

# Noise Estimation from a Single Image

Ce Liu William T. Freeman  
CS and AI Lab, MIT  
{celiu, billf}@csail.mit.edu

Richard Szeliski Sing Bing Kang  
Microsoft Research  
{szeliski, sbkang}@microsoft.com

## Abstract

*In order to work well, many computer vision algorithms require that their parameters be adjusted according to the image noise level, making it an important quantity to estimate. We show how to estimate an upper bound on the noise level from a single image based on a piecewise smooth image prior model and measured CCD camera response functions. We also learn the space of noise level functions—how noise level changes with respect to brightness—and use Bayesian MAP inference to infer the noise level function from a single image. We illustrate the utility of this noise estimation for two algorithms: edge detection and feature-preserving smoothing through bilateral filtering. For a variety of different noise levels, we obtain good results for both these algorithms with no user-specified inputs.*

## 1. Introduction

Many computer vision algorithms can work well only if the parameters of the algorithm are hand-tweaked to account for characteristics of the particular images under study. One of the most common needs for algorithm parameter adjustment is to account for variations in noise level over input images. These variations can be caused by changes in the light level or acquisition modality, and the amount of variation can be severe. An essential step toward achieving reliable, automatic computer vision algorithms is the ability to accurately estimate the noise level of images. Examples of algorithms requiring noise level estimates include motion estimation [1], denoising [12, 11], super-resolution [5], shape-from-shading [19], and feature extraction [9].

Estimating the noise level from a single image seems like an impossible task: we need to recognize whether local image variations are due to color, texture, or lighting variations from the image itself, or due to the noise. It might seem that accurate estimation of the noise level would require a very sophisticated prior model for images. However, in this work we use a very simple image model—piecewise smooth over segmented regions—to derive a *bound* on the image noise

level. We fit the image data in each region with a smooth function and estimate the noise using the residual. Such estimates are guaranteed to be over-estimates of the true noise variance, since the regions can contain image variations that are not being modeled. However, there are usually some regions in each image for which the simple model holds, and we find that our noise bounds are usually tight. While we assume that auxiliary camera information is unknown, the algorithm proposed in this paper can easily be augmented with such information.

Unfortunately, real CCD camera noise is not simply additive; it is strongly dependent on the image intensity level. We call the noise level as a function of image intensity the *noise level function*, or NLF. Since it may not be possible to find image regions at all the desired levels of intensity or color needed to describe the noise level function, we use the Columbia camera response function database [6] to model the NLF using principal components and bounds on derivatives. These principal components serve as a prior model to estimate the NLF over regions of missing data. We use a Bayesian MAP framework to estimate the noise level functions in RGB channels, using the bounds derived from the noise estimates over each image region. Experiments on both synthetic and real data demonstrate that the proposed algorithm can reliably infer noise levels.

To illustrate that estimating the noise can make vision algorithms more robust, we apply our noise inference to two algorithms: bilateral filtering for feature-preserving smoothing, and edge detection. The resulting algorithms, properly accounting for image noise, show robust behavior over a wide range of noise conditions.

The paper is organized as follows. We review related work in Sect. 2. In Sect. 3, we describe our camera noise model and show how it can be combined with a set of parameterized camera response functions to develop a prior on the noise response function. We show how to estimate the noise level function in Sect. 4 using a combination of per-region variance estimation and the prior model on NLFs. We provide experimental results in Sect. 5 and apply those estimates to the two computer vision algorithms in Sect. 6. We conclude in Sect. 7.

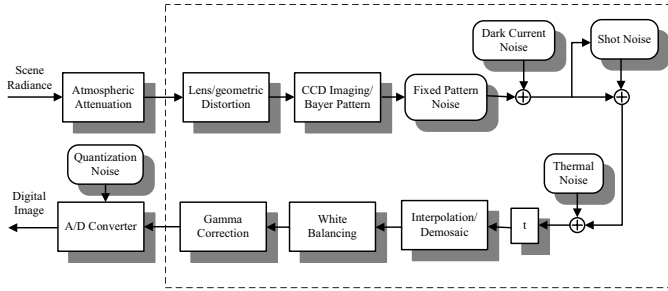


Figure 1. CCD camera imaging pipeline, redrawn from [18].

## 2. Related Work

There is a large literature on image denoising. Although very promising denoising results have been achieved using a variety of methods, such as wavelets [12], anisotropic diffusion [11] and bilateral filtering [17], the noise level is often assumed known and constant for varying brightness values.

In contrast, the literature on noise estimation is very limited. Noise can be estimated from multiple images or a single image. Estimation from multiple image is an over-constrained problem, and was addressed in [7]. Estimation from a single image, however, is an under-constrained problem and further assumptions have to be made for the noise. In the image denoising literature, noise is often assumed to be additive white Gaussian noise (AWGN). A widely used estimation method is based on mean absolute deviation (MAD) [3]. In [15], the authors proposed three methods to estimate noise level based on training samples and the statistics (Laplacian) of natural images. However, real CCD camera noise is intensity-dependent.

## 3. Noise Study

In this section, we build a model for the noise level functions of CCD cameras. In Subsect. 3.1 we introduce the terms of our camera noise model, showing the dependence of the noise level function (the noise variance as function of brightness) on the camera response function (the image brightness as function of scene irradiance). Given a camera response function, described in Subsect. 3.2, we can synthesize realistic camera noise, as shown in Subsect. 3.3. Thus, in Subsect. 3.4, from a parameterized set of camera response functions (CRFs), we derive the set of possible noise level functions. This restricted class of NLFs allows us to accurately estimate the NLF from a single image, as described in Sect. 4.

### 3.1. Noise Model of CCD Camera

The CCD digital camera converts the irradiance, the photons coming into the imaging sensor, to electrons and finally to bits. See Figure 1 for the imaging pipeline of CCD cam-

era. There are mainly five noise sources as stated in [7], namely *fixed pattern noise*, *dark current noise*, *shot noise*, *amplifier noise* and *quantization noise*. These noise terms are simplified in [18]. Following the imaging equation in [18], we propose the following noise model of a CCD camera

$$I = f(L + n_s + n_c) + n_q, \quad (1)$$

where  $I$  is the observed image brightness,  $f(\cdot)$  is camera response function (CRF),  $n_s$  accounts for all the noise components that are dependent on irradiance  $L$ ,  $n_c$  accounts for the independent noise before gamma correction, and  $n_q$  is additional quantization and amplification noise. Since  $n_q$  is the minimum noise attached to the camera and most cameras can achieve very low noise,  $n_q$  will be ignored in our model. We assume noise statistics  $E(n_s) = 0$ ,  $\text{Var}(n_s) = L\sigma_s^2$  and  $E(n_c) = 0$ ,  $\text{Var}(n_c) = \sigma_c^2$ . Note the linear dependence of the variance of  $n_s$  on the irradiance  $L$  [18].

### 3.2. Camera Response Function (CRF)

The camera response function models the nonlinear processes in a CCD camera that perform tonescale (gamma) and white balance correction [14]. There are many ways to estimate camera response functions given a set of images taken under different exposures. To explore the common properties of many different CRFs, we downloaded 201 real-world response functions from <http://www.cs.columbia.edu/CAVE> [6]. Note that we only chose 190 saturated CRFs since the unsaturated curves are mostly synthesized. Each CRF is a 1024-dimensional vector that represents the discretized  $[0, 1] \rightarrow [0, 1]$  function, where both irradiance  $L$  and brightness  $I$  are normalized to be in the range  $[0, 1]$ . We use notation  $\text{crf}(i)$  to represent the  $i^{\text{th}}$  function in the database.

### 3.3. Synthetic CCD Noise

In principle, we could set up optical experiments to measure precisely for each camera how the noise level changes with image brightness. However, this would be time consuming and might still not adequately sample the space of camera response functions. Instead, we use numerical simulation to estimate the noise function. The basic idea is to transform the image  $\mathbf{I}$  by the inverse camera response function  $f^{-1}$  to obtain an irradiance plane  $\mathbf{L}$ . We then take  $\mathbf{L}$  through the processing blocks in Figure 1 to obtain the noisy image  $\mathbf{I}_N$ .

A direct way from Eqn. (1) is to reverse transform  $\mathbf{I}$  to irradiance  $\mathbf{L}$ , add noise independently to each pixel, and transform to brightness to obtain  $\mathbf{I}_N$ . This process is shown in Figure 2 (a). The synthesized noise image, for the test pattern (c), is shown in Figure (d).

Real CCD noise is not white, however; there are spatial correlations introduced by “demosaicing” [13], i.e., the re-

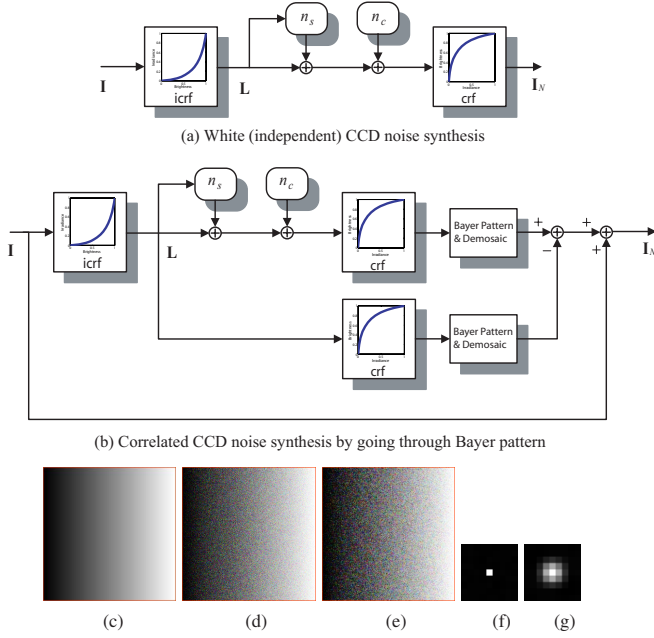


Figure 2. Block diagrams showing noise simulations for color camera images. (a) shows independent white noise synthesis; (b) adds CCD color filter pattern sensing and demosaicing to model spatial correlations in the camera noise [8]. (c): test pattern. (d) and (e): the synthesized images of (a) and (b). (f) and (g): the corresponding autocorrelation.

construction of three colors at every pixel from the single-color samples measured under the color filter array of the CCD. We simulate this effect for a common color filter pattern (Bayer) and demosaicing algorithm (gradient-based interpolation [8]); we expect that other filter patterns and demosaicing algorithms will give comparable noise spatial correlations. We synthesized CCD camera noise in accordance with 2 (b) and took the difference between the demosaiced images with and without noise, adding that to the original image to synthesize CCD noise. The synthesized noise is shown in Figure 2 (e). The autocorrelation functions for noise images (d) and (e) are shown in (f) and (g), respectively, showing that the simulated CCD noise shows spatial correlations after taking into account the effects of demosaicing.

### 3.4. The Space of Noise Level Functions

We define the *noise level function* (NLF) as the variation of the standard deviation of noise with respect to image intensity. This function can be computed as

$$\tau(I) = \sqrt{E[(I_N - I)^2]}, \quad (2)$$

where  $I_N$  is the observation and  $I = E(I_N)$ . Essentially this is a function of how standard deviation changes with respect to the mean value.

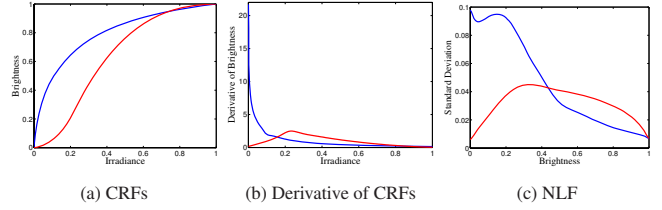


Figure 3. The blue and red curves correspond to  $\text{crf}(50)$  and  $\text{crf}(60)$ . The NLFs are estimated at  $\sigma_s = 0.06$ ,  $\sigma_c = 0.02$ .

Based on the CCD camera noise model Eqn. (1) and noise synthesis process,  $I_N$  is a random variable dependent on the camera response function  $f$  and noise parameters  $\sigma_s$  and  $\sigma_c$ . Because  $L = f^{-1}(I)$ , the noise level function can also be written as

$$\tau(I; f, \sigma_s, \sigma_c) = \sqrt{E[(I_N(f^{-1}(I), f, \sigma_s, \sigma_c) - I)^2]}, \quad (3)$$

where  $I_N(\cdot)$  is the noise synthesis process.

We use numerical simulation to estimate the noise function given  $f$ ,  $\sigma_s$  and  $\sigma_c$ , for each of red, green and blue channels. The smoothly changing pattern in Figure 2 (c) is used to estimate Eqn. (3). To reduce statistical fluctuations, we use an image of dimension  $1024 \times 1024$  and take the mean of 20 samples for each estimate.

To see how the noise function changes with respect to parameter  $f$ , we examine two CRFs,  $\text{crf}(50)$  and  $\text{crf}(60)$ , as shown in Figure 3 (a). Their derivatives are plotted in (b). The corresponding NLFs, which have been computed at  $\sigma_s = 0.06$ ,  $\sigma_c = 0.02$ , are plotted in (c). Notice that the derivative of  $\text{crf}(50)$  is significant at zero irradiance, resulting in high noise standard deviation there. It is clear that the NLFs are highly dependent on the derivative function.

To represent the whole space of noise level functions, we draw samples of  $\tau(\cdot; f, \sigma_s, \sigma_c)$  from the space of  $f$ ,  $\sigma_s$  and  $\sigma_c$ . The downloaded 190 CRFs are used to represent the space of  $f$ . We found that  $\sigma_s = 0.16$  and  $\sigma_c = 0.06$  result in very high noise, so these two values are set as the maximum of the two parameters. We sample  $\sigma_s$  from 0.00 to 0.16 with step size 0.02, and sample  $\sigma_c$  from 0.01 to 0.06 with step size 0.01. We get a dense set of samples  $\{\tau_i\}_{i=1}^K$  of NLFs, where  $K = 190 \times 9 \times 6 = 10,260$ . Using principal component analysis (PCA), we get mean noise level function  $\bar{\tau}$ , eigenvectors  $\{w_i\}_{i=1}^m$  and the corresponding eigenvalues  $\{v_i\}_{i=1}^m$ . Thus, a noise function can be represented as

$$\tau = \bar{\tau} + \sum_{i=1}^m \beta_i w_i, \quad (4)$$

where coefficient  $\beta_i$  is Gaussian distributed  $\beta_i \sim \mathcal{N}(0, v_i)$ , and the function must be positive everywhere, i.e.,

$$\bar{\tau} + \sum_{i=1}^m \beta_i w_i \geq 0, \quad (5)$$

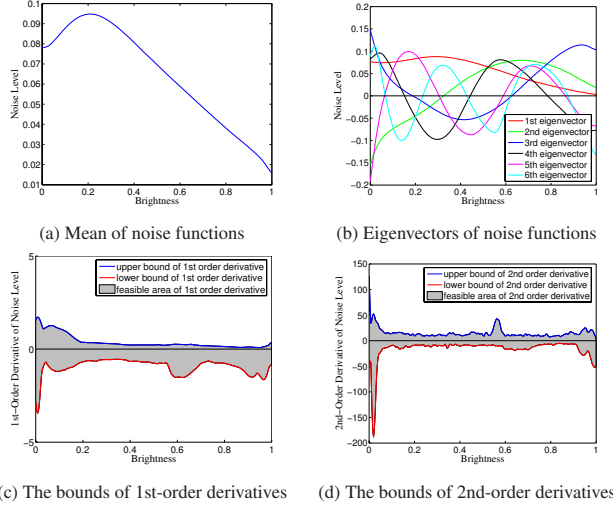


Figure 4. Camera noise function properties.

where  $\bar{\tau}, w_i \in \mathbb{R}^d$  and  $d = 256$ . This inequality constraint implies that noise functions lie inside a cone in  $\beta$  space. The mean noise function and eigenvectors are displayed in Figure 4 (a) and (b), respectively.

Eigenvectors serve as basis functions to impose smoothness to the function. We also impose upper and lower bounds on 1st and 2nd order derivatives to further constrain noise functions. Let  $\mathbf{T} \in \mathbb{R}^{(d-1) \times d}$  and  $\mathbf{K} \in \mathbb{R}^{(d-2) \times d}$  be the matrix of 1st- and 2nd-order derivatives [16]. The constraints can be represented as

$$b_{min} \leq \mathbf{T}\tau \leq b_{max}, \quad h_{min} \leq \mathbf{K}\tau \leq h_{max}, \quad (6)$$

where  $b_{min}, b_{max} \in \mathbb{R}^{d-1}$ ,  $h_{min}, h_{max} \in \mathbb{R}^{d-2}$  are estimated from the training data set  $\{\tau_i\}_{i=1}^K$ .

#### 4. Noise Estimation from a Single Color Image

Noise estimation and image denoising are in general chicken and egg problems. To estimate the noise level, the underlying signal must be known, which could be estimated using a denoising algorithm; conversely, most denoising algorithms depend on knowing the noise level. Sophisticated algorithms such as EM might find a stationary point, but algorithms including denoising as an inner loop might be too slow to be practical. Instead, we propose a rough estimate of the signal using a piecewise smooth image prior. This approach is reasonable since we aim at estimating an upper bound of the noise.

##### 4.1. Piecewise Smooth Image Prior

The estimation is fairly easy if the underlying clean image  $\mathbf{I}$  of the noisy input  $\mathbf{I}_N$  is constant. Although this is not true in practice, we assume in our algorithm a piecewise

smooth prior for natural images. We group pixels into regions based on both range and spatial similarities using a K-means clustering method as described in [20]. Each segment is represented by a linear slope and spatial extent. The spatial extent is computed so that the shape of the segment is biased towards convex shapes and so that all segments have similar size.

We assume that the mean estimate of each segment accounts for brightness  $I$  and the variance estimation accounts for  $\hat{\sigma}$ . From all the segments, we can get samples  $\{I_i, \hat{\sigma}_i\}$ , which are then used to estimate the noise level function.

##### 4.2. Likelihood Model

The goal of noise estimation is to fit a lower envelope to the sample set as shown in Figure 6. We could simply fit the noise function in the learnt space to lie below all the sample points yet close to them. However, because the estimates of variance in each segment are noisy, extracting these estimates with hard constraints could result in bias due to a bad outlier. Instead, we follow a probabilistic inference framework to let every data point contribute to the estimation.

Let the estimated standard deviation of noise from  $k$  pixels be  $\hat{\sigma}$ , with  $\sigma$  being the true standard deviation. When  $k$  is large, the square root of chi-square distribution is approximately  $\mathcal{N}(0, \sigma^2/k)$  [4]. In addition, we assume a noninformative prior for large  $k$ , and obtain the posterior of the true standard deviation  $\sigma$  given  $\hat{\sigma}$ :

$$\begin{aligned} p(\sigma|\hat{\sigma}) &\propto \frac{1}{\sqrt{2\pi\sigma^2/k}} \exp\left\{-\frac{(\hat{\sigma}-\sigma)^2}{2\sigma^2/k}\right\} \\ &\approx \frac{1}{\sqrt{2\pi\hat{\sigma}^2/k}} \exp\left\{-\frac{(\sigma-\hat{\sigma})^2}{2\hat{\sigma}^2/k}\right\}. \end{aligned} \quad (7)$$

Let the cumulative distribution function of a standard normal distribution be  $\Phi(z)$ . Then, given the estimate  $(I, \hat{\sigma})$ , the probability that the underlying standard deviation  $\sigma$  is larger than  $u$  is

$$\Pr[\sigma \geq u|\hat{\sigma}] = \int_u^\infty p(\sigma|\hat{\sigma}) d\sigma = \Phi\left(\frac{\sqrt{k}(\hat{\sigma}-u)}{\hat{\sigma}}\right). \quad (8)$$

To fit the noise level function to the lower envelope of the samples, we discretize the range of brightness  $[0, 1]$  into uniform intervals  $\{nh, (n+1)h\}_{n=0}^{\frac{1}{h}-1}$ . We denote the set  $\Omega_n = \{(I_i, \hat{\sigma}_i) | nh \leq I_i \leq (n+1)h\}$ , and find the pair  $(I_n, \hat{\sigma}_n)$  with the minimum variance  $\hat{\sigma}_n = \min_{\Omega_n} \hat{\sigma}_i$ . Lower envelope means that the fitted function should most probably be lower than all the estimates while being as close as possible to the samples. Mathematically, the likelihood function is the probability of seeing the observed image intensity and noise variance measurements given a particular



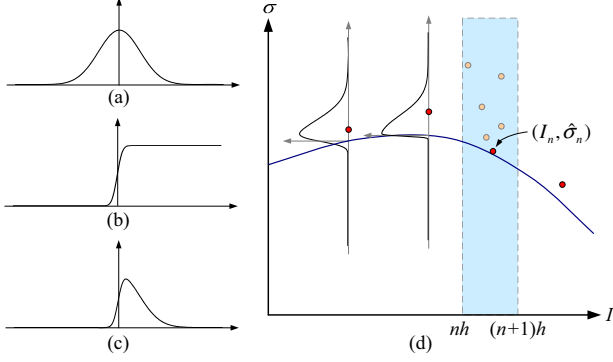


Figure 5. The likelihood function of Eq. 9. Each single likelihood function (c) is a product of a Gaussian pdf (a) and Gaussian cdf (b).

noise level function. It is formulated as

$$\begin{aligned} \mathcal{L}(\tau(I)) &= P(\{I_n, \hat{\sigma}_n\} | \tau(I)) \\ &\propto \prod_n \Pr[\sigma_n \geq \tau(I_n) | \hat{\sigma}_n] \exp\left\{-\frac{(\tau(I_n) - \hat{\sigma}_n)^2}{2s^2}\right\} \\ &= \prod_n \Phi\left(\frac{\sqrt{k_n}(\hat{\sigma}_n - \tau(I_n))}{\hat{\sigma}_n}\right) \exp\left\{-\frac{(\tau(I_n) - \hat{\sigma}_n)^2}{2s^2}\right\}, \quad (9) \end{aligned}$$

where  $s$  is the parameter to control how close the function should approach the samples. This likelihood function is illustrated in Figure 5, where each term (c) is a product of a Gaussian pdf with variance  $s^2$  (a) and a Gaussian cdf with variance  $\hat{\sigma}_n^2$  (b). The red dots are the samples of minimum in each interval. Given the function (blue curve), each red dot is probabilistically beyond but close to the curve with the pdf in (c).

### 4.3. Bayesian MAP Inference

The parameters we want to infer are actually the coefficients on the eigenvectors  $x_l = [\beta_1 \dots \beta_m]^T \in \mathbb{R}^m$ ,  $l = 1, 2, 3$  of the noise level function for RGB channels. We denote the sample set to fit  $\{(I_{ln}, \hat{\sigma}_{ln}, k_{ln})\}$ . Bayesian inference turns out to be an optimization problem

$$\begin{aligned} \{x_l^*\} = \arg \min_{\{x_l\}} & \sum_{l=1}^3 \left\{ \sum_n \left[ -\log \Phi\left(\frac{\sqrt{k_{ln}}}{\hat{\sigma}_{ln}}(\hat{\sigma}_{ln} - e_n^T x_l - \bar{\tau}_n)\right) \right. \right. \\ & \left. \left. + \frac{(e_n^T x_l + \bar{\tau}_n - \hat{\sigma}_{ln})^2}{2s^2} \right] + x_l^T \Lambda^{-1} x_l + \right. \\ & \left. \sum_{j=1, j>l}^3 (x_l - x_j)^T \mathbf{E}^T (\gamma_1 \mathbf{T}^T \mathbf{T} + \gamma_2 \mathbf{K}^T \mathbf{K}) \mathbf{E} (x_l - x_j) \right\} \quad (10) \end{aligned}$$

subject to

$$\bar{\tau} + \mathbf{E}x_l \geq 0, \quad (11)$$

$$b_{min} \leq \mathbf{T}(\bar{\tau} + \mathbf{E}x_l) \leq b_{max}, \quad (12)$$

$$h_{min} \leq \mathbf{K}(\bar{\tau} + \mathbf{E}x_l) \leq h_{max}. \quad (13)$$

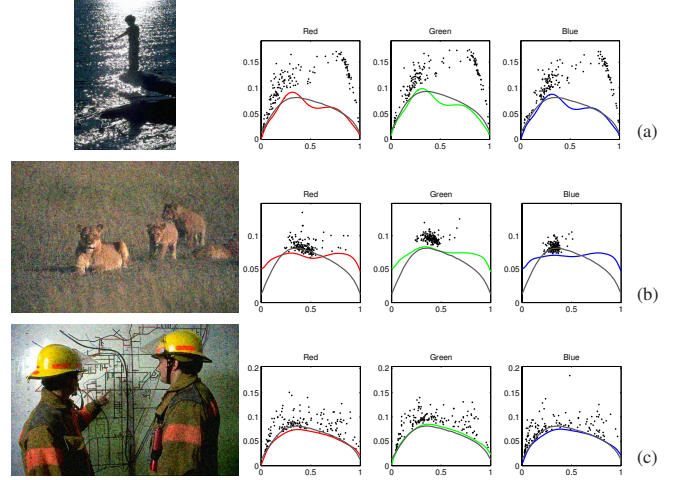


Figure 6. Synthesized noisy images and their corresponding noise level functions (noise standard deviation as a function of image brightness). The red, green and blue curves are estimated using the proposed algorithm, whereas the gray curves are the true values for the synthetically generated noise.

In the above formula, the matrix  $\mathbf{E} = [w_1 \dots w_m] \in \mathbb{R}^{dxm}$  contains the principal components,  $e_n$  is the  $n^{th}$  row of  $\mathbf{E}$ , and  $\Lambda = \text{diag}(v_1, \dots, v_m)$  is the diagonal eigenvalue matrix. The last term in the objective function accounts for the similarity of the NLF for RGB channels. Their similarity is defined as a distance on 1st and 2nd order derivative. Since the dimensionality of the optimization is low, we use the MATLAB standard nonlinear constrained optimization function `fmincon` for optimization. The function was able to find an optimal solution for all the examples we tested.

### 5. Experimental Results

We have conducted experiments on both synthetic and real noisy images to test the proposed algorithm. First, we applied our CCD noise synthesis algorithm in Sect 3.3 to 17 randomly selected pictures from the Berkeley image segmentation database [10] to generate synthetic test images. To generate the synthetic CCD noise, we specify a CRF and two parameters  $\sigma_s$  and  $\sigma_c$ . From this information, we also produce the ground truth noise level function using the training database in Sect 3.4. For this experiment, we selected `crf(60)`,  $\sigma_s = 0.10$  and  $\sigma_c = 0.04$ . Then, we applied our method to estimate the NLF from the synthesized noisy images. Both  $L^2$  and  $L^\infty$  norms are used to measure the distance between the estimated NLF and the ground truth. The error statistics under the two norms are listed in Table 1, where the mean and maximum value of the ground truth are 0.0645 and 0.0932, respectively.

Some estimated NLFs are shown in Figure 6. In (a) we observe many texture regions especially at high intensity values, which implies high signal variance. The estimated

| Norm       | mean   | std. deviation |
|------------|--------|----------------|
| $L^2$      | 0.0048 | 0.0033         |
| $L^\infty$ | 0.0110 | 0.0120         |

Table 1. The statistics of the  $L^2$  and  $L^\infty$  norms between the estimated NLF and the ground truth.

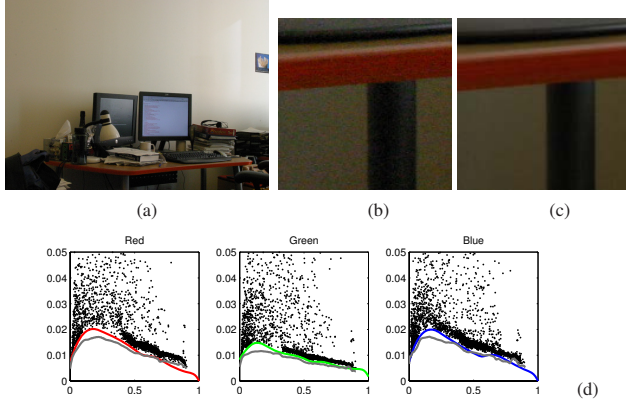


Figure 7. Comparison of estimated camera noise with experimental measurement. (a) shows one of the 29 images taken with a Canon<sup>TM</sup>EOS 10D. An enlarged patch is shown for (b) a single image, and (c) the mean image. (d) is the estimated NLF from a single image (color), showing good agreement with the ground truth (gray), measured from the noise variations over all 29 images.

curves (in red, green and blue) do not tightly follow the lower envelope of the samples at high intensities, although they deviate from the true noise function (in gray) slightly. In (b) the samples do not span the full intensity range, so our estimate is only reliable where the samples appear. This shows a limit of the prior model: the samples are assumed to be well-distributed. The estimation is reliable if the color distribution span the full range of the spectrum and there are textureless regions, as in (c).

We conducted a further experiment as a sanity check. We took 29 images of a static scene by Canon<sup>TM</sup>EOS 10D (ISO 1600, exposure time 1/30 s and aperture f/19) and computed the mean image. One sample is shown in Figure 7 (a). A close-up view of sample (a) and the mean image is shown in (b) and (c), respectively. Clearly the noise is significantly reduced in the mean image. Using the variance over the 29 images as a function of mean intensity, we calculated the “ground truth” NLF and compared that to the NLF estimated by our method from only one image. The agreement between the NLFs in each color band is very good, see Figure 7 (d).

We also applied the algorithm to estimating noise level functions from the other images taken by a CCD camera. We evaluated our results based on repeatability: pictures taken by the same camera with the same setting on the same date should have the same noise level function, independent

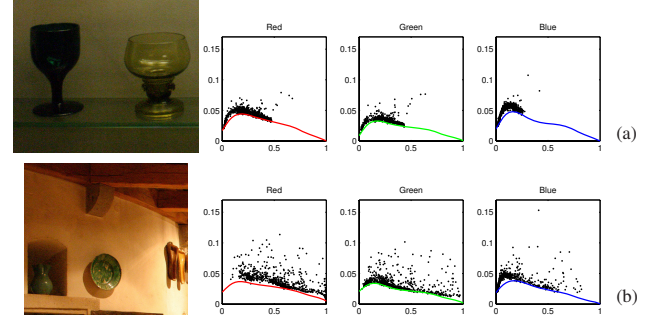


Figure 8. The two images are taken by a Canon<sup>TM</sup>EOS DIGITAL REBEL and the estimated noise level functions. Very similar noise level functions are derived, even though the two images have very different tonescales.

of the image content. We collected two pictures taken by a Canon<sup>TM</sup>EOS DIGITAL REBEL and estimated the corresponding noise level functions, as shown in Figure 8 (a) and (b). Even though image (a) is missing high intensity values, the estimated NLFs are similar.

The performance of our unoptimized MATLAB code is 28.8s on a Pentium(R) D CPU 3.00GHz 2.00GB RAM PC when the samples span the full spectrum of intensity.

## 6. Computer Vision Applications

Many computer vision problems are noise-dependent, and the typical approach assumes the noise level is known or supplied by the user. Our technique provides a way to automatically infer the noise level function directly from the image.

### 6.1. Adaptive Bilateral Filtering

Bilateral filtering [17] is a simple and popular algorithm for feature-preserving image smoothing. To take advantage of our image noise estimates, we implemented the bilateral filter in two steps. The first step is per pixel denoising, based on  $\Sigma_n$ , the diagonal matrix of the noise variance of each color band, obtained from the NLF. We can estimate the covariance of the signal,  $\Sigma_s$  from  $\Sigma_s = \Sigma_z - \Sigma_n$ , where  $\Sigma_z$  is the covariance matrix in each segment. Let  $z$  be the observation of the pixel, and  $\mu$  the value of piecewise smooth function at the pixel, then by downweighting we can estimate the signal for that pixel [12]

$$s = (\Sigma_n^{-1} + \Sigma_s^{-1})^{-1}(\Sigma_n^{-1}z + \Sigma_s^{-1}\mu) \quad (14)$$

The bilateral filter has two parameters, the spatial and range standard deviations,  $\sigma_s$  and  $\sigma_r$ . In the second step we apply a bilateral filter, separately in each color band, to the per-pixel denoised image. We obtained visually pleasing results using  $\sigma_s = 3$ , and  $\sigma_r = 1.95\sigma_n$ , where  $\sigma_n$  is the noise level at each pixel in each color band obtained from the NLF.

Four synthetic noisy images were generated by gradually increasing the noise level, as shown in Fig 9 (a). After estimating the NLFs shown in (b), we obtained the denoised results shown in (d). For comparison, we show denoising results of classical bilateral filtering with constant parameter setting  $\sigma_r = 3$  and  $\sigma_s = 0.12$  in (c). Clearly our adaptive method was able to remove more noise while maintaining image structure.

## 6.2. Canny Edge Detection

The basic idea of Canny edge detection [2] is to find an optimal filter so that the most salient edges can be found despite noise. The optimal filter is theoretically independent of noise, but the threshold is noise dependent. We designed an adaptive Canny edge detector where the low pass filter is fixed and the higher threshold is set to be  $0.5\bar{\sigma}_n^2 + 2\bar{\sigma}_n + 0.1$ , where  $\bar{\sigma}_n$  is the average noise of the input image. The lower threshold is set to be 0.4 of the higher threshold. The results of the adaptive Canny edge detection are shown in Figure 9 (f). For comparison, the results of classical Canny edge detection with automatic parameter setting in MATLAB<sup>TM</sup> are shown in (e). Although the edges are distorted by the noise in (f) as the noise level is increased, the adaptive edge detector does a better job of detecting true edges while suppressing false ones.

## 7. Conclusion

In this paper, we have shown how to infer the noise as a function of image intensity (the noise level function, NLF) from a single image. We used a very simple prior model for images to estimate the noise level without knowing the image content, by deriving bounds on the noise level. We verified that these bounds are tight in three ways: (1) by showing good agreement with experimentally measured noise from repeated exposures of the same image, (2) by repeatedly measuring the same NLF's with the same camera for different image content, and (3) by accurately estimating known synthetic noise functions.

The inferred NLFs are useful for computer vision algorithms that require knowing the noise level to set algorithm parameters (or to select a relevant training set). We have shown examples for two algorithms, bilateral filtering and Canny edge detection. Using our noise estimates to set algorithm parameters, we achieved robust results over a wide range of image noise levels.

Our work addresses the fundamental issue of making computer vision algorithms more reliable. We feel that such reliability can only be obtained by considering and modeling the sources of variability in real-world images. Automatically inferring the image noise level and taking it into account in vision algorithms is an important step.

## References

- [1] S. Baker and I. Matthews. Lucas-Kanade 20 years on: a unifying framework. *Int'l Journal on Computer Vision*, 56(3):221–255, 2004.
- [2] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [3] D. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.
- [4] M. Evans, N. Hastings, and B. Peacock. *Statistical Distributions*. A Wiley-Interscience Publication, 2000.
- [5] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *Int'l Journal on Computer Vision*, 40(1):25–47, 2000.
- [6] M. D. Grossberg and S. K. Nayar. Modeling the space of camera response functions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(10):1272–1282, October 2004.
- [7] G. E. Healey and R. Kondepudy. Radiometric CCD camera calibration and noise estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(3):267–276, March 1994.
- [8] C. Laroche and M. Prescott. *Apparatus and Methods for Adaptively Interpolating a Full Color Image Utilizing Chrominance Gradients*. U.S. Patent No. 5,373,322 (1994).
- [9] D. Lowe. Object recognition from local scale-invariant features. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1150–1157, 1999.
- [10] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. IEEE Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.
- [11] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.
- [12] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, Nov 2003.
- [13] R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. Sander III. Demosaicking methods for Bayer color arrays. *Journal of Electronic Imaging*, 11(3):306–315, July 2002.
- [14] W. F. Schreiber. *Fundamentals of electronic imaging systems*. Springer-Verlag, 1986.
- [15] A. Stefano, P. White, and W. Collis. Training methods for image noise level estimation on wavelet components. *EURASIP Journal on Applied Signal Processing*, 16:2400–2407, 2004.
- [16] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, 1986.
- [17] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 839–846, 1998.
- [18] Y. Tsin, V. Ramesh, and T. Kanade. Statistical calibration of CCD imaging process. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 480–487, 2001.



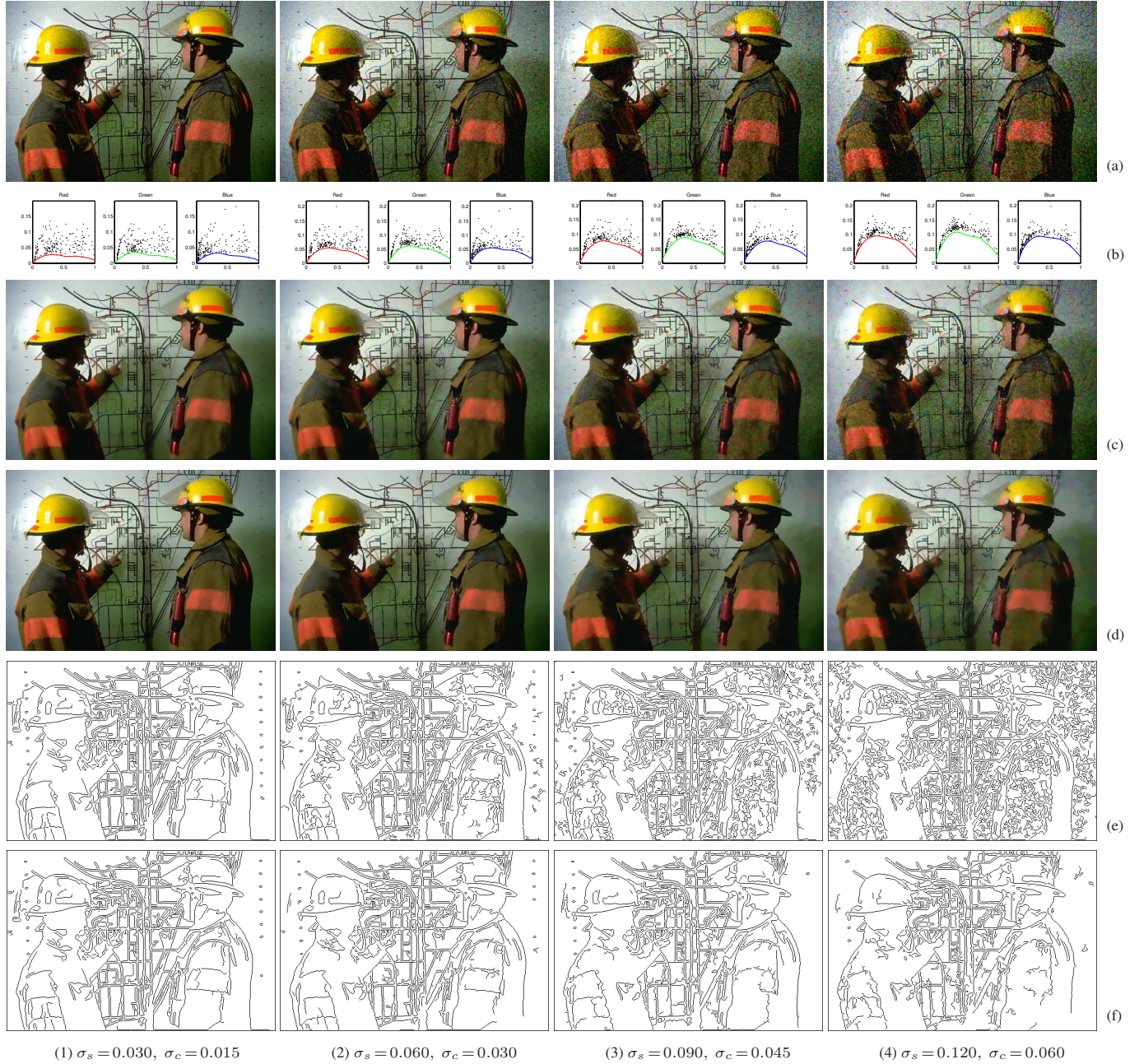


Figure 9. Noise estimation can make computer vision algorithms more robust to noise. Four synthetic noise contaminated images (a) are obtained by increasing  $\sigma_s$  and  $\sigma_c$ . Noise level functions as inferred by our algorithm from each image (b). (c) Classical bilateral filtering with constant parameter setting. Note the sensitivity of the output to the input noise level. (d) Adaptive bilateral filtering exploiting inferred noise levels, leading to robust performance independent of noise level. (e) Canny edge detection using the automatic parameter settings supplied in MATLAB<sup>TM</sup> program, which attempt to adapt to the input image. Despite adaptation, the algorithm does not adequately compensate for the changes in image noise level. (f) Canny edge detection after adapting algorithm parameters using inferred noise level functions. Algorithm performance is now substantially independent of the image noise levels.

[19] R. Zhang, P. Tsai, J. Cryer, and M. Shah. Shape from shading: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.

[20] C. L. Zitnick, N. Jojic, and S. B. Kang. Consistent segmen-

tation for optical flow estimation. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1308–1315, 2005.