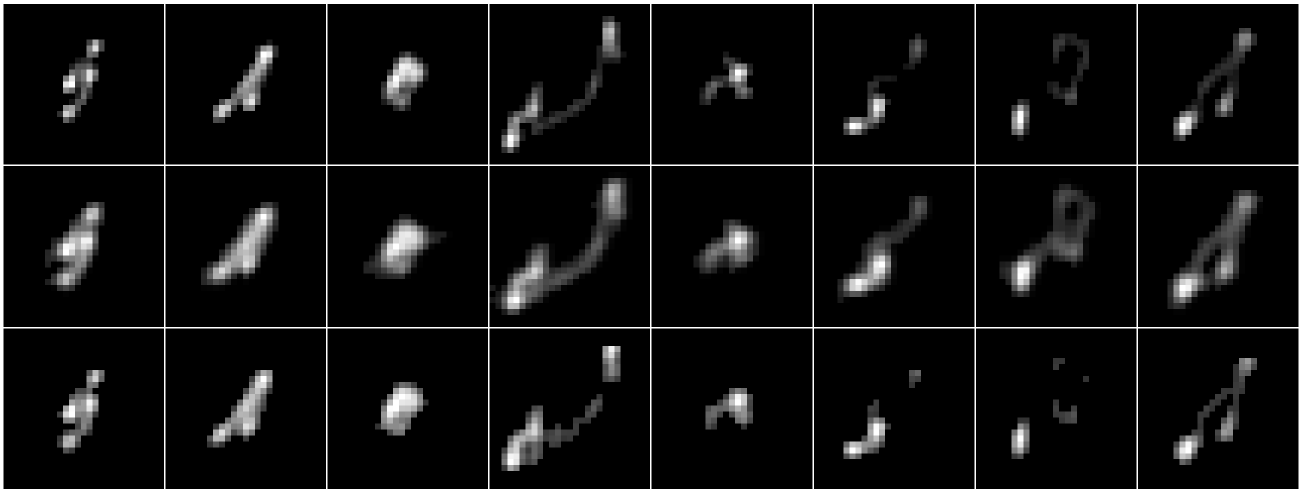
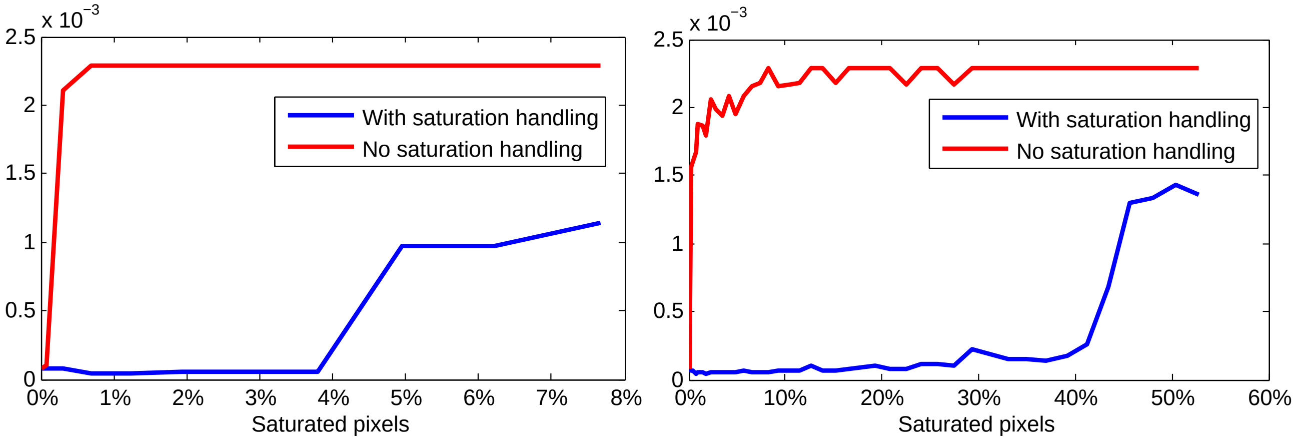


Figure 3. Estimated PSFs, image 1. Rows from top to bottom: our method, method of,[19] method of.[8] Compare with ground truth in Fig. 1.



(a)

(b)

Figure 4. MSE of estimated blur kernels depending on percentage of saturated pixels in the image, with two different layouts of saturated pixels. Left: Random dot–like pattern of saturated pixels spread throughout the entire image. Right: Single large region of saturated pixels.

that is rather small and even if we unjustly remove a small number of pixels from the data term, the estimation result is not harmed.

Figure 4a shows the resulting MSE of blur estimation depending on percentage of image pixels for which the convolutional model does not hold. Red curve represents the estimation error with no extra handling of saturated pixels. Blue curve is the result of the same method (incl. parameters) but with our extended model in which saturated pixels are excluded from the data term. We see that traditional blur estimation fails very quickly (values of MSE above $2 \cdot 10^{-3}$ indicate complete fail) as the number of saturated pixels increases, while our extended method is much more robust. The fraction of corrupted pixels for which our method starts to fail (around 5%) is seemingly too small, but because the corrupted pixels are somewhat regularly spread throughout the entire image, such image is already visually perceived as significantly degraded, as can be seen in Fig. 5.

In our next experiment with intensity clipping, we repeated the same process of pixel saturation, image blurring and intensity clipping but instead of a randomized pattern of saturated dots spread in the entire image, we placed a single large region of saturated pixels in the image. As in the previous experiment, we repeated this process with 4 images and 10 different placements of the saturated region and averaged the results. Figure 4b shows the MSE of estimated PSFs. It is apparent that while standard deconvolution with no extra care for saturation fails almost immediately, our method can cope with significant degree of model violation. It is interesting, yet quite understandable, that small corruptions spread in the entire image have far more damaging effect on blur estimation than single large corruption (see the respective x-axis limits in Fig. 4a and 4b).

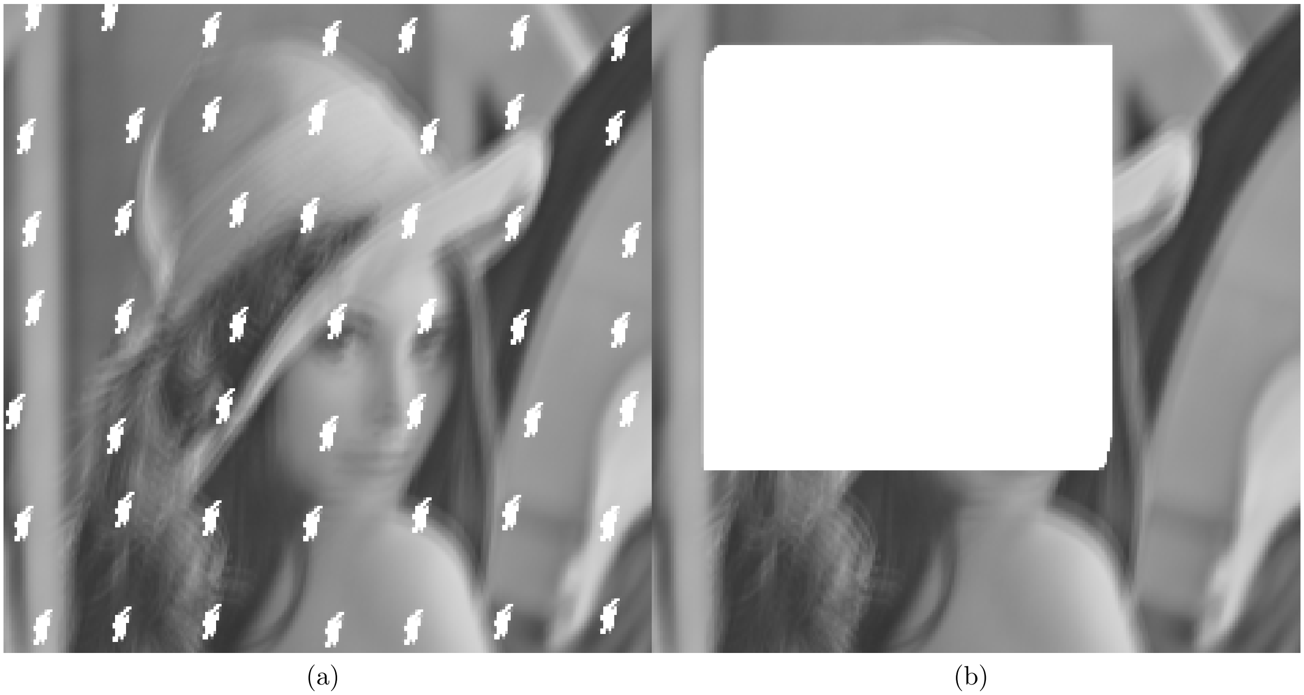


Figure 5. Example input images for our experiments with intensity clipping, for which the blur estimation is still successful. Left: Random dot–like pattern corresponding to 4% of saturated pixels. Right: Single large region corresponding to 41% of saturated pixels.

Figure 5 shows input images, for which our blur estimation method with saturation handling was still successful. The images are damaged to the similar degree in terms of achievable blur estimate accuracy, yet quantitatively the right image contains an order of magnitude more pixels which violate the blurring model than the left image. This is understandable, because convolution spreads information in the image and evenly distributed error thus affects much more pixels.

Our last experiment in Figure 6 shows the deconvolution result of genuinely motion-blurred photo captured by hand-held camera, the improvement in quality is evident.

5. CONCLUSION

We proposed a blind image blur estimation method based on pure Bayesian MAP estimation with no other tricks or heuristics involved. Our method uses heavy–tailed derivative prior and correct boundary handling and automatically handles saturated pixels, which occur quite often in low-light photography. Because of fast matrix inversion in the Fourier domain and utilization of fast-converging augmented Lagrangian method, our method can handle large images and blurs. We showed that our method compares favorably to current state-of-the-art methods for blur estimation. Next, we have experimentally verified that our handling of saturated pixels vastly improves blur estimation from images with overshoots.

In our ongoing work, we want to naturally extend our method so that it automatically determines and exploits regions of the input image which are most suitable for correct blur estimation.

REFERENCES

[1] Chang, M., Tekalp, A., and Erdem, A., "Blur identification using the bispectrum," *IEEE Trans. Signal Processing* **39**, 2323–2325 (Oct. 1991).

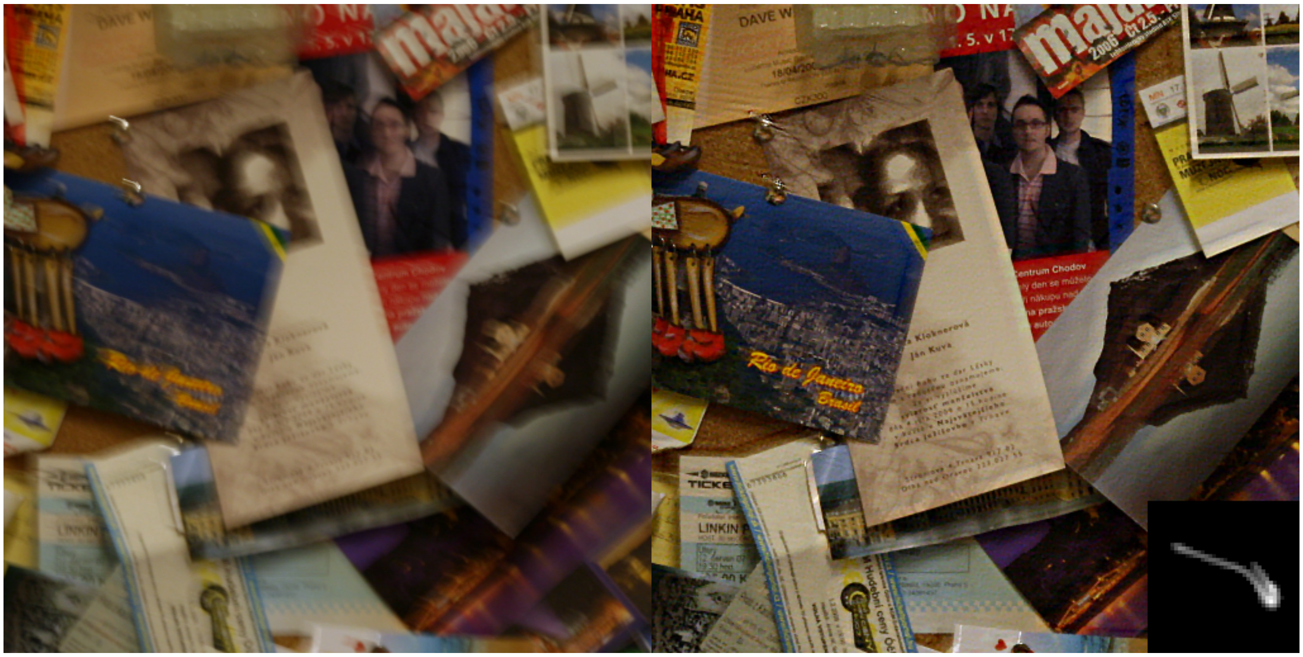


Figure 6. Deconvolution of true motion blur. Left: Captured image. Right: Deconvolved result (estimated PSF superimposed).

[2] Shan, Q., Xiong, W., and Jia, J., "Rotational motion deblurring of a rigid object from a single image," in [*Proc. IEEE 11th International Conference on Computer Vision ICCV 2007*], 1–8 (14–21 Oct. 2007).

[3] Stern, A. and Kopeika, N., "Analytical method to calculate optical transfer functions for image motion and vibrations using moments," *J. Opt. Soc. Am. A* **14**, 388–396 (Feb. 1997).

[4] Yitzhaky, Y. and Kopeika, N., "Identification of blur parameters from motion blurred images," *Graphical Models and Image Processing* **59**, 310–320 (Sept. 1997).

[5] Godard, D., "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Transactions on Communications* **28**(11), 1867–1875 (1980).

[6] Chan, T. and Wong, C., "Total variation blind deconvolution," *IEEE Trans. Image Processing* **7**, 370–375 (Mar. 1998).

[7] Molina, R., Mateos, J., and Katsaggelos, A. K., "Blind deconvolution using a variational approach to parameter, image, and blur estimation," *IEEE Transactions on Image Processing* **15**, 3715–3727 (Dec. 2006).

[8] Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., and Freeman, W. T., "Removing camera shake from a single photograph," in [*SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*], 787–794, ACM, New York, NY, USA (2006).

[9] Jia, J., "Single image motion deblurring using transparency," in [*Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '07*], 1–8 (17–22 June 2007).

[10] Shan, Q., Jia, J., and Agarwala, A., "High-quality motion deblurring from a single image," in [*SIGGRAPH '08: ACM SIGGRAPH 2008 papers*], 1–10, ACM, New York, NY, USA (2008).

[11] Levin, A., Weiss, Y., Durand, F., and Freeman, W., "Understanding and evaluating blind deconvolution algorithms," in [*Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '09*], 1964–1971 (2009).

[12] Tzikas, D., Likas, A., and Galatsanos, N. P., "Variational bayesian sparse kernel-based blind image deconvolution with student's-t priors.," *IEEE Transactions on Image Processing* **18**(4), 753–764 (2009).

[13] Almeida, M. and Almeida, L., "Blind and semi-blind deblurring of natural images," *Image Processing, IEEE Transactions on* **19**, 36 –52 (Jan. 2010).

[14] Krishnan, D., Tay, T., and Fergus, R., "Blind deconvolution using a normalized sparsity measure.," in [*CVPR*], 233–240, IEEE (2011).

[15] Levin, A., Weiss, Y., Durand, F., and Freeman, W. T., "Understanding blind deconvolution algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(12), 2354–2367 (2011).

[16] Miskin, J. and MacKay, D. J. C., "Ensemble learning for blind image separation and deconvolution.," in [*Advances in Independent Component Analysis*], Girolani, M., ed., 123–142, Springer-Verlag (2000).

[17] Galatsanos, N. P., Mesarovic, V. Z., Molina, R., and Katsaggelos, A. K., "Hierarchical Bayesian image restoration from partially known blurs," *IEEE Transactions on Image Processing* **9**(10), 1784–1797 (2000).

[18] Joshi, N., Szeliski, R., and Kriegman, D. J., "PSF estimation using sharp edge prediction," in [*Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008*], 1–8 (23–28 June 2008).

[19] Xu, L. and Jia, J., "Two-phase kernel estimation for robust motion deblurring," in [*Proceedings of the 11th European conference on Computer vision: Part I*], *ECCV'10*, 157–170, Springer-Verlag, Berlin, Heidelberg (2010).

[20] Liu, R. and Jia, J., "Reducing boundary artifacts in image deconvolution," in [*Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*], 505508, IEEE (2008).

[21] Šorel, M., "Removing boundary artifacts for real-time iterated shrinkage deconvolution," *IEEE Transactions on Image Processing* **21**, 2329–2334 (Apr 2012).

[22] Matakos, A., Ramani, S., and Fessler, J., "Accelerated edge-preserving image restoration without boundary artifacts," *Image Processing, IEEE Transactions on* **22**, 2019–2029 (May 2013).

[23] Almeida, M. and Figueiredo, M., "Deconvolving images with unknown boundaries using the alternating direction method of multipliers," *Image Processing, IEEE Transactions on* **22**, 3074–3086 (Aug 2013).

[24] Nocedal, J. and Wright, S., [*Numerical Optimization*], Springer Series in Operations Research, Springer (2006).

[25] Eckstein, J. and Bertsekas, D. P., "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.* **55**, 293–318 (June 1992).