

# Image restoration by matching gradient distributions

Taeg Sang Cho, *Student Member, IEEE*, C. Lawrence Zitnick, *Member, IEEE*, Neel Joshi, Sing Bing Kang, *Senior Member, IEEE*, Richard Szeliski, *Fellow, IEEE*, and William. T. Freeman, *Fellow, IEEE*,

**Abstract**—A common image restoration method is to use a MAP estimator, which maximizes a posterior probability to reconstruct a clean image from a degraded image. A MAP estimator, when used with a sparse gradient image prior, reconstructs piecewise smooth images and typically removes textures that are important for visual realism. We present an alternative deconvolution method called *iterative distribution reweighting (IDR)* which imposes a global constraint on gradients so that a reconstructed image should have a gradient distribution similar to a reference distribution. In natural images, a reference distribution not only varies from one image to another, but also within an image depending on texture. We estimate a reference distribution directly from an input image for each texture segment. Our algorithm is able to restore 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.

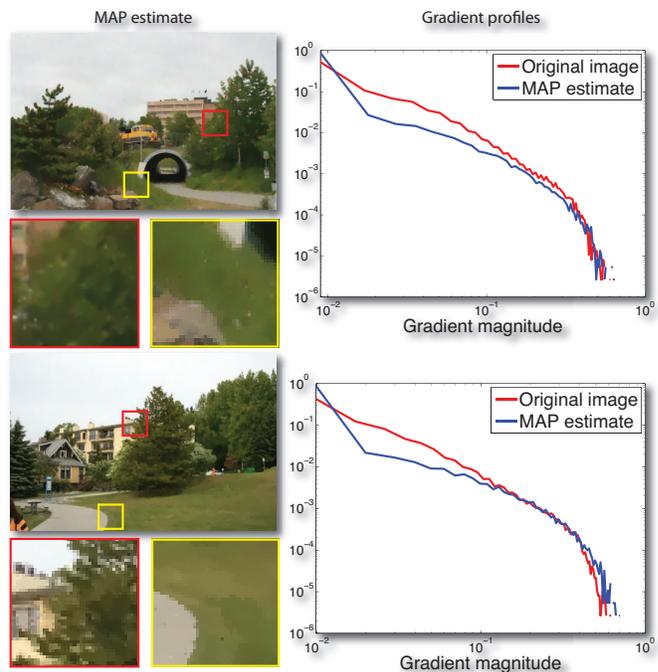
**Index Terms**—Non-blind deconvolution, image prior, image deblurring, image denoising

## 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.



**Fig. 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 [9] 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

• T. S. Cho and W. T. Freeman are with Computer Science and Artificial Intelligence Lab (CSAIL), Massachusetts Institute of Technology, Cambridge, MA 02139.

• C. L. Zitnick, N. Joshi, S. B. Kang and R. Szeliski are with Microsoft Research, Redmond, WA 98004

those image gradients are not consistent with a Gaussian distribution.

Bouman and Sauer [1], Chan and Wong [2], and more recently Fergus *et al.* [6] and Levin *et al.* [15], use a heavy-tailed gradient prior such as a generalized Gaussian distribution [1], [15], a total variation [2], or a mixture of Gaussians [6]. 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 [11] synthesize textures by matching wavelet sub-band histograms to those of the desired texture. Portilla and Simoncelli [18] match joint statistics of wavelet coefficients to synthesize homogeneous textures. Kopf *et al.* [13] 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 [16] introduce a two-step image restoration algorithm that first reconstructs an image using an exemplar-based technique similar to Freeman *et al.* [8], and warp the reconstructed image's gradient distribution to a reference gradient distribution using Heeger and Bergen's method [11]. Woodford *et al.* [24] propose a MAP estimation framework called a marginal probability field (MPF) that matches a histogram of low-level features, such as gradients or texels, for computer vision tasks including denoising. MPF requires that one bins features to form a discrete histogram; we propose a distribution matching method that by-passes this binning process. Also, Woodford *et al.* [24] use an image prior estimated from a database of images and use the same global prior to reconstruct images with different textures. In contrast, we estimate the image prior directly from the degraded image for each textured region. Schmidt *et al.* [21] match the gradient distribution through sampling. As with Woodford *et al.* [24], Schmidt *et al.* also use a single global prior to reconstruct images with different textures, which causes noisy renditions in smooth regions. HaCohen *et al.* [10] explicitly integrate texture synthesis to image restoration, specifically for an image up-sampling problem. To restore textures, they segment a degraded image and replace each texture segment with textures in a database of images.

### 3 CHARACTERISTICS OF MAP ESTIMATORS

In this section, we illustrate why MAP estimators with a sparse prior recover unrealistic, piecewise smooth renditions as illustrated in Figure 1. Let  $B$  be a degraded image,  $k$  be a blur kernel,  $\otimes$  be a convolution operator, and  $I$  be a latent image. A MAP estimator solves the following regularized problem:

$$\hat{I} = \operatorname{argmin}_I \left\{ \frac{\|B - k \otimes I\|^2}{2\eta^2} + w \sum_m \rho(\nabla_m I) \right\} \quad (1)$$

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

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

$\Gamma$  is a Gamma function and shape parameters  $\gamma, \lambda$  determine the shape of the distribution. In most MAP-based image reconstruction algorithms, gradients are assumed to be independent for computational efficiency:  $p(\nabla I; \gamma, \lambda) = \frac{1}{Z} \prod_{i=1}^N p(\nabla I_i; \gamma, \lambda)$ , where  $i$  is a pixel index,  $Z$  is a partition function, and  $N$  is the total number of pixels in an image.

A MAP estimator balances two competing forces: the reconstructed image  $\hat{I}$  should satisfy the observation model while conforming to the image prior. Counter-intuitively, the image prior term, assuming independence among gradients, *always* favors a flat image to any other images, even a natural image. Therefore, the more the MAP estimator relies on the image prior term, which is often the case when the image degradation is severe, the more the reconstructed image becomes piecewise smooth.

One way to explain this property is that the independence among local gradients fails to capture the global statistics of gradients for the whole image. The image prior tells us that gradients in a natural image *collectively* exhibit a sparse gradient profile, whereas the independence assumption of gradients forces us to minimize each gradient *independently*, always favoring a flat image. Nikolova [17] provides a theoretic treatment of MAP estimators in general to show its deficiency.

We could remove the independence assumption and impose a joint prior on all gradients, but this approach is computationally expensive. This paper introduces an alternative method to impose a global constraint on gradients – that a reconstructed image should have a gradient distribution similar to a reference distribution.

## 4 IMAGE RECONSTRUCTION

In this section, we develop an image reconstruction algorithm that minimizes the KL divergence between the reconstructed image's gradient distribution and its reference distribution. This distance penalty plays the role of a global image prior that steers the solution away from piecewise smooth images.

Let  $q_E(\nabla I)$  be an empirical gradient distribution of an image  $I$ , and  $q_R$  be a reference distribution. We measure the distance between distributions  $q_E$  and  $q_R$  using the Kullback-Leibler (KL) divergence:

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

An empirical distribution  $q_E$  is parameterized using a generalized Gaussian distribution  $p(\nabla I; \gamma, \lambda)$  (Eq. 2). Given gradient

**Algorithm 1** MAP with KL penalty

---

```

% Initial image estimate to start iterative minimization
 $\hat{I}^0 = \operatorname{argmin}_I \left\{ \frac{\|B - k \otimes I\|^2}{2\eta^2} + w_1 \lambda_D \|\nabla I\|^{\gamma_D} \right\}$ 
Update  $q_E^0$  using Eq. 4
% Iterative minimization
for  $l = 1 \dots 10$  do
  % KL distance penalty term update
   $\rho_G^l(\nabla I) = \frac{1}{N} \ln \left( \frac{q_E^{(l-1)}(\nabla I)}{q_R(\nabla I)} \right)$ 
  % Image reconstruction
   $\hat{I}^l = \operatorname{argmin}_I \left\{ \frac{\|B - k \otimes I\|^2}{2\eta^2} + w_1 \lambda_D \|\nabla I\|^{\gamma_D} + w_2 \rho_G^l(\nabla I) \right\}$ 
  Update  $q_E^l$  using Eq. 4
end for
 $\hat{I} = \hat{I}^{10}$ 

```

---

samples,  $\nabla I_i$ , where  $i$  indexes samples, we estimate the shape parameters  $\gamma_E, \lambda_E$  of an empirical gradient distribution  $q_E$  by minimizing the log-likelihood:

$$[\gamma_E, \lambda_E] = \operatorname{argmin}_{\gamma, \lambda} \left\{ - \sum_{i=1}^N \frac{1}{N} \ln(p(\nabla I_i; \gamma, \lambda)) \right\} \quad (4)$$

This is equivalent to minimizing the KL divergence between gradient samples  $\nabla I$  and a generalized Gaussian distribution. We use the Nelder-Mead optimization method [14] to solve Eq. 4.

#### 4.1 Failure of penalizing KL divergence directly

To motivate our algorithm in Section 4.2, we first introduce a method that penalizes the KL divergence between an empirical gradient distribution  $q_E$  and a reference distribution  $q_R$ . We show that the performance of this algorithm is sensitive to the parameter setting and that the algorithm may not always converge. In Section 4.2, we extend this algorithm to a more stable algorithm called Iterative Distribution Reweighting (IDR).

We augment the MAP estimator (Eq. 1) with KL divergence:

$$\hat{I} = \operatorname{argmin}_I \left\{ \frac{\|B - k \otimes I\|^2}{2\eta^2} + w_1 \lambda_D \|\nabla I\|^{\gamma_D} + w_2 KL(q_E || q_R) \right\} \quad (5)$$

where  $w_2$  determines how much to penalize the KL divergence.<sup>1</sup> It's hard to directly solve Eq. 5 because the KL divergence is a non-linear function of a latent image  $I$ . Therefore we solve Eq. 5 iteratively.

Algorithm 1, shown in pseudocode, solves Eq. 5 iteratively. We can describe Algorithm 1 qualitatively as follows: if  $q_E$  has more gradients of a certain magnitude than  $q_R$ ,  $\rho_G$  penalizes those gradients *more* in the following iteration; if  $q_E$  has fewer gradients of a certain magnitude than  $q_R$ ,  $\rho_G$  penalizes those gradients *less* in the following iteration. Therefore, at each

1. In Eq. 5, we have replaced the summation over multiple filters in Eq. 1, i.e.  $\sum_m \lambda_m \|\nabla_m I\|^{\gamma_m}$ , with a single derivative filter to reduce clutter, but the derivation can easily be generalized to using multiple derivative filters. We use four derivative filters in this work: x, y derivative filters and x-y, and y-x diagonal derivative filters.

iteration, the solution will move in the ‘‘correct’’ direction. Figure 2 illustrates the procedure. The full derivation of the algorithm details is available in Appendix A.

We can show that the penalty function  $\rho_G$  in Algorithm 1 is one way to evaluate the KL divergence between the empirical distribution  $q_E$  and the reference distribution  $q_R$ .

*Proposition 1:* Let  $q_E$  be a parametric distribution of samples  $x_i, i = 1 \dots N$  and let  $q_R$  be a fixed parametric distribution. Then we can represent the KL divergence between samples  $q_E$  and  $q_R$  as follows:

$$KL(q_E || q_R) = \sum_i^N \rho_G(x_i) = \sum_i^N \left\{ \frac{1}{N} \ln \left( \frac{q_E(x_i)}{q_R(x_i)} \right) \right\} \quad (6)$$

*proof:* The KL divergence between  $q_E$  and  $q_R$  is defined as follows:

$$KL(q_E || q_R) = \int_z q_E(z) \ln \left( \frac{q_E(z)}{q_R(z)} \right) dz \quad (7)$$

There are different ways to represent the parametric distribution  $q_E$ . We can parameterize the distribution of samples using a generalized Gaussian distribution as follows:

$$q_E(z) = \frac{\gamma_E \lambda_E \left( \frac{1}{\gamma_E} \right)}{2\Gamma\left(\frac{1}{\gamma_E}\right)} \exp(-\lambda_E \|z\|^{\gamma_E}) \quad (8)$$

where the shape parameters  $\gamma_E, \lambda_E$  are fitted to samples  $x_i$  using Eq. 4. We can also parameterize the distribution of samples  $x_i$  as follows:

$$\tilde{q}_E(z) = \frac{1}{N} \sum_i^N \delta(z - x_i) \quad (9)$$

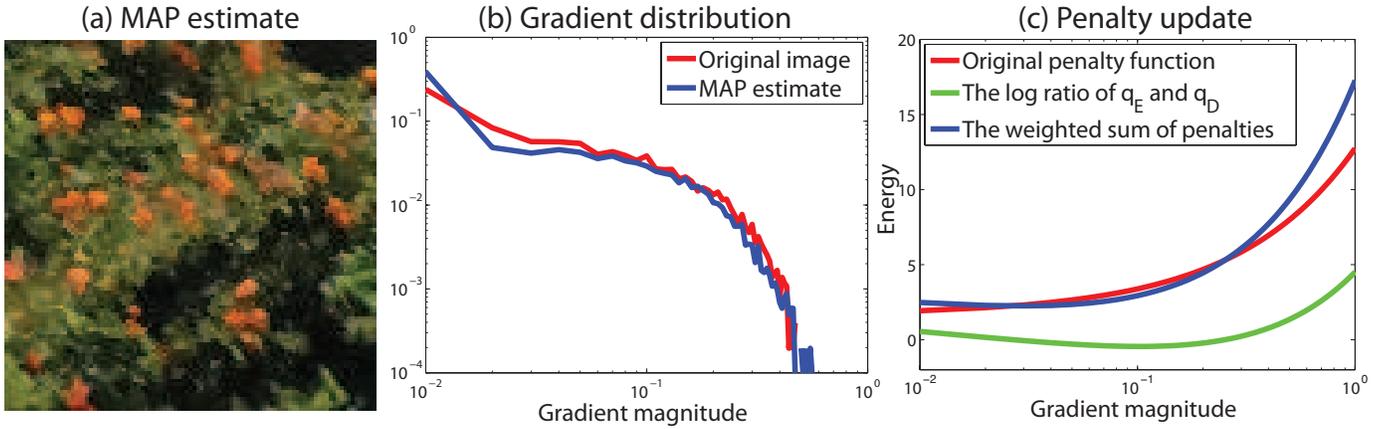
Therefore,

$$\begin{aligned} KL(q_E || q_R) &= \int_z q_E(z) \ln \left( \frac{q_E(z)}{q_R(z)} \right) dz \\ &= \int_z \tilde{q}_E(z) \ln \left( \frac{q_E(z)}{q_R(z)} \right) dz \\ &= \sum_i^N \left\{ \frac{1}{N} \ln \left( \frac{q_E(x_i)}{q_R(x_i)} \right) \right\} \\ &= \sum_i^N \rho_G(x_i) \end{aligned} \quad (10)$$

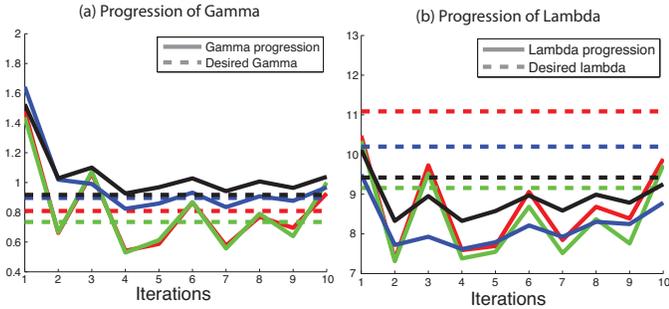
□

##### 4.1.1 Algorithm analysis

The behavior of Algorithm 1 depends on the value of  $w_2$ . When  $w_2$  is small, the reconstructed image is similar to the MAP estimate. On the other hand, when  $w_2$  is large, the algorithm oscillates around the desired solution (Figure 3): the algorithm ‘‘ping-pong’s’’ between a noisy solution and a piecewise smooth solution. For instance, suppose the current image estimate is piecewise smooth. The algorithm would then encourage more pixels with larger derivatives in the next iteration, which makes the subsequent solution noisier. In the



**Fig. 2:** This figure illustrates Algorithm 1. Suppose we deconvolve a degraded image using a MAP estimator. (b) shows that the  $x$ -gradient distribution of the MAP estimate in (a) does not match that of the original image. (c) Our algorithm adds the log ratio of  $q_E$  and  $q_R$  to the original penalty (i.e.  $\lambda_D \|\nabla I\|^{\gamma_D}$ ) such that the weighted sum of the two penalty terms encourages a better distribution match in the following iteration.



**Fig. 3:** We illustrate the operation of Algorithm 1 in terms of the  $\gamma_E, \lambda_E$  progressions. Different colors correspond to different gradient filters. Oftentimes, Algorithm 1 does not converge to a stable point, but oscillates around the desired solution.

following iteration, to reduce the derivative magnitudes to smooth noise, the algorithm penalizes gradients more severely to better match the reference distribution, in which case the image becomes piecewise smooth again, exhibiting an oscillatory behavior. In fact, when  $w_2$  is very large, the linearized system (in Appendix A, Eq. 11) becomes indefinite, in which case the minimum residual method [20] cannot be used to solve the linearized system. To mitigate the reliability issue and to damp possible oscillations around the desired solution, we develop an iterative distribution reweighting algorithm.

## 4.2 The iterative distribution reweighting (IDR)

We extend Algorithm 1 to reduce oscillations around the correct solution and to reduce sensitivity to parameter values. We achieve this by modifying the regularization function  $\rho_G$  in Algorithm 1. Our technique is motivated by perceptron algorithms [5] that iteratively adjust a decision boundary to minimize classification error. In our case, we iteratively adjust the regularization function to match the empirical gradient distribution to the reference gradient distribution.

---

### Algorithm 2 The iterative distribution reweighting (IDR)

---

```

% Initial image estimate to start iterative minimization
 $\hat{I}^0 = \operatorname{argmin}_I \left\{ \frac{\|B - k \otimes I\|^2}{2\eta^2} + w_1 \lambda_D \|\nabla I\|^{\gamma_D} \right\}$ 
Update  $q_E^0$  using Eq. 4
% Iterative minimization
for  $l = 1 \dots 10$  do
    % Accumulating the KL divergence
     $\rho_G^l(\nabla I) = \rho_G^{(l-1)}(\nabla I) + \frac{1}{N} \ln \left( \frac{q_E^{(l-1)}(\nabla I)}{q_R(\nabla I)} \right)$ 
    % Image reconstruction
     $\hat{I}^l = \operatorname{argmin}_I \left\{ \frac{\|B - k \otimes I\|^2}{2\eta^2} + w_1 \lambda_D \|\nabla I\|^{\gamma_D} + w_2 \rho_G^l(\nabla I) \right\}$ 
    Update  $q_E^l$  using Eq. 4
end for
 $\hat{I} = \hat{I}^{10}$ 
    
```

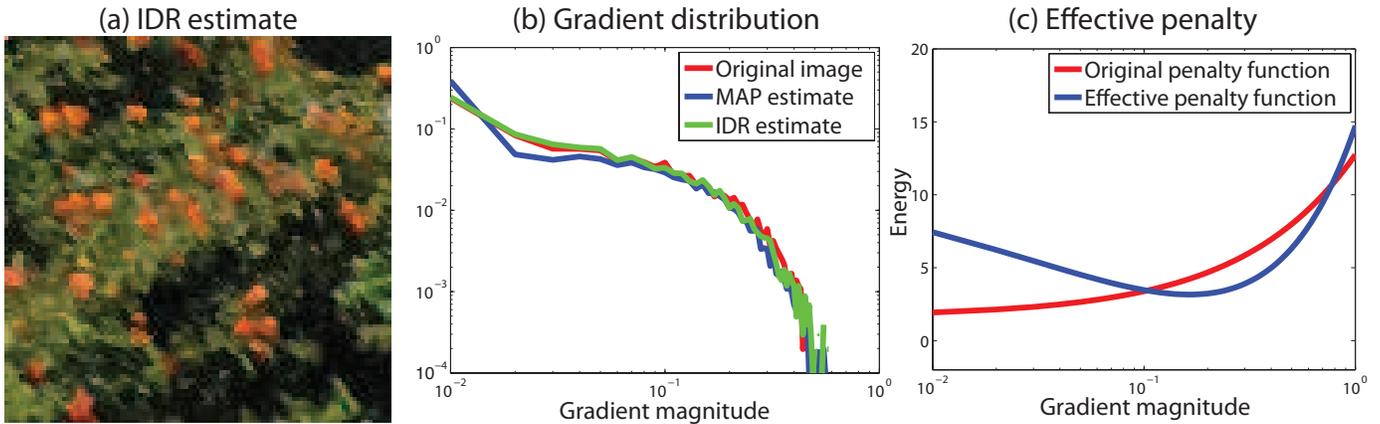
---

To do so, instead of using the KL divergence as a regularization term  $\rho_G$  as in Algorithm 1, we set  $\rho_G$  as the *sum* of the KL divergences over previous iterations. Algorithm 2 shows the pseudocode for IDR.

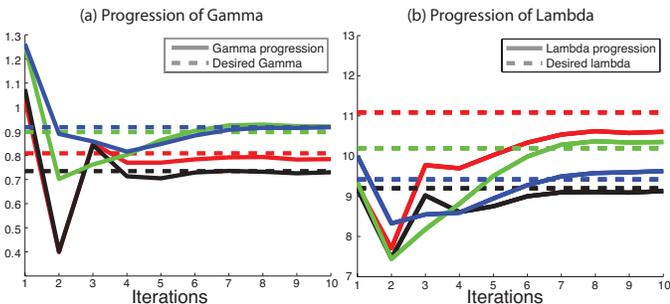
IDR iteratively adjusts the penalty function  $\rho_G$  by the ratio of distributions  $q_E$  and  $q_R$  using the formulation of KL divergence from Eq. 6, thus the name the *iterative distribution reweighting (IDR)*. If the ratio between the empirical and reference distributions is large at a given iteration, the gradients are penalized more heavily at the next iteration, and less if the ratio is small. As shown in the pseudocode for IDR in Algorithm 2, our new regularization term  $\rho_G$  is set to the sum of the KL divergences computed from Eq. 6 over previous iterations:

$$\rho_G^l(\nabla I) = \rho_G^{(l-1)}(\nabla I) + \frac{1}{N} \ln \left( \frac{q_E^{(l-1)}(\nabla I)}{q_R(\nabla I)} \right) \quad (11)$$

The benefit of IDR is that it reaches convergence when



**Fig. 4:** The IDR deconvolution result. (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 (i.e.  $w_1 \lambda_D \|\nabla I\|^{y_D} + w_2 \sum_{l=1}^{10} \frac{1}{N} \ln \left( \frac{q_E(\nabla I)}{q_R(\nabla I)} \right)$ ) penalizes gradients with small and large magnitude more than gradients with moderate magnitude.



**Fig. 5:** This figure shows how the  $\gamma_E, \lambda_E$  progress 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.

$q_E = q_R$ .<sup>2</sup> We can also view the  $\rho_G$  update equation as damping the KL divergence with the sum of previous KL divergences, thereby smoothing oscillations. We can easily modify derivations in Appendix A to derive details for Algorithm 2. We illustrate the operation of IDR in Figure 4, and show how  $\gamma_E, \lambda_E$  changes from one iteration to the next in Figure 5. Observe that  $\gamma_E, \lambda_E$  no longer oscillate as in Figure 3.

In Figure 6, we test IDR for deblurring a single texture, assuming that the reference distribution  $q_R$  is known a priori. We synthetically blur the texture using the blur kernel shown in Figure 8 and add 5% Gaussian noise to the blurred image. We deblur the image using a MAP estimator and using IDR, and compare the reconstructions. For all examples in this paper, we use  $w_1 = 0.025, w_2 = 0.0025$ . We observe that the gradient distribution of the IDR estimate matches the reference distribution better than that of the MAP estimate, and visually, the texture of the IDR estimate better matches the original image's texture. Although visually superior, the peak signal-to-noise ratio (PSNR) / gray-scale SSIM [23] of the IDR estimate are lower than those of the MAP estimate. This occurs because

2. This statement does not mean that the algorithm will converge only if  $q_E = q_R$ ; the algorithm can converge to a local minimum.

IDR may not place the gradients at exactly the right position. Degraded images do not strongly constrain the position of gradients, in which case our algorithm disperses gradients to match the gradient distribution, which would lower the PSNR / SSIM.

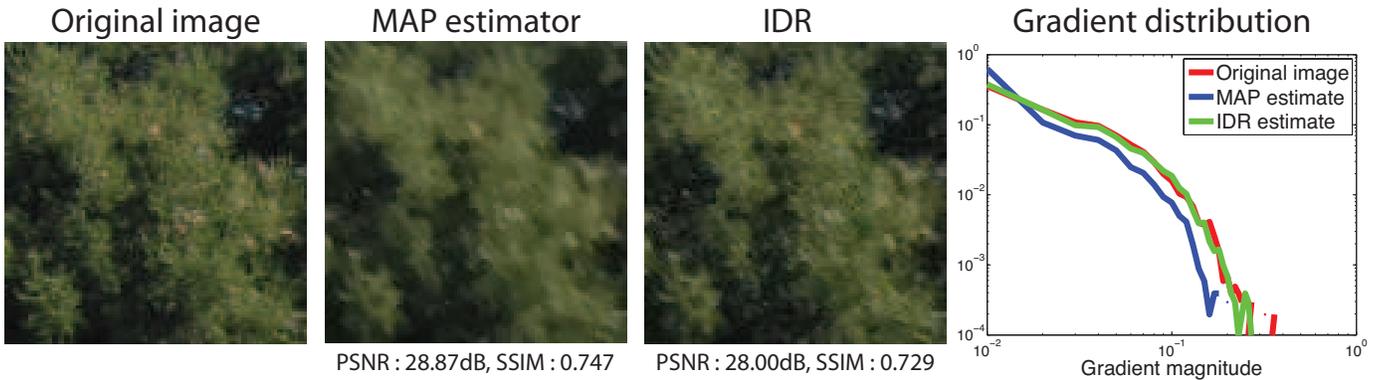
#### 4.2.1 Algorithm analysis

IDR matches a *parametrized* gradient distribution, and therefore the algorithm is inherently limited by the accuracy of the fit. The behavior of IDR is relatively insensitive to the weighting term  $w_2$ , but a large  $w_2$  can destabilize the minimum residual algorithm [20] that solves the linearized system in Eq. 11.

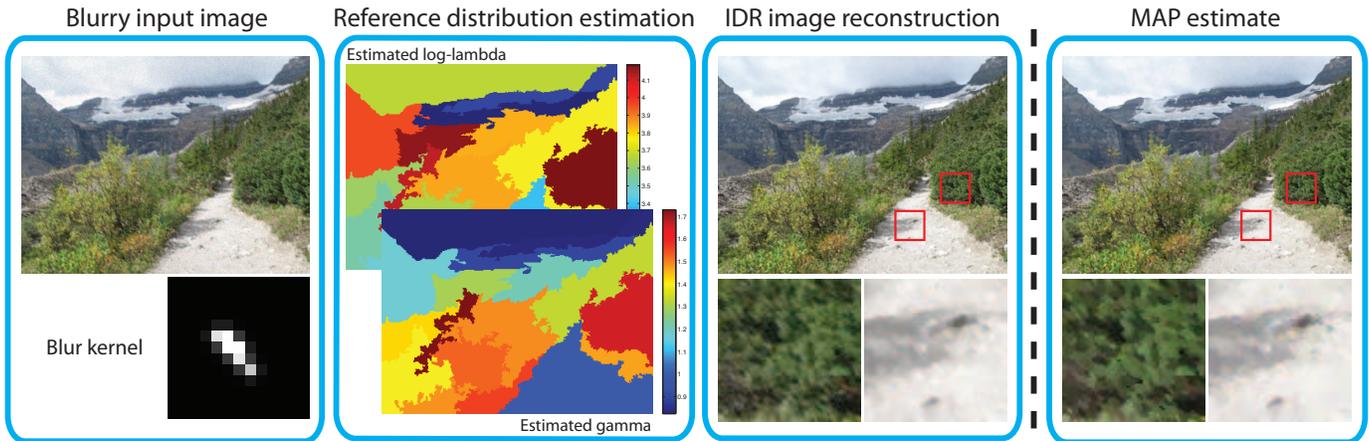
In most cases, IDR reliably reconstructs images with the reference gradient distribution. However, there are cases in which the algorithm settles at a local minimum that does not correspond to the desired texture. This usually occurs when the support of derivative filters is large and when we use many derivative filters to regularize the image. For instance, suppose we want to match the gradient histogram of a  $3 \times 3$  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 increases the complexity. To control the complexity, we match four two-tap derivative filters. Adapting derivative filters to local image structures using steerable filters [3], [7], [19] may further improve the rendition of oriented textures, but it is not considered in this work.

### 4.3 Reference distribution $q_R$ estimation

We parameterize a reference distribution  $q_R$  using a generalized Gaussian distribution. Unfortunately, one often does not know *a priori* what  $q_R$  should be. Previous work estimates  $q_R$  from a database of natural images [6], [24] or hand-picks  $q_R$  through trial and error [15]. We adopt the image prior



**Fig. 6:** We compare the deblurring performance of a MAP estimator and IDR. IDR reconstructs visually more pleasing mid-frequency textures compared to a MAP estimator.



**Fig. 7:** For an image with spatially varying texture, our algorithm segments the image into regions of homogeneous texture and matches the gradient distribution in each segment independently. Compared to MAP estimators, our algorithm reconstructs visually more pleasing textures.

estimation technique introduced in Cho *et. al.* [3] to estimate  $q_R$  directly from a degraded image, as we will now describe.

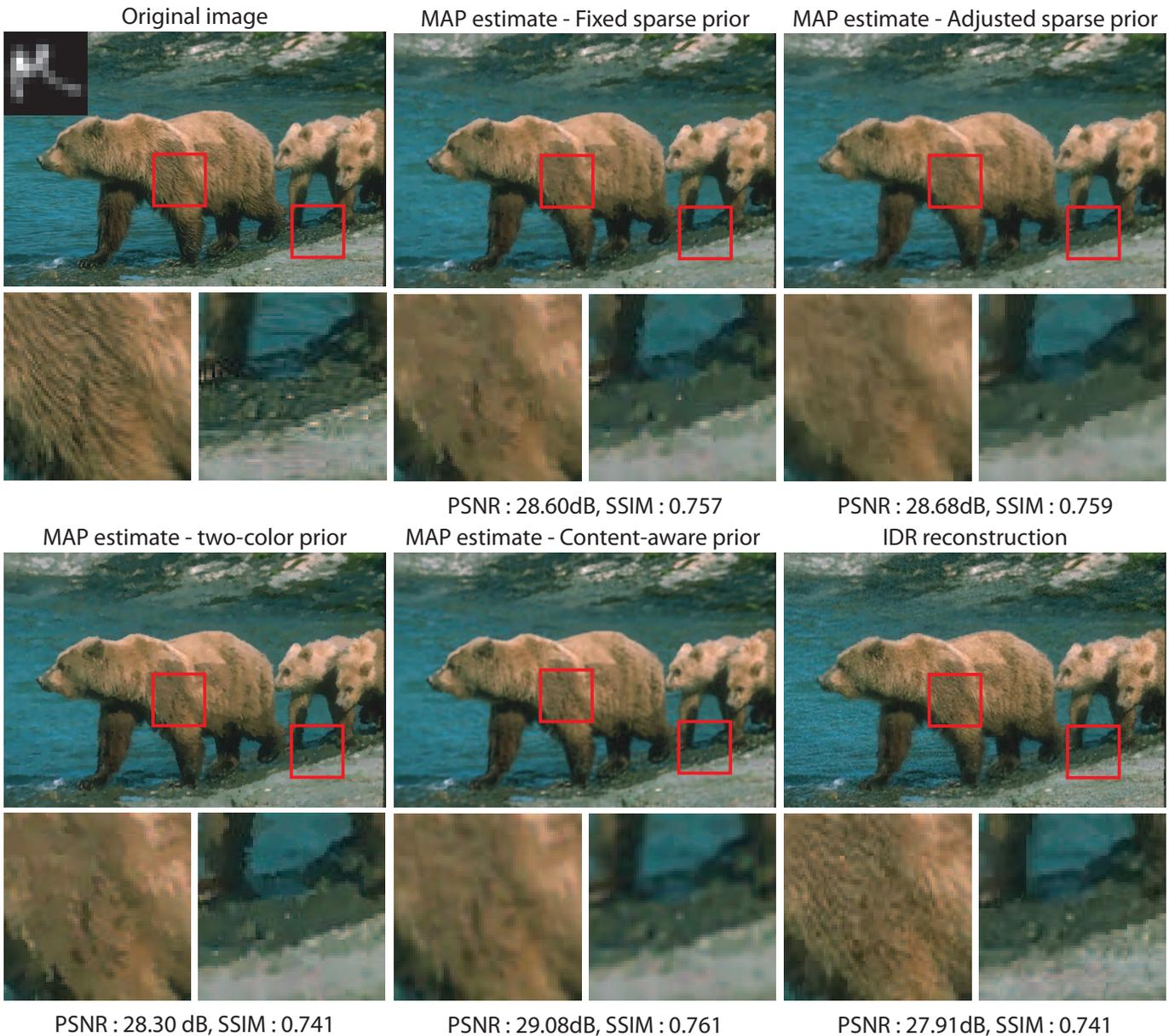
We first deconvolve a degraded image  $B$  using a MAP estimator (Eq. 1) with a hand-picked image prior, tuned to restore different textures reasonably well at the expense of a slightly noisy image reconstruction (i.e. a relatively small gradient penalty). In this paper, we set the parameters of the image prior as  $[\gamma = 0.8, \lambda = 4, w_1 = 0.01]$  for all images. To reduce deconvolution noise, we down-sample the reconstructed image. We fit gradients from the down-sampled image to a generalized Gaussian distribution, as in Eq. 4, to estimate the reference distribution  $q_R$ . While fine details can be lost through down-sampling, empirically, the estimated reference distribution  $q_R$  is accurate enough for our purpose.

Our image reconstruction algorithm assumes that the texture is homogeneous (i.e. a single  $q_R$ ). In the presence of multiple textures within an image, we segment the image and estimate separate reference distribution  $q_R$  for each segment: we use the EDISON segmentation algorithm [4] to segment an image into about 20 regions. Figure 7 illustrates the image deconvolution process for spatially varying textures.

## 5 EXPERIMENTS

### 5.1 Deconvolution experiments

We synthetically blur sharp images with the blur kernel shown in Figure 8, add 2% noise, and deconvolve them using competing methods. We compare the performance of IDR against four other competing methods: (i) a MAP estimator with a sparse gradient prior [15], (ii) a MAP estimator with a sparse prior adapted to each segment (iii) a MAP estimator with a two-color prior [12] (iv) a MAP estimator with a content-aware image prior [3]. We blur a sharp image using the kernel shown on the right, add 2% noise to it, and restore images using the competing methods. Figure 8 shows experimental results. As mentioned in Section 4.2, IDR does not perform the best in terms of PSNR / SSIM. Nevertheless, IDR reconstructs mid-frequency textures better, for instance fur details. Another interesting observation is that the content-aware image prior performs better, in terms of PSNR/SSIM, than simply adjusting the image prior to each segment's texture. By using the segment-adjusted image prior, we observe segmentation boundaries that are visually disturbing. Another set of comparisons is shown in Figure 9.

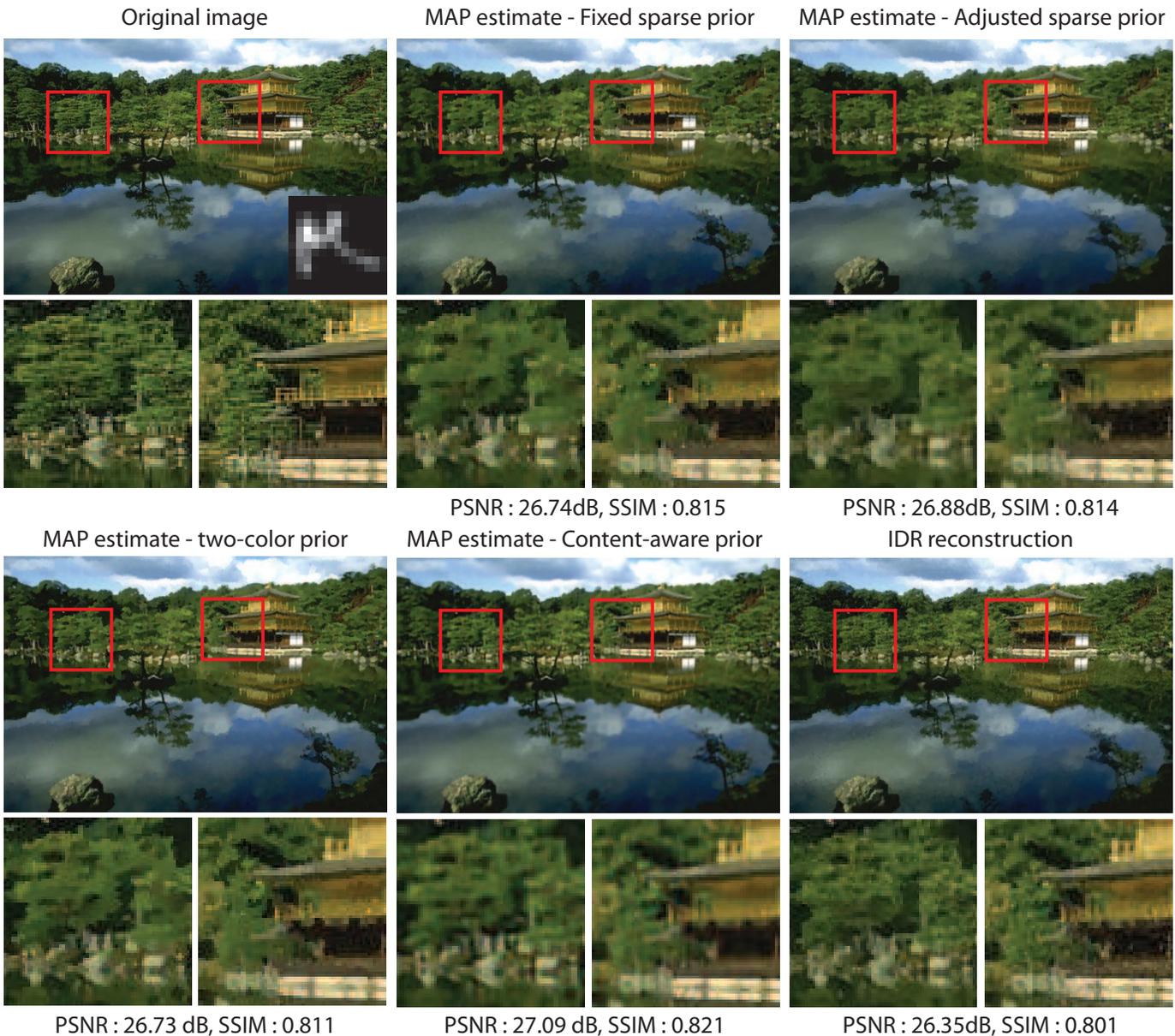


**Fig. 8:** We compare the performance of IDR against four other competing methods: (i) a MAP estimator with a sparse gradient prior [15], (ii) a MAP estimator with a sparse prior adapted to each segment (iii) a MAP estimator with a two-color prior [12] (iv) a MAP estimator with a content-aware image prior. The red box indicate the cropped regions. Although the PSNR and the SSIM of our results are often lower than those of MAP estimators, IDR restores more visually pleasing textures (see bear furs).

In Figure 10, we compare the denoising performance of IDR to that of a marginal probability field (MPF) by Woodford *et al.* [24] at two noise levels (their implementation only handles grayscale, square images). Using MPF for denoising has two drawbacks. First, MPF quantizes intensity levels and gradient magnitudes to reduce computation. MPF quantizes 256 (8-bit) intensity levels to 64 intensity levels (6-bit), and it bins 256 (8-bit) gradient magnitudes to 11 slots. These quantizations would accentuate spotty noise in reconstructed images. IDR adopts a continuous optimization scheme that does not require any histogram binning or intensity quantization, therefore it does not suffer from quantization noise. Second, Woodford

*et al.* [24] estimate the reference 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 reference distributions  $q_R$ , but MPF would enforce the same gradient profile on them. Also, MPF does not adapt the image prior to the underlying texture, treating different textures the same way. Therefore, MPF distributes gradients uniformly across the image, even in smooth regions, which can be visually disturbing. IDR addresses these issues by estimating a reference distribution  $q_R$  from an input image and by adapting  $q_R$  to spatially varying texture.

At a high degradation level, such as a noise level of 31.4%, our



**Fig. 9:** We compare the performance of IDR against four other competing methods. As in Figure 8, IDR’s PSNR/SSIM are lower than those of MAP estimators, but IDR restores visually more pleasing textures.

reference distribution estimation algorithm can be unstable. In Figure 10(a), our  $q_R$  estimation algorithm returns a distribution that has more “large” derivatives and less “small” derivatives (dotted line in Figure 10), which manifests itself as a noisy IDR reconstruction. In contrast, MPF restores a plausible image, but this is somewhat coincidental in that the reference distribution that MPF imposes is quite similar to that of the original image.

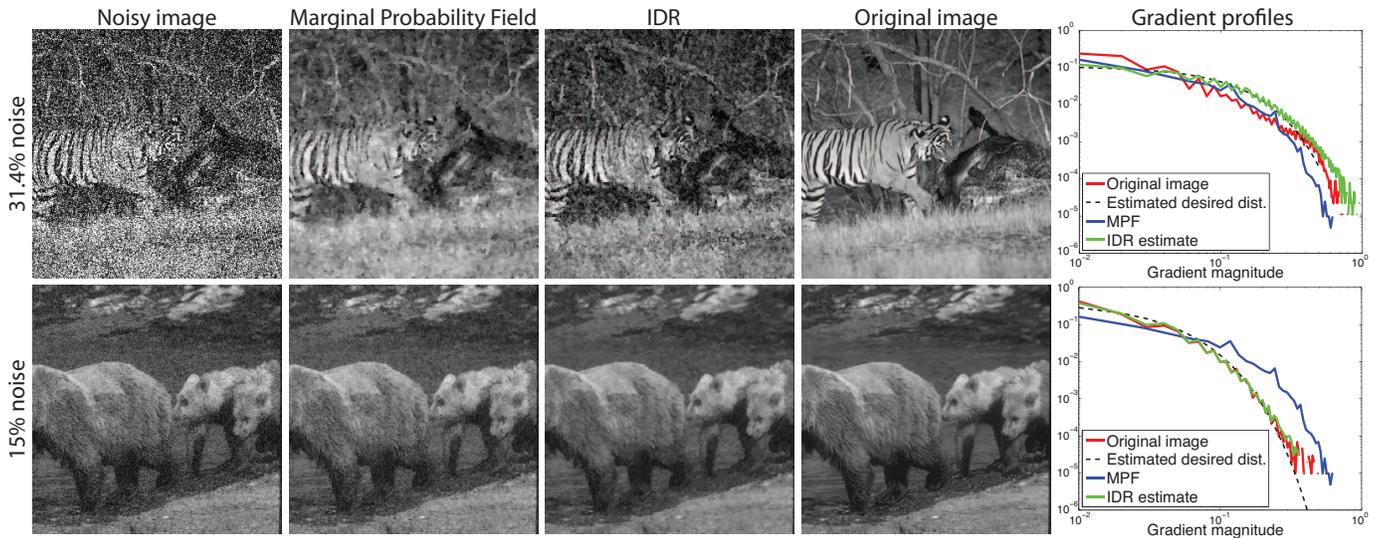
At a more reasonable degradation level (15% noise), shown in Figure 10(b), our algorithm estimates a reference distribution that is very similar to that of the original image. Given a more accurate reference distribution, IDR restores a visually pleasing image. On the other hand, MPF restores a noisy rendition because the reference distribution is quite different from that of the original image. Also note that the gradient

distribution of the restored image in Figure 10(b) is very similar to that of the restored image in Figure 10(a), which illustrates our concern that using a single image prior for different images would degrade the image quality.

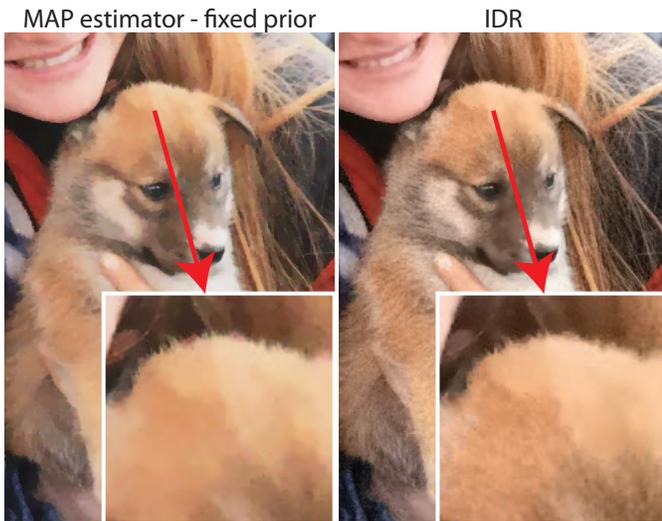
Segmenting images to regions and deconvolving each region separately may generate artificial texture boundaries, as in Figure 11. While this rarely occurs, we could mitigate these artifacts using a texture-based segmentation algorithm rather than EDISON [4], which is a color-based segmentation algorithm.

## 5.2 User study

IDR generates images with rich texture but with lower PSNR/SSIM than MAP estimates. To test our impression that



**Fig. 10:** Comparing the denoising performance of IDR to the marginal probability field (MPF) [24]. IDR generates a better rendition of the spatially variant texture.

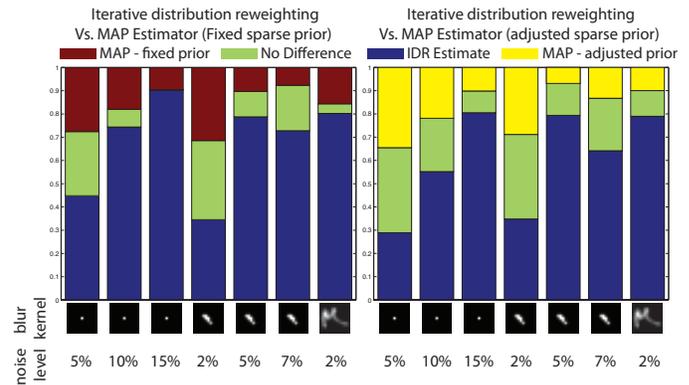


**Fig. 11:** We could observe an artificial boundary when the estimated prior is different in adjacent segments that have similar textures. While this rarely occurs, we could remove such artifacts using a texture segmentation algorithm instead of a color-based segmentation algorithm.

images reconstructed by IDR are 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 and 2% noise. For each degradation scenario, we randomly selected 4 images from a dataset of 13 images (roughly  $700 \times 500$  pixels), and reconstructed images using a MAP estimator with a fixed sparse prior (i.e. the same sparse prior across the whole image), an adjusted sparse prior, and IDR.

We showed users two images side-by-side, one reconstructed using our algorithm and another reconstructed using one of



**Fig. 12:** We conducted a user study to test our impression that IDR reconstructions are visually more 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.

the two MAP estimators (i.e. fixed or adjusted). We asked users to select an 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 images side by side.

We collected more than 25 user inputs for each comparison, and averaged user responses for each degradation scenario (Figure 12). When the degradation level is low (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 and/or the reconstruction algorithm. When the degradation level is high, 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 MAP

estimates preferred clean renditions of flat regions and were not disturbed by piecewise smooth textures (some even found it artistic.) Individual users consistently favored either our result or MAP estimates, suggesting that image evaluation is subjective in nature.

## 6 CONCLUSION

We have developed an iterative deconvolution algorithm that matches the gradient distribution. Our algorithm bridges the energy minimization methods for deconvolution and texture synthesis. We show through a user study that matching derivative distribution improves the perceived quality of reconstructed images. The fact that a perceptually better image receives lower PSNR/SSIM suggests that there is a room for improvement in image quality assessment.

## APPENDIX A

We sketch the details of the optimization procedures in Algorithm 1. 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 Algorithm 1 with respect to  $X$ :

$$\begin{aligned}
& -\frac{K^T(Y - KX)}{\eta^2} + 2w_1\lambda_D\gamma_D G^T \|GX\|^{\gamma_D-1} \\
& + w_2 \left( 2\lambda_D\gamma_D G^T \|GX\|^{\gamma_D-1} - 2\lambda_E\gamma_E G^T \|GX\|^{\gamma_E-1} \right. \\
& + \left. \left( \frac{1}{\gamma_E} - \frac{\ln(\lambda_E)}{\gamma_E^2} + \frac{\Psi\left(\frac{1}{\gamma_E}\right)}{\gamma_E^2} - \lambda_E|GX|^{\gamma_E} \ln(|GX|) \right) \circ \frac{\partial\gamma_E}{\partial X} \right. \\
& + \left. \left( \frac{1}{\gamma_E\lambda_E} - |GX|^{\gamma_E} \right) \circ \frac{\partial\lambda_E}{\partial X} \right) = 0
\end{aligned} \tag{12}$$

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:

$$\begin{aligned}
\frac{\partial\gamma_E}{\partial X} &= \frac{\gamma_E^2\lambda_E\left(\frac{2}{\gamma_E}\right)\Gamma\left(\frac{1}{\gamma_E}\right)}{N\Gamma\left(\frac{3}{\gamma_E}\right)\left(\Psi\left(\frac{1}{\gamma_E}\right) - 3\Psi\left(\frac{3}{\gamma_E}\right) + 2\ln(\lambda_E)\right)} 2G^T GX \\
\frac{\partial\lambda_E}{\partial X} &= -\frac{\Gamma(1/\gamma_E)\gamma_E\lambda_E^{(1+2/\gamma_E)}}{N\Gamma(3/\gamma_E)} G^T GX
\end{aligned} \tag{13}$$

$N$  is the dimension of  $X$ .

We solve Eq. 12 by iteratively approximating it with a linear equation [15], [22]. We use a minimum residual method [20] to solve the *linearized* system in Eq. 12.

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

[3] T. S. Cho, N. Joshi, C. L. Zitnick, S. B. Kang, R. Szeliski, and W. T. Freeman. A content-aware image prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.  
[4] C. M. Christoulias, B. Georgescu, and P. Meer. Synergism in low level vision. In *IEEE ICPR*, 2002.  
[5] R. O. Duda, P. E. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, 2000.  
[6] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM TOG (Proc. SIGGRAPH)*, 2006.  
[7] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE TPAMI*, 1991.  
[8] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *IJCV*, 40(1):25 – 47, 2000.  
[9] Gonzalez and Woods. *Digital image processing*. Prentice Hall, 2008.  
[10] Y. HaCohen, R. Fattal, and D. Lischinski. Image upsampling via texture hallucination. In *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*, 2010.  
[11] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *ACM SIGGRAPH*, 1995.  
[12] N. Joshi, C. L. Zitnick, R. Szeliski, and D. Kriegman. Image deblurring and denoising using color priors. In *IEEE CVPR*, 2009.  
[13] 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.  
[14] 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.  
[15] 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.  
[16] Y. Li and E. H. Adelson. Image mapping using local and global statistics. In *SPIE EI*, 2008.  
[17] M. Nikolova. Model distortions in Bayesian MAP reconstruction. *Inverse Problems and Imaging*, 1(2):399–422, 2007.  
[18] 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.  
[19] S. Roth and M. J. Black. Steerable random fields. In *IEEE ICCV*, 2007.  
[20] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM JSSC*, 1986.  
[21] U. Schmidt, Q. Gao, and S. Roth. A generative perspective on MRFs in low-level vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.  
[22] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Reviews*, 41(3):513 – 537, Sept. 1999.  
[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004.  
[24] O. J. Woodford, C. Rother, and V. Kolmogorov. A global perspective on MAP inference for low-level vision. In *IEEE ICCV*, 2009.