Revisiting Loss-Specific Training of Filter-Based MRFs for Image Restoration

Yunjin Chen, Thomas Pock, René Ranftl, and Horst Bischof*

Institute for Computer Graphics and Vision, TU Graz

Abstract. It is now well known that Markov random fields (MRFs) are particularly effective for modeling image priors in low-level vision. Recent years have seen the emergence of two main approaches for learning the parameters in MRFs: (1) probabilistic learning using sampling-based algorithms and (2) loss-specific training based on MAP estimate. After investigating existing training approaches, it turns out that the performance of the loss-specific training has been significantly underestimated in existing work. In this paper, we revisit this approach and use techniques from bi-level optimization to solve it. We show that we can get a substantial gain in the final performance by solving the lower-level problem in the bi-level framework with high accuracy using our newly proposed algorithm. As a result, our trained model is on par with highly specialized image denoising algorithms and clearly outperforms probabilistically trained MRF models. Our findings suggest that for the loss-specific training scheme, solving the lower-level problem with higher accuracy is beneficial. Our trained model comes along with the additional advantage, that inference is extremely efficient. Our GPU-based implementation takes less than 1s to produce state-of-the-art performance.

1 Introduction and Previous Work

Nowadays the MRF prior is quite popular for solving various inverse problems in image processing in that it is a powerful tool for modeling the statistics of natural images. Image models based on MRFs, especially higher-order MRFs, have been extensively studied and applied to image processing tasks such as image denoising [14,16,15,7,19,18], deconvolution [17], inpainting [14,16,15], super-resolution [21], etc.

Due to its effectiveness, higher-order filter-based MRF models using the framework of the Field of Experts (FoE) [14], have gained the most attention. They are defined by a set of linear filters and the potential function. Based on the observation that responses of mean-zero linear filters typically exhibit heavy-tailed distributions [9] on natural images, three types of potential functions have been investigated, including the Student-t distribution (ST), generalized Laplace distribution (GLP) and Gaussian scale mixtures (GSMs) function.

* This work was supported by the Austrian Science Fund (project no. P22492) and the Austrian Research Promotion Agency (project no. 832366).

J. Weickert, M. Hein, and B. Schiele (Eds.): GCPR 2013, LNCS 8142, pp. 271–281, 2013.

Table 1. Summary of various typical MRF-based systems and the average denoising results on 68 test images [14] with $\sigma = 25$

model	potential	training	inference	PSNR
5×5 FoE	ST&Lap.	contrastive divergence	MAP, CG	27.77[14]
3×3 FoE	GSMs	contrastive divergence	Gibbs sampling	27.95[16]
5×5 FoE	GSMs	persistent contrastive divergence	Gibbs sampling	28.40[7]
5×5 FoE	ST	loss-specific(truncated optimization)	MAP, GD	28.24[2]
5×5 FoE	ST	loss-specific(truncated optimization)	MAP, lbfgs [11]	28.39[5]
5×5 FoE	ST	loss-specific(implicit differentiation)	MAP, CG	27.86[15]

In recent years several training approaches have emerged to learn the parameters of the MRF models [8,20,14,16,15,2,5,7]. Table 1 gives a summary of several typical methods and the corresponding average denoising PSNR results based on 68 test images from Berkeley database with $\sigma = 25$ Gaussian noise.Existing training approaches typically fall into two main types: (1) probabilistic training using (persistent) contrastive divergence ((P)CD); (2) loss-specific training. Roth and Black [14] first introduced the concept of FoE and proposed an approach to learn the parameters of FoE model which uses a sampling strategy and the idea of CD to estimate the expectation value over the model distribution. Schmidt *et al.* [16] improved the performance of their previous FoE model [14] by changing (1) the potential function to GSMs and (2) the inference method from MAP estimate to Bayesian minimum mean squared error estimate (MMSE). The same authors present their latest results in [7], where they achieve significant improvements by employing an improved learning scheme called PCD instead of previous CD.

Samuel and Tappen [15] present a novel loss-specific training approach to learn MRF parameters under the framework of bi-level optimization [3]. They use a plain gradient-descent technique to optimize the parameters, where the essence of this learning scheme - the gradients, are calculated by using implicit differentiation technique. Domke [5] and Barbu [2] propose two similar approaches for the training of MRF model parameters also under the framework of bi-level optimization. Their methods are some variants of standard bi-level optimization method [15]. In the modified setting, the MRF model is trained in terms of results after optimization is truncated to a fixed number of iterations, *i.e.*, they do not solve the energy minimization problem exactly; instead, they just run some specific optimization algorithm for a fixed number of steps.

In a recent work [10], the bi-level optimization technique is employed to train a non-parametric image restoration framework based on Regression Tree Fields (RTF), resulting a new state-of-the-art. This technique is also exploited for learning the so-called analysis sparsity priors [13], which is somewhat related to the FoE model.

2 Motivation and Contributions

Arguments: The loss-specific training criterion is formally expressed as the following bi-level optimization problem

$$\begin{cases} \arg\min_{\vartheta} L(x^*(\vartheta), g) \\ \text{subject to } x^*(\vartheta) = \arg\min_{x} E(x, f, \vartheta). \end{cases} \tag{1}$$

The goal of this model is to find the optimal parameters ϑ to minimize the loss function $L(x^*(\vartheta), g)$, which is called the upper-level problem in the bi-level framework. The MRF model is defined by the energy minimization problem $E(x, f, \vartheta)$, which is called the lower-level problem. The essential point for solving this bi-level optimization problem is to calculate the gradient of the loss function $L(x^*(\vartheta), g)$ with respect to the parameters ϑ. As aforementioned, [15] employs the implicit differentiation technique to calculate the gradients explicitly; in contrast, [5] and [2] make use of an approximation approach based on truncated optimization. All of them use the same ST-distribution as potential function; however, the latter two approaches surprisingly obtain much better performance than the former, as can be seen in Table 1.

In principle, Samuel and Tappen should achieve better (at least similar) results compared to the approximation approaches, because they use a “full” fitting training scheme, but actually they fail in practice. Therefore, we argue that there must exist something imperfect in their training scheme, and we believe that we will very likely achieve noticeable improvements by refining this “full” fitting training scheme.

Contributions: Motivated by the above investigation, we think it is necessary and worthwhile to restudy the loss-specific training scheme and we expect that we can achieve significant improvements. In this paper, we do not make any modifications to the training model used in [15] - we use exactly the same model capacity, potential function and training images. The only difference is the training algorithm. We exploit a refined training algorithm that we solve the lower-level problem in the loss-specific training with very high accuracy and make use of a more efficient quasi-Newton's method for model parameters optimization. We conduct a series of playback experiments and we show that the performance of loss-specific training is indeed underestimated in previous work [15]. We argue that the the critical reason is that they have not solved the lower-level problem to sufficient accuracy. We also demonstrate that solving the lower-level problem with higher accuracy is indeed beneficial. This argument about the loss-specific training scheme is the major contribution of our paper.

We further show that our trained model can obtain slight improvement by increasing the model size. It turns out that for image denoising task, our optimized MRF (opt-MRF) model of size 7×7 has achieved the best result among existing MRF-based systems and been on par with state-of-the-art methods. Due to the simplicity of our model, it is easy to implement the inference algorithm on parallel computation units, *e.g.*, GPU. Numerical results show that our GPU-based implementation can perform image denoising in near real-time with clear state-of-the-art performance.

3 Loss-Specific Training Scheme: Bi-level Optimization

In this section, we firstly present the loss-specific training model. Then we consider the optimization problem from a more general point of view. Our derivation shows that the implicit differentiation technique employed in previous work [15] is a special case of our general formulation.

3.1 The Basic Training Model

Our training model makes use of the bi-level optimization framework, and is conducted based on the image denoising task. For image denoising, the ST-distribution based MRF model is expressed as

$$\arg\min_{x} E(x) = \sum\nolimits_{i=1}^{N_f} \alpha_i \sum\nolimits_{p=1}^{N_p} \rho((K_i x)_p) + \frac{\lambda}{2}\|x - f\|_2^2. \tag{2}$$

This is the lower-level problem in the bi-level framework. Wherein N_f is the number of filters, N_p is the number of pixels in image x, K_i is an $N_p \times N_p$ highly sparse matrix, which makes the convolution of the filter k_i with a two-dimensional image x equivalent to the product result of the matrix K_i with the vectorization form of x, *i.e.*, $k_i * x \Leftrightarrow K_i x$. In our training model, we express the filter K_i as a linear combination of a set of basis filters $\{B_1, \cdots, B_{N_B}\}$, *i.e.*, $K_i = \sum_{j=1}^{N_B} \beta_{ij} B_j$. Besides, $\alpha_i \geq 0$ is the parameters of ST-distribution for filter K_i, and λ defines the trade-off between the prior term and data fitting term. $\rho(\cdot)$ denotes the Lorentzian potential function $\rho(z) = \log(1 + z^2)$, which is derived from ST-distribution.

The loss function $L(x^*, g)$ (upper-level problem) is defined to measure the difference between the optimal solution of energy function and the ground-truth. In this paper, we make use of the same loss function as in [15], $L(x^*, g) = \frac{1}{2}\|x^* - g\|_2^2$, where g is the ground-truth image and x^* is the minimizer of (2).

Given the training samples $\{f_k, g_k\}_{k=1}^N$, where g_k and f_k are the k^{th} clean image and the associated noisy version respectively, our aim is to learn an optimal MRF parameter $\vartheta = (\alpha, \beta)$ (we group the coefficients β_{ij} and weights α_i into a single vector ϑ), to minimize the overall loss function. Therefore, the learning model is formally formulated as the following bi-level optimization problem

$$\begin{cases} \min\limits_{\alpha \geq 0, \beta} L(x^*(\alpha, \beta)) = \sum_{k=1}^N \frac{1}{2}\|x_k^*(\alpha, \beta) - g_k\|_2^2 \\ \text{where } x_k^*(\alpha, \beta) = \arg\min\limits_{x} \sum_{i=1}^{N_f} \alpha_i \rho(K_i x) + \frac{1}{2}\|x - f_k\|_2^2, \end{cases} \tag{3}$$

where $\rho(K_i x) = \sum_{p=1}^{N_p} \rho((K_i x)_p)$. We eliminate λ for simplicity, since it can be incorporated into weights α.

3.2 Solving the Bi-level Problem

In this paper, we consider the bi-level optimization problem from a general point of view. In the following derivation we only consider the case of a single training sample for convenience, and we show how to extend the framework to multiple training samples in the end.

According to the optimality condition, the solution of the lower-level problem in (3) is given by x^*, such that $\nabla_x E(x^*) = 0$. Therefore, we can rewrite problem (3) as following constrained optimization problem

$$\begin{cases} \min\limits_{\alpha \geq 0, \beta} L(x(\alpha, \beta)) = \frac{1}{2}\|x(\alpha, \beta) - g\|_2^2 \\ \text{subject to } \nabla_x E(x) = \sum_{i=1}^{N_f} \alpha_i K_i^T \rho'(K_i x) + x - f = 0, \end{cases} \tag{4}$$

where $\rho'(K_i x) = (\rho'((K_i x)_1), \cdots, \rho'((K_i x)_p))^T \in \mathbb{R}^{N_p}$. Now we can introduce Lagrange multipliers and study the Lagrange function

$$\mathcal{L}(x, \alpha, \beta, p, \mu) = \frac{1}{2}\|x - g\|_2^2 + \langle -\alpha, \mu \rangle + \langle \sum\nolimits_{i=1}^{N_f} \alpha_i K_i^T \rho'(K_i x) + x - f, p \rangle, \tag{5}$$

where $\mu \in \mathbb{R}^{N_f}$ and $p \in \mathbb{R}^{N_p}$ are the Lagrange multipliers associated to the inequality constraint $\alpha \geq 0$ and the equality constraint in (4), respectively. Here $\langle \cdot, \cdot \rangle$ denotes the standard inner product. Taking into account the inequality constraint $\alpha \geq 0$, the first order necessary condition for optimality is given by

$$G(x, \alpha, \beta, p, \mu) = 0, \tag{6}$$

where

$$G(x, \alpha, \beta, p, \mu) = \begin{pmatrix} (\sum_{i=1}^{N_f} \alpha_i K_i^T \mathcal{D}_i K_i + \mathcal{I})p + x - g \\ (\langle K_i^T \rho'(K_i x), p \rangle)_{N_f \times 1} - \mu \\ (\langle B_j^T \rho'(K_i x) + K_i^T \mathcal{D}_i B_j x, p \rangle)_{n \times 1} \\ \sum_{i=1}^{N_f} \alpha_i K_i^T \rho'(K_i x) + x - f \\ \mu - \max(0, \mu - c\alpha) \end{pmatrix}.$$

Wherein $\mathcal{D}_i(K_i x) = \operatorname{diag}(\rho''((K_i x)_1), \cdots, \rho''((K_i x)_p)) \in \mathbb{R}^{N_p \times N_p}$, $(\langle \cdot, p \rangle)_{N \times 1} = (\langle (\cdot)_1, p \rangle, \cdots, \langle (\cdot)_r, p \rangle)^T$, in the third formulation $n = N_f \times N_B$. Note that the last formulation is derived from the optimality condition for the inequality constraint $\alpha \geq 0$, which is expressed as $\alpha \geq 0, \mu \geq 0, \langle \alpha, \mu \rangle = 0$. It is easy to check that these three conditions are equivalent to $\mu - \max(0, \mu - c\alpha) = 0$ with c to be any positive scalar and max operates coordinate-wise.

Generally, we can continue to calculate the generalized Jacobian of G, *i.e.*, the Hessian matrix of Lagrange function, with which we can then employ a Newton's method to solve the necessary optimality system (6). However, for this problem calculating the Jacobian of G is computationally intensive; thus in this paper we do not consider it and only make use of the first derivatives.

Since what we are interested in is the MRF parameters $\vartheta = \{\alpha, \beta\}$, we can reduce unnecessary variables in (6). By solving for p and x in (6), and substituting them into the second and the third formulation, we arrive at the gradients of loss function with respect to parameters ϑ

$$\begin{cases} \nabla_{\beta_{ij}} L = -(B_j^T \rho'(K_i x) + K_i^T \mathcal{D}_i B_j x)^T (H_E(x))^{-1} (x - g) \\ \nabla_{\alpha_i} L = -(K_i^T \rho'(K_i x))^T (H_E(x))^{-1} (x - g) \\ \text{where } \nabla_x E(x) = \sum_{i=1}^{N_f} \alpha_i K_i^T \rho'(K_i x) + x - f = 0. \end{cases} \tag{7}$$

In (7), $H_E(x)$ denotes the Hessian matrix of $E(x)$,

$$H_E(x) = \sum\nolimits_{i=1}^{N_f} \alpha_i K_i^T \mathcal{D}_i K_i + \mathcal{I}. \tag{8}$$

In (7), we also eliminate the Lagrange multiplier μ associated to the inequality constraint $\alpha \geq 0$, as we utilize a quasi-Newton's method for optimization, which can easily handle this type of box constraints. We can see that (7) is equivalent to the results presented in previous work [15] using implicit differentiation.

Considering the case of N training samples, in fact it turns out that the derivatives of the overall loss function in (3) with respect to the parameters ϑ are just the sum of (7) over the training dataset.

As given by (7), we have collected all the necessary information to compute the required gradients, so we can now employ gradient descent based algorithms for optimization, *e.g.*, steepest-descent algorithm. In this paper, we turn to a more efficient non-linear optimization method–the LBFGS quasi-Newton's method [11]. In our experiments, we will make use of the LBFGS implementation distributed by L. Stewart[1]. In our work, the third equation in (7) is completed the L-BFGS algorithm, since this problem is smooth, to which L-BFGS is perfectly applicable. The training algorithm is terminated when the relative change of the loss is less than a tolerance, *e.g.*, $tol = 10^{-5}$ or a maximum number of iterations *e.g.*, $maxiter = 500$ is reached or L-BFGS can not find a feasible step to decrease the loss.

4 Training Experiments

In order to demonstrate that the loss-specific training scheme was undervalued in previous work [15], we conducted a playback experiment using (1) the same 40 images for training and 68 images for testing; (2) the same model capacity–24 filters of size 5×5; (3) the same basis –"inverse" whitened PCA [14], as in Samuel and Tappen's experiments. We randomly sampled four 51×51 patches from each training image, resulting in a total of 160 training samples. We then generated the noisy versions by adding Gaussian noise with standard deviation $\sigma = 25$.

The major difference between our training experiment and previous one is the training algorithm. In our refined training scheme, we employed (1) our proposed algorithm to solve the lower-level problem with very high accuracy, and (2) LBFGS to optimize the model parameters, but in contrast, Samuel and Tappen used non-linear conjugate gradient and plain gradient descent algorithm, respectively. In our refined training algorithm, we used the normalized norm of the gradient, *i.e.*, $\frac{\|\nabla_x E(x^*)\|_2}{\sqrt{N}} \leq \varepsilon_l$ (N is the pixel number of the training patch) as the stopping criterion for solving the lower-level problem. In our training experiment, we set $\varepsilon_l = 10^{-5}$ (gray-value in range [0 255]), which implies a very accurate solution.

[1] `http://www.cs.toronto.edu/~liam/software.shtml`

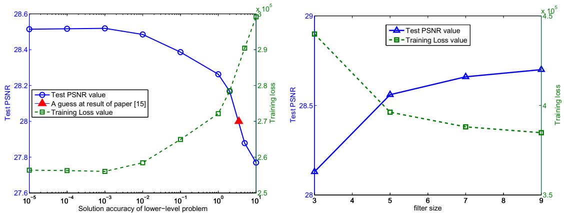

Fig. 1. Performance curves (test PSNR value and training loss value) *vs.*{the solution accuracy of the lower-level problem ε_l & the filter size}. It is clear that solving the lower-level problem with higher accuracy is beneficial and larger filter size can normally bring some improvement.

Based on this training configuration, we learned 24 filters of size 5×5, then we applied them to image denoising task to estimate the inference performance using the same 68 test images. Finally, we got an average PSNR value of 28.51dB for noise level $\sigma = 25$, which is significantly superior to previous result of 27.86dB in [15]. We argue that the major reason lies in our refined training algorithm that we solve the lower-level problem with very high accuracy.

To make this argument more clear, we need to eliminate the possibility of training dataset, because we did not exploit exactly the same training dataset as previous work (unfortunately we do not have their dataset in hand). Since the training patches were randomly selected, we could run the training experiment multiple times by using different training dataset. Finally, we found that the deviation of test PSNR values based on 68 test images is within 0.02dB, which is negligible. Therefore, it is clear that training dataset is not the reason for this improvement, and the only remaining reason is our refined training scheme.

The Influence of ε_l: To investigate the influence of the solution accuracy of the lower-level problem ε_l more detailedly, we conducted a series of training and testing experiments by setting ε_l to different magnitudes. Based on a fixed training dataset (160 patches of size 51×51) and 68 test images, we got the performance curves with respect to the solution accuracy ε_l, as shown in Figure 1 (left). From Figure 1 (left), we can clearly see that it is indeed the high solution accuracy that helps us to achieve the above siginificant improvement. This finding is the main contribution of our paper. We also make a guess how accurate Samuel and Tappen solve the lower-level problem according to their result and our performance curve, which is marked by a red triangle in Figure 1 (left). The argument that higher solution accuracy of the lower-level problem is helpful is explicable, the reason is described below.

As we know, the key aspect of our approach is to calculate the gradients of the loss function with respect to the parameters ϑ. According to (7), there is a precondition to obtain accurate gradients: both the lower-level problem and the inverse matrix of Hessian matrix H_E must be solved with high accuracy, *i.e.*, we need to calculate a x^* such that $\nabla_x E(x^*) = 0$ and compute $(H_E)^{-1}$ explicitly. Since the Hessian matrix H_E is highly sparse, we can solve the linear system $H_E x = b$ efficiently with very high accuracy (we use the "backslash" operator in

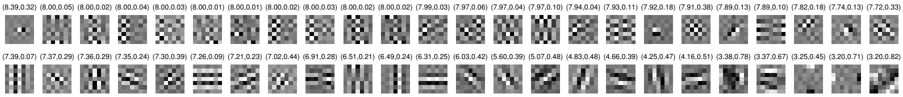

Fig. 2. 48 learned filters (7×7). The first number in the bracket is the weight α_i and the second one is the norm of the filter.

Matlab). However, for the lower-level problem, in practice we can only solve it to finite accuracy by using certain algorithms, *i.e.*, $\frac{\|\nabla_x E(x^*)\|_2}{\sqrt{N}} \leq \varepsilon_l$. If the lower-level problem is not solved to sufficient accuracy, the gradients $\nabla_\vartheta L$ are certainly inaccurate which will probably affect the training performance. This has been demonstrated in our experiments. Therefore, for the bi-level training framework, it is necessary to solve the lower-level problem as accurately as possible, *e.g.*, in our training we solved it to a very high accuracy with $\varepsilon_l = 10^{-5}$.

The Influence of Basis: In our playback experiments, we used the "inverse" whitened PCA basis to keep consistent with previous work. However, we argue that the DCT basis is a better choice, because meaningful filters should be mean-zero according to the findings in [9], which is guaranteed by DCT basis without the constant basis vector. Therefore, we will exploit the DCT filters excluding the filter with uniform entries from now on. Using this modified DCT basis, we retrained our model and we got a test PSNR result of 28.54dB.

The Influence of Training Dataset: To verify whether larger training dataset is beneficial, we retrained our model by using (1) 200 samples of size 64×64 and (2) 200 samples of size 100×100, which is about two times and four times larger than our previous dataset, respectively. Finally, we got a test PSNR result of 28.56dB for both cases. As shown before, the influence of training dataset is marginal.

The Influence of Model Capacity: In above experiments, we concentrated on the model of size 5×5 to keep consistent with previous work. We can also train models of different filter sizes, *e.g.*, 3×3, 7×7 or 9×9, to investigate the influence of model capacity. Based on the training dataset of 200 patches of size 64×64, we retrained our model with different filter size; the training results and testing performance are summarized in Figure 1 (right). We can see that normally increasing the filter size can bring some improvement. However, the improvement of filter size 9×9 is marginal compared to filter size 7×7, yet the former is much more time consuming. The training time for the model with 48 filters of size 7×7 was approximately 24 hours on a server (Intel X5675, 3.07GHz, 24 cores), but in contrast, the model of size 9×9 took about 20 days. More importantly, the inference time of the model of size 9×9 is certainly longer than the model of size 7×7, in that it involves more filters of larger size. Therefore, the model of size 7×7 offers the best trade-off between speed and quality, and we use it for the following applications. The learned 48 filters together with their associated weights and norms are presented in Figure 2.

Table 2. Summary of denoising experiments results (average PSNRs over 68 test images from the Berkeley database). We highlighted the state-of-the-art results.

σ	KSVD	FoE	BM3D	LSSC	EPLL	Ours
15	30.87	30.99	31.08	**31.27**	31.19	31.18
25	28.28	28.40	28.56	**28.70**	**28.68**	**28.66**
50	25.17	25.35	25.62	**25.72**	**25.67**	**25.70**

Table 3. Typical run time of the denoising methods for a 481×321 image ($\sigma = 25$) on a server (Intel X5675, 3.07GHz). The highlighted number is the run time of GPU implementation.

	KSVD	FoE	BM3D	LSSC	EPLL	Ours
T(s)	30	1600	4.3	700	99	12 (**0.3**)
psnr	28.28	28.40	28.56	28.70	28.68	28.66

5 Application Results

An important question for a learned prior model is how well it generalizes. To evaluate this, we directly applied the above 48 filters of size 7×7 trained based on image denoising task to various image restoration problems such as image deconvolution, inpainting and super-resolution, as well as denoising. Due to space limitation, here we only present denoising results and the comparison to state-of-the-arts. The other results will be shown in the final version [1].

We applied our opt-MRF model to image denoising problem and compared its performance with leading image denoising methods, including three state-of-the-art methods: (1) BM3D [4]; (2) LSSC [12]; (3) GMM-EPLL [22] along with two leading generic methods: (4) a MRF-based approach, FoE [7]; and (5) a synthesis sparse representation based method, KSVD [6] trained on natural image patches. All implementations were downloaded from the corresponding authors' homepages. We conducted denoising experiments over 68 test images with various noise levels $\sigma = \{15, 25, 50\}$. To make a fair comparison, we used exactly the same noisy version of each test image for different methods and different test images were added with distinct noise realizations. All results were computed per image and then averaged over the test dataset. We used L-BFGS to solve the MAP-based MRF model (2). When (2) is applied to various noise level σ, we need to tune the parameter λ (empirical choice $\lambda = 25/\sigma$).

Table 2 shows the summary of results. It is clear that our opt-MRF model outperforms two leading generic methods and has been on par with three state-of-the-art methods for any noise level. Comparing the result of our opt-MRF model with results presented in Table 1, our model has obviously achieved the best performance among all the MRF-based systems. To the best of our knowledge, this is the first time that a MRF model based on generic priors of natural images has achieved such clear state-of-the-art performance. We provide image denoising examples in the final version [1].

In additional, our opt-MRF model is well-suited to GPU parallel computation in that it only contains the operation of convolution. Our GPU implementation based on NVIDIA Geforce GTX 680 accelerates the inference procedure significantly; for a denoising task with $\sigma = 25$, typically it takes 0.42s for image size 512×512, 0.30s for 481×321 and 0.15s for 256×256. In Table 3, we show the average run time of the considered denoising methods on 481×321 images. Considering the speed and quality of our model, it is a perfect choice of the base methods in the image restoration framework recently proposed in [10], which leverages advantages of existing methods.

6 Conclusion

In this paper, we revisited the loss-specific training approach proposed by Samuel and Tappen in [15] by using a refined training algorithm. We have shown that the performance of the loss-specific training was indeed undervalued in previous work. We argued that the major reason lies in the solution accuracy of the lower-level problem in the bi-level framework, and we have demonstrated that solving the lower-level problem with higher accuracy is beneficial. We have shown that we can further improve the performance of the learned model a little bit by using larger filters. For image denoising task, our learned opt-MRF model of size 7×7 presented the best performance among existing MRF-based systems, and has already been on par with state-of-the-art denoising methods. The performance of our opt-MRF model proves two issues: (1) the loss-specific training scheme under the framework of bi-level optimization, which is convergence guaranteed, is highly effective for parameters learning; (2) MAP estimate should be still considered as one of the leading approaches in low-level vision.

References

1. http://gpu4vision.icg.tugraz.at/
2. Barbu, A.: Training an active random field for real-time image denoising. IEEE Trans. on Image Proc. 18(11), 2451–2462 (2009)
3. Colson, B., Marcotte, P., Savard, G.: An overview of bilevel optimization. Annals OR 153(1), 235–256 (2007)
4. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.O.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Trans. on Image Proc. 16(8), 2080–2095 (2007)
5. Domke, J.: Generic methods for optimization-based modeling. Journal of Machine Learning Research - Proceedings Track 22, 318–326 (2012)
6. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. on Image Proc. 15(12), 3736–3745 (2006)
7. Gao, Q., Roth, S.: How well do filter-based MRFs model natural images? In: Pinz, A., Pock, T., Bischof, H., Leberl, F. (eds.) DAGM and OAGM 2012. LNCS, vol. 7476, pp. 62–72. Springer, Heidelberg (2012)
8. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Computation 14(8), 1771–1800 (2002)

9. Huang, J., Mumford, D.: Statistics of natural images and models. In: CVPR, Fort Collins, CO, USA, pp. 541–547 (1999)
10. Jancsary, J., Nowozin, S., Rother, C.: Loss-specific training of non-parametric image restoration models: A new state of the art. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VII. LNCS, vol. 7578, pp. 112–125. Springer, Heidelberg (2012)
11. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Mathematical Programming 45(1), 503–528 (1989)
12. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: ICCV, pp. 2272–2279 (2009)
13. Peyré, G., Fadili, J.: Learning analysis sparsity priors. In: Proc. of Sampta 2011 (2011), http://hal.archives-ouvertes.fr/hal-00542016/
14. Roth, S., Black, M.J.: Fields of experts. International Journal of Computer Vision 82(2), 205–229 (2009)
15. Samuel, K.G.G., Tappen, M.: Learning optimized MAP estimates in continuously-valued MRF models. In: CVPR (2009)
16. Schmidt, U., Gao, Q., Roth, S.: A generative perspective on MRFs in low-level vision. In: CVPR, pp. 1751–1758 (2010)
17. Schmidt, U., Schelten, K., Roth, S.: Bayesian deblurring with integrated noise estimation. In: CVPR, pp. 2625–2632 (2011)
18. Tappen, M.F., Liu, C., Adelson, E.H., Freeman, W.T.: Learning gaussian conditional random fields for low-level vision. In: CVPR, pp. 1–8 (2007)
19. Tappen, M.F.: Utilizing variational optimization to learn markov random fields. In: CVPR, pp. 1–8 (2007)
20. Weiss, Y., Freeman, W.T.: What makes a good model of natural images? In: CVPR (2007)
21. Zhang, H., Zhang, Y., Li, H., Huang, T.S.: Generative bayesian image super resolution with natural image prior. IEEE Trans. on Image Proc. 21(9), 4054–4067 (2012)
22. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: ICCV, pp. 479–486 (2011)