Optimal Learning High-Order MRF Priors of Color Image

No Author Given

No Institute Given

Please print in color

Abstract. In this paper, we present an optimised learning algorithm for high-order Markov random fields (MRF) color image priors that capture the statistics of natural scenes and can be used for a variety of computer vision tasks. The proposed optimal learning algorithm is achieved by simplifying the estimation of partition function in the learning model. The parameters in MRF color image priors are learned alteratively and iteratively by maximising their likelihood. We demonstrate the capability of the proposed learning algorithm of high-order MRF color image priors with the application of color image denoising. Experimental results show the superior performance of our algorithm compared to the state–of–the–art of color image priors in [1], although we use a much smaller training image set.

Keywords: MRF, image prior, color image denoising

1 Introduction

The need for prior models of image structure occurs in many computer vision problems including stereo, optical flow, denoising, super-resolution and image-based rendering to name a few. Whenever a observed“scene” must be inferred from noisy, degraded or loss partial image information, a natural image prior is required [2]. Modeling image priors is a challenging work, because of the high-dimensionality of images, their non-Gaussian statistics and the need to model correlations in image structure over extended image neighborhoods [3]. Some researchers attempted to use sparse coding approaches to address the modeling of complex image structure. Based on a variety of simple assumptions, they have obtained sparse representations of local image structure in terms of the statistics of filters that are local in position, orientation, and scale [4][5]. However, these methods which focus on image patches provide no direct way of modeling the statistics of whole images [3].

Markov random fields on the other hand have been widely used in computer vision but exhibit serious limitations. In particular, as MRF priors typically exploit handcrafted clique potentials and small neighborhood systems, it is limited the expressiveness of the models, and only crudely capture the statistics of natural images [6]. Since typical MRF models consider simple nearest neighbor relations and model first derivative filter response, extremely local (e.g. first order) priors employed by most MRF methods may not show any advantages, comparing with rich, patch based priors obtained by sparse coding methods [3].

However, Roth and Black [3] went beyond this limitation with a model, called the *Fields of Experts* (FoE) model, which is a generic Markov random field model of image priors over extended neighborhoods. For availability of application, they represented the MRF potentials as a *Product of Experts* (PoE) [7]. As FoE takes the product over all neighborhoods of each image patch, the number of parameters is only determined by the size of the maximal cliques in the MRF model and the number of filters defining the potential [3]. Furthermore, because of the homogeneity of the potential functions, the model does not have any restriction for the size of images [3]. As shown in their experiment, FoE can achieve the state–of–the–art performance for monochromatic image denoising and inpainting. Based on the work of Roth and Black [3], McAuley et al. [1] proposed a MRF color image prior model by generalising FoE model to capture the correlations between the different color channels. Their model was compared with the original FoE monochromatic prior for color image denoising and evidenced peformance improvements although their learning algorithm is clearly sub-optimal.

In this paper we build on McAuley et al.'s [1] contribution to further improve the learning algorithm for color image priors. As the learning model of color image priors is significantly simpler than the one in [1], by improving the estimation of the model partition function, both high dimensional filters and their corresponding weights can be optimal learned by maximising the likelihood. The experimental results show improvements of results reported in McAuley et al. [1] on color image denoising, although we use a much smaller training image set.

The remainder of this paper is organised as follows. In Section 2, we briefly illustrate the MRF image prior models and their original learning approaches. Our optimised learning algorithm is introduced in Section 3. In Section 4, we demonstrate the performance of our learning algorithm and compare the denoising quality with three other methods (McAuley et al's [1], Bilateral Filtering and Wavelet-based denoising approach). Finally, Section 5 concludes this paper.

2 MRF Prior Model

2.1 The Monochromatic MRF Prior

In [3], Roth and Black have merged the ideas of learning in MRF and sparse image coding in order to develop a high-order MRF prior model where the cliques are square image patches [8][4]. According to the Hammersley–Clifford theorem, the joint probability distribution of a MRF with clique set C can be written as

$$P(\mathbf{x}) = \frac{1}{Z(\Theta)} \prod_{c \in C} \phi_c(\mathbf{x}_c), \tag{1}$$

where $\phi_c(\mathbf{x}_c)$ is a potential function and $Z(\Theta)$ is the partition function.

The potential functions over these cliques are assumed to be *Products of Experts* [7], i.e., products of individual function ϕ_f (with a parameter α_f) given the response of a filter J_f to the image patch $\mathbf{x}_c$:

$$\phi_c(\mathbf{x}_c; J, \alpha) = \prod_{f=1}^{F} \phi_f(\mathbf{x}_c; J_f, \alpha_f), \tag{2}$$

In the prior model, the potential functions are assumed to be stationary, i.e., every clique in the image has the same parameter set $\Theta = \{J_f, \alpha_f : 1 \leq f \leq F\}$.

The particular form they postulated for the expert is related to the Student–T distribution, and is given by [3]:

$$\phi_f(\mathbf{x}_c; J_f, \alpha_f) = (1 + \frac{1}{2} < J_f, \mathbf{x}_c >^2)^{-\alpha_f}. \tag{3}$$

Because the computing of partition function $Z(\Theta)$ is intractable, Roth and Black [3] used contrastive divergence to learn both J's and α's.

2.2 The Color Image MRF Prior

(Is tense used in this subsection correct?) McAuley et al. [1] extended the monochromatic MRF prior to color images. They proposed the "higher" order MRF prior model, i.e., using $3 \times 3 \times 3$ clique instead of 3×3 clique, to represent the correlations of color channels over the local neighbourhood. To deal with the unacceptable computing load due to the significant rise of data dimensions in this color model, they adopted a simple gradient-ascent-based learning algorithm rather than learning by maximising the likelihood. In their learning algorithm, they perform singular value decomposition (SVD) over the covariance matrix of training data to learn filters J's, and only update α's alone the gradient direction.

The estimation of the α's in their model takes the following form: Let $D = \{X_1, X_2, \cdots, X_M\}$ a set of training images, $R = \{Y_1, Y_2, \cdots, Y_N\}$ a set of random images, $P(D|\Theta)$ the likelihood of the training images given the model. Let $\Theta = \{\theta_1, \theta_2, \cdots, \theta_F\}$ where $\theta_f = (J_f, \alpha_f)$ a set of filters and their corresponding weights. Then the likelihood of training images, $P(D|\Theta)$ is given by:

$$P(D|\Theta) = \prod_{i=1}^{M} \frac{1}{Z(\Theta)} \prod_{c \in C} \phi_c(\mathbf{x}_c^i; J, \alpha), \tag{4}$$

where $Z(\Theta)$ is partition function. They use a arithmetic mean, $\hat{Z}_{am}(\Theta)$, to estimate the real value of partition function, given by

$$Z(\Theta) \propto \hat{Z}_{am}(\Theta) = \frac{1}{N} \sum_{i=1}^{N} \prod_{c \in Y_i} \phi_c(\mathbf{x}_c^i; J, \alpha). \tag{5}$$

By taking the derivative with respect to a particular α_k, the gradient of the α_k is obtained:

$$\frac{\partial}{\partial \alpha_k} \log P(D|\Theta) = \sum_{i=1}^{M} \sum_{c \in X_i} \psi_c(J_k, \mathbf{x}_c^i) - M \frac{\partial}{\partial \alpha_k} \log Z(\Theta), \tag{6}$$

where $\psi(a,b) = -\log(1 + \frac{1}{2} < a, b >^2)$.

According to Eq.5, the derivative of the log partition function is given by

$$\frac{\partial}{\partial \alpha_k} \log Z(\Theta) = \frac{\sum_{i=1}^{N} [\sum_{c \in Y_i} \psi_c(J_k, \mathbf{x}) (\prod_{c \in Y_i} \phi_c(\mathbf{x_c}; J, \alpha))]}{\sum_{i=1}^{N} \prod_{c \in Y_i} \phi_c(\mathbf{x_c}; J, \alpha)}, \tag{7}$$

and the α's are updated along the gradient ascent direction which can be obtained by Eq. 6 and Eq. 7.

(Could you please improve the part of problem presentation?) Note that the algorithm proposed by McAuley et al. [1] only updates the α's with fixed values of filters. For several reasons their learning algorithm is not optimal: (1) the filters obtained from SVD are sub–optimal because they ideally should be learned by maximising the likelihood; (2) the α's in their learning algorithm must be initialised to zero [1], and the gradient ascent dose not work (absolute values of α's are not convergent and their relative values remain the same) after the first iteration according to our implementation. Our aim is to learn a set of filters and their corresponding weights that maximises the likelihood. This can be achieved via standard gradient ascent method given initial estimates of the model parameters. However, as the $Z(\Theta)$ estimated in [1] is just proportional to the true $Z(\Theta)$, we can not obtain a reliable model likelihood by Eq.4. Furthermore, when we perform the partial derivative with respect to J and implemented it, we found that the J–α gradient iteration is not convergent.

3 An Optimised Learning Algorithm

3.1 Estimation of Partition Function

Although the approximation of $Z(\Theta)$, $\hat{Z}_{am}(\Theta)$, has a clear physical meaning (when the images used to approximate $Z(\Theta)$ cover all possible images, this estimation represents the true form of $Z(\Theta)$), it is the main cost in the parameters updates, i.e. the complicated form of Eq. 7, and may occur the gradient iteration invalid.

For solving the problems described above, we use geometric mean $\hat{Z}_{gm}(\Theta)$ which has more robust performance in the case of non-Gaussian distributions [9], instead of arithmetic mean $\hat{Z}_{am}(\Theta)$ (Eq.5). The geometric mean of the partition function can be expressed as:

$$\hat{Z}_{gm}(\Theta) = (\prod_{i=1}^{N} \prod_{c \in Y_i} \phi_c(\mathbf{x}_c^i; J, \alpha))^{\frac{1}{N}}. \tag{8}$$

According to Jensen's inequality [10], we can obtain the upper and lower boundaries of $\hat{Z}_{am}(\Theta)$ and $\hat{Z}_{gm}(\Theta)$:

$$(\frac{\sum_{i=1}^{N} f_i(\varepsilon)^2}{N})^{\frac{1}{2}} \geq \frac{1}{N} \sum_{i=1}^{N} f_i(\varepsilon) \geq (\prod_{i=1}^{N} f_i(\varepsilon))^{\frac{1}{N}} \geq N(\sum_{i=1}^{N} \frac{1}{f_i(\varepsilon)})^{-1}, \tag{9}$$

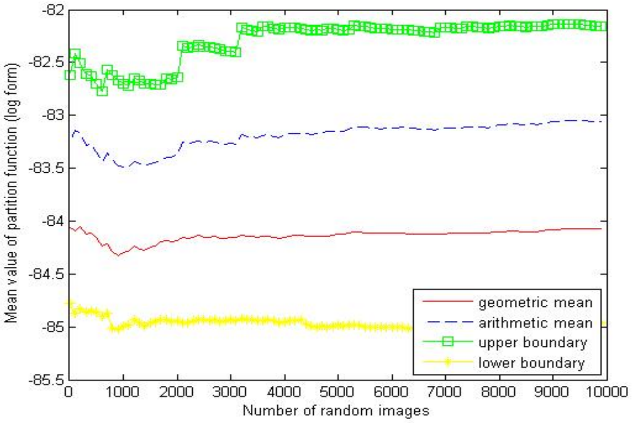

Fig. 1. The log values of two $Z(\Theta)$ estimation methods along the increasing number of random images.

where $f_i(\varepsilon) = \prod_{c \in Y_i} \phi_c(\mathbf{x}_c^i; J, \alpha)$.

As shown in Fig. 1, the log values of $\hat{Z}_{gm}(\Theta)$ and $\hat{Z}_{am}(\Theta)$ are very closed along the increasing number of random images. Furthermore, we found that the standard deviations of $\log \hat{Z}_{gm}(\Theta)$ is smaller than those of $\log \hat{Z}_{am}(\Theta)$ over various amount of random images tests. Thus, we can say that $\hat{Z}_{gm}(\Theta)$ is a robust approximation to the mean of the partition function, and can use a small set of random images to estimate the mean values of partition function in log form. In the calculation of the model likelihood given a set of parameters, we can use $\hat{Z}(\Theta) = T \times \hat{Z}_{gm}(\Theta)$ to estimate the true value of $Z(\Theta)$. In there, T is the number of assignments for the all possible pixel values of the images patch, i.e., $T = 256^{3\times3\times3}$ (3×3 clique size) or $T = 256^{5\times5\times3}$ (5×5 clique size). Based on the assumption above, the approximation of log–partition function $\hat{Z}(\Theta)$ can be rewritten as:

$$\begin{aligned} \log \hat{Z}(\Theta) &= \log T + \log \hat{Z}_{gm}(\Theta) \\ &= \log T + \frac{1}{N} \sum_{i=1}^{N} \sum_{c \in X_i} \sum_{f=1}^{F} \alpha_f \psi_f(J_f, \mathbf{x}_c^i). \end{aligned} \tag{10}$$

3.2 Proposed Learning Algorithm

The log–likelihood of MRF prior model (Eq.4) can be rewritten as follows:

$$
\begin{aligned}
\log P(D|\Theta) = & \sum_{i=1}^{M} \sum_{c \in X_i} \sum_{f=1}^{F} \alpha_f \psi_f(J_f, \mathbf{x}_c^i) - M \log T \\
& - \frac{M}{N} \sum_{i=1}^{N} \sum_{c \in Y_i} \sum_{f=1}^{F} \alpha_f \psi_f(J_f, \mathbf{x}_c^i),
\end{aligned} \tag{11}
$$

where parameters have the same denotations in the Section 2.1.

As we can expect to obtain a more accurate value of the log model likelihood than the one suggested by [1], 11 can be an indicator that determines whether a given set of parameters has higher likelihood in our learning algorithm. Furthermore, based on Eq.11, the partial derivative with respect to both filters J's and their corresponding weights α's are significantly simplified:

$$
\begin{aligned}
\frac{\partial \log P(D|\Theta)}{\partial J_k} = & \sum_{i=1}^{M} \sum_{c \in X_i} \frac{-\alpha_k \mathbf{x}_c^i < J_k, \mathbf{x}_c^i >}{(1 + \frac{1}{2} < J_k, \mathbf{x}_c^i >^2)} \\
& - \frac{M}{N} \sum_{i=1}^{N} \sum_{c \in Y_i} \frac{-\alpha_k \mathbf{x}_c^i < J_k, \mathbf{x}_c^i >}{(1 + \frac{1}{2} < J_k, \mathbf{x}_c^2 >^2)}.
\end{aligned} \tag{12}
$$

$$
\frac{\partial \log P(D|\Theta)}{\partial \alpha_k} = \sum_{i=1}^{M} \sum_{c \in X_i} \psi_c(J_k, \mathbf{x}_c^i) - \frac{M}{N} \sum_{i=1}^{N} \sum_{c \in Y_i} \psi_c(J_k, \mathbf{x}_c^i). \tag{13}
$$

To summarise, our learning algorithm is described as follows:

1. Initialise the filters (J's) by performing SVD over training images. The initial values of α's are randomly generated.
2. Update α's by applying a line search in the gradient direction given by Eq. 13. The step size μ_α is chosen such that the highest log–likelihood in Eq. 11 is reached.

$$
\alpha \leftarrow \alpha + \mu_\alpha \{ \frac{\partial}{\partial \alpha_i} \log P(D|\Theta) \} \tag{14}
$$

3. Update J's by applying a line search in the gradient direction given by Eq. 12. The step size μ_J is, again, chosen by maximising the log–likelihood in Eq. 11.

$$
J \leftarrow J + \mu_J \{ \frac{\partial}{\partial J_i} \log P(D|\Theta) \} \tag{15}
$$

4. Repeat steps 2–3 until the log–likelihood of the model dose not change.

Since the update step sizes (μ_α and μ_J) are very sensitive to the input parameters, it is quite difficult to specify it with any fixed value. In our implementation, we employ back–tracking line search to find the optimal solution in each update step [11].

3.3 Inference

After we get the MRF prior model, in order to perform inference (i.e. denoising in our experiments), we adopted a standard gradient based approach, as in McAuley et al [1]. Gradient ascent is a valid technique in the case of denoising, since the noisy image is 'close to' the original image, meaning that a local maximum is likely to be a global one [1]. In the denoising problem, the purpose is to infer the most likely correction for the image given the image prior and the noise model. The noise model assumed in our experiments, as in [1], is i.i.d. Gaussian: $P(\mathbf{y}|\mathbf{x}) \propto \prod_j \exp(-\frac{1}{2\sigma^2}(\mathbf{y}_j - \mathbf{x}_j)^2)$. Here, j ranges over all the pixels in the image, $\mathbf{y}_j$ denotes the real color value of the noisy image at pixel j, and $\mathbf{x}_i$ denotes the color to be estimated at pixel j; σ denotes the variance of Gaussian noise.

Combining the noise model and the MRF prior (Eq.1), the gradient of the log–posterior becomes [1]:

$$\nabla_{\mathbf{x}} \log P(\mathbf{x}|\mathbf{y}) = \sum_{f=1}^{F} \alpha_f J_f^- * \frac{(J_f * \mathbf{x})}{1 + \frac{1}{2}(J_f * \mathbf{x})^2} + \frac{\lambda}{\sigma^2}(\mathbf{y} - \mathbf{x}) \tag{16}$$

where $*$ denotes matrix convolution, and the algebraic operations above are performed in an elementwise fashion on the corresponding convolution matrix. J_f^- denotes the mirror image of J_f in two dimensions. λ is a critical parameter that gauges the relative importance of the prior and the image terms.

The updated image is then simply computed by

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \delta \frac{\partial}{\partial \mathbf{x}} \log P(\mathbf{x}|\mathbf{y}) \tag{17}$$

where δ is the step size of the gradient ascent. We found it is not sensitive to the inference result and can be selected empirically.

4 Experimental Results

In our experiments, to initialise our filters J's, we randomly selected 8,000 $3{\times}3{\times}3$ and $5 \times 5 \times 3$ patches, cropped from 200 images in the Berkeley Segmentation Database, and performed singular value decomposition (SVD) over their covariance matrices [12]. Thus, we obtained 27 and 75 filters for the two clique sizes, with 27 and 75 dimensions for each kind of filters respectively. α's were initialised to be a set of random values with the same dimension as the number of filters. There is no constraint on the scale of the initial α's, and they converge for both absolute and relative values after several steps of update. In the updating process, we randomly selected 2,000 training image patches and 2,000 random images patches from the same image database for each update step. The sizes of training/random image patches for $3{\times}3{\times}3$ and $5{\times}5{\times}3$ cliques are, respectively, $7 \times 7 \times 3$ and $13 \times 13 \times 3$.

In the inference, we do not need to eliminate the filter with highest variance since in our algorithm α's are normalised (with range of [0,1]). The least important filter will be ignored automatically in the denoising process because its corresponding weight will be zero. In the selection of λ, we used images other than those involving in our experiments. This was done by denoising a test image with several candidate λ–value, and selecting whichever one yields the best results [1]. The step size δ has been chosen to grow linearly with the noise level, which was found to work well in practice. The denoising performances are evaluated by PSNR ($10\log_{10}(\frac{255^2}{MSE})$, where $MSE = \frac{\sum_{i,j,k}^{I,J,K}(R_{i,j,k}-O_{i,j,k})^2}{IJK}$; R denotes restored image and O denotes original image). In Figure 2 we show results obtained for denoising an image in which a different amount of noise is applied to each of the three channels, and compare these with the state-of-the-art [1], simple bilateral filtering (using the MATLAB code from [13]), and Wavelet-based denoising [14]. In the bilateral filtering and Wavelet-based denoising experiments, the RGB test images were converted into YCbCr format before processing, which has less correlation between color channels. As we show by using different priors, the denoising performance of the priors learned by our algorithm has significant improvements comparing that of using McAuley et al.'s [1] and other two denoising approaches.

Tables 1 and 2 show results obtained for denoising 50 test color images (from the Berkeley Segmentation Database) in which all three channels have been equally corrupted, and compare our algorithm with McAuley et al.'s priors and other two well known methods in the $3 \times 3 \times 3$ and $5 \times 5 \times 3$ case, respectively. As results show, the performance of priors learned by the proposed algorithm for both model sizes is statistically significantly (paired T-test at the 0.05 level) superior to all others which use the same training/random image set. Furthermore, the performance of our priors learned from 2,000/2,000 training images/random image patches is comparable with the priors learned in [1], which used 100,000/50,000 training/random image patches.

Table 1. ($3 \times 3 \times 3$ window) Average denoising performance over 50 testing images. Results are measured in PSNR.

image/σ	**5**	**15**	**25**	**50**
Noisy image	34.10	24.70	20.15	14.16
McAuley et al^1	36.19	29.17	26.04	22.45
Our algorithm ^{1}	37.12*	29.78*	26.98*	23.38*
McAuley et al^2	36.83	29.74	26.69	23.15
Bilateral filtering	28.11	27.18	25.75	21.87
Wavelet-based denoising	36.11	28.99	25.98	22.41

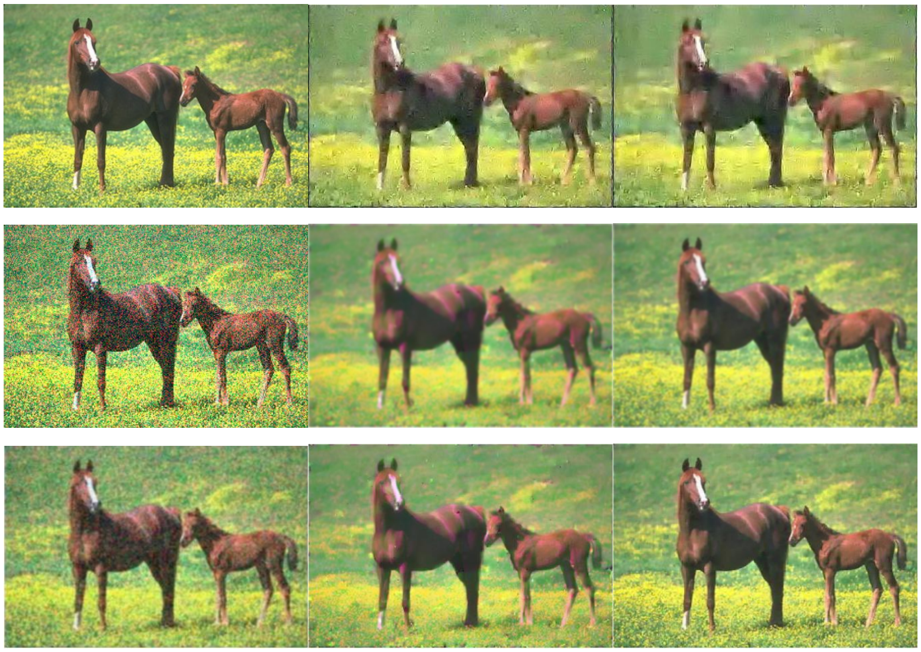

Fig. 2. The first column displays the original image (up), the noisy image (middle) with $\sigma = 75$ (red), 25 (green), 15 (blue) (PSNR=14.97) and the result of bilateral filtering 5×5 window (down, PSNR=23.55); the second column shows the result using Wavelet-based approach 3×3 window (up, PSNR=23.32), the results of McAuley et al's 3×3 prior (middle, PSNR=25.03) and our 3×3 prior (down, PSNR=25.90). the third column shows the result of 5×5 window Wavelet-based approach (up, PSNR=23.84), McAuley et al's 5×5 prior (25.99) and our 5×5 prior (down, PSNR=26.82).

5 Conclusion

In this paper, we have proposed the learning algorithm of high-order MRF prior models for color image denoising. By collecting a relatively small set of sample color image patches from a standard color image database, we have learned priors specified for color images using several steps of gradient ascent update with the rule of maximum likelihood. Results comparing the color prior models learned by our algorithm to a state–of–the-art color image prior model [1] performance improvements.

[1] indicates priors learned from 2,000/2,000 training/random pathes

[2] indicates priors learned from100,000/50,000 training/random patches

[3] $*$ indicates significant difference in performance compared with the upper one

Table 2. ($5 \times 5 \times 3$ window) Average denoising performance over 50 testing images. Results are measured in PSNR.

image/σ	**5**	**15**	**25**	**50**
Noisy image	34.10	24.70	20.15	14.16
McAuley et al^1	36.57	29.59	26.55	22.79
Our algorithm ^{1}	37.56*	30.11*	27.41*	23.69*
McAuley et al^2	37.28	30.08	27.16	23.41
Bilateral filtering	29.32	27.78	25.82	21.90
Wavelet-based denoising	36.41	29.50	26.32	22.46

References

1. McAuley, J., Caetano, T., Smola, A., Franz, M.: Learning high-order MRF priors of color images. In: ICML '06. (2006) 617–624
2. Freeman, W., Pasztor, E., Carmichael, O.: Learning low-level vision. International Journal of Computer Vision **40** (2000) 25–47
3. Roth, S., Black, M.: Fields of experts: A framework for learning image priors. In: ICCV. (2005) 860–867
4. Olshausen, B., Field, D.: Sparse coding with an overcomlete basis set: a strategy employed by v1? Vision Research **37** (1997) 3311–3325
5. Welling, M., Hinton, G., Osindero, S.: Learning sparse topographic representations with products of student-t distributions. In: NIPS 15. (2003) 1359–1366
6. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. PAMI **6(6)** (1984) 721–741
7. Hinton, G.: Products of experts. In: 9th ICANN. (1999) 1–6
8. Zhu, S., Wu, Y., Mumford, D.: Filters, random fields and maximum entropy (frame): Towards a unified theory of texture modeling. International Journal of Computer Vision **27** (1998) 107–126
9. Abramowitz, M., Stegun, I.E.: The process of the arithmetic–geometric mean. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, 9th printing (1972) 571
10. Krantz, S. In: Handbook of Complex Variables. Boston, MA:Brikhauser (1999) p.118
11. Moré, J., Thuente, D.: Line search algorithms with guaranteed sufficient decrease. ACM Trans. Math. Software **20** (1994) 286–307
12. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and ites application to evaluating segmentation algorithms and measuring ecological statistics. In: 8th ICCV. (2001) 416–423
13. (http://mesh.brown.edu/dlanman/photos/Bilateral)
14. Portilla, J., Strela, V., Wainwright, M., Simoncelli, E.: Image denoising using scale mixtures of gaussians in the wavelet domain. IEEE Trans. Image Processing. **12(11)** (2003) 1338–1351