Efficient Marginal Likelihood Optimization in Blind Deconvolution - Supplementary File

Anat Levin[1], Yair Weiss[2], Fredo Durand[3], William T. Freeman[3]
[1]Weizmann Institute of Science, [2]Hebrew University, [3]MIT CSAIL

Abstract

In blind deconvolution one aims to estimate from an input blurred image y a sharp image x and an unknown blur kernel k. Recent research shows that a key to success is to consider the overall shape of the posterior distribution $p(x,k|y)$ and not only its mode. This leads to a distinction between MAP$_{x,k}$ strategies which estimate the mode pair x,k and often lead to undesired results, and MAP$_k$ strategies which select the best k while marginalizing over all possible x images.

The MAP$_k$ principle is significantly more robust than the MAP$_{x,k}$ one, yet, it involves a challenging marginalization over latent images. As a result, MAP$_k$ techniques are considered complicated, and have not been widely exploited. This paper derives a simple approximated MAP$_k$ algorithm which involves only a modest modification of common MAP$_{x,k}$ algorithms. We show that MAP$_k$ can, in fact, be optimized easily, with no additional computational complexity.

1. Introduction

Blind deblurring is the problem of recovering a sharp version of a blurred input image when the blur parameters are unknown. Under certain motion types, a blurred input y can be modeled as convolution of a latent sharp image x with a blur kernel k

$$y = k \otimes x \tag{1}$$

where both x and k are unknown. Since there is an infinite set of pairs (x,k) that can explain an input image y, additional assumptions are required. The common approach is to utilize prior knowledge about the statistics of natural images, such as their sparse derivative distribution [6, 12, 20, 2, 4, 8, 7, 21, 3, 23]. However, the prior itself is usually not enough, and the estimation strategy should be chosen with caution.

The direct approach is to look for a MAP$_{x,k}$ estimate, that is, a pair $(\hat{x},\hat{k})$ with maximal a posteriori probability

$$(\hat{x},\hat{k}) = \arg\max \log p(x,k|y). \tag{2}$$

The MAP$_{x,k}$ pair should minimize the convolution error, and have sparse derivatives. However, as shown by Levin *et al*. [15], the total contrast of all derivatives in a blurred image is usually lower than in a sharp one. As a result, the MAP$_{x,k}$ score tends to favor the no-blur explanation, for which k is a delta kernel and x is the input blurred image y. The MAP$_{x,k}$ score does favor sharp signals at the vicinity of step edges, and thus steering it towards the sharp solution is usually sensitive to a careful detection of step edges and the boosting of their contribution.

While a simultaneous MAP estimation of both image and kernel is ill-posed, estimating the kernel alone is better conditioned because the number of parameters to estimate is small relative to the number of image pixels measured [15]. This leads to MAP$_k$ estimation:

$$\hat{k} = \arg\max p(k|y) = \arg\max \int p(x,k|y)dx. \tag{3}$$

The challenge of the MAP$_k$ score is that computing $p(k|y)$ in Eq. (3) involves a computationally intractable marginalization over all possible x explanations. The best practical MAP$_k$ algorithm is that of Fergus *et al*. [6], but this algorithm is sometimes viewed as challenging to implement. In general, despite the superior robustness of the MAP$_k$ estimation principle, only a few recent approaches to blind deconvolution have taken this direction [6, 22, 18], whereas many research attempts are devoted to the MAP$_{x,k}$ approach [20, 2, 4, 8, 7, 21, 3, 23].

The main contribution of this paper is to show that an approximation to MAP$_k$ can, in fact, be optimized easily using a simple modification to MAP$_{x,k}$ algorithms. Similar to most MAP$_{x,k}$ approaches, we alternate between solving for the kernel and solving for the image. The critical difference is that our kernel update system accounts for the covariance around the current latent image estimate, and not only for the central x estimate itself. Furthermore, an efficient approximation to this covariance can be computed with no extra computational complexity. We derive this simple algorithm by casting the MAP$_k$ problem in the Expectation-Minimization framework where the latent variable is the sharp image x.

We build on the algorithm of Fergus *et al*. [6], but provide a significantly simpler derivation. As a result we shed new light on the success of this algorithm and lead to improved performance.

To isolate the effect of the various algorithmic components, we compare experimentally multiple algorithmic versions. In particular, we show that the use of independent x and y derivative images, which was originally thought of

as an approximation to the correct use of a real derivative field, significantly improves performance. To encourage follow up research, we include our `matlab` implementation.

2. MAP$_k$ blind deconvolution

In blind deconvolution, one observes a blurred image y which is the convolution of a latent sharp image x with a latent blur kernel k, corrupted by measurement noise n:

$$y = k \otimes x + n \tag{4}$$

We denote the number of unknowns in x, k by N, M respectively, where typically $M \ll N$. Fergus *et al*. [6], formulate the problem in derivative space, and consider:

$$f_h \otimes y = k \otimes (f_h \otimes x) + n_h, \quad f_v \otimes y = k \otimes (f_v \otimes x) + n_v. \tag{5}$$

with $\{f_h, f_v\} = \{[-1, 1], [-1, 1]^T\}$. In their formulation, the "blurred input" is taken as $y = [f_h \otimes y; f_v \otimes y]$, and one solves for the derivative image $x = [f_h \otimes x; f_v \otimes x]$, without enforcing $\{f_h \otimes x, f_v \otimes x\}$ to integrate into a single image x. While ignoring integrability neglects an important constraint on the problem, we show that the derivative representation significantly improves the results in practice.

Our goal is to estimate x and k from the blurred input y. Since there are many pairs x, k which can explain the y observation, one should utilize some prior knowledge. A common natural image prior is to assume that the image derivatives are sparse. In this article we express the sparse prior as a mixture of J Gaussians (MOG):

$$p(x) = \Pi_i \Pi_\gamma \rho(f_{i,\gamma}(x)) \tag{6}$$

$$\rho(f_{i,\gamma}(x)) = \sum_j \frac{\pi_j}{\sqrt{2\pi}\sigma_j} e^{-\frac{1}{2\sigma_j^2}\|f_{i,\gamma}(x)\|^2} \tag{7}$$

where $f_{i,\gamma}(x)$ denotes the output of $f_\gamma \otimes x$ at the i'th pixel. In the image space formulation (Eq. (4)), $\{f_\gamma\}_{\gamma=1}^{\Gamma}$ are a set of derivative filters. In the derivative space formulation (Eq. (5)), $\{f_\gamma\}$ consists of the delta filter.

Most blind deconvolution algorithms use a sparsity prior on the kernel, and in practice our implementation employs a weak sparsity prior as well. However, the contribution of this term is usually small and for the simplicity of the derivation, we consider here a uniform prior on k and only enforce all entries of k to be non negative.

Assuming an i.i.d. Gaussian imaging noise with variance η^2, we can write

$$p(y|x,k) = \frac{1}{\left(\sqrt{2\pi}\eta\right)^N} e^{-\frac{\|k\otimes x - y\|^2}{2\eta^2}} \tag{8}$$

where N is the number of image pixels.

Combining Eqs. (6)–(8) we express

$$p(y, x, k) = p(y, x|k)p(k) = p(y|x, k)p(x)p(k)$$

Thus,

$$-\log p(y, x|k) = \frac{\|k \otimes x - y\|^2}{2\eta^2} - \sum_{i,\gamma} \log \rho(f_{i,\gamma}(x)) + c \tag{9}$$

where c denotes a constant[1], and $p(k)$ is assumed uniform and ignored.

The straightforward approach to blind deconvolution is to search for the MAP$_{x,k}$ solution:

$$(\hat{x}, \hat{k}) = \arg\max p(x, k|y) = \arg\max p(x, y, k) \tag{10}$$

However, as analyzed by Levin *et al*. [15], for priors of the form of Eq. (6), MAP$_{x,k}$ does not provide the expected answer and favors the no blur explanation. Instead, they suggest that since the kernel size is significantly smaller than the image size, a MAP estimation of the kernel alone is well conditioned. Thus, one should look for a MAP$_k$ estimate, marginalizing over all latent images:

$$\begin{aligned} \hat{k} &= \arg\max p(k|y) = \arg\max p(y|k) \\ p(y|k) &= \int p(x, y|k)dx. \end{aligned} \tag{11}$$

However, computing the integral of Eq. (11) is not trivial, and the remainder of this paper discusses approximation strategies.

2.1. EM optimization

To optimize the MAP$_k$ score, we consider an Expectation-Maximization framework which treats the latent image as a hidden variable and marginalizes over it. In a nutshell, this algorithm alternates between two main steps. In the E-step one solves a non-blind deconvolution problem and estimates the mean image given the current kernel, with the covariance around it. In the M-step one solves for the best kernel given the image. However, it accounts for the covariance around the image estimate and not only for the mean image estimate itself. Accounting for the covariance is the crucial difference distinguishing the EM MAP$_k$ approach from the MAP$_{x,k}$ approach. Formally, the algorithm is defined as follows:

1. E-step: Set $q(x) = p(x|y, k)$, and compute μ, C, the mean and covariance of $q(x)$, which are the mean image given a kernel and the covariance around it.

2. M-step: Find k minimizing

$$E_q\left[\|k \otimes x - y\|^2\right]. \tag{12}$$

   As explained below, since Eq. (12) integrates a quadratic term, the mean and covariance computed in the E-step are the sufficient statistics of $q(x)$ required for that minimization.

The standard EM derivation shows that if the E-step is exact, every step of this algorithm improves $\log p(y|k)$ [9]. The M-step minimization can be done easily, by solving a quadratic programming problem. This requires knowledge of the mean and covariance of q alone and not the full distribution.

[1] Through this paper, we overload the variable c to denote any additive constant independent of the variables of current interest.

Claim 1 *Eq. (12) is minimized by the solution to the quadratic programming problem*

$$\min_k \ \frac{1}{2}k^T \bar{A}_k k - \bar{b}_k^T k, \qquad s.t.\ k \geq 0 \tag{13}$$

where

$$\bar{A}_k(i_1,i_2) = \sum_i \mu(i+i_1)\mu(i+i_2) + C(i+i_1, i+i_2) \tag{14}$$
$$\bar{b}_k(i_1) = \sum_i \mu(i+i_1)y(i). \tag{15}$$

Proof: For a fixed x, the convolution error is quadratic in k and therefore can be written as

$$\|k \otimes x - y\|^2 = k^T A_k k - b_k^T k \tag{16}$$

If k is an $m \times m$ kernel and $M = m^2$, A_k is an $M \times M$ matrix representing the covariance of all $m \times m$ windows in x, and b_k the correlation with y:

$$A_k(i_1,i_2) = \sum_i x(i+i_1)x(i+i_2), \quad b_k(i_1) = \sum_i x(i+i_1)y(i) \tag{17}$$

where i sums over all image pixels, and i_1, i_2 are kernel indexes (in practice these are 2D indexes but we use the 1D vectorized version of the image and kernel). Averaging Eq. (17) over x values coming from the distribution $q(x)$ provides Eqs. (14) and (15). Therefore, minimizing Eq. (12) with respect to k is equivalent to minimizing Eq. (13). □

EM MAP$_k$ v.s. MAP$_{x,k}$: MAP$_{x,k}$ algorithms usually alternate between two main steps: 1) set k constant and solve for the best x (a non-blind deconvolution problem), and 2) set x constant and solve for the best k. The EM algorithm is not more complicated: finding the mean image in the E-step is equivalent to solving for x given k. In the M-step one solves for k, where the only difference is that solving for k in Eq. (13) takes into account not only the best x, but also the covariance around it. However, this small covariance term has a crucial effect on the results. Deleting the covariance term from Eq. (14) will move us from the desired MAP$_k$ result to the problematic MAP$_{x,k}$ one. We show that an approximated covariance can be computed efficiently.

2.1.1 The E-step

For general sparse priors, computing the mean and covariance of the distribution q is hard, and below we discuss our approximation strategy. For simplicity, we start with the case of a Gaussian prior on x. For a Gaussian prior, the covariance can be computed in closed form, resulting in the Gaussian blind deconvolution algorithm of [16].

E-step under a Gaussian prior: A Gaussian prior on x can be expressed using Eq. (7) with a single mixture component. $p(y, x|k)$ is then Gaussian as well, and Eq. (9) reads as:

$$\begin{aligned} -\log p(y,x|k) &= \frac{\|k \otimes x - y\|^2}{2\eta^2} + \sum_{i,\gamma} \frac{\|f_{i,\gamma}(x)\|^2}{2\sigma^2} + c \\ &= \frac{1}{2}x^T A_x x - b_x^T x + c \end{aligned} \tag{18}$$

where c denotes an additive constant and:

$$A_x = \tfrac{1}{\eta^2}T_k^T T_k + \tfrac{1}{\sigma^2}\textstyle\sum_\gamma T_{f_\gamma}^T T_{f_\gamma} \tag{19}$$
$$b_x = \tfrac{1}{\eta^2}T_k^T y \tag{20}$$

where T_ϕ denotes a Toeplitz (convolution) matrix with the filter ϕ. The conditional distribution $p(x|y,k)$ is also Gaussian, and its mean and covariance can be shown to be:

$$C = A_x^{-1} \qquad \mu = C b_x. \tag{21}$$

This implies that μ is the solution of the linear system $A_x \mu = b_x$, which is essentially a non-blind deconvolution problem: find an image μ such that its convolution with k approximates y, plus a regularization term on the derivatives. The deconvolution system can be solved efficiently in the frequency domain. We show in Sec. 3 that this simple Gaussian prior already provides good results, but sparse priors can further improve performance.

Approximate E-step using sampling: Unfortunately, there is no closed-form formula for the mean and covariance under a general sparse prior. One approach is to approximate these using samples. We tried the MOG sampling algorithm of Levi and Weiss [11, 19]. However, this sampling algorithm is quite slow. A better option discussed in the next section, is to consider variational free-energy approximations.

2.2. Variational free energy strategies

Since for a sparse prior the mean and covariance cannot be computed in closed form, we approximate the conditional distribution with a simpler one using variational optimization. The major algorithmic steps are summarized in Algorithm 1. In practice, this algorithm is very simple to implement and involves steps which are anyway computed by MAP$_{x,k}$ algorithms. Given k it solves a non-blind deconvolution problem, at which a mean latent image estimate μ is computed using iterative reweighted least squares [13, 14]. In each iteration, one finds μ by solving an $N \times N$ linear system $A_x \mu = b_x$. This system seeks μ minimizing the convolution error plus a weighted regularization term on the derivatives (compare Eq. (19) with Eq. (26)). The weights are selected to provide a quadratic upper bound on the MOG negative log likelihood based on the previous μ solution. This iterative reweighted least squares algorithm is a standard strategy for finding x in a MAP$_{x,k}$ approach. The covariance approximation uses the weighted deconvolution system A_x which was already computed anyhow. A

full covariance would be the $N \times N$ inverse

$$C = A_x^{-1}. \tag{22}$$

However, for efficiency, we show that a diagonal approximation is sufficient. This diagonal approximation can be computed easily in $O(N)$, by inverting the diagonal elements of A_x alone

$$C(i,i) = \frac{1}{A_x(i,i)}. \tag{23}$$

Given μ, C, one employs the M-step described in the previous section, and solves for the kernel as a quadratic programming problem. This is again a standard step in MAP$_{x,k}$ algorithms with the important difference that one accounts for the covariance and not only the single x solution. However, including the covariance can be done at no extra computational complexity. We usually iterate steps 1&2 (solving for x) of Algorithm 1 three times before proceeding to step 3 (solving for k).

For completeness, we provide below a formal derivation of the variational free-energy algorithm. Similar derivations can be found in [17, 1]. The reader who is interested in experimental evaluation can directly read Sec. 3.

2.2.1 Hidden mixture component variables

Before introducing the variational framework, we rewrite the MOG prior of Eq. (7) with a slight change. We associate with each derivative a hidden variable $h_{i,\gamma}$ indicating the mixture component from which it arises. $h_{i,\gamma}$ can take each of J discrete values $j \in \{1, \ldots, J\}$. Then

$$p(f_{i,\gamma}(x)|h_{i,\gamma}) = \sum_j \frac{h_{i,\gamma,j}}{\sqrt{2\pi}\sigma_j} e^{-\frac{1}{2\sigma_j^2}\|f_{i,\gamma}(x)\|^2} \tag{31}$$

where $h_{i,\gamma,j}$ is a short notation for $\delta(h_{i,\gamma} - j)$. The prior on the hidden variables is the mixture component prior

$$p(h_{i,\gamma,j}) = \pi_j. \tag{32}$$

Therefore

$$\begin{aligned} p(f_{i,\gamma}(x)) &= \sum_j p(h_{i,\gamma,j}) p(f_{i,\gamma}(x)|h_{i,\gamma,j}) \\ &= \sum_j \frac{\pi_j}{\sqrt{2\pi}\sigma_j} e^{-\frac{1}{2\sigma_j^2}\|f_{i,\gamma}(x)\|^2} \end{aligned} \tag{33}$$

which is exactly the original prior definition in Eq. (7).

The main advantage in introducing the hidden variables is that given their values, things become Gaussian. For example, since $h_{i,\gamma,j}$ are binary, the log of Eq. (31) involves no exponents:

$$\log p(f_{i,\gamma}(x)|h_{i,\gamma}) = \sum_j h_{i,\gamma,j} \left(-\frac{\|f_{i,\gamma}(x)\|^2}{2\sigma_j^2} - \log(\sqrt{2\pi}\sigma_j) \right) \tag{34}$$

Algorithm 1 Blind deconvolution using free-energy

Iterate:

1. Update weights:

$$w_{i,\gamma,j_0} = \frac{\frac{\pi_{j_0}}{\sigma_{j_0}} e^{-\frac{E[\|f_{i,\gamma}(x)\|^2]}{2\sigma_{j_0}^2}}}{\sum_j \frac{\pi_j}{\sigma_j} e^{-\frac{E[\|f_{i,\gamma}(x)\|^2]}{2\sigma_j^2}}} \tag{24}$$

   with $E[\|f_{i,\gamma}(x)\|^2]$ given by μ, C.
   Set W_γ to be a diagonal matrix with:

$$W_\gamma(i,i) = \sum_j \frac{w_{i,\gamma,j}}{\sigma_j^2}. \tag{25}$$

2. Update x: set

$$A_x = \frac{1}{\eta^2} T_k^T T_k + \sum_\gamma T_{f_\gamma}^T W_\gamma T_{f_\gamma} \tag{26}$$

$$b_x = \frac{1}{\eta^2} T_k^T y \tag{27}$$

   solve: $A_x \mu = b_x$.
   set diagonal covariance: $C(i,i) = \frac{1}{A_x(i,i)}$.

3. Update k: set

$$\bar{A}_k(i_1,i_2) = \sum_i \mu(i+i_1)\mu(i+i_2) + C(i+i_1, i+i_2). \tag{28}$$

$$\bar{b}_k(i_1) = \sum_i \mu(i+i_1) y(i). \tag{29}$$

   solve the quadratic program

$$min_k \ \frac{1}{2} k^T \bar{A}_k k + \bar{b}_k^T k \qquad s.t. \ k \geq 0 \tag{30}$$

Similarly, with h included, the joint distribution of Eq. (9) simplifies to:

$$\begin{aligned} -\log p(y,x,h|k) &= \frac{\|k \otimes x - y\|^2}{2\eta^2} \\ &+ \sum_{i,\gamma,j} h_{i,\gamma,j} \left(\frac{\|f_{i,\gamma}(x)\|^2}{2\sigma_j^2} + \frac{1}{2}\log(\sigma_j^2) - \log(\pi_j) \right) + c \end{aligned} \tag{35}$$

2.2.2 The free energy

The idea behind the variational framework is to search for a distribution $q(x)$ approximating $p(x|y,k)$. While $p(x|y,k)$ cannot be computed in closed form, the trick is to select $q(x)$ from some simpler parametric family, which allows for tractable computation. In our case we choose q to be a distribution on both x and h, of the form

$$q(x,h) = q(x) \prod_{i,\gamma} q(h_{i,\gamma}) \tag{36}$$

$q(x)$ is chosen to be a Gaussian distribution, characterized by a mean μ and covariance C. $q(h_{i,\gamma})$ is just a J-dimensional vector whose elements sum to 1 (to be a valid

distribution), the j'th element of this vector is $p(h_{i,\gamma} = j)$. To fully express $q(h)$ we need to define a separate J-dimensional vector for each image pixel, resulting in a table of $N \times \gamma \times J$ elements.

The variational optimization then alternates between two main steps which approximate the E and M steps. In the first step, we hold k constant, find a distribution $q(x)q(h)$ (within the simpler parametric family) which best approximates $p(x|y,k)$, and compute its mean and covariance. The second step is equivalent to the M-step: find the best k with respect to the distribution q (Eq. (12)).

More precisely, we attempt to minimize the free energy:

$$\begin{aligned} F(q) = &- \int q(x,h) \log p(y,x,h|k) dh dx \\ &+ \int q(x,h) \log q(x,h) dh dx \end{aligned} \tag{37}$$

We note that since

$$\log p(y,x,h|k) = \log p(x,h|y,k) + \log p(y|k), \tag{38}$$

we can write the free energy as

$$\begin{aligned} F(q) = &- \int q(x,h) \log p(x,h|y,k) dh dx \\ &- \log p(y|k) \int q(x,h) dh dx \\ &+ \int q(x,h) \log q(x,h) dh dx \\ = &\; D_{KL}\left(q(x,h) || p(x,h|y,k)\right) - \log p(y|k) \end{aligned} \tag{39}$$

That is, the free energy is the KL-divergence between $q(x,h)$ and the correct conditional $p(x,h|y,k)$, minus $\log p(y|k)$. Since the KL-divergence is non-negative, minimizing the free energy minimizes an upper bound on the term $-\log p(y|k)$ we wish to minimize. If the family of q distributions includes $p(x,h|y,k)$ such as in the Gaussian case, and k is fixed, the best q in the family is exactly $p(x,h|y,k)$. If the q family is not expressive enough, the best approximation should be chosen.

To minimize the free energy we use an alternate optimization over the parameters $k, \mu, C, q(h_{i,\gamma})$. In each step it selects the optimal value for one of the parameters while holding the others fixed. The update equations are derived below.

2.2.3 Update equations

To derive the update equations, let us substitute Eqs. (35) and (36) in Eq. (37) and express the blind deconvolution free energy explicitly:

$$\begin{aligned} &F(q) = \\ &\int q(x) \left(\frac{\|k \otimes x - y\|^2}{2\eta^2} + \sum_{i,\gamma,j} q(h_{i,\gamma,j}) \frac{\|f_{i,\gamma}(x)\|^2}{2\sigma_j^2} \right) dx \\ &+ \sum_{i,\gamma,j} q(h_{i,\gamma,j}) \left(\tfrac{1}{2} \log(\sigma_j^2) - \log(\pi_j) + \log(q(h_{i,\gamma,j})) \right) \\ &- \tfrac{1}{2} \log |C| + c. \end{aligned} \tag{40}$$

We now attempt to minimize Eq. (40) with respect to each of its variables while fixing the others.

Updating $q(h_{i,\gamma})$: Fixing μ, C, k, for each i, γ we can isolate from Eq. (40) the terms which involve $h_{i,\gamma}$:

$$\sum_j q(h_{i,\gamma,j}) \left(\frac{E[\|f_{i,\gamma}(x)\|^2]}{2\sigma_j^2} + \frac{1}{2} \log(\sigma_j^2) - \log(\pi_j) + \log(q(h_{i,\gamma,j})) \right) \tag{41}$$

Where $E[\|f_{i,\gamma}(x)\|^2] = \int q(x) \|f_{i,\gamma}(x)\|^2 dx$, is the expected derivative magnitude according to the current q distribution, which can be easily computed using the mean and covariance μ, C, e.g. if f_γ is a delta filter, $E[\|f_{i,\gamma}(x)\|^2] = \mu(i)^2 + C(i,i)$.

$q(h_{i,\gamma})$ should be a unit sum J-dimensional vector. By writing the Lagrangian of the problem, one can show that Eq. (41) is minimized by

$$q(h_{i,\gamma,j_0}) = \frac{\pi_{j_0}}{\sigma_{j_0}} e^{-\frac{E[\|f_{i,\gamma}(x)\|^2]}{2\sigma_{j_0}^2}} \Big/ \sum_j \frac{\pi_j}{\sigma_j} e^{-\frac{E[\|f_{i,\gamma}(x)\|^2]}{2\sigma_j^2}} \tag{42}$$

Updating μ: We hold $k, q(h_{i,\gamma})$ fixed and isolate from Eq. (40) the terms which involve x. We can write:

$$F(q) = \int q(x) \left(\frac{1}{2} x^T A_x x - b_x^T x \right) dx - \frac{1}{2} \log |C| + c \tag{43}$$

with A_x, b_x defined in Eq. (26). Since $q(x)$ is Gaussian, the integral of Eq. (43) can be computed easily:

$$F(q) = \frac{1}{2} \mu^T A_x \mu - b_x^T \mu + \frac{1}{2} Tr(A_x C) - \frac{1}{2} \log |C| + c \tag{44}$$

Since Eq. (44) is quadratic in μ, it is minimized by the solution to the linear system:

$$A_x \mu = b_x. \tag{45}$$

Note that iterating Eqs. (42) and (45) is essentially an iterative reweighted least squares non-blind deconvolution [13, 14]. In Eq. (45) we solve a weighted non-blind deconvolution- find an image μ, such that its convolution with k approximates y, plus a regularization term on the derivatives. The weights on the derivatives are updated in every iteration by Eq. (42).

For the specific case of a Gaussian prior the, filter weights are uniform and one can solve for μ efficiently in the frequency basis. Otherwise, we would like to employ a fast numerical solver, and our implementation uses the conjugate gradient algorithm. One can also consider the fast solver of [10], but we found that for this application, conjugate gradient converges faster. Another solver discussed below is the simple Gauss-Seidel solver, which is employed by the classical mean-field approach [6, 17].

Updating C: The following claim derives a formula for the best update of C, by differentiating Eq. (44) with respect to C.

Claim 2 *The covariance matrix minimizing the free energy of Eq. (44) is $C = A_x^{-1}$, for A_x defined in Eq. (26).*

Proof: Fixing $k, \mu, q(h_{i,\gamma})$, the free energy of Eq. (44) can be written as:

$$F(q) = \frac{1}{2} Tr(A_x C) - 0.5 \log |C| + c. \quad (46)$$

Since log det is a convex function (see e.g. [5]), Eq. (46) has a global minimum and it is enough to show that at $C = A_x^{-1}$, the derivative of Eq. (46) with respect to each of the entries of C is zero.

We recall that for every square matrix B

$$\frac{\log |B|}{\partial B(i_1, i_2)} = B^{-1}(i_1, i_2). \quad (47)$$

Thus, differentiating Eq. (46) at $C = A_x^{-1}$ provides

$$\frac{F(q, y)}{\partial C(i_1, i_2)}\Big|_{C=A_x^{-1}} = A_x(i_1, i_2) - A_x(i_1, i_2) = 0. \quad (48)$$

□

Covariance approximations: The drawback of the above approach is that to compute C one needs to invert an $N \times N$ matrix. For large images, this is computationally intractable. To simplify computation, one can search for a C matrix with a simpler parametric form. The simplest choice would be a zero covariance, but ignoring the variance around μ completely leads to the undesirable $\text{MAP}_{x,k}$ solution. A more reasonable alternative we derive below is to constrain C to be diagonal. While not derived here, one could consider several other simplified covariance forms, for example, a block diagonal covariance, or a Toeplitz (convolution) covariance which is diagonal in the frequency domain.

How should we update a diagonal C matrix? Let us fix $k, \mu, q(h_{i,\gamma,j})$ and also fix all the off-diagonal elements of C to 0. We then isolate from Eq. (44) the terms involving $C(i, i)$:

$$F(q) = \frac{1}{2} A_x(i, i) C(i, i) - \frac{1}{2} \log C(i, i) + c. \quad (49)$$

Differentiating Eq. (49) shows that it is minimized by:

$$C(i, i) = \frac{1}{A_x(i, i)}. \quad (50)$$

Therefore, a diagonal C can be updated efficiently, in $o(N)$.

Updating k: Given the mean and covariance computed above, we update k by solving the quadratic programming problem of Eq. (13).

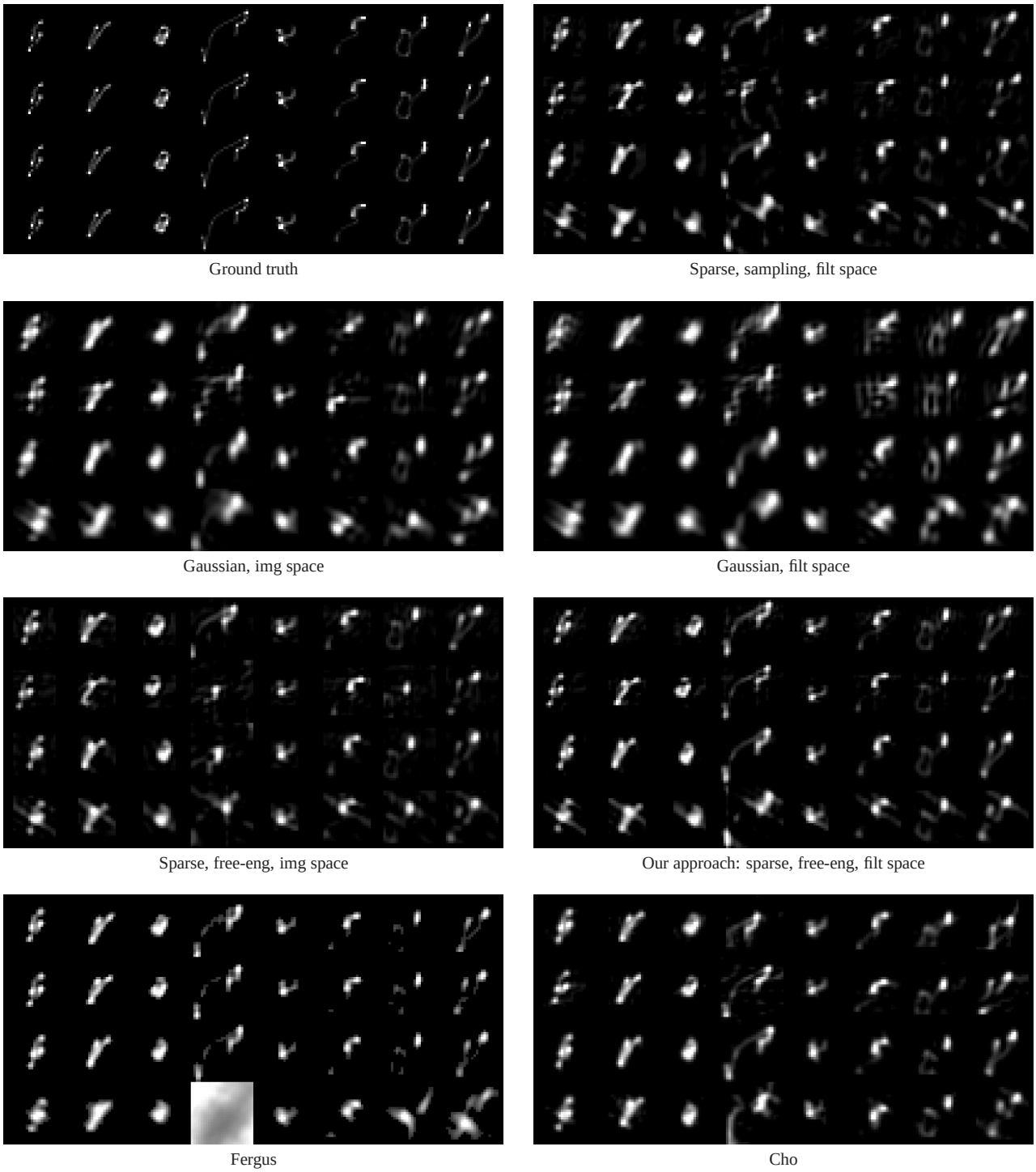
2.3. Fergus et al.'s algorithm

Our algorithm is related to the successful Fergus *et al*. approach [6], and our analysis is aimed to alleviate some of its components and simplify extensions [22, 18]. Fergus *et al*. [6] algorithm is similar to the diagonal free-energy approach, and represents the problem in derivative space (Eq. (5)). The main differences are summarized below.

Free energy definition: Fergus *et al*. [6] and the original Miskin and MacKay [17] algorithms use a more general free energy function, which aims to approximate the joint distribution $p(x, k|y)$ and not just the conditional $p(x|k, y)$. In practice, this means that they also estimate the variance around k, while our approach considers a single k estimate at each iteration. However, since Fergus' algorithm works in derivative domain, the x estimated by their variational approach is an independent set of derivatives and not the desired image. This x derivative estimate cannot be used directly, leading Fergus *et al*. to a MAP_k approach. That is, they picked only the k estimate resulting from their variational $p(x, k|y)$ approximation, and used it to deconvolve the input image. Later, Levin *et al*. [15] showed that this MAP_k approach is actually a major reason for their success. In this paper we have observed that once the goal is directly expressed as computing MAP_k, the full conditional distribution $p(x, k|y)$ is not required, which significantly simplifies the update equations.

Mean field: The algorithms of [6, 17] employ a mean-field approach. The classical mean field approach is basically a specific simplified choice of approximate distribution q, which factorizes as an independent product over pixels $q(x) = \Pi_i q(x_i)$, where each $q(x_i)$ is a 1D Gaussian, whose mean and variance should be estimated. This is essentially the case if a diagonal covariance is assumed. However, in the mean field framework, one typically updates only a single $q(x_i)$ at a time, holding all other pixels fixed. On the other hand, since we view $q(x)$ as a joint distribution on all pixels, we update all of them simultaneously. Solving Eq. (45) with respect to a single pixel $\mu(i)$ at a time is equivalent to the Gauss-Seidel linear solver, which is known as a slow numerical solver. If all variables can be updated simultaneously, stronger solvers can be employed. In our implementation we have observed that, with a sufficient number of iterations, the Gauss-Seidel approach leads to good results, but stronger solvers converge much faster.

Noise estimate: Fergus *et al*. algorithm also automatically estimates the noise variance. We have observed this is often a source of problems since their optimization diverges when the noise estimate decreases too much. Our implementation alleviates this component by assuming the noise variance is known, and we used $\eta = 0.01$ in all experiments. However, one reason for a noise update is that EM algorithms are known to converge slowly at low noise levels and faster at higher ones. To speed convergence, we start with a high noise variance and gradually reduce it during optimization, dividing by a factor of 1.15, until the desired $\eta = 0.01$ value is reached.



Ground truth | Sparse, sampling, filt space

Gaussian, img space | Gaussian, filt space

Sparse, free-eng, img space | Our approach: sparse, free-eng, filt space

Fergus | Cho

Figure 2. Recovered kernels, for the set of 32 test images, including 4 test images blurred with 8 different kernels.

3. Experiments

A `matlab` implementation of the algorithms derived in this paper is available online[2]. This unoptimized implementation processes the 255×255 test images of [15] in about 2-4 minutes.

The MAP_k algorithms described in the previous section involve three main choices. First, whether we express the problem in the image (Eq. (4)) or filter spaces (Eq. (5)).

[2] www.wisdom.weizmann.ac.il/~levina/papers/LevinEtalCVPR2011Code.zip

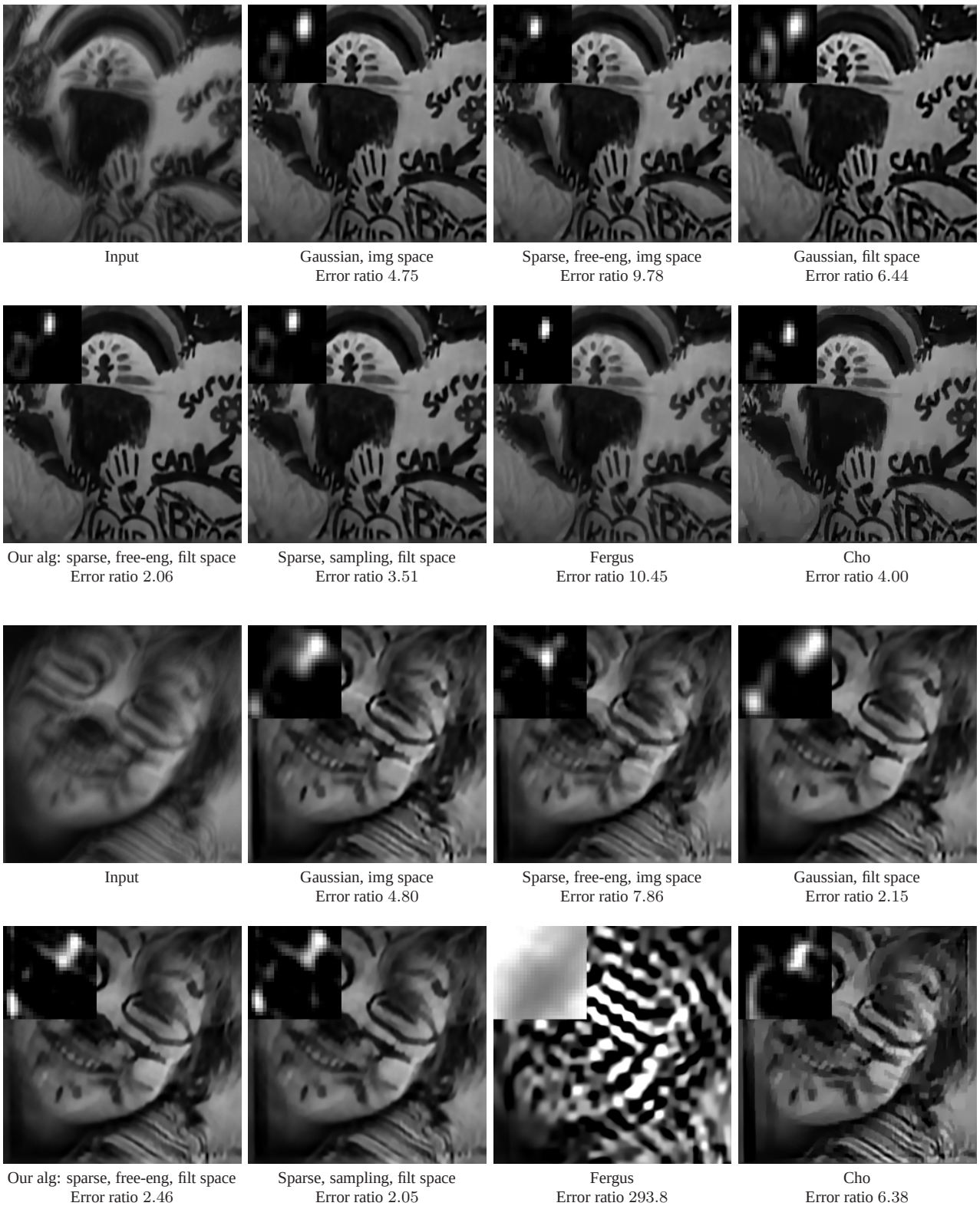


Figure 3. Recovered images, 1. We empirically observe that deconvolution results are visually plausible when the ratio of errors between deconvolution with the estimated kernel and deconvolution with the ground truth kernel is below 3.

Input | Gaussian, img space, Error ratio 2.32 | Sparse, free-eng, img space, Error ratio 1.74 | Gaussian, filt space, Error ratio 2.39

Our alg: sparse, free-eng, filt space, Error ratio 1.33 | Sparse, sampling, filt space, Error ratio 2.05 | Fergus, Error ratio 1.24 | Cho, Error ratio 1.46

Input | Gaussian, img space, Error ratio 3.07 | Sparse, free-eng, img space, Error ratio 2.76 | Gaussian, filt space, Error ratio 4.14

Our alg: sparse, free-eng, filt space, Error ratio 1.86 | Sparse, sampling, filt space, Error ratio 2.63 | Fergus, Error ratio 1.91 | Cho, Error ratio 9.21

Figure 4. Recovered images, 2

Input | Gaussian, img space, Error ratio 3.62 | Sparse, free-eng, img space, Error ratio 2.88 | Gaussian, filt space, Error ratio 3.84

Our alg: sparse, free-eng, filt space, Error ratio 2.10 | Sparse, sampling, filt space, Error ratio 1.97 | Fergus, Error ratio 3.34 | Cho, Error ratio 5.30

Input | Gaussian, img space, Error ratio 1.68 | Sparse, free-eng, img space, Error ratio 1.49 | Gaussian, filt space, Error ratio 2.69

Our alg: sparse, free-eng, filt space, Error ratio 1.27 | Sparse, sampling, filt space, Error ratio 1.18 | Fergus, Error ratio 1.30 | Cho, Error ratio 1.28

Figure 5. Recovered images, 3

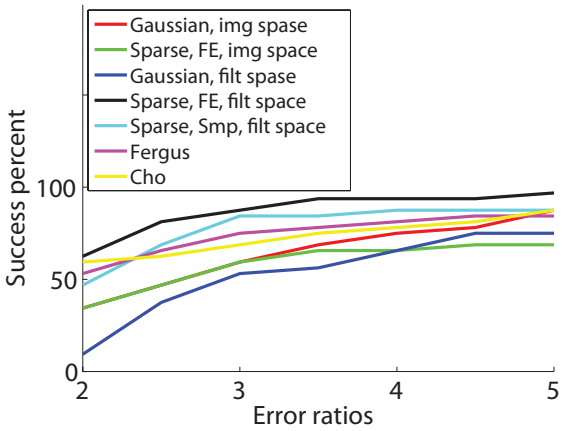


Figure 1. Evaluation results: Cumulative histogram of the deconvolution error ratio across test examples (the r'th bin counts the percentage of test examples achieving error ratio below r).

Second, the type of prior used– Gaussian or sparse. And finally, the choice of covariance approximation. To isolate the effect of the different factors we have compared five different algorithmic versions. First, a Gaussian prior [16] in both image and filter domains. In this case the covariance can be computed exactly and efficiently in the frequency basis. Second, we used a sparse MOG prior in the image and filter domains. We use the free energy approach to compute a diagonal covariance. The last algorithm used the filter domain and estimated a covariance using the sampling algorithm of [11, 19]. Like most recent blind deconvolution algorithms, we used a coarse to fine approach. We also compare our results with Cho and Lee [2], the best available $\text{MAP}_{x,k}$ algorithm, and with the original implementation of Fergus *et al*. [6].

We used the 32 test images of [15]. To evaluate the results we used the SSD ratio test of [15], and measured the ratio of error between deconvolution with the estimated and correct kernels. The idea is to normalize for the fact that harder kernels achieve a larger reconstruction error even when estimated correctly. In Fig. 1 we plot the cumulative histogram of error ratios (e.g. bin $r = 3$ counts the percentage of test examples with error ratio below 3). Fig. 2 visualizes the estimated kernels. Figs. 3–5, visualize deconvolution results for some test images. Other images are included in the code package.

The best results are obtained by the diagonal free energy approach in the derivative space. The original results of Fergus *et al*. [6] slightly outperform Cho and Lee [2]. The evaluation shows that the derivative-space approach clearly outperforms the image-domain approach, and we discuss this success below. The simple Gaussian prior performs surprisingly well and, in the image domain, it even outperforms the sparse one (our Gaussian results improve over the original results of [15]). The sampling approach is significantly slower than the free energy approach, and produces slightly less accurate results. We observe that deconvolution results are usually visually plausible when their error ratio is below 3. Thus, the error ratios in Fig. 1 show 88% success for our diagonal free energy deconvolution, compared with 75% success for the original Fergus *et al*. implementation and 69% for Cho and Lee. Despite the subtle differences, all these algorithms perform relatively well. Most importantly, they significantly outperform a naive $\text{MAP}_{x,k}$ approach with no extra computational complexity.

The success of the derivative space approach: The derivative space solution assumes independence between derivatives and ignores the important integrability constraint. Despite this problematic assumption, it largely improves the results in practice.

One advantage of the derivative representation is that it fits better with the variational model which considers an independent product over variables. Another advantage is that the deconvolution system solved in each iteration is better conditioned, since the regularization is placed on the unknowns themselves and not on their derivatives.

Another possible explanation is that the prior parameter fitting of independent derivative signals is more accurate, since it is enough to match the observed derivative histogram. In contrast, learning prior models which correctly encode dependencies between horizontal and vertical derivatives is not trivial. Thus, it is possible that the prior we used in the images space representation is not sufficiently accurate. In fact, a Gaussian prior in the image domain might provide a better approximation to the distribution than a sparse prior with wrong parameters, as suggested by its superior performance in Fig. 1.

The filter domain approach which ignores integrability is used only when estimating the kernel. Given k, the sharp image x is recovered using standard non-blind deconvolution in the image domain.

4. Discussion

The MAP_k blind deconvolution principle is significantly more robust than the $\text{MAP}_{x,k}$ principle. Yet, it is considered hard to implement and has not been widely exploited. In this paper we argue that the MAP_k approach can actually be optimized easily, and present simple and practical MAP_k algorithms. While popular $\text{MAP}_{x,k}$ strategies basically alternate between latent image estimation given a kernel and kernel estimation given an image, our MAP_k algorithm employs the same steps, where the only difference is that the kernel estimation accounts for the covariance around x and not only for the mean solution. While an exact estimation of the covariance is challenging, a diagonal approximation can be computed efficiently in $O(N)$ as the inverse diagonal of the deconvolution system.

While we have presented the basic principles of MAP_k optimization, there are many more algorithmic choices to explore, such as the choice of filters, the choice of covari-

ance approximation, and the prior model. We hope that the basic principles laid in this paper will open the door for follow up research on these important questions.

Acknowledgments: We thank the Israel Science Foundation, European Research Council, US-Israel Bi-National Science Foundation, the Royal Dutch/Shell Group, NGA NEGI-1582-04-0004, MURI N00014-06-1-0734, NSF 0964004 and Quanta.

References

[1] H. Attias. Independent Factor Analysis. *Neural Computation*, 11(4), 1999.

[2] S. Cho and S. Lee. Fast motion deblurring. *SIGGRAPH ASIA*, 2009.

[3] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from image. In *ICCV*, 2007.

[4] T. S. Cho. *Removing motion blur from photographs*. PhD thesis, Massachusetts Institute of Technology, 2010.

[5] C. Davis. All convex invariant functions of hermitian matrices. In *American Mathematical Society*, 1957.

[6] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, and W.T. Freeman. Removing camera shake from a single photograph. *SIGGRAPH*, 2006.

[7] Jiaya Jia. Single image motion deblurring using transparency. In *CVPR*, 2007.

[8] N. Joshi, R. Szeliski, and D. Kriegman. PSF estimation using sharp edge prediction. In *CVPR*, 2008.

[9] Steven M. Kay. *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[10] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, 2009.

[11] E. Levi. Using natural image priors - maximizing or sampling? Masters thesis, The Hebrew University of Jerusalem, 2009.

[12] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, 2006.

[13] A. Levin, R. Fergus, F. Durand, and W. Freeman. Image and depth from a conventional camera with a coded aperture. *SIGGRAPH*, 2007.

[14] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. *PAMI*, 2007.

[15] A Levin, Y. Weiss, F. Durand, and W. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, 2009.

[16] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman. Understanding and evaluating blind deconvolution algorithms. Technical report, MIT-CSAIL-TR-2009-014, 2009.

[17] J. Miskin and D. MacKay. Ensemble learning for blind image separation and deconvolution, 2000.

[18] F. Romeiro and T. Zickler. Blind reflectometry. In *ECCV*, 2010.

[19] U. Schmidt, Q. Gao, and S. Roth. A generative perspective on mrfs in low-level vision. In *CVPR*, 2010.

[20] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *SIGGRAPH*, 2008.

[21] Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *ICCV*, 2007.

[22] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *CVPR*, 2010.

[23] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, 2010.