A Holistic Approach to Cross-Channel Image Noise Modeling and its Application to Image Denoising

Seonghyeon Nam* Yonsei University
Youngbae Hwang* KETI
Yasuyuki Matsushita Osaka University
Seon Joo Kim Yonsei University

Abstract

Modelling and analyzing noise in images is a fundamental task in many computer vision systems. Traditionally, noise has been modelled per color channel assuming that the color channels are independent. Although the color channels can be considered as mutually independent in camera RAW images, signals from different color channels get mixed during the imaging process inside the camera due to gamut mapping, tone-mapping, and compression. We show the influence of the in-camera imaging pipeline on noise and propose a new noise model in the 3D RGB space to accounts for the color channel mix-ups. A data-driven approach for determining the parameters of the new noise model is introduced as well as its application to image denoising. The experiments show that our noise model represents the noise in regular JPEG images more accurately compared to the previous models and is advantageous in image denoising.

1. Introduction

Noise is one of the most fundamental problems in computer vision and image processing. In computer vision literature, many works mention noise as one of the main sources that explain the error of their systems and emphasize the necessity of establishing robustness against noise. But what really is image noise and how can we explain it?

Informally, noise explains the uncertainty of the light measurement in an image. A low noise value would indicate that the observed intensity value is highly likely to be the ground truth intensity and vice versa. Since many computer vision algorithms rely on accurate light measurement, it is important to have a noise model that explains the properties of image noise accurately.

What is interesting is that all of the existing image noise models fall short of explaining what is really happening to noise in the images that most people (both regular consumers and vision researchers) use today - JPEG images in the sRGB color space. A key assumption used in existing noise models is that the noise is independent between different color channels. While this channel independency is valid for linear vision cameras or RAW images, which are unprocessed sensor-level measurements, the assumption breaks down as the R,G,B values are heavily mixed during the in-camera image processing [3, 15]. Furthermore, image compression, typically in JPEG format, significantly affects the noise characteristics; however, the effect of the compression noise has not been explicitly considered in the past[1].

This work attempts to seek deeper understanding about image noise and to come up with a better explanation about the noise compared to the previous noise models. Previous noise models either only fit to a limited number of cases, or they were only validated with synthetic images created with their own models. Even in the image denoising works, the quantitative performance of denoising is evaluated with synthetic images created with a per channel independent Gaussian model [5], which does not describe the noise in real photographs as we will see later in this paper. Therefore, we argue for a new image noise model that better explains the properties of noise in images that most people use today.

The contributions of this paper are as follows:

- We provide observations and analysis on the effect of the in-camera imaging process on noise and introduce a new *cross-channel* noise model. We develop a color-depenent noise model in the 3D RGB space that can simultaneously take into account the correlation between the color channels and the JPEG compression effect.
- We further propose a data-driven approach for automatically determining the noise in the 3D RGB space from observed color images. Specifically, we use a simple multi-layer perceptron (MLP) to infer the parameters of the noise model. Note that we use the neu-

[*] Authors contributed equally to this work.

[1] Note that our problem is to analyze the effect of compression on the noise level, which is different from dealing with the blocky JPEG noise or artifacts[18].

ral network (NN) to compute the noise model, which is fundamentally different from using NNs for image denoising [25, 2].

- We validate our model and the parameter estimation method using real images instead of synthetic images and show that applying our new noise model can improve the image denoising performance.

2. Related Work

The most common noise model used in computer vision is the channel-independent Gaussian model [21, 22, 24] because of its simplicity. However, the Gaussian noise model is found inflexible to describe the actual noise in real images, and therefore, several more sophisticated noise models have been recently proposed. In [6], Foi *et al.* proposed a Poissonian-Gaussian noise model for single-image raw data to consider signal-dependent and signal-independent noise components separately. Granados *et al.* [8] presented a noise model that takes into account both temporal and the spatial noise for reconstructing high-dynamic range (HDR) images. Their weighting function produces statistically optimal estimates under the assumption of compound Gaussian noise. Hwang *et al.* [12] presented a difference based noise model using the Skellam distribution to represent the distribution of intensity differences. They showed that the difference-based modeling has a more significant linear relationship between the intensity and the noise parameters.

The methods described above all operate with RAW images or images from a linear vision camera [12]. While noise modeling of RAW image data is useful for some specific tasks, most of the images used in computer vision go through an in-camera imaging pipeline. In a seminal work by Healey and Kondepudy [10], five main sources for image noise in the camera imaging process were identified as photon shot, fixed pattern, dark current, readout, and quantization noise, and they presented a statistical model in which the variance of noise is linearly proportional to the observed intensity.

In [19], Liu *et al.* presented a more general noise model that fits the in-camera imaging pipeline, which includes processes such as white balancing and camera response functions (gamma correction). They used the in-camera imaging model from [23] and defined the noise level function (NLF) as the variation of standard deviation of the noise distribution according to image brightness. Using the space of camera response functions [9], they used a Bayesian MAP estimation to infer the NLF from a single image. In spite of insufficient information to estimate image noise, their method showed good performance when applied to noise removal [19] and image deblurring [13].

While the methods described above are effective, each color channel is still treated independently in these works. In [15], Kim *et al.* described a new in-camera imaging model that well fits modern cameras by showing the effect of the gamut mapping step, which is a 3D nonlinear mapping (RGB to RGB). Their work intrinsically indicates the limitation of channel-independent noise modeling because of the mixture of color channels due to the gamut mapping and color space transformations. In addition to such mixtures, JPEG compression process [4, 18] adds further mixing of color channels. Our goal is to accurately model and determine such cross-channel image noise.

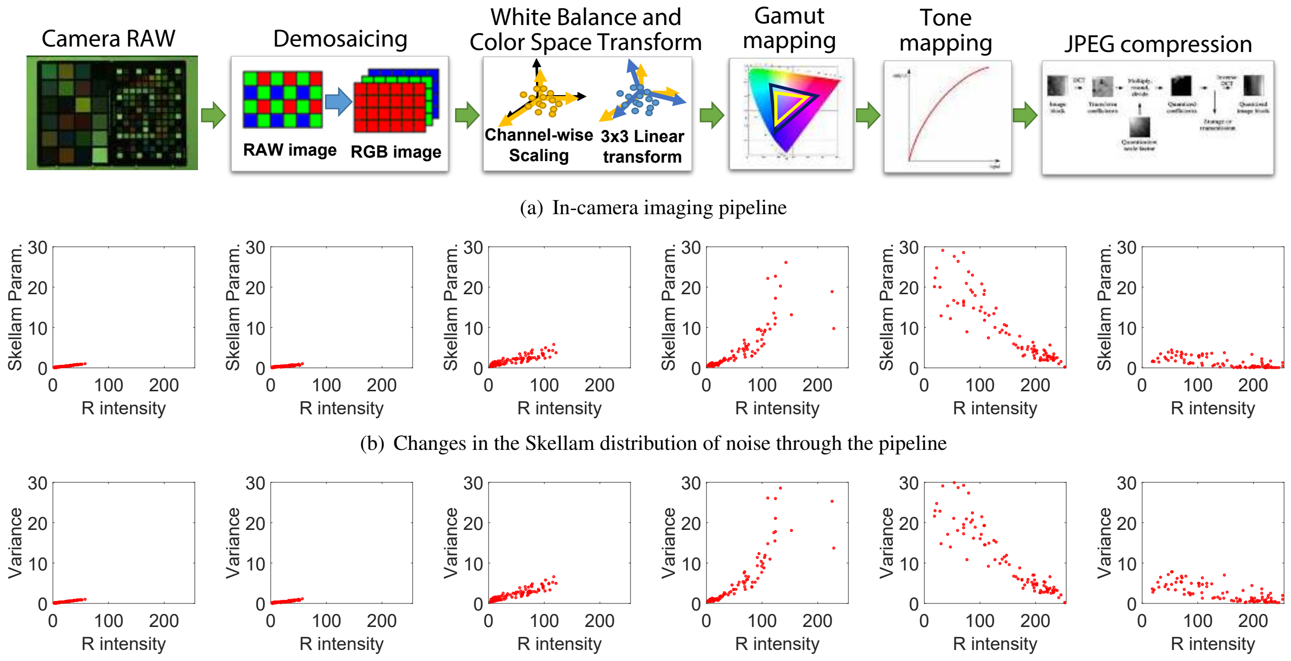
3. Noise Model in the 3D RGB Space

This section analyzes the effects of the in-camera imaging process and JPEG compression on noise and propose a noise model that can accurately represent them.

3.1. Noise through the In-camera Imaging Process

Figure 1 shows the influence of each procedure in the in-camera imaging pipeline on noise. The top row of the figure lists the imaging steps described in [15]. To verify how the noise characteristics are altered through the procedures, we first took a RAW image with many homogeneous color patches with a Canon EOS-5D Mark III camera. We then simulated the imaging pipeline using the calibrated camera parameters from [15] and observed the changes of the noise distributions. As shown in the first plots in Fig. 1 (b) and (c), both the Skellam parameter [12] and the variance [19] linearly increase with the intensity value in the RAW image. While this linear relationship is still largely maintained after going through the demosaicing and white balancing/linear color transformations, the linear relationship drastically breaks down with the gamut mapping and tone mapping processes.

We ran another experiment to see how the mix-up of R,G,B channel values (mainly due to 3D gamut mapping) influence the image noise. We took images as shown in Fig. 2 (a) with a Nikon D800 camera, which records images in three different formats: RAW, uncompressed TIFF, and JPEG. The uncompressed TIFF image allows us to analyze the effect of the whole imaging pipeline without the compression effect. With these different formats of images, we computed the covariance matrices for each pixel from 1,000 temporal images of a static scene, some of which are shown in Fig. 2 (b-d). They show the magnitude of elements in the covariance matrix; from the variances of R,G,B to the covariance R/G, R/B, and G/B. At the RAW level, the noise in different channels are indeed independent. However, we can observe that the covariance scores significantly increase as the image goes through the imaging pipeline, and reach the point where we cannot simply ignore the cross-channel noise.

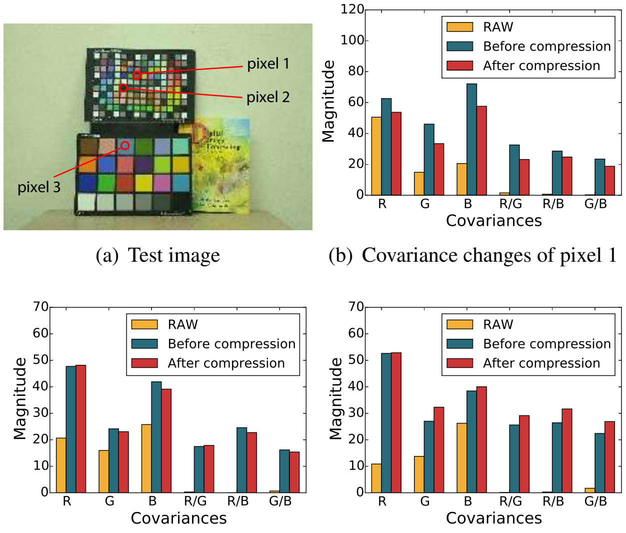


(a) In-camera imaging pipeline

(b) Changes in the Skellam distribution of noise through the pipeline

(c) Changes in the noise variance distribution through the pipeline

Figure 1. In-camera imaging pipeline and changes in noise distribution through the pipeline. (a) shows block diagram sequences of in-camera imaging pipeline. (b) and (c) show the changes in the Skellam and the variance distribution through the pipeline, respectively. Noise chracteristic drastically changes with the gamut mapping and the tone mapping processes.



(a) Test image

(b) Covariance changes of pixel 1

(c) Covariance changes of pixel 2

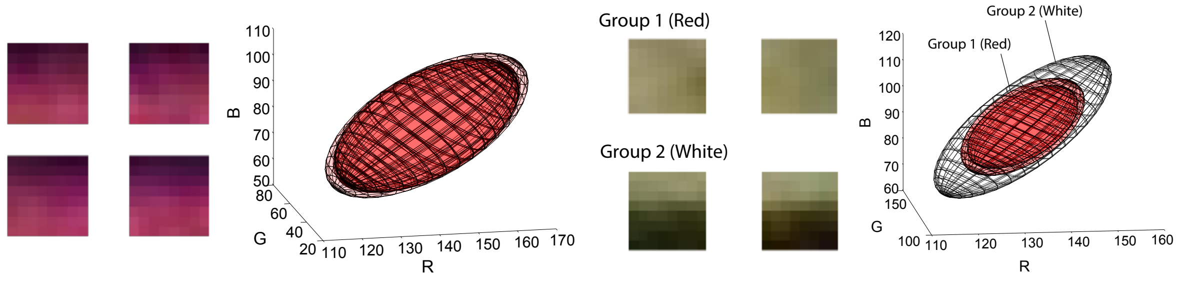
(d) Covariance changes of pixel 3

Figure 2. Covariance magnitude changes of some pixels after in-camera imaging pipeline and JPEG compression. (a) is the test image scene which was captured in a RAW, an uncompressed TIFF and a JPEG format by Nikon D800. (b)-(d) shows the changes in the covariance terms during the imaging pipeline.

3.2. Effect of the JPEG Compression on Noise

Figures 1 and 2 show that JPEG compression has a significant effect on the noise characteristics. With a typical JPEG compression, an image is compressed by dividing the image region into 8×8 patches and processing those patches separately. Therefore, the level of compression may differ patch by patch, and the influence of the compression to noise would also depend on the patch content. The noise characteristics depend not only on a single pixel RGB value but also on other pixels in the patch. Therefore, even for a particular RGB value, the noise characteristics would vary according to the other pixels in the patch.

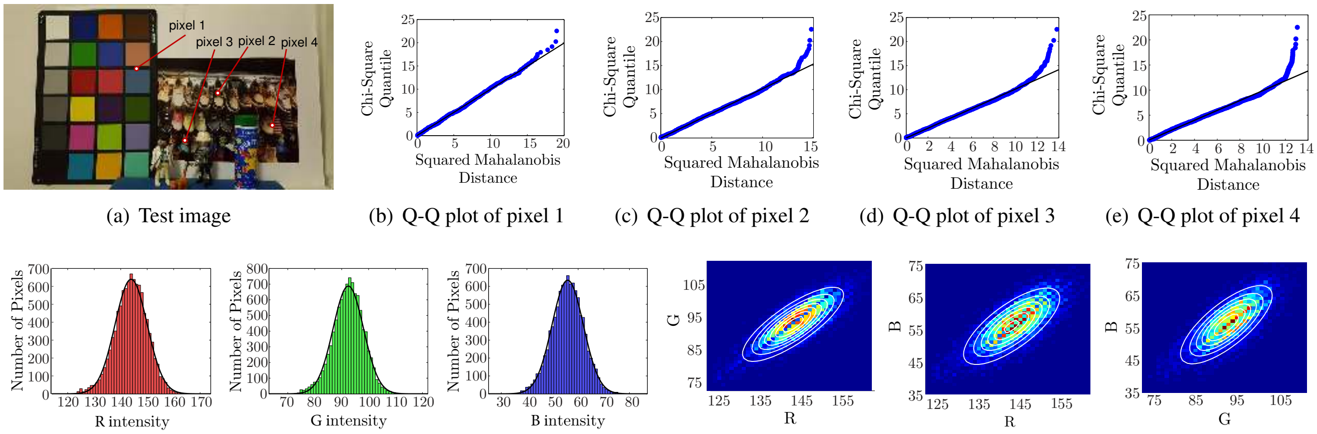
The examples of the effect of the JPEG compression on noise are visualized in Fig. 3. After recording 1,000 JPEG images of a static scene, we fit covariance matrices to different pixels. Figure 3 shows the covariance matrices of several pixels that have the same RGB value in the form of ellipsoids. As expected, the pixels (having the same RGB value) that are located in similar patches show the similar covariance structures (Fig. 3 (a)), while pixels in visually different patches exhibit diverse covariance structures (Fig. 3 (b)). These examples indicate that a good noise model should be able to explain noise's dependence on both the scene and the pixel color.



(a) Similar patches

(b) Different patches

Figure 3. JPEG compression effect on 8×8 patch. (a) shows the ellipsoids of covariance matrices of the same RGB value in similar patches have similar shapes. (b) shows the ellipsoids vary according to the patch the color is in. The ellipsoids represent 95% confidence interval of distribution.



(a) Test image

(b) Q-Q plot of pixel 1

(c) Q-Q plot of pixel 2

(d) Q-Q plot of pixel 3

(e) Q-Q plot of pixel 4

(f) Univariate and bivariate Gaussian distributions of pixel 4

Figure 4. Multivariate Gaussian fitting test on real data. (a) A test image captured 10,000 times using a Samsung Galaxy S6 smartphone camera (ISO 800, 80% compression). (b)-(e) Multivariate Q-Q plot of four selected pixels. The linear relationships indicate that the samples follow multivariate Gaussian distribution [11]. (f) Shows the color distribtion of pixel 4. It empirically shows that the noise should be modelled as a multivariate (3D) Gaussian distribution.

3.3. Noise Model in the 3D RGB Space

To properly account for the image noise discussed in previous subsections, we propose a noise model that is characterized by a covariance matrix in the RGB color space. Based on the observations as shown in Fig. 4, we model the noise as the signal-dependent multivariate Gaussian distribution. In Fig. 4 (b)-(e), the Q-Q plots show the ordered squared Mahalanobis distance of samples versus the estimated quantiles from a chi-square distribution with 3 degrees of freedom. The linear relationships as shown on the plots mean that the samples follow a multivariate Gaussian distribution [11]. The empirical example in Fig. 4 (f) also supports for the multivariate (3D) Gaussian model.

In addition to the multivariate Gaussian model, we also consider the local patch contents in our model to deal with the content dependency due to the JPEG compression. With this consideration, the noise of an image pixel (x, y) is determined by the (R,G,B) values of the pixel $\mathbf{I}(x, y)$ as well as the 8×8 patch in which the pixel (x, y) is located. We ignore the pixel position in the patch for simplicity.

Putting it altogether, our noise model in the 3D RGB space is written mathematically as:

$$\begin{aligned} \mathbf{I}(x,y) &= \bar{\mathbf{I}}(x,y) + \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}(\bar{\mathbf{I}}(x,y), \mathbf{p}_{xy})\right), \\ \mathbf{\Sigma}(\bar{\mathbf{I}}(x,y), \mathbf{p}_{xy}) &= \begin{bmatrix} \sigma_r^2 & \sigma_{rg} & \sigma_{rb} \\ \sigma_{rg} & \sigma_g^2 & \sigma_{gb} \\ \sigma_{rb} & \sigma_{gb} & \sigma_b^2 \end{bmatrix}, \end{aligned} \tag{1}$$

where $\bar{\mathbf{I}}(x,y)$ is the true intensity of $\mathbf{I}(x,y)$, and $\mathbf{p}_{xy}$ is the 8×8 color patch value. $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}(\bar{\mathbf{I}}(x,y), \mathbf{p}_{xy}))$ is the zero mean multivariate Gaussian distribution of noise with the covariance matrix $\mathbf{\Sigma}(\bar{\mathbf{I}}(x,y), \mathbf{p}_{xy})$ that is the function of the true intensity $\bar{\mathbf{I}}(x,y)$ and its corresponding patch $\mathbf{p}_{xy}$.

4. Data-driven Noise Estimation Algorithm

In theory, the proposed model in Eq. (1) should be defined for every possible patch with each pixel value in the

3D RGB space, which is computed as $256^{(8\times8\times3)}$. It is unrealistic to compute and store the noise model parameters for all those colors and patch values. Therefore, we employ a data-driven approach based on a multi layer perceptron (MLP) for determining the noise parameters of pixels in a given image.

4.1. Data Collection

For the MLP to perform well, collecting a large number of quality data is essential. We captured training images for 11 static scenes, 500 JPEG images per scene, and computed the mean image of each scene to generate the ground truth noise-free images.[2] Some of the captured scenes are shown in Fig. 5. For each dataset, the covariance for each pixel, which is computed using the temporal stack of images, is fed into the system for training along with its (R,G,B) values and the $8 \times 8 \times 3$ patch information. The training is done per camera model and settings such as ISO, and the total amount of data per set is about 13 million patches for an image with resolution 7360×4912 (98% of the data is used for the training and 2% for the validation).

4.2. MLP based Noise Estimation Method

A multi-layer perceptron (MLP) is a feed-forward neural network that trains a nonlinear transformation of vector-valued input layer. The input layer is mapped to the output layer via several hidden layers. Formally, MLPs are defined as

$$\mathbf{x}^{(n+1)} = g(b^{(n)} + \mathbf{W}^{(n)}\mathbf{x}^{(n)}), \tag{2}$$

where $\mathbf{x}^{(n+1)}$ is the value of $(n+1)$-th layer ($\mathbf{x}^{(1)}$ is the input layer). $\mathbf{W}^{(n)}$ and $b^{(n)}$ are trainable weights and a bias. For the nonlinear activation function g, a sigmoid, a tanh, or ReLU [16] is used. When n is more than 2, the MLPs can be used as a universal approximator, which is able to learn any nonlinear mapping. Therefore, we use an MLP with our training data to find the complex nonlinear mapping from the RGB value of a pixel and its surrounding patch to its corresponding covariance matrix.

At first, we had expected that the MLP for our problem would require a large number of layers and units. However, we have found that a single hidden layer MLP is enough to learn the nonlinear correlation of our data. Specifically, the structure of our MLP is (195, 200, 6), which are the number of units in each layer[3]. Note that the number of parameters of the MLP is considerably small compared to the original problem space where we need a covariance matrix per combination of color and patch ($256^{(8\times8\times3)}$). With the small number of parameters and its regression power, MLP serves as an efficient and accurate modeling tool for our noise modelling problem.

Figure 5. Some samples of the scenes in our dataset.

The trained MLP can be seen as a regressor, which can predict the covariance matrix for any given (R,G,B) value and its patch. We can formally express the MLP-based noise estimation method as

$$\boldsymbol{\Sigma}(\bar{\mathbf{I}}(x,y),\mathbf{p}_{xy}) = h(f(\bar{\mathbf{I}}(x,y),\mathbf{p}_{xy}))), \tag{3}$$

where the input $\bar{\mathbf{I}}(x,y)$ and $\mathbf{p}_{xy}$ are the same as those in Eq. (1). Because the output covariance matrix is symmetric, we only use the half of the matrix. h is a function that converts the 6 dimensional output to the 3×3 covariance matrix. To ensure that the covariance matrix is positive-definite, we replace zero or negative eigenvalues of the matrix with a small positive value. Our MLP f is trained by minimizing the following cost function:

$$\begin{aligned} L = \frac{1}{N}\sum_i \| h^{-1}(\boldsymbol{\Sigma}(\bar{\mathbf{I}}_i(x,y),\mathbf{p}_{xy,i})) \\ - f(\bar{\mathbf{I}}_i(x,y),\mathbf{p}_{xy,i})\|^2, \end{aligned} \tag{4}$$

where h^{-1} is the inverse function of h. In our implementation, we use the ReLU as the activation function and stochastic gradient descent [17] as the optmization method. Also, the learning rate was set to 0.0001 and the training iteration was set to a million with 64 batch size. On average, the learning took 20 minutes with the machine with NVIDIA GTX Titan X GPU.

5. Experiments

5.1. Experimental Results

To verify the accuracy of our noise estimation, we compare the estimated noise covariance with the ground truth covariance computed from training set. We used the following distance measure introduced in [7] as the similarity between covariance matrices:

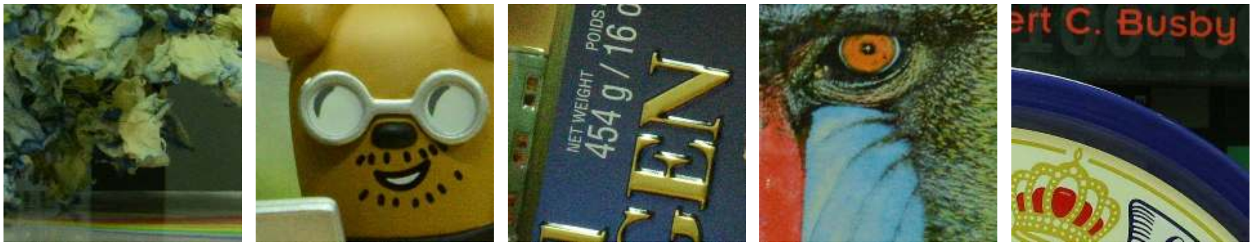
$$d(\mathbf{A},\mathbf{B}) = \sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i(\mathbf{A},\mathbf{B})}, \tag{5}$$

[2] Using the mean of temporal images has been used as noise-free image [19, 20].

[3] The input layer is the concatenation of RGB color (3) and vectorized 8×8 color patch (192).

Image #	Camera Settings			NLF [19] from GT		Ours	
	Camera	ISO	JPEG	Mean	Median	Mean	Median
1	Nikon D800	1600	Normal	4.86	4.88	**1.82**	**1.76**
2				4.60	4.49	**1.71**	**1.58**
3				5.61	5.41	**3.16**	**2.43**
Average				5.02	4.93	**2.23**	**1.92**
4	Nikon D800	3200	Normal	5.32	5.19	**2.39**	**1.89**
5				5.60	5.46	**2.03**	**1.89**
6				5.50	5.50	**2.31**	**1.92**
Average				5.47	5.38	**2.24**	**1.90**
7	Nikon D800	6400	Normal	5.80	5.72	**1.99**	**1.80**
8				6.05	6.16	**2.28**	**2.20**
9				6.04	6.04	**2.06**	**1.97**
Average				5.96	5.97	**2.11**	**1.99**
10	Nikon D600	3200	Normal	5.99	6.05	**1.68**	**1.63**
11				5.60	5.66	**1.83**	**1.70**
12				5.07	4.98	**1.59**	**1.44**
Average				5.55	5.56	**1.70**	**1.59**
13	Canon 5D Mark III	3200	Fine	3.72	3.39	**2.36**	**2.24**
14				4.75	4.65	**2.75**	**2.55**
15				4.80	4.81	**2.66**	**2.53**
Average				4.42	4.28	**2.59**	**2.44**

Table 1. Noise model evaluations for test images shown in Fig. 6. The values are the median and the mean value of covariance matrix errors in Eq. 5. Small value means better performance. Regardless of the scene, ISO and the camera, our model represent the noise more accurately.



(a) (b) (c) (d) (e)

Figure 6. Test images used in Table 1. From left to right, 1, 4, 7, 10, and 13 (bold).

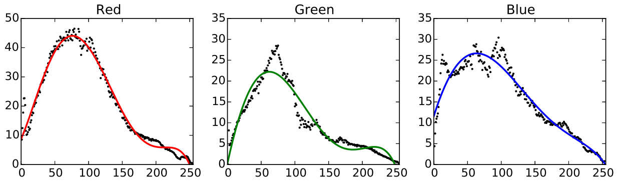


Figure 7. The NLF computed from our ground truth. The points are minimum variances of each intensity. We obtain the NLF by fitting the intensity-variance pairs of each channel.

where $\lambda_i(\mathbf{A}, \mathbf{B})$ is the i-th generalized eigenvalue of $\mathbf{Ax} = \lambda\mathbf{Bx}$.

Since no previous noise model can be fit to real data as ours, direct comparisons with previous models are difficult. The closest model that can be used is the noise level function (NLF) in [19], so we compared our model to NLF as shown in Table 1. For NLF, we computed NLFs separately for each color channel by obtaining the lower bound of intensity-variance pairs from 500 static images, which is the upper bound of noise as shown in Fig. 7. From the variances of three channels, we generated the covariance matrix with zero off-diagonal terms. The table validates our model esstimation process and also shows that it represents the noise in real images better than the NLF.

Figure 8 shows a qualitative analysis of our new noise model and its parameter estimation. It verifies that our multivariate Gaussian model which is an ellipsoid fits well with the observed data samples.

5.2. Image Denoising Application

To study the effectiveness of our model and its estimation method, we apply our method to image denoising, which is a process of estimating the true intensity corresponding to the scene radiance from the noisy observations. For image denoising, we adopt the Bayesian non-local means (BNLM) method in [14], which is an extension of the non-local means denoising algorithm [1] with more robust similarity measures. Non-local means algorithms denoise the

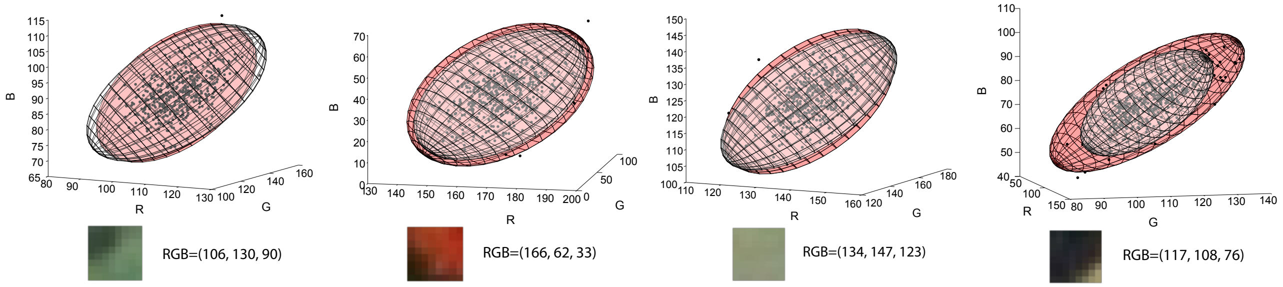


Figure 8. Some noise estimation results of our model. Each shows two covariance ellipsoids which are our estimation (red) and the ground-truth (white). Also, black dots are actual color points from 500 temporal images. Our estimation is quite accurate compared with the ground truth and fits well with the noise distribution of real JPEG image.

images based on the nature of self-similarity in natural images.

Let $\mathbf{I}$ be the noisy observation of a pixel (r, g, b) and $\bar{\mathbf{I}}$ be the denoised color. We can apply the BNLM to compute $\bar{\mathbf{I}}$, which is expressed as

$$\bar{\mathbf{I}}_i = \frac{\sum_{j\in N_i} e^{-\frac{1}{2}\left(\sqrt{d^2(i,j)}-\sqrt{2M-1}\right)^2}\mathbf{I}_j}{\sum_{j\in N_i} e^{-\frac{1}{2}\left(\sqrt{d^2(i,j)}-\sqrt{2M-1}\right)^2}}, \qquad (6)$$

where i and j are pixel positions, N_i is the set of neighboring pixels of i, and M is the number of pixels in patch times the number of channels. The squared dissimilarity measure $d^2(i, j)$ is originally the squared Euclidean distance normalized by σ^2. But, in our problem it is changed to the squared Mahalanobis distance, which is expressed as

$$\begin{aligned} d^2(i,j) &= (\mathbf{I}_{P_i} - \mathbf{I}_{P_j})^T \mathbf{\Sigma}_{P_i}^{-1} (\mathbf{I}_{P_i} - \mathbf{I}_{P_j}) \\ &= \sum_{d\in P} (\mathbf{I}_{i+d} - \mathbf{I}_{j+d})^T \mathbf{\Sigma}_{i+d}^{-1} (\mathbf{I}_{i+d} - \mathbf{I}_{j+d}), \end{aligned} \qquad (7)$$

where $\mathbf{I}_{P_i}$ and $\mathbf{I}_{P_j}$ are M dimensional vectorized patches whose center pixel is i and j, respectively and $\mathbf{\Sigma}_{P_i}$ is the covariance matrix of $\mathbf{I}_{P_i}$. $d^2(i, j)$ can be rewritten as the sum of each pixel distance, where P is the set of disparities d from center pixel within patch. For each pixel $i + d$, $\mathbf{\Sigma}_{i+d}$ is the 3×3 covariance matrix of the pixel.

Table 2 and Figure 9 show the experimental results of image denoising both quantitatively and qualitatively. BNLM denoising with our noise model is compared with BM3D [5], original BNLM [14], and BNLM with NLF noise model [19]. For BM3D and original BNLM, σ is computed by averaging the ground truth (GT) σ of every pixels in the whole image. For all noise models applied to BNLM, 5×5 patch and 35×35 search window are used, which are sufficient for both the quality and the time complexity.

For vast majority of cases, denoising using our noise model outperformed other models quantitatively. The quality of denoising using our noise model is more apparent in the qualitative examples shown in Figure 9. These experiments support the need for a new image noise model that are both color and content dependent. We would also like to point out that the experiments in Table 2 and Figure 9 are meaningful as the first noise evaluation on real image data (to the best of our knowledge) as most previous noise evaluations were done on either RAW images or on simulated data.

6. Conclusion

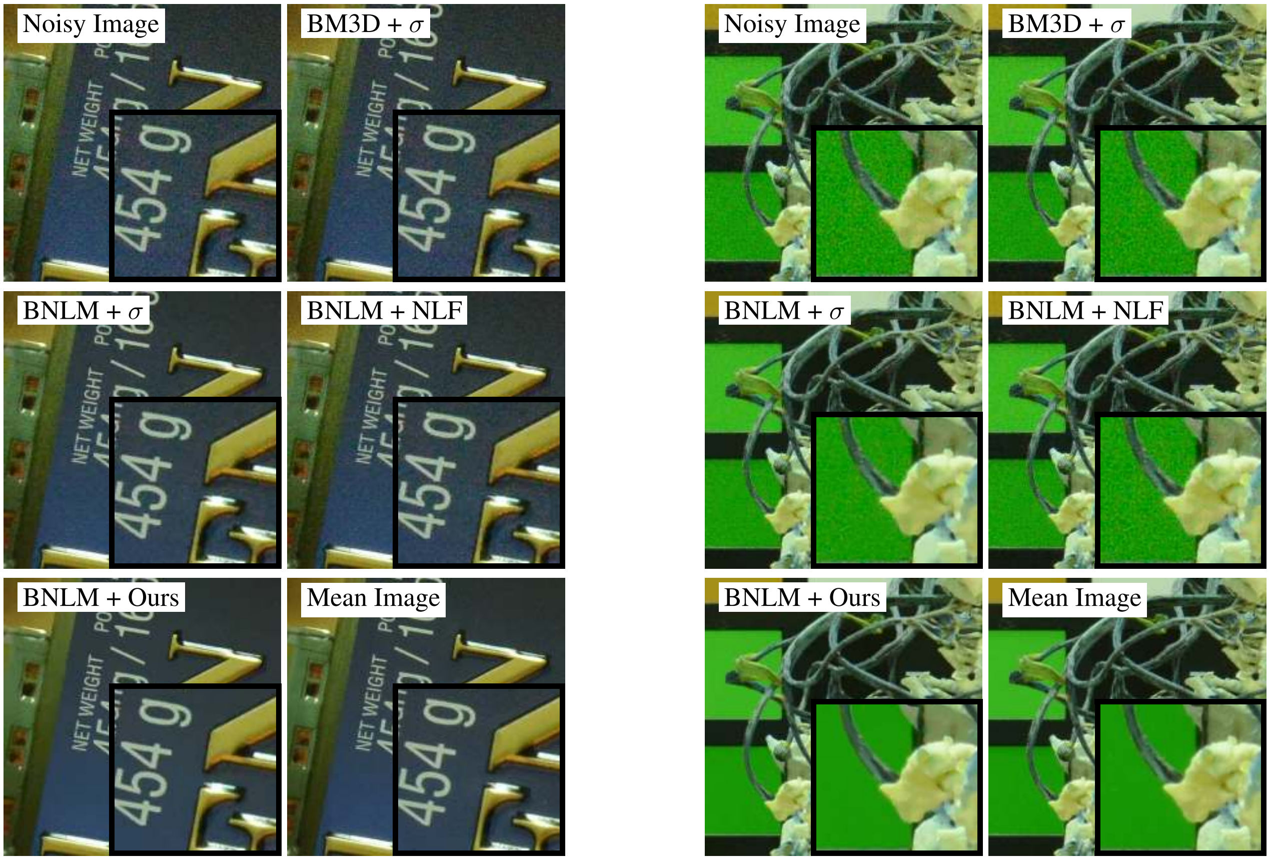
In this paper, we presented a new noise model in the 3D RGB space considering both the cross-channel dependency and the scene dependency of the noise in consumer camera images. We empirically showed that the noise characteristics change through the in-camera imaging process and the JPEG compression. The observations showed that a new noise model and the estimation method are necessary as previous noise models cannot explain those factors. To estimate the noise, we collected training image sets for various scenes and proposed a data-driven noise estimation algorithm using a multi layer perceptron. We validated our method using real images and applied it to image denoising, which showed large improvement over the previous work. In the future, we are interested in applying our work to other computer vision applications including radiometric calibration and HDR imaging.

Acknowledgement

This work was supported by Global Ph.D. Fellowship Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2015H1A2A 1033924), the center for Integrated Smart Sensors funded by the Ministry of Science, ICT & Future Planning as Global Frontier Project (CISS-2013M3A6A6073718), and Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (B0101-16-0552, Development of Predictive Visual Intelligence Technology).

Image #	Camera Settings		Noisy Image		BM3D [5]		BNLM [14]					
					σ from GT		σ from GT		NLF [19] from GT		Ours	
	Camera	ISO	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	Nikon D800	1600	35.47	0.957	36.15	0.964	37.59	0.980	36.61	0.972	**37.99**	**0.982**
2			35.71	0.954	36.57	0.964	39.42	0.990	37.61	0.981	**40.36**	**0.992**
3			34.81	0.989	35.47	0.991	37.40	0.995	35.91	0.993	**38.30**	**0.996**
Average			35.33	0.967	36.06	0.973	38.14	0.988	36.71	0.982	**38.89**	**0.990**
4	Nikon D800	3200	33.26	0.978	34.00	0.982	38.10	0.992	35.99	0.988	**39.01**	**0.993**
5			32.89	0.988	33.43	0.989	35.17	0.995	33.84	0.991	**36.75**	**0.996**
6			32.91	0.951	33.53	0.957	38.33	0.987	35.92	0.976	**39.06**	**0.990**
Average			33.02	0.972	33.65	0.976	37.20	0.991	35.25	0.985	**38.27**	**0.993**
7	Nikon D800	6400	29.63	0.862	29.97	0.872	33.35	0.954	31.91	0.933	**34.61**	**0.963**
8			29.97	0.921	30.33	0.928	32.25	0.967	30.94	0.950	**33.21**	**0.970**
9			29.87	0.914	30.21	0.921	32.67	0.962	31.13	0.940	**33.22**	**0.970**
Average			29.82	0.899	30.17	0.907	32.76	0.961	31.33	0.941	**33.68**	**0.968**
10	Nikon D600	3200	33.28	0.968	33.70	0.972	34.74	0.978	34.27	0.975	**34.98**	**0.979**
11			33.77	0.990	34.33	0.992	**36.20**	**0.995**	35.54	**0.995**	35.95	**0.995**
12			35.21	0.939	35.75	0.954	40.57	0.987	38.42	0.979	**41.15**	**0.989**
Average			34.09	0.966	34.59	0.973	37.17	0.987	36.08	0.983	**37.36**	**0.988**
13	Canon 5D Mark III	3200	37.00	0.976	37.79	0.984	**38.44**	0.986	37.97	0.987	38.37	**0.988**
14			33.88	0.983	34.34	0.986	35.27	0.988	34.39	0.986	**35.37**	**0.990**
15			33.83	0.977	34.27	0.979	34.78	0.982	34.13	0.979	**34.91**	**0.983**
Average			34.90	0.979	35.47	0.983	36.16	0.985	35.50	0.984	**36.22**	**0.987**

Table 2. Denoising performance comparisons. In vast majority of cases, our noise model outperforms other models for denoising in both the PSNR and the SSIM.



(a) (b)

Figure 9. Qualitative denoising performance comparisons. (a) and (b) are data 7 and 8 in Table 2, respectively. In addition to the quantitative values, we can observe that denoising with our noise model outperform others in this example.

References

[1] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[2] H. Burger, C. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[3] A. Chakrabarti, Y. Xiong, B. Sun, T. Darrell, D. Scharstein, T. Zickler, and K. Saenko. Modeling radiometric uncertainty for vision with tone-mapped color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2185–2198, 2014.

[4] I. Choi, S. Kim, M. Brown, and Y.-W. Tai. A learning-based approach to reduce JPEG artifacts in image matting. In *Proc. IEEE International Conference on Computer Vision*, 2013.

[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, Aug 2007.

[6] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008.

[7] W. Förstner and B. Moonen. A metric for covariance matrices. In *Geodesy-The Challenge of the 3rd Millennium*, pages 299–309. Springer, 2003.

[8] M. Granados, B. Ajdin, M. Wand, and C. Theobalt. Optimal HDR reconstruction with linear digital cameras. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[9] M. Grossberg and S. Nayar. Modeling the space of camera response functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1272–1282, Oct 2004.

[10] G. E. Healey and R. Kondepudy. Radiometric CCD camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, 1994.

[11] M. J. R. Healy. Multivariate normal plotting. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 17(2):157–161, 1968.

[12] Y. Hwang, J. S. Kim, and I. S. Kweon. Difference-based image noise modeling using skellam distribution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1329–1341, 2012.

[13] N. Joshi, C. L. Zitnick, R. Szeliski, and D. Kriegman. Image deblurring and denoising using color priors. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1550–1557. IEEE, 2009.

[14] C. Kervrann, J. Boulanger, and P. Coup. Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal. In *Proc. Conf. Scale-Space and Variational Meth.*, 2007.

[15] S. J. Kim, H. T. Lin, Z. Lu, S. Susstrunk, S. Lin, and M. S. Brown. A new in-camera imaging model for color computer vision and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2289–2302, 2012.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems*, 2012.

[17] Y. LeCun, L. Bottou, G. Orr, and K. Muller. Efficient backprop. In G. Orr and M. K., editors, *Neural Networks: Tricks of the trade*. Springer, 1998.

[18] Y. Li, F. Guo, R. T. Tan, and M. S. Brown. A contrast enhancement framework with JPEG artifacts suppression. In *Proc. European Conference on Computer Vision*, 2014.

[19] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman. Automatic estimation and removal of noise from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):299–314, 2008.

[20] X. Liu, M. Tanaka, and M. Okutomi. Practical signal-dependent noise parameter estimation from a single noisy image. *IEEE Transactions on Image Processing*, 23(10):4361–4371, Oct 2014.

[21] N. Ohta. A statistical approach to background subtraction for surveillance systems. In *Proc. IEEE International Conference on Computer Vision*, pages 481–486, 2001.

[22] P. L. Rosin. Thresholding for change detection. In *Proc. IEEE International Conference on Computer Vision*, pages 274–279, 1998.

[23] Y. Tsin, V. Ramesh, and T. Kanade. Statistical calibration of ccd imaging process. In *Proc. IEEE International Conference on Computer Vision*, July 2001.

[24] C. Wren, A. Azarbaygaui, T. Darrel, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:780–785, July 1997.

[25] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Proc. Advances in Neural Information Processing Systems*, 2012.