Image reconstruction by matching gradient distributions

Anonymous CVPR submission

Paper ID 0090

Abstract

Most Bayesian deconvolution methods maximize the posterior probability to reconstruct a clean image from a degraded image. The MAP estimator, in conjunction with a sparse gradient image prior, favors piecewise smooth images and typically removes mid-frequency textures that are important for visual realism.

We present an alternative deconvolution method called iterative distribution reweighting (IDR) *that matches the restored image's gradient distribution to the desired gradient distribution. The desired gradient distribution is different from one image to another, but may also vary within an image, depending on the texture. We estimate the desired gradient distribution directly from the degraded input image for each segment. Our algorithm is able to reconstruct or hallucinate rich mid-frequency textures. A large scale user study supports the conclusion that our algorithm improves the visual realism of reconstructed images compared to those of MAP estimators.*

1. Introduction

Image restoration is typically an under-constrained problem. Information lost during a lossy observation process needs to be restored with prior information about natural images to achieve visual realism. Most Bayesian image restoration algorithms reconstruct images by maximizing the posterior probability, abbreviated MAP. Reconstructed images are called the MAP estimates.

One of the most popular image priors exploits the heavy-tailed characteristics of the gradient distribution, which are often parameterized using a mixture of Gaussians or a generalized Gaussian distribution. The MAP estimator balances the observation likelihood with the gradient penalty from the sparse gradient prior, reducing image deconvolution artifacts such as ringing and noise. Unfortunately, the MAP estimator also removes mid-frequency textures, often giving an unnatural and cartoonish look to the reconstructed image.

In this paper, we introduce an alternative image restoration strategy that is capable of reconstructing visually pleasing textures. The key idea is not to penalize gradients, but to match the reconstructed image's gradient distribution to the desired distribution. We introduce a method to estimate the desired distribution directly from the degraded input image. A user study substantiates the claim that images reconstructed by matching gradient distributions are visually more pleasing compared to those reconstructed using the MAP estimator.

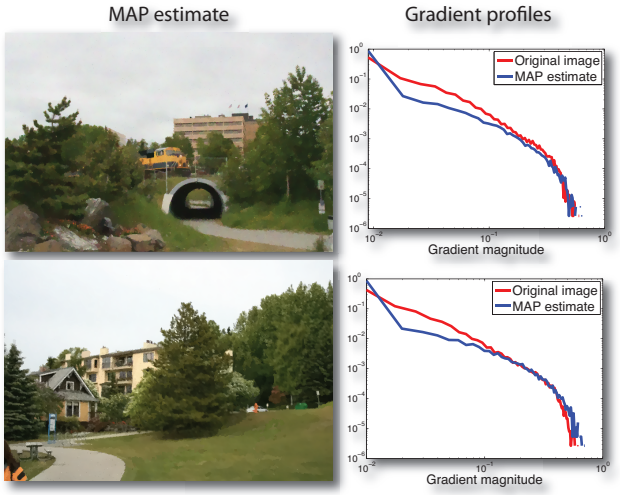

Figure 1. The gradient distribution of images reconstructed using the MAP estimator can be quite different from that of the original images. We present a method that matches the reconstructed image's gradient distribution to that of the desired gradient distribution (in this case, that of the original image) to hallucinate visually pleasing textures.

2. Related work

The Wiener filter [7] is a popular image reconstruction method with a closed form solution. The Wiener filter is a MAP estimator with a Gaussian prior on image gradients, which tends to blur edges and causes ringing around edges because those image gradients are not consistent with a Gaussian distribution.

Bouman and Sauer [1], Chan and Wong [2], and more recently Fergus *et al*. [4] and Levin *et al*. [13], use a heavy-tailed gradient prior such as a generalized Gaussian distribu-

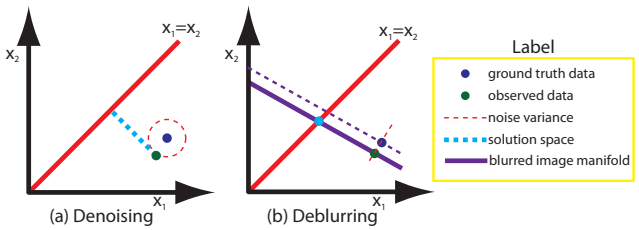

Figure 2. We illustrate the solution space of the MAP estimator for the special case of a two-pixel image (x_1, x_2). (a) In denoising, the MAP estimates lie on a line segment connecting the observed image (the green dot) and $x_1 = x_2$. The weighting term w in Eq 1 determines $\hat{x}$. (b) We can model a blurred image as a weighted average of pixels x_1, x_2. The MAP estimate is always on $x_1 = x_2$ regardless of w.

tion [1, 13], a total variation [2], or a mixture of Gaussians [4]. MAP estimators using sparse gradient priors preserve sharp edges while suppressing ringing and noise. However, they also tend to remove mid-frequency textures, which causes a mismatch between the reconstructed image's gradient distribution and that of the original image.

Matching gradient distributions has been addressed in the texture synthesis literature. Heeger and Bergen [8] synthesize textures by matching wavelet sub-band histograms to those of the desired texture. Portilla and Simoncelli [16] match joint statistics of wavelet coefficients to synthesize homogeneous textures. Kopf *et al*. [10] introduce a non-homogeneous texture synthesis technique by matching histograms of texels (or elements of textures).

Matching gradient distributions in image restoration is not entirely new. Li and Adelson [14] introduced a two-step image restoration algorithm that first reconstructs an image using an exemplar-based technique similar to Freeman *et al*. [6], and warps the reconstructed image's gradient distribution to the desired gradient distribution using Heeger and Bergen's method [8]. Recently, Woodford *et al*. [21] proposed a MAP estimation framework called the marginal probability field (MPF) that matches the histogram of low-level features, such as gradients or texels, for computer vision tasks including denoising. The technique in Woodford *et al*. [21] requires that we bin features to form a discrete histogram; we propose a distribution matching method that bypasses the binning process. Also, Woodford *et al*. used an image prior derived from a database of images, and used the same global prior to reconstruct images with different textures. We estimate our image priors directly from the degraded image and for each textured region independently.

3. Characteristics of MAP estimators

We study the solution space of MAP estimators, and show that MAP estimators favor piecewise smooth reconstructions in the absence of image information. Let y be the observed (degraded) image, k be the blur kernel, $\otimes$ be the convolution operator, and x be the latent image. The MAP estimator solves the following regularized problem:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \left\{ \frac{\|y - k \otimes x\|^2}{2\eta^2} + w \sum_m \rho(\nabla^m x) \right\} \quad (1)$$

where η^2 is the observation noise variance, m indexes gradient filters, and ρ is a robust function that favors sparse image gradients. We parameterize the gradient distribution using a generalized Gaussian distribution. In this case, $\rho(\nabla x) = -\ln(p(\nabla x))$, where $p(\nabla x)$ is given as follows:

$$p(\nabla x) = \frac{\gamma \lambda^{(\frac{1}{\gamma})}}{2\Gamma(\frac{1}{\gamma})} \exp(-\lambda \|\nabla x\|^{\gamma}) \quad (2)$$

Γ is a Gamma function, and shape parameters γ, λ determine the shape of the distribution. In most MAP-based image reconstruction algorithms, gradients are assumed to be independent for computational efficiency: $p(\nabla x) = \prod_{i=1}^{N} p(\nabla x_i)$, where i indexes pixels and N is the total number of pixels in an image.

To provide intuition, we graphically analyze the MAP estimator for the special case of a two-pixel image. Let us first consider the denoising task (Figure 2(a)). We observe a noisy version y (the green dot) of a sharp image x (the blue dot), and we want to reconstruct x from y. The observation term favors latent images that are close to y, but the image prior favors images that lie on the $x_1 = x_2$ axis. The weighting term balances the two competing "forces", and reconstructs an image that lies on the dotted blue line. Even if we optimally set w, we cannot remove the noise component parallel to the $x_1 = x_2$ axis.

Next, we consider the deblurring task (Figure 2(b)). Suppose the blur kernel is rank-deficient so that we effectively observe only one value $\tilde{y}$, which is a linear combination of the two pixel values x_1, x_2. Given the blur kernel, the set of x_1, x_2 that can regenerate the observation $\tilde{y}$ forms a linear manifold (the dotted purple line.) Any observation noise shifts the linear manifold orthogonal to the original manifold. Let y be the blurred noisy observation (the solid purple line.) The observation error is zero for x_1, x_2 that lie on this purple line, whereas the prior term favors images with lower gradients. Therefore, the MAP solution space is the blue dot on the $x_1 = x_2$ line, regardless of w. In other words, in the presence of a rank-deficient blur, the MAP estimator always favors a piecewise smooth image, and generates an image that has a different gradient distribution from the original image's gradient distribution (Figure 1).

These toy examples illustrate the discrepancy between the image prior and the MAP estimate. The image prior tells us that *collectively*, gradients in a natural image generate a sparse gradient profile, whereas the MAP estimator minimizes each gradient independently, favoring piecewise smooth images. This is unsatisfying for textured regions.

Therefore, we need a mechanism to ensure that the reconstructed image's gradient profile matches the gradient distribution insisted by the image prior.

4. Image reconstruction

We develop an image reconstruction algorithm that minimizes the distance between the reconstructed image's gradient distribution and the desired distribution. The gradient distribution penalty acts as a global image prior that steers the solution away from piecewise smooth images.

Let $q_E(\nabla x)$ be the empirical gradient distribution of the image x, and q_D be the distribution we want to match to. We measure the distance between distributions q_E, q_D using the Kullback-Leibler (KL) divergence:

$$KL(q_E||q_D) = \int_{\nabla x} q_E(\nabla x) \ln\left(\frac{q_E(\nabla x)}{q_D(\nabla x)}\right) d(\nabla x) \quad (3)$$

In Section 4.1, we first introduce a method that penalizes the KL distance between the empirical gradient distribution q_E and the desired gradient distribution q_D. We show that this algorithm is sensitive to the parameter setting; in the following section, we extend this result to a more stable algorithm called iterative distribution reweighting (IDR).

4.1. Penalizing the KL divergence

We can reconstruct an image $\hat{x}$ with a gradient distribution close to q_D by penalizing the KL divergence:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \frac{\|y - k \otimes x\|^2}{2\eta^2} + w_1\lambda_D\|\nabla x\|^{\gamma_D} + w_2 KL(q_E||q_D) \quad (4)$$

where w_2 determines how much to penalize the KL divergence. In the above equation, we have replaced the summation over multiple filters in Eq 1, i.e. $\sum_m \lambda_m\|\nabla x^m\|^{\gamma_m}$, with a single gradient filter to reduce clutter, but the derivation can easily be generalized to using multiple gradient filters. We use four gradient filters in this work: x, y gradient filters, x-y, and y-x diagonal gradient filters.

It's hard to directly optimize Eq 4 because the KL divergence is a non-linear function of the latent image x. Therefore we optimize Eq 4 iteratively.

We iterate the following procedures to solve Eq 4 :

$$\hat{x}^L = \quad (5)$$
$$\underset{x}{\operatorname{argmin}} \left\{ \frac{\|y - k \otimes x\|^2}{2\eta^2} + w_1\lambda_D\|\nabla x\|^{\gamma_D} + w_2\rho_G^L(\nabla x) \right\},$$
$$\rho_G^L(\nabla x) = \ln\left(\frac{{q_E}^{(L-1)}(\nabla x)}{q_D(\nabla x)}\right) \quad (6)$$

We show in the supplemental material that $\sum_i \rho_G^L(\nabla x_i)$ is one way of computing the KL divergence, where i indexes gradient samples.

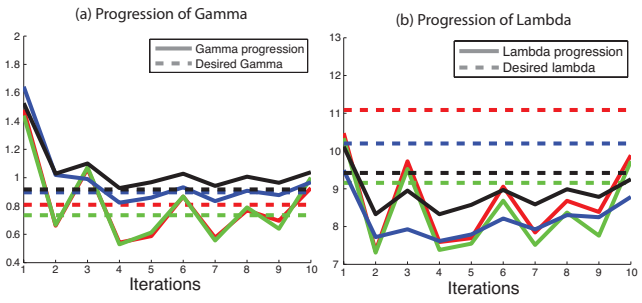

Figure 4. This figure shows how γ_E, λ_E change from one iteration to the next. Different colors correspond to different gradient filters. The algorithm in Eq 5 does not converge to a stable point, but rather oscillates around the desired solution.

In words, we fit the gradient distribution of the previous image estimate $\hat{x}^{L-1}$ to a generalized Gaussian distribution to get ${q_E}^{(L-1)}$, and compute ρ_G^L using ${q_E}^{(L-1)}$. Then we reconstruct a new image estimate $\hat{x}^L$ that minimizes Eq 4 assuming the penalty function ρ_G^L is fixed. Given the new estimate $\hat{x}^L$, we re-estimate the empirical distribution ${q_E}^L$ and iterate the process until convergence.

Intuitively, if q_E has more gradients of a certain magnitude compared to q_D, ρ_G penalizes those gradients more in the following iteration; if q_E has fewer gradients of a certain magnitude compared to q_D, ρ_G penalizes those gradients less in the following iteration. Figure 3 illustrates the procedure.

We estimate the shape parameters γ_E, λ_E of an empirical gradient distribution q_E by minimizing the log-likelihood of a generalized Gaussian distribution $p(\nabla x)$ (Eq 2):

$$[\gamma_E, \lambda_E] = \underset{\gamma,\lambda}{\operatorname{argmin}} \left\{ -\sum_{i=1}^{N} \frac{1}{N} \ln\left(p(\nabla x_i)\right) \right\} \quad (7)$$

where i indexes over gradient samples. This is equivalent to minimizing the KL divergence between the gradient samples ∇x and a generalized Gaussian distribution (see the supplemental material). We use the Nelder-Mead optimization method [11] to solve Eq 7. We sketch the algorithmic details of the energy minimization (Eq 5) in Appendix A; the full derivation is available in the supplemental material.

Algorithm analysis The behavior of the algorithm in Eq 5 depends on the value of w_2. When w_2 is small, the reconstructed image resembles the MAP estimate; when w_2 is large, the algorithm may oscillate around the desired solution (Figure 4). When w_2 is very large, the linearized objective function (in Appendix A, Eq 10) becomes indefinite, in which case the minimum residual method [18] cannot solve the linearized system. To mitigate the reliability issue, we develop an iterative distribution reweighting algorithm that is conceptually similar to penalizing the KL divergence .

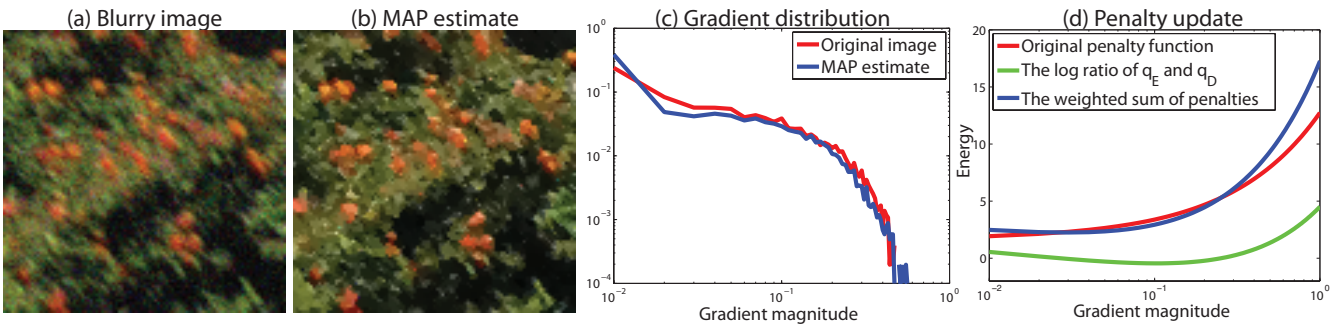

Figure 3. This figure illustrates the KL divergence regularization algorithm in Section 4.1. Suppose we deconvolve a degraded image in (a) using the MAP estimator. (c) shows that the x-gradient distribution of the deconvolved image in (b) does not match that of the original sharp image. Our algorithm adds the log ratio of q_E and q_D to the original gradient penalty such that the weighted sum of the two penalty terms encourages a better gradient match in the following iteration.

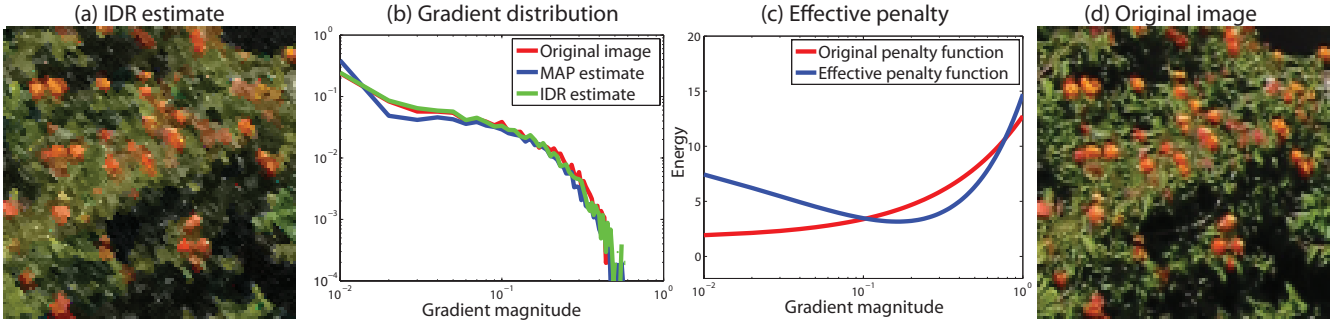

Figure 5. Our algorithm iteratively modifies the penalty function such that it encourages missing gradients and penalizes surplus gradients. (a) shows the deconvolved image using IDR, and (b) compares the gradient distribution of images reconstructed using the MAP estimator and IDR. (c) The effective penalty after convergence penalizes small gradients and large gradients more than mid-level gradients.

4.2. Iterative distribution reweighting (IDR)

Our objective is to find a regularization function ρ_G in Eq 5 such that minimizing the energy reconstructs the image with the desired gradient distribution. We propose an iterative strategy to search for the desired regularization function ρ_G. This technique is motivated by perceptron algorithms that iteratively adjust a decision boundary to minimize the classification error. In our case, we iteratively adjust the regularization function to match the empirical gradient distribution to the desired gradient distribution.

To do so, instead of penalizing just the KL distance as in the previous section, we penalize the sum of KL distances over previous iterations. In other words, we adjust the penalty ρ_G by the KL divergence:

$$\hat{x}^L = \tag{8}$$

$$\underset{x}{\operatorname{argmin}} \left\{ \frac{\|y - k \otimes x\|^2}{2\eta^2} + w_1 \lambda_D \|\nabla x\|^{\gamma_D} + w_2 \rho_G^L(\nabla x) \right\},$$

$$\rho_G^L(\nabla x) = \rho_G^{(L-1)}(\nabla x) + \ln\left(\frac{{q_E}^{(L-1)}(\nabla x)}{q_D(\nabla x)}\right) \tag{9}$$

Eq 9 iteratively adjusts the penalty function ρ_G by reweighting each gradient by the ratio of q_E and q_D, thus the name *iteratively distribution reweighting (IDR)*. The benefit of IDR is that it reaches convergence when $q_E = q_D$. We can easily modify the algorithmic details in Appendix A to solve Eq 8. We illustrate IDR in Figure 5, and show how γ_E, λ_E changes from one iteration to the next in Figure 6.

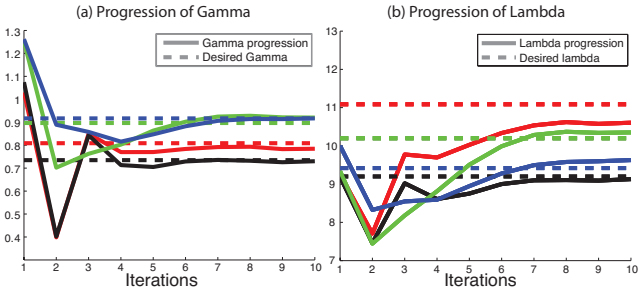

Figure 6. This figure shows how γ_E, λ_E estimates of gradient filters change from one iteration to the next. Different colors correspond to different gradient filters. We observe that the algorithm converges to a stable point in about 8 iterations.

In Figure 7, we test IDR for deblurring a single texture, assuming a known correct gradient distribution q_D. We synthetically blur the texture and add 5% noise to the blurred image. We deblur the image using the MAP estimator and IDR assuming a known blur kernel, and compare the reconstructed images. For all examples in this paper, we run 10 iterations of IDR, and we use $w_1 = 0.025, w_2 = 0.0025$.

The MAP estimator over-smooths textures whereas IDR reconstructs rich textures. The peak-signal-to-noise ratio (PSNR) / gray-scale SSIM [20] of our result may actually

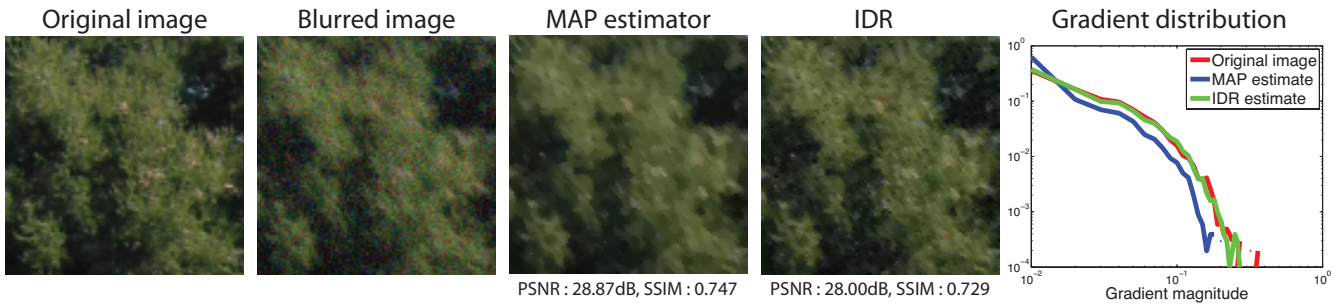

Figure 7. We compare the deblurring performance of the MAP estimator and IDR. IDR matches the gradient distribution of the original image better than the MAP estimator. IDR reconstructs visually more pleasing mid-frequency textures compared to the MAP estimator.

be lower than those of the MAP estimate. Visually, however, the texture reconstructed using our algorithm matches the original image better.

Algorithm analysis In most cases, IDR reliably reconstructs images with the desired gradient distribution. However, there are cases in which the algorithm settles at a local minimum away from the desired texture. This usually occurs when the support of the gradient filters is large and many gradient filters are used to regularize the image. Suppose we want to match the gradient histogram of a 3×3 Laplacian filter. The algorithm needs to update 9 pixels to change the filter response at the center pixel, but updating 9 pixels also affects filter responses of 8 neighboring pixels. Having to match multiple gradient distributions at the same time adds on another degree of complexity. To control the complexity, we match four two-tap gradient filters.

IDR matches a *parametrized* gradient distribution, and therefore the algorithm is inherently limited by the accuracy of the fit. In general, a generalized Gaussian is a good parameterization of natural image gradients.

The behavior of IDR is relatively insensitive to the weighting term w_2, but a large w_2 can destabilize the minimum residual algorithm [18] that solves the linearized system in Eq 10.

4.3. Desired distribution q_D estimation

We often do not know a priori what the desired gradient distribution q_D should be, so previous work estimated q_D from other natural images [4, 21] or hand-picked the parameters through trial and error [13]. We estimate q_D from the degraded input image directly.

To do so, we first deconvolve the degraded image using a MAP estimator using a hand-picked set of parameters, tuned to restore different textures reasonably well (i.e. a relatively small gradient penalty). In this paper, we set them as $[\gamma = 0.8, \lambda = 4, w_1 = 0.01]$ for all images. Then we down-sample the reconstructed image to reduce the deconvolution noise. We fit the gradients from the down-sampled image to a generalized Gaussian distribution to estimate the desired gradient distribution q_D. Fractal textures such as trees or piecewise smooth surfaces such as buildings exhibit scale-invariant gradient statistics [12, 15], so in theory the downsampling should not induce estimation error in such regions. While fine details in images can be lost through down-sampling, we observed empirically that the estimated gradient distribution q_D is accurate enough to reconstruct details in the final image.

IDR assumes that the texture is homogeneous (see the algorithm derivation in the supplemental material). When we have different textures within a single image, we segment the image using the EDISON algorithm [3] and deconvolve each region separately. We set the parameters (fixed for all images) such that each image generates about 20 segments. We illustrate the full deconvolution process in Figure 8.

5. Experiments

Deconvolution experiments We synthetically blur sharp images with the kernel in Figure 9, add 2% noise, and deconvolve them using competing methods. Figure 9 compares the deconvolution result of a MAP estimator, using a sparse gradient prior [13] and a two-color prior [9], and our result. We also present the deconvolution result of a MAP estimator using the estimated image prior for each segment. We name this prior an adjusted sparse prior.

IDR does not perform the best in terms of PSNR / SSIM. One explanation is that the degraded image may not strongly constrain the position of gradients, in which case our algorithm spreads gradients to match the gradient distribution, lowering PSNR / SSIM. Nevertheless, IDR reconstructs mid-frequency textures well and produces visually more compelling results compared to MAP estimators.

In Figure 10, we compare the denoising performance of IDR to that of the MPF [21] at two noise levels (their implementation only handles gray-scale, square images). Woodford *et al.* [21] estimate the desired gradient distribution from a database of images, and use the same prior to denoise different images. This can be problematic because different images have different gradient profiles, but the MPF enforces the same gradient profile on them. Also, the MPF does not adapt the image prior to the underlying texture, treating different textures the same way. Therefore, the MPF distributes gradients uniformly across the image, even in

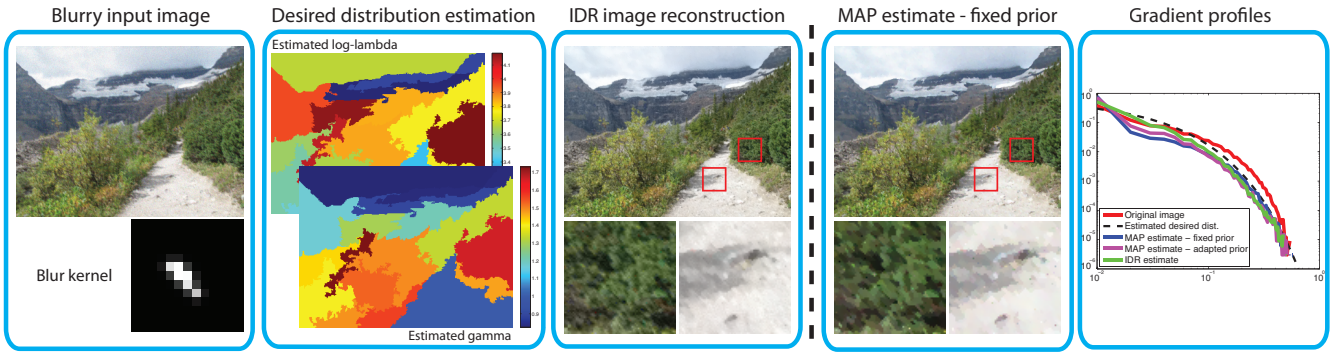

Figure 8. We illustrate the IDR deconvolution process. We first deblur the degraded image using the MAP estimator with a fixed prior. Down-sampled version of the deconvolved image is used to estimate the desired image prior q_D (i.e. the shape parameters γ, λ) in each segment, which is then used to deconvolve each segment separately using IDR. The red box denotes the cropped region. Compared to MAP estimators, IDR better preserves natural textures.

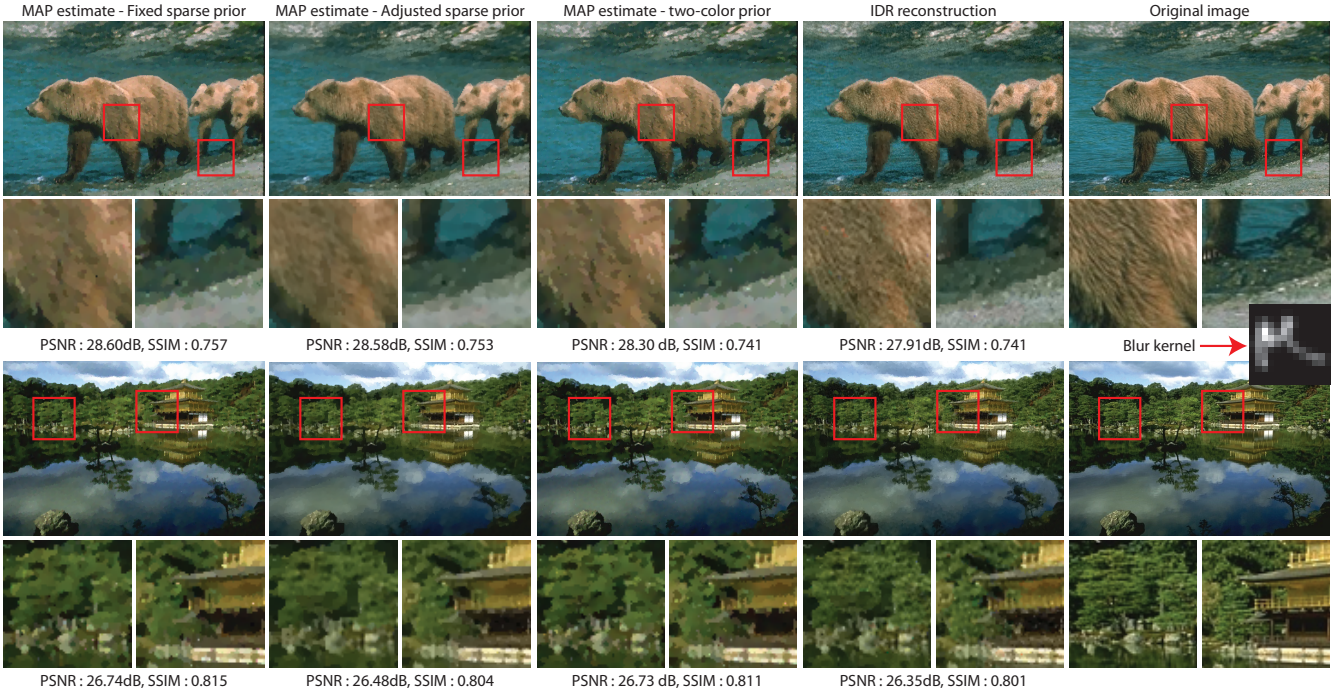

Figure 9. We compare the visual quality of deblurred images using IDR and the MAP estimators with the sparse gradient prior, the adjusted gradient prior and the two-color prior. We blur the sharp image using the kernel shown on the right, and add 2% noise to it. The red box denotes the cropped region. Although the PSNR and SSIM of our results are sometimes lower than those of MAP estimators, IDR reconstructs visually pleasing textures such as bear furs or trees surrounding the lake.

smooth regions, which can be visually disturbing. IDR addresses these issues by estimating the desired gradient prior from the input image, adapting to the spatially varying texture. Adapting gradient filters to local image structures using steerable filters [5, 17] may further improve the rendition of oriented textures such as hair and fur, but it is not considered in this work. The MPF quantizes 256 (8-bit) intensity levels to 64 intensity levels (6-bit) and bins gradient magnitudes to 11 slots for computational reasons, which reduces the contrast and accentuates spotty noise in reconstructed images. IDR adopts a continuous optimization scheme that does not require any histogram binning or intensity quantization.

Segmenting images to regions and deconvolving each region independently can sometimes generate artificial texture boundaries. We show such an example in Figure 11 where the artificial boundary between two segments is visually disturbing. While this rarely occurs, we could mitigate the artifacts using a texture-based segmentation algorithm rather than EDISON [3], which is a color-based segmentation algorithm.

User study IDR generates images with rich texture but with lower PSNR/SSIM compared to MAP estimates. To test our impression that images reconstructed by IDR are

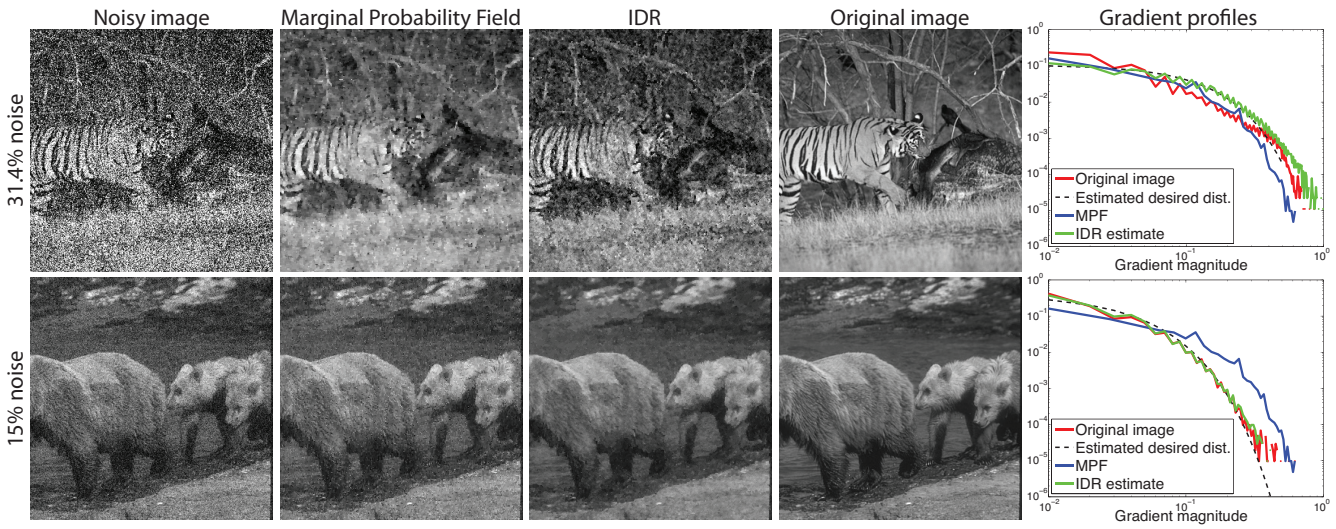

Figure 10. Comparing the denoising performance of IDR to the marginal probability field (MPF) [21]. IDR generates a better rendition of the spatially variant texture.

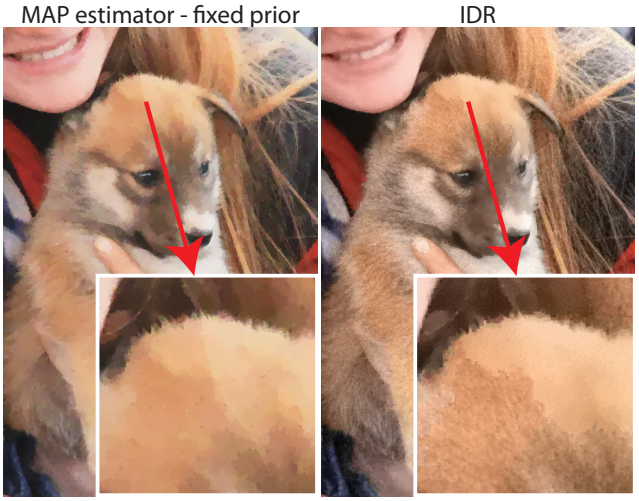

Figure 11. We may observe a disturbing artifact when the estimated prior is different in adjacent segments that have similar textures. While this rarely occurs, we could remove such artifacts using a texture segmentation algorithm instead of a color-based segmentation algorithm.

more visually pleasing, we performed a user study on Amazon Mechanical Turk.

We considered seven image degradation scenarios: noisy observations with $5\%, 10\%, 15\%$ noise, blurry observations with a small blur and $2\%, 5\%, 7\%$ noise, and a blurry observation with a moderate-size blur with 2% noise. For each degradation scenario, we randomly selected 4 images from a dataset of 13 images (roughly 700×500 pixels), and reconstructed images using the MAP estimator with a fixed sparse prior, an adjusted sparse prior, and IDR. A subset of reconstructed images are included in the supplemental material for visual comparisons.

We showed users two images side-by-side, one reconstructed using our algorithm and another reconstructed using one of the two MAP estimators. We asked users to select the image that is more visually pleasing, and give reasons for their choice. Users were also given a *"There is no difference."* option. We randomized the order in which we place the images side by side. To ensure that users can view both images on the same page while examining details, we cropped test images to 320×240. We selected the crop region directly from original sharp images *without* referring to reconstructed images, to minimize the selection bias.

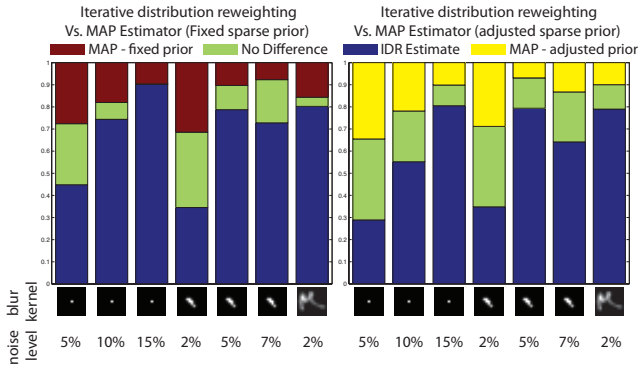

Figure 12. We conducted a user study to test our impression that IDR reconstructions are more visually pleasing than MAP estimates. The blue region corresponds to the fraction of users that favored IDR over MAP estimators. When the image degradation level is small, users did not show a particular preference, but as the image degradation level increases, users favored images reconstructed using IDR.

We collected at least 25 user inputs for each comparison, and averaged the user preference for each degradation scenario (Figure 12). When the degradation level is small (5% noise or a small blur with 2% noise), users did not prefer a particular algorithm. In such cases, the observation term is strong enough to reconstruct visually pleasing images regardless of the prior or the reconstruction algorithm.

When the degradation is large, however, many users clearly favored our results. User comments pointed out that realistic textures in trees, grass, and even in seemingly flat regions such as gravel paths are important for visual realism. Users who favored the MAP estimates preferred clean renditions of flat regions, and were not disturbed by piecewise smooth textures (some even found it artistic.) Another interesting observation is that individual users consistently favored either our result or the MAP estimates, suggesting that the image evaluation is subjective in nature. We show a subset of user comments in the supplemental material.

6. Conclusion

We have developed an iterative deconvolution algorithm that matches the gradient distribution in an energy minimization framework. Our deconvolution framework bridges the energy minimization framework for deconvolution and texture synthesis framework. We show through a user study that matching gradient distribution is beneficial for the perceptual quality of reconstructed images.

Appendix A

We sketch the details of the optimization procedures in Eq 5. Let Y be a rasterized vector of the observed image y, X be a rasterized vector of the image x, and K be the convolution matrix of the blur kernel k. We take the derivative of the optimization function in Eq 5 with respect to X:

$$
\begin{aligned}
&-\frac{K^T(Y-KX)}{\eta^2}+2w_1\lambda_D\gamma_D G^T\|GX\|^{\gamma_D-1}\\
&+w_2\Big(2\lambda_D\gamma_D G^T\|GX\|^{\gamma_D-1}-2\lambda_E\gamma_E G^T\|GX\|^{\gamma_E-1}\\
&+\left(\frac{1}{\gamma_E}-\frac{\ln(\lambda_E)}{{\gamma_E}^2}+\frac{\Psi\left(\frac{1}{\gamma_E}\right)}{{\gamma_E}^2}-\lambda_E|GX|^{\gamma_E}\ln(|GX|)\right)\circ\frac{\partial\gamma_E}{\partial X}\\
&+\left(\frac{1}{\gamma_E\lambda_E}-|GX|^{\gamma_E}\right)\circ\frac{\partial\lambda_E}{\partial X}\Big)=0
\end{aligned}
\tag{10}
$$

where G is a gradient operator, and $\circ$ is a Hadamard element-wise matrix multiplication operator. $\frac{\partial\gamma_E}{\partial X}$ and $\frac{\partial\lambda_E}{\partial X}$ can be derived as follows:

$$
\begin{aligned}
\frac{\partial\gamma_E}{\partial X}&=\frac{{\gamma_E}^2\lambda_E{}^{(\frac{2}{\gamma_E})}\Gamma(\frac{1}{\gamma_E})}{N\Gamma(\frac{3}{\gamma_E})\left(\Psi(\frac{1}{\gamma_E})-3\Psi(\frac{3}{\gamma_E})+2\ln(\lambda_E)\right)}2G^TGX\\
\frac{\partial\lambda_E}{\partial X}&=-\frac{\Gamma(1/\gamma_E)\gamma_E\lambda_E{}^{(1+2/\gamma_E)}}{N\Gamma(3/\gamma_E)}G^TGX
\end{aligned}
\tag{11}
$$

N is the dimension of X.

We solve Eq 10 by iteratively approximating it with a linear equation [13, 19]. We use a minimum residual method [18] to solve the *linearized* system in Eq 10. The full derivation is available in the supplemental material.

References

[1] C. A. Bouman and K. Sauer. A generalized Gaussian image model for edge-preserving MAP estimation. *IEEE TIP*, 2(3):296 – 310, Mar. 1993. 1, 2

[2] T. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE TIP*, 7(3):370 – 375, Mar. 1998. 1, 2

[3] C. M. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In *IEEE ICPR*, 2002. 5, 6

[4] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM TOG (Proc. SIGGRAPH)*, 2006. 1, 2, 5

[5] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE TPAMI*, 1991. 6

[6] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *IJCV*, 40(1):25 – 47, 2000. 2

[7] Gonzalez and Woods. *Digital image processing*. Prentice Hall, 2008. 1

[8] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *ACM SIGGRAPH*, 1995. 2

[9] N. Joshi, C. L. Zitnick, R. Szeliski, and D. Kriegman. Image deblurring and denoising using color priors. In *IEEE CVPR*, 2009. 5

[10] J. Kopf, C.-W. Fu, D. Cohen-Or, O. Deussen, D. Lischinski, and T.-T. Wong. Solid texture synthesis from 2D exemplars. *ACM TOG (Proc. SIGGRAPH)*, 26(3), 2007. 2

[11] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal of Optimization*, 1998. 3

[12] A. B. Lee, D. Mumford, and J. Huang. Occlusion models for natural images: a statistical study of a scale-invariant dead leaves model. *IJCV*, 41:35–59, 2001. 5

[13] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM TOG (Proc. SIGGRAPH)*, 2007. 1, 2, 5, 8

[14] Y. Li and E. H. Adelson. Image mapping using local and global statistics. In *SPIE EI*, 2008. 2

[15] G. Matheron. *Random Sets and Integral Geometry*. John Wiley and Sons, 1975. 5

[16] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *IJCV*, 40(1):49 – 71, Oct. 2000. 2

[17] S. Roth and M. J. Black. Steerable random fields. In *IEEE ICCV*, 2007. 6

[18] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM JSSC*, 1986. 3, 5, 8

[19] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Reviews*, 41(3):513 – 537, Sept. 1999. 8

[20] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004. 4

[21] O. J. Woodford, C. Rother, and V. Kolmogorov. A global perspective on MAP inference for low-level vision. In *IEEE ICCV*, 2009. 2, 5, 7