Image Denoising using Dark Frames

Rahul Garg

December 18, 2009

1 Introduction

In digital images there are multiple sources of noise. Typically, the noise increases on increasing ths ISO but some noise is still observable at lower ISOs as well, especially in underexposed regions of the image. Also, it is often the case that there are patterns in noise specific to the camera being used. While it's true that on increasing the ISO, random noise dominates; fixed pattern noise is more observable at lower ISOs. While there exists a large body of work that models noise as independent and Guassian at every pixel, it is not completely true, especially at lower ISOs.

Camera noise can be observed in isolation by capturing dark frames, i.e., by taking images with shutter closed or lens cap on. A naive method to remove noise is to simply subtract the dark frame from a captured image [2]. One can improve upon it by capturing a number of dark frames, taking their average and then subtracting it from the captured image. However, in this work we aim to model the noise statistics better and couple it with natural image statistics to perform denoising.

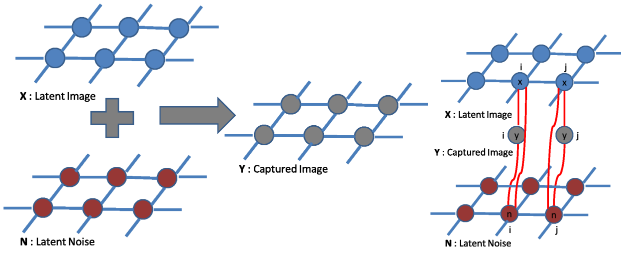
Image denoising is a hot area of research in image processing community. However, the focus of majority of work has been to model natural image statistics while assuming Gaussian per pixel independent noise [3]. Here, we take a complimentary approach where we primarily focus on learning and modeling noise.

2 Overview

We assume that in the image formation process, noise is additive (Figure 1a). The captured image $\mathbf{Y}$ is generated by adding together latent (denoised) image $\mathbf{X}$ and latent noise $\mathbf{N}$, i.e.,

$$\mathbf{Y} = \mathbf{X} + \mathbf{N}$$

The corresponding graphical model is shown in Figure 1b. Note that the graphical model assumes only pairwise interactions between the pixels in the latent noise and the latent image. From the graphical model, one can write



(a)

(b)

Figure 1: 1a shows image formation model. $\mathbf{X}$ is the latent image, $\mathbf{N}$ is the latent noise, which add to give the captured image $\mathbf{Y}$. 1b shows the corresponding graphical model.

$$P(\mathbf{X}, \mathbf{Y}, \mathbf{N}) = \frac{1}{Z} \prod_{i,j} (\phi(n_i, n_j)) \prod_{i,j} (\phi(x_i, x_j)) \prod_{i} (\phi(x_i, y_i, n_i))$$

where the first two products are over all pairs of adjacent pixels (in noise image and original image respectively) and the third product runs over all pixels involving triplets of noise, image and captured data. However, under our additive model $\phi(x_i, y_i, n_i)$ is non zero only if $y_i = x_i + n_i$. Under that assumption, one can write the above model in simplified equivalent form:

$$P(\mathbf{X}, \mathbf{Y}, \mathbf{N}) = \frac{1}{Z'} \prod_{i,j} (\phi(n_i, n_j)) \prod_{i,j} (\phi(y_i - n_i, y_j - n_j))$$

Given $\mathbf{Y}$, we want to infer $\mathbf{X}$ and $\mathbf{N}$ that maximize the above probablity, or equivalently minimize the negative log of above probability. Hence the solution is given by

$$\arg\min_{\mathbf{N}} (-\sum_{i,j} log(\phi(n_i, n_j)) - \sum_{i,j} log(\phi(y_i - n_i, y_j - n_j)))$$

or, one can write it as

$$\arg\min_{\mathbf{N}} \sum \psi(n_i, n_j)) + \sum \psi(y_i - n_i, y_j - n_j)$$

where $\psi(n_i, n_j)$ and $\psi(y_i - n_i, y_j - n_j)$ are arbitrary functions. Note that while the first term captures the statistics of noise, the second term depends on the statistics of natural images. Lets look at each of them separately.

1. **Noise statistics**

   The naive technique of subtracting dark frames suggests that certain pixels are likely to have more noise than other pixels. Given a number of dark frames, one can calculate the probablility distribution over noise for each pixel, i.e., one can calculate $P_i(n_i)$. Given this, one simple choice of $\psi(n_i, n_j)$ is

   $$\psi(n_i, n_j) = -log(P_i(n_i)) - log(P_j(n_j))$$

   However, noise in adjacent pixels tends to be correlated. Hence, one can also add another term to the above function that encourages adjacent pixels to have similar noise.

   $$\psi(n_i, n_j) = -log(P_i(n_i)) - log(P_j(n_j)) + f_{i,j}(n_i, n_j)$$

   For e.g. a very simple choice for $f_{i,j}(n_i, n_j)$ could be $|n_i - n_j|$. However, ideally one would like to calculate a separate $f_{i,j}$ for every pair of adjacent pixels, i.e., learn the distribution $P_{i,j}(n_i, n_j)$ for every pair of adjacent pixels i and j. This would also require a large amount of training data.

2. **Image statistics**

   Gradients in natural image are believed to have a sparse distribution (A heavy tailed Gaussian to be precise [7]). Hence, a common prior imposed on natural image gradients is the sparsity prior. Motivated by that, we use the following simple function

   $$\psi(y_i - n_i, y_j - n_j) = \lambda|(y_i - n_i) - (y_j - n_j)|$$

   where λ contains the relative importance being given to the image statistics vs. the noise statistics. In the actual implementation, the above is made robust by putting an upper threshold on the function.

3 Implementation Details

3.1 Generating Statistics

I used a Canon Rebel XTi camera which is known to exhibit significant banding noise. 168 dark frames were captured in raw format and were converted to 16 bit TIFFs. An ISO setting of 400 was used which is high enough to exhibit some banding noise yet low enough to avoid random sensor noise. Aperture was set to $f8.0$ to make sure that it is not near either of the extreme ends where lenses typically show abnormal behaviour. An example dark frame (with intensity boosted 50 times to make the noise visible) is shown in Figure 2. Similar settings were used for capturing an actual image used as a test image for denoising results (Figure 4).

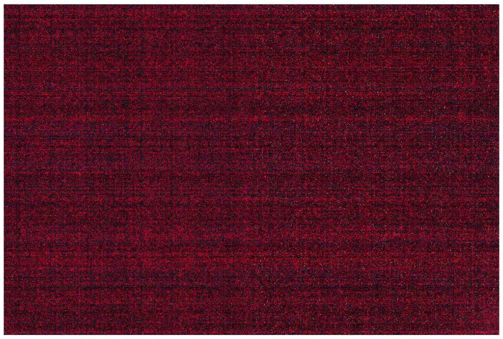


Figure 2: An example dark frame with intensity boosted 50 times.

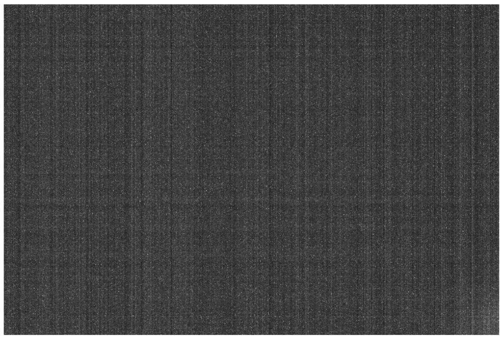


Figure 3: Average noise in red channel of dark frames (boosted 50 times).

Then for every pixel, a frequency histogram over noise values was computed. Noise values upto 2550 were considerer (in 16 bit format) and were scaled down by a factor of 255 to reduce them to 8 bit format. More precisely $P_i(n_i)$ was computed as *number of dark frames with value n_i at the i^{th} pixel* divided by the total number of dark frames. Since the noise has been scaled down to 8 bits, $n_i \in \{0, 1, 2, ..., 9\}$.

Also, the average noise for red channel is shown in Figure 3 showing that significant vertical banding exists for this particular camera. The bands are still prominent after averaging over 168 frames hint that these bands tend to occur at specific places.

Statistics were computed independently for the three channels. MATLAB was used to implement this part of the project and the computed statistics (frequency histograms) were written down to text files.

3.2 Running Inference

From Section 2, the function which we aim to minimize for infering the MAP solution is given by

$$\sum_{i,j}(-log(P_i(n_i)) - log(P_j(n_j)) + f_{ij}(n_i, n_j))$$

$$+\lambda \sum_{i,j} |(y_i - n_i) - (y_j - n_j)|$$

It was mentioned in Section 3.1 how $P_i(n_i)$'s are learnt. However, even 168 dark frames are not enough to learn $f_{ij}(n_i, n_j)$ reliably (Assuming that the noise lies in the range $\{0, 9\}$, one requires to compute a frequency distribution over 100 values, suggesting that we need much more than 168 frames to come up with reliable estimates). It was also mentioned in Section 2 that a simple choice could be to set $f_{ij}(n_i, n_j) = |n_i - n_j|$. However, I feel that it's not a good choice as f_{ij} appears to be highly dependent on the location of the pixels. For e.g., the presence of vertical banding suggests that the correlation is stronger in vertical direction that in horizontal direction. One may get away by learning correlation in horizontal and vertical direction separately. But again, since the bands occur at specific locations, the vertical correlation itself varies with columns. Would learning vertical correlation for each column (same across the complete length of column) and similar horizontal correlation constant along the complete row work? For now, I chose to drop the correlation term altogether and the function which I hence minimize is

$$\sum_{i,j}(-log(P(n_i))-log(P(n_j))+\lambda \sum_{i,j} |(y_i-n_i)-(y_j-n_j)|$$

As mentioned before, an image was captured using the same settings. The RAW image was converted to an 8 bit bitmap using linear scaling (It would be computationally inefficient to run inference on 16 bit images as noise value would range from 1 to 2550, which reduces to 1 to 10 in case of 8 bit images).

For running the max-product loopy belief propagation I used the code made available online by Szeliski et. al. [4] which is optimized for grid graphs used commonly in images. The code provides several methods to do inference (LBP [5], Graph cuts [1], Tree Reweighted Message Passing (TRW) [6], etc.). While certain methods like Graph Cuts which rely on the objective function being submodular are not applicable in this case as the objective function here may be non-submodular, I used a variant of the max-product LBP from the library.

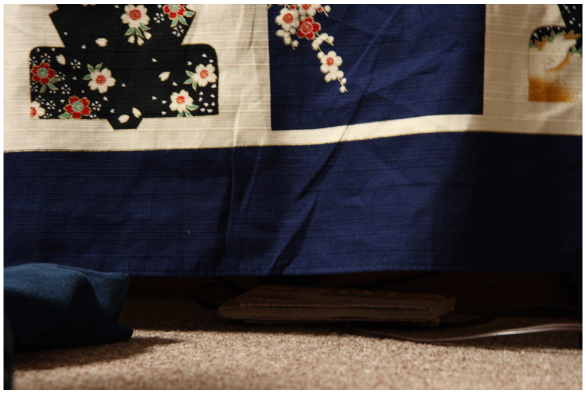
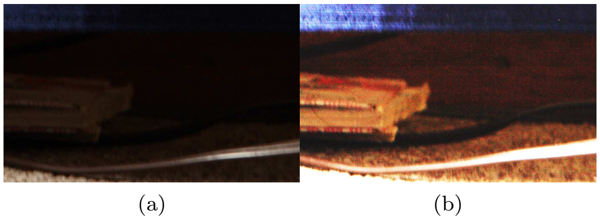


Figure 4: Test image to test denoising.



(a)

(b)

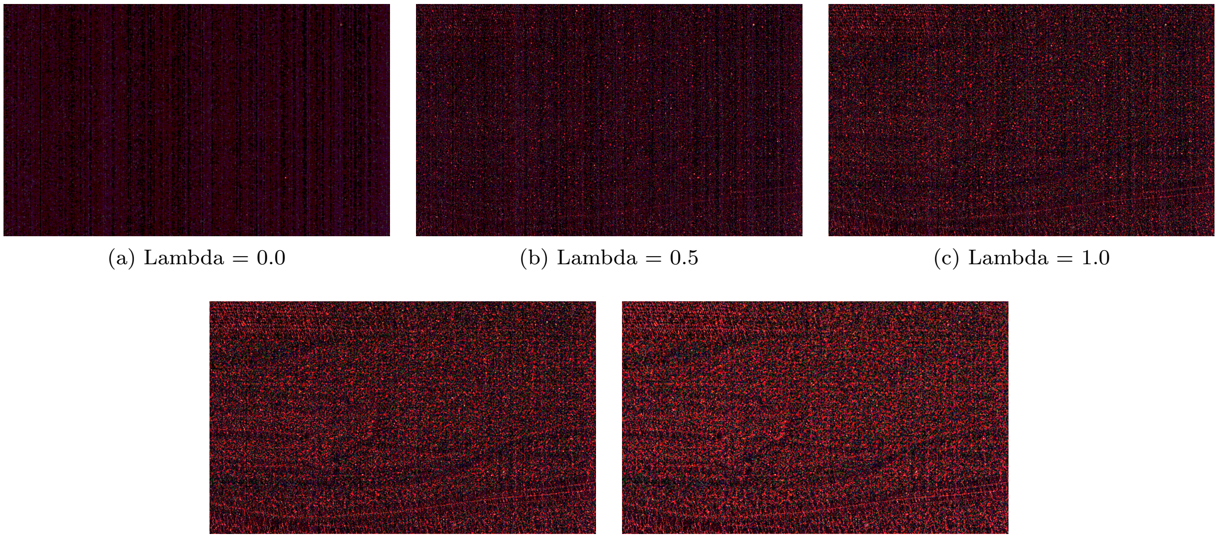
Figure 5: A crop of of bottom right corner of the image shown in Figure 4. Contrast and brightness has been increased in the second image to make the noise visible.

4 Results

Figure 4 shows the image used for denoising purposes. In fact, I used only the bottom right corner of the image which is significantly underexposed (Figure 5a) and one can see noise on increasing the contrast (Figure 5b).

Figure 6 shows the recovered noise as λ is increased. Note that if $\lambda = 0.0$, the inference is equivalent to having the most likely noise at each pixel. In that sense, it is closer to subtracting the average noise from the image. As λ is increased, one can see the banding pattern in the noise starts disappearing as the image statistic term dominates and pulls the noise away from the noise statistic. In fact, for high enough λ, one starts to see image structures appearing in noise indicating that the denosing is over smoothing the image at this point.

Figure 7 shows crops of denoised image (The magnitude of noise is still too small to be seen, hence enlarged crops are shown and contrast has been further increased). While results with $\lambda = 0.5$ and $\lambda = 1.0$



(a) Lambda = 0.0

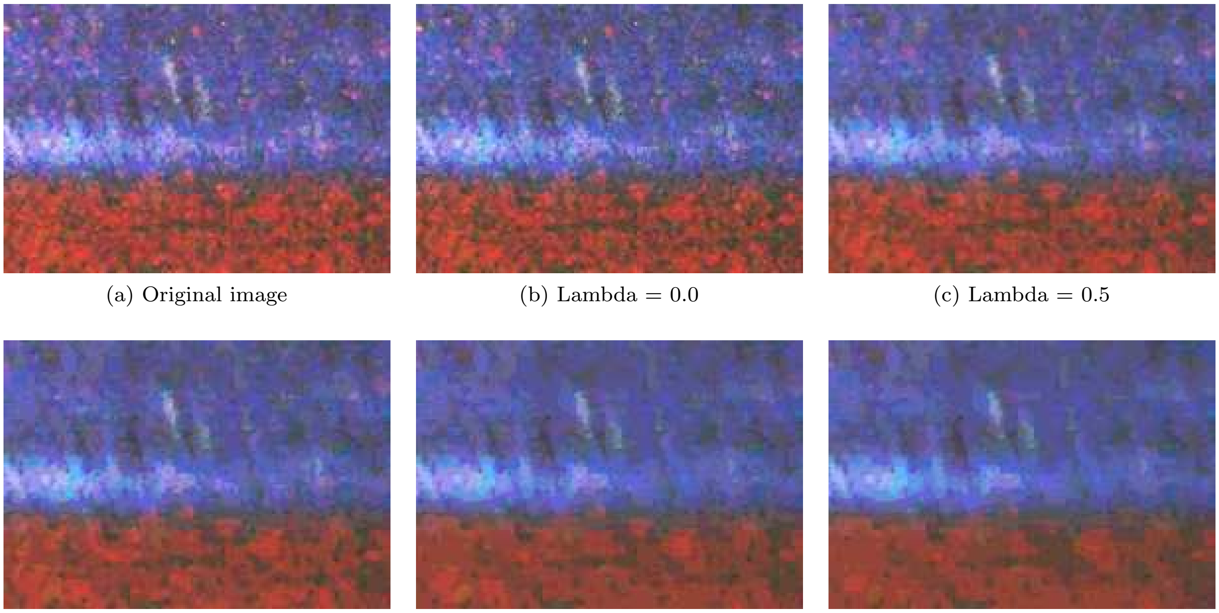
(b) Lambda = 0.5

(c) Lambda = 1.0

(d) Lambda = 2.0

(e) Lambda = 4.0

Figure 6: A crop of of bottom right corner of the image shown in Figure 4. Contrast and brightness has been increased in the second image to make the noise visible.



(a) Original image

(b) Lambda = 0.0

(c) Lambda = 0.5

(d) Lambda = 1.0

(e) Lambda = 2.0

(f) Lambda = 4.0

Figure 7: Denoised results with varying λ (zoomed in crop, contrast has been further increased to make the difference visible). Results with $\lambda = 0.5$ and $\lambda = 1.0$ are significantly better than those with $\lambda = 0.0$ (look at the red noise on blue bacground). However, increasing λ beyond that leads to observable smoothing of the output image.

are better than those with $\lambda = 0.0$, one starts to see noticable smoothing of image on increasing λ beyond that.

4.1 Running Time

Running loopy belief propagation takes on average 10 seconds for each channel for a 0.6 megapixel image. Hence it takes a total of 30 seconds for processing all channels but it can be easily parallelized as the processing of three channels is independent.

However, the statistics files are also quite large ($\sim$ 300 MB) and it takes a while to parse them. But that time can be easily cut down by storing the statistics in a format easier to parse than plain text.

5 Conclusion and Future Work

While it was encouraging to see some pattern in noise which goes against the commonly made assumption of independent Gaussian noise, the magnitude of noise is still too small to see in any appreciable difference in most images under normal circumstances (without stretching the contrast to extreme limits). Also, it seems that testing on ISO 100 is probably worthwhile where the banding noise would dominate even more.

But in general case, image noise would be sum of banding (patterned) noise augmented with random noise. Hence while simply removing the banding noise does not make much of difference, augmenting it with other denoising algorithms that remove random noise as well should result in a better performance.

The parameter λ, even though helpful, is bound to lead to some loss of sharpness of the image. Ideally, one would not want to make any assumption about statistics of the underlying image. However, we need some way to use the captured data, i.e., the evidence (For e.g. in our case, if we set $\lambda = 0$, inference is independent of $\mathbf{Y}$ which implies that it'll lead to the same solution for every image). I am not sure how exactly to go about it. One approach could be an interactive one, where the user specifies smooth regions of image so that some noise values can be grounded. With this evidence, one can hopefully infer the rest of noise values solely using the noise statistics. However, we might need to capture statistics over a larger neighborhood than simply pairwise interactions consequentially implying much larger training data.

There is another interesting aspect of this approach. Inference based denoising algorithms often directly used image intensities as labels implying that there are 256 possible labels for each pixel. However, if we use noise values as labels then we end up with a much smaller label set (10 in this case). Hence, there might be significant efficiency gains to be had from this approach. On the other hand, the objective function here is non-submodular disallowing certain fast algorithms like Graph Cuts.

References

[1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, Nov 2001.

[2] http://en.wikipedia.org/wiki/dark_frame_subtraction.

[3] Stefan Roth and Michael J. Black. Fields of experts: A framework for learning image priors. In *Proc. CVPR*, pages 860–867, Washington, DC, USA, 2005. IEEE Computer Society.

[4] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *PAMI*, 30(6):1068–1080, 2008.

[5] M.F. Tappen and W.T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Proc. ICCV*, pages 900–906 vol.2, Oct. 2003.

[6] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. Map estimation via agreement on trees: message-passing and linear programming. *Information Theory, IEEE Transactions on*, 51(11):3697–3717, Nov. 2005.

[7] Y. Weiss and W.T. Freeman. What makes a good model of natural images? In *Proc. CVPR*, pages 1–8, June 2007.