Experimental report

Article: *Multi-Scale DCT Denoising,* Pierazzo, Nicola and Morel, Jean-Michel and Facciolo, Gabriele

Article: *Analysis and Extension of the Ponomarenko et al Method Estimating a Noise Curve from a Single Image*, Colom, Miguel and Buades, Antoni

Ponomarenko et al Method:

The Ponomarenko method is an algorithm whose objective is to estimate the noise of an image. It relies on the assumption that the noise is additive, signal-independent and Gaussian.

The underlying principle of the algorithm is to find low frequency patches, i.e. patches that are almost pixel uniform, without any edge. These patches are likely to contain only noise. Thus, the high frequency in those patches is used to determine the noise.

The noise is estimated through 5 key steps:

1. Compute the Discrete Cosine Transform (DCT) on every $w \times w$ patches of an image.
2. Classify the low and high frequencies of the DCT decomposition according to a threshold.
3. Compute an empirical variance based on the low frequencies on every $w \times w$ patches.
4. Select the K patches having the lowest empirical variance based on the low frequencies.
5. Compute the median of empirical variances of those K patches, based on high frequencies. This is the final variance of the noise of the image.

We can note that K can either be selected manually, or by repeating the step 4 multiple times.

Extension of the method:

The assumption of a signal-independent and Gaussian noise cannot always be hold, especially in case of very dark photos. The authors propose an extension of the Ponomarenko method for signal dependent noise. In this new method, the noise is supposed to have a Poisson distribution, whose mean is the value of the noiseless image.

The idea is to classify the $w \times w$ patches of an image into B bins, according to their mean values. Each bin is now composed of patches sharing the same noise. The authors then apply the Ponomarenko algorithm to each bin. We can notice that if $B = 1$, this extension is simply the Ponomarenko algorithm.

Experiments on the extension of the Ponomarenko method.

This extension now allows to compute an estimation of the noise per pixel intensity, as shown in the experiments below. We will apply the extension to two images: a contrasted image and a textured image.

Contrasted image

The first image is a white and black image. The applied noise is signal dependent:

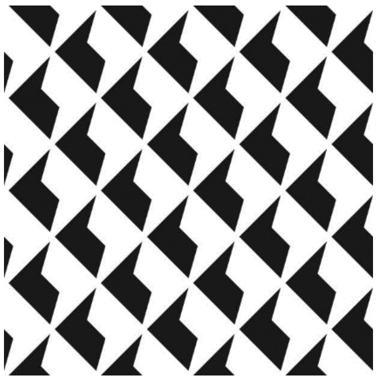

Figure 1: Original image

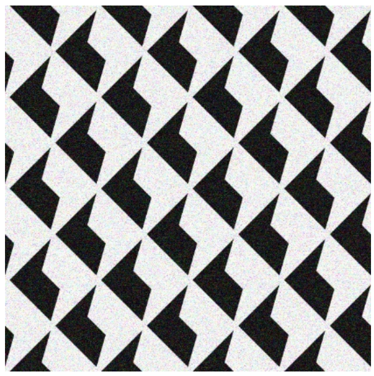

Figure 2: Noisy image

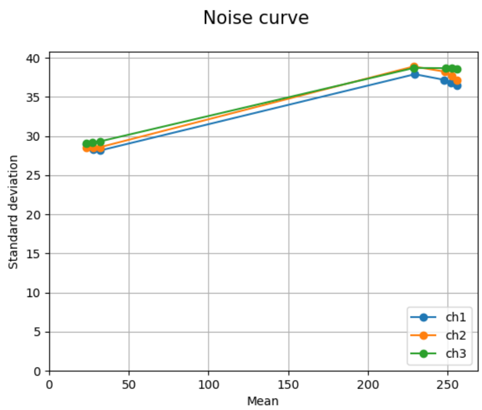

Figure 3: Noise curve for the noisy image

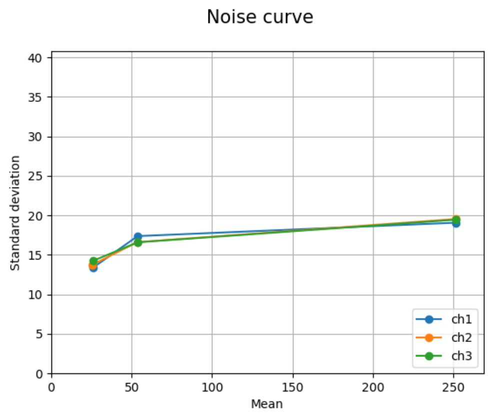

Figure 4: Noise curve for the noisy image with in smaller scale

Textured image

The next image is an image of a tree. The applied noise is signal-dependent.

Figure 5: Original image

Figure 6: Noisy image

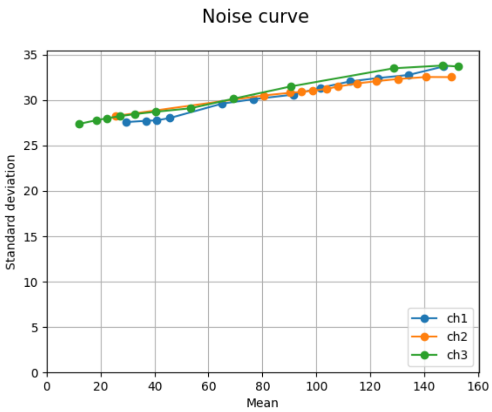

Figure 7: Noise curve for the noisy image

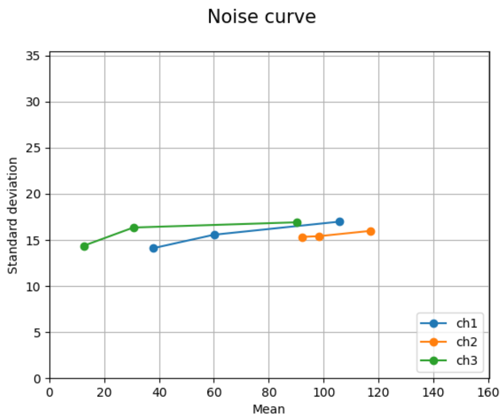

Figure 8: Noise curve for the noisy image with in smaller scale

<u>Interpretation of the results:</u>

The extension is good at estimating the signal-dependent noise. Indeed, the estimated standard deviation tends to increase with the increase of the pixel intensity. Moreover, we notice that, every time we use a smaller scale, the noise is divided by 2 on every channel. This is exactly how the ground truth noise behave.

We notice that the extension is good at smoothing the noise curve. Taking a good size for the patch, deleting the saturated pixels, and applying a curve filter all help to obtain an affine noise curve with respect to the intensity, like the ground truth model.

However, we notice that the curve noise is made for natural images. Indeed, the perfect noise curve for the white and black image has a hollow between intensities 20-255. This is not the case here. The resulting problem is that the curve filter will interpolate between values that don't exist in the original image.

Experiments on the Multi-Scale DCT Denoising

This algorithm provides an improvement of the classical DCT denoising algorithm. Indeed, this algorithm eliminates the low and high frequency noise using the same method, which can lead to poor results.

Contrasted image

Once again, the first image is the same white and black image. The noise is uniform. The images below are zooms in the top left corner.

Figure 9: Zoomed original image

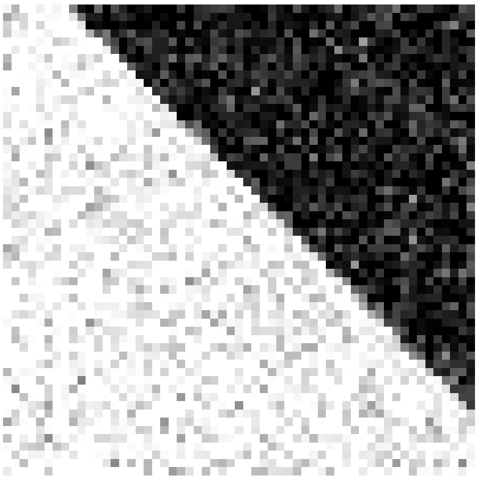

Figure 10: Zoomed noisy image

Figure 11: Zoomed DCT image on single scale

Figure 12: Zoomed DCT image on multiple scales (proposed algorithm)

Textured image

The next image is an image of a tree.

Figure 13: Original image

Figure 14: Noisy image

Figure 15: Single scale DCT denoising

Figure 16: Multiscale DCT denoising

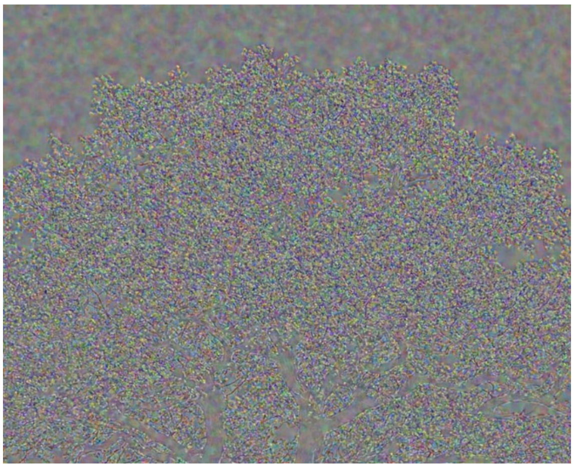

Figure 17: Difference Fig 14 – Fig 15

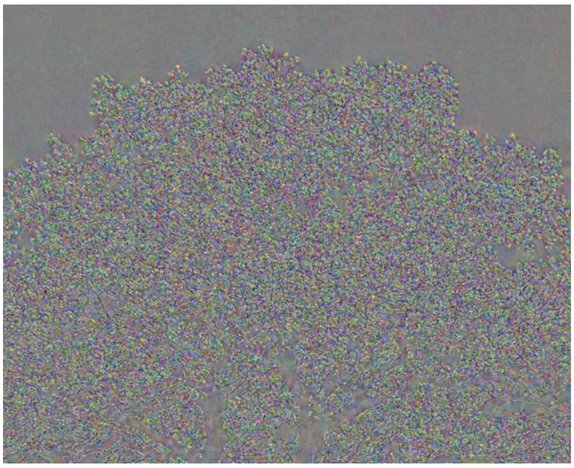

Figure 18: Difference Fig 14 - Fig 16

Interpretation of the results:

The experiment with the white and black image clearly shows a good improvement brought by the multi scale DCT denoising. The multiscale DCT denoises particularly well on uniform patches, where there is no edge nor texture. This improvement on the uniform spaces can be highlighted on Fig 17 and Fig 18. The multiscale DCT uniformly denoises every part of the noisy image that contains the sky. Thus, we can achieve very good denoising even between the tree leaves where the sky is visible. It can therefore be seen that multiscale DCT is continuously denoising to all noise frequencies.

In addition, we see that multi-scale denoising maintains good performance at the edges of the image, notably thanks to the aggregation algorithm. Finally, we notice that the PSNR is 0.5dB higher but this is already enough to see improvements on the uniform areas. If we modify the hyperparameters of the program, we realise that the gain offered by multiscale denoising is greater the smaller the patches used. Indeed, with large patches, single scale denoising will be able to understand low frequency noise and correct it.

However, if we compare to the original image, we still notice that the edges are less sharp, like in the first white and black image.