Project 1: Statistics of Natural Images

Yufei Hu (404944367)

January 19, 2018

1 Objectives

This is a small project aiming at verifying some properties of natural image statistics that were discussed in Chapter 2.

2 High kurtosis and scale invariance

For computational considerations, the image is firstly converted to grey level and re-scale the intensity to [0, 31] i.e. 32 grey levels. Convolve the image with a gradient filter $\nabla_x I$, i.e. the intensity difference between two adjacent (horizontally or vertically). I only pick one single image for the following steps.

Question 1 and 2:

Plot the histogram $H(z)$ for the difference against the horizontal axis $z \in [-31, +31]$. Then do a log-plot $\log H(z)$. Then compute the mean, variance, and kurtosis for this histogram.

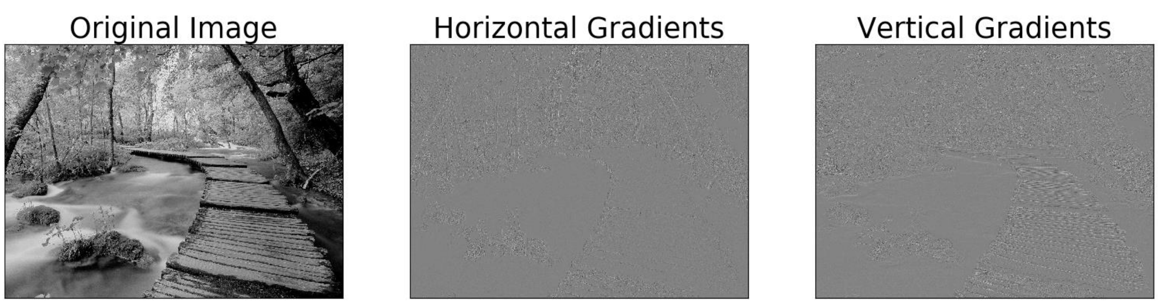

Figure 1: Original image together with its horizontal and vertical gradients

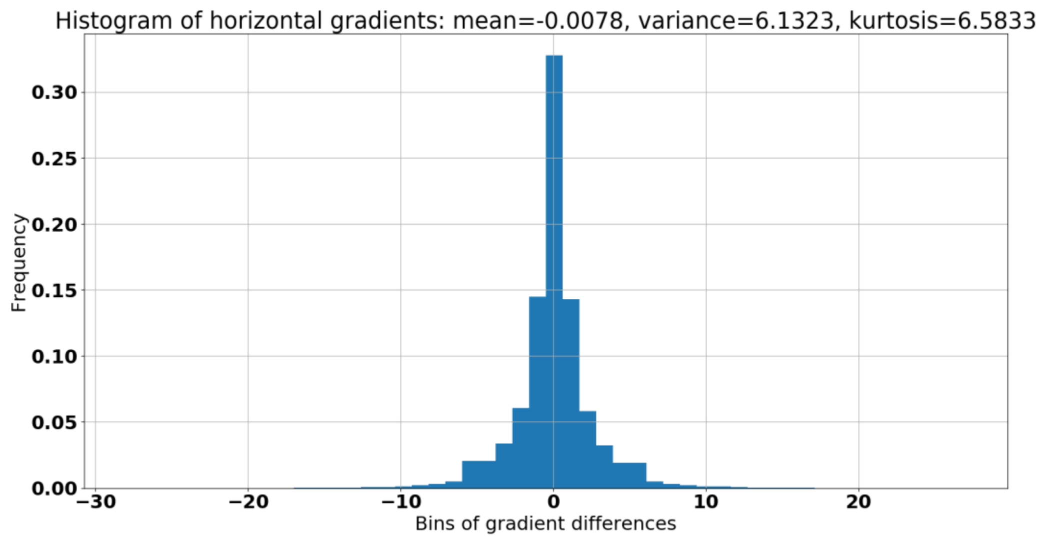

Figure 2: Histogram of horizontal gradients

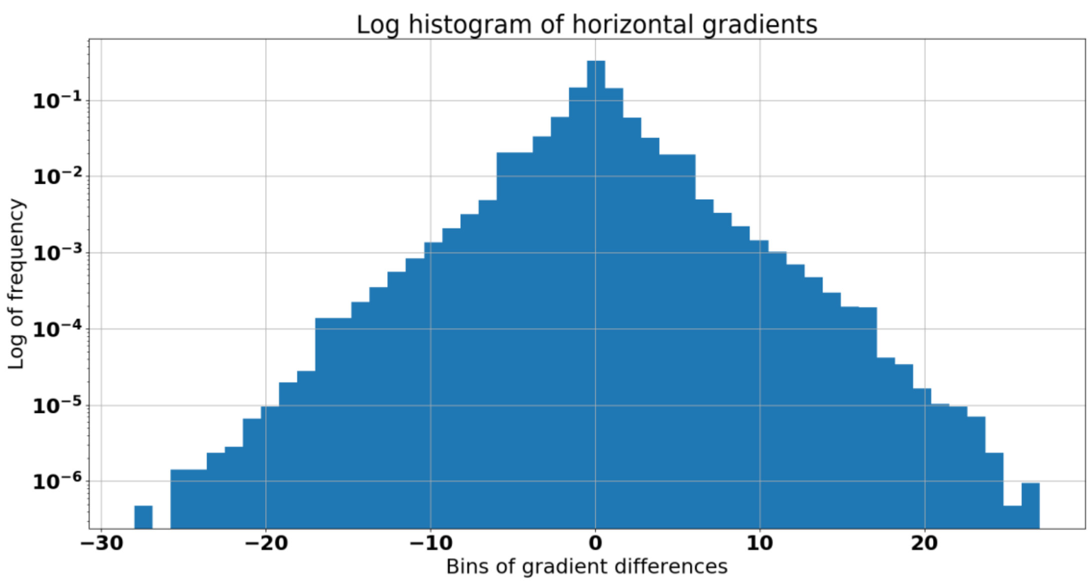

Figure 3: Histogram of horizontal gradients under log scale

According to Figure 2 and 3, most of the horizontal gradients sit in the range between -3 and +3, indicating that most frequency components of a natural image have a rather low frequency, i.e. the greyscale changes between adjacent pixels are rather gentle.

The three important statistics of the natural image are listed in the table below:

Table 1: mean, variance, and kurtosis of the horizontal gradients of the natural image

Mean	**Variance**	**Kurtosis**
-0.0078	6.1323	6.5833

Question 3:

Fit this histogram to a Generalized Gaussian distribution $e^{|z/\sigma|^{\gamma}}$ and plot the fitted curves super-imposed against the histogram. What is the value of γ in the fitted generalized Gaussian?

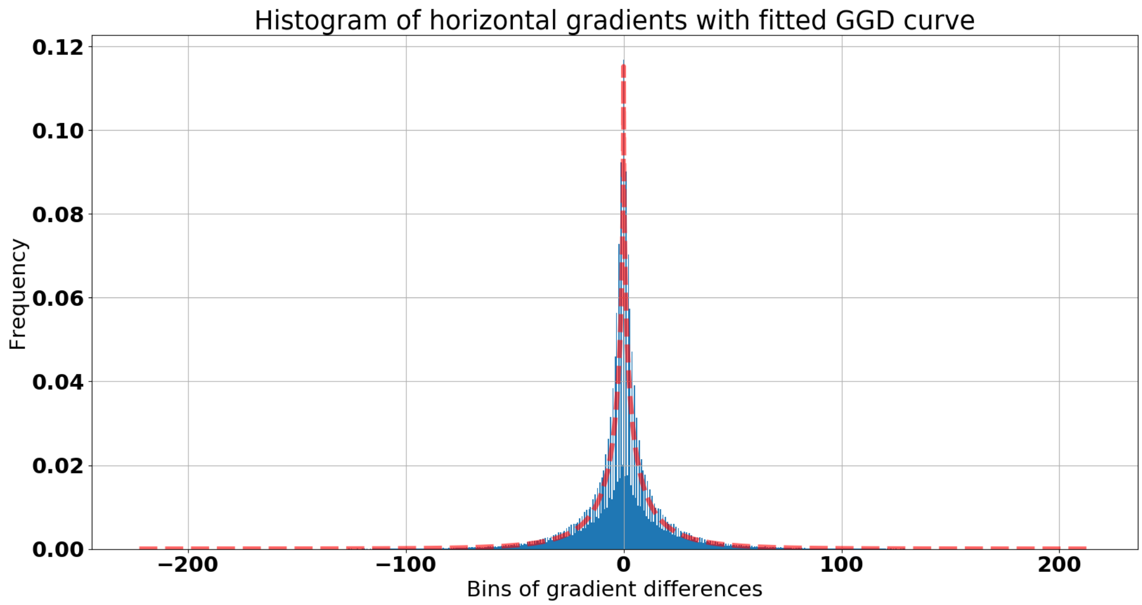

Figure 4: Histogram of horizontal gradients with fitted GGD curve

As seen in Figure 4, the fitted Generalized Gaussian distribution (GGD) curve matches with the histogram pretty well. Notice here I used the original horizontal gradients with values in the range between -255 and 255 to increase the histogram resolution. The necessity of doing so is because the GGD curve fits better on a data with higher distribution resolution. Besides, I have used 800 bins to plot the histogram.

The fitted γ is 0.5122, suggesting that the histogram is rather sharp.

Question 4:

Plot the Gaussian distribution using the mean and the variance above, and super-impose this plot with the plots in Figure 2 and 3.

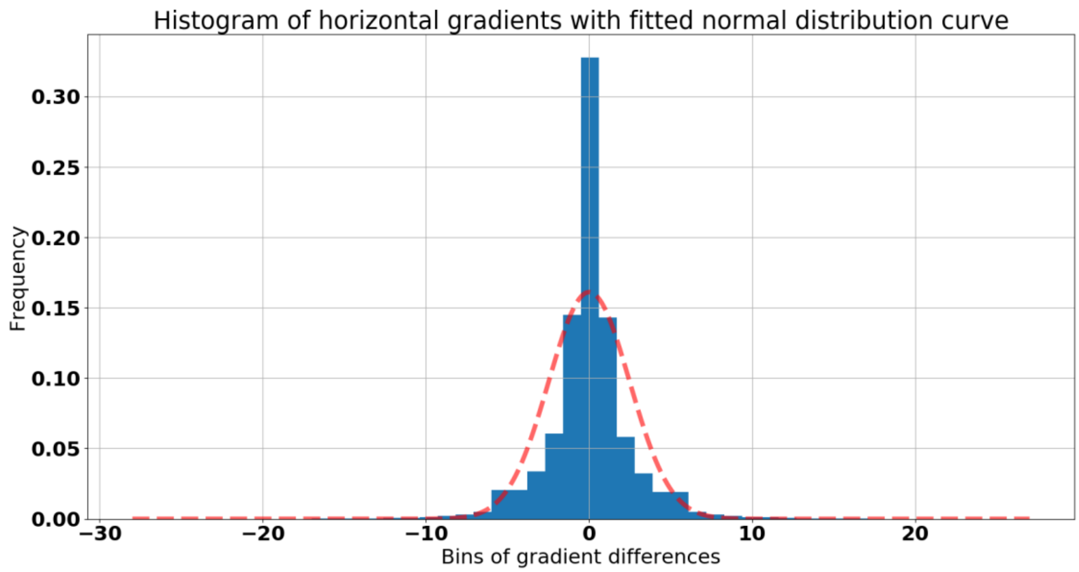

Figure 5: Histogram of horizontal gradients with fitted normal distribution curve

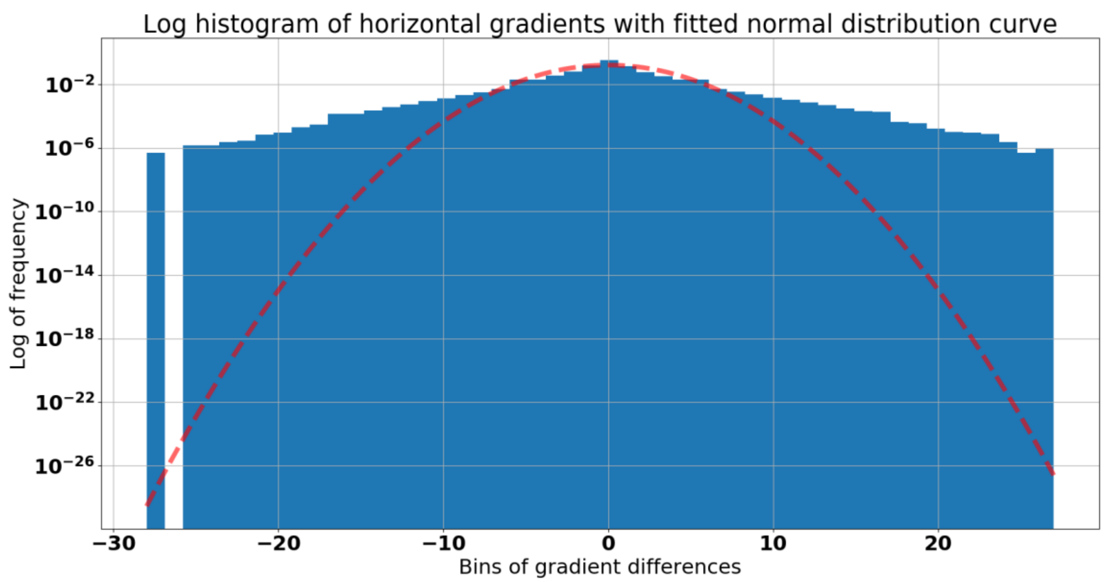

Figure 6: Histogram of horizontal gradients with fitted normal distribution curve under log scale

As shown in Figure 5 and 6, the normal distribution curve cannot fit the histogram well. This is even more obvious when the curve is shown under log scale. The reason is because the normal distribution only has two statistics to decide its shape while GGD has an extra γ to further adjust its shape.

Though by comparison with GGD, the normal distribution curve is not a perfect fit to the histogram of a natural image, it still manages to approximate the histogram in an efficient way considering its rather low computation complexity. This is probably the reason why many researchers tend to use normal distribution to study computer vision problem.

Question 5:

Down-sample the image by a 2×2 averaging. Plot the histogram and log histogram, and impose with the plots in Question 1, to compare the difference. Repeat this down-sampling process 2-3 times.

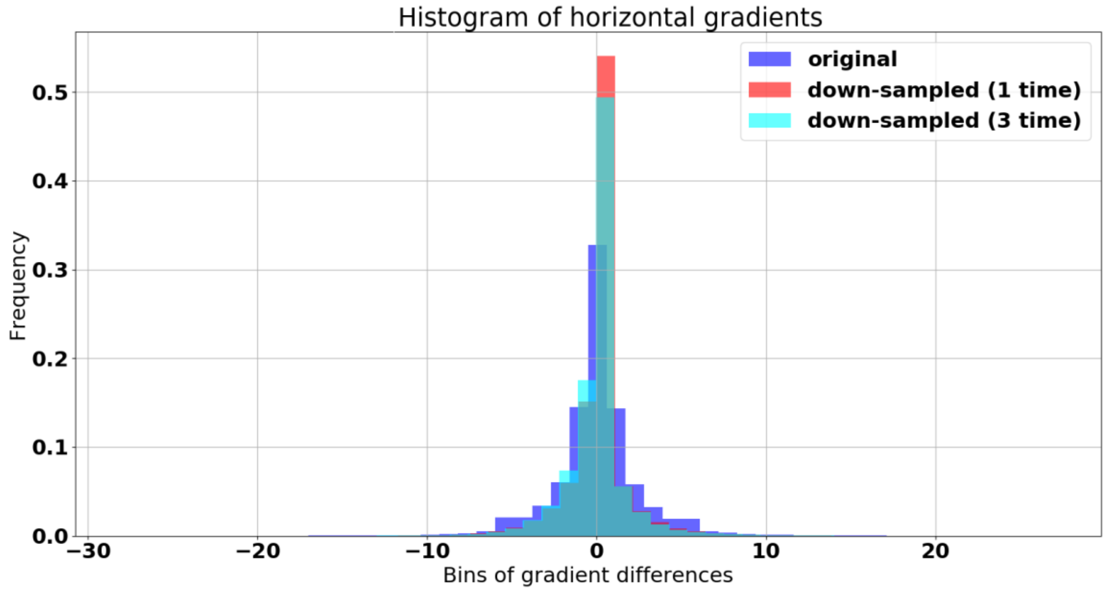

Figure 7: Histogram of horizontal gradients with down-sampled versions

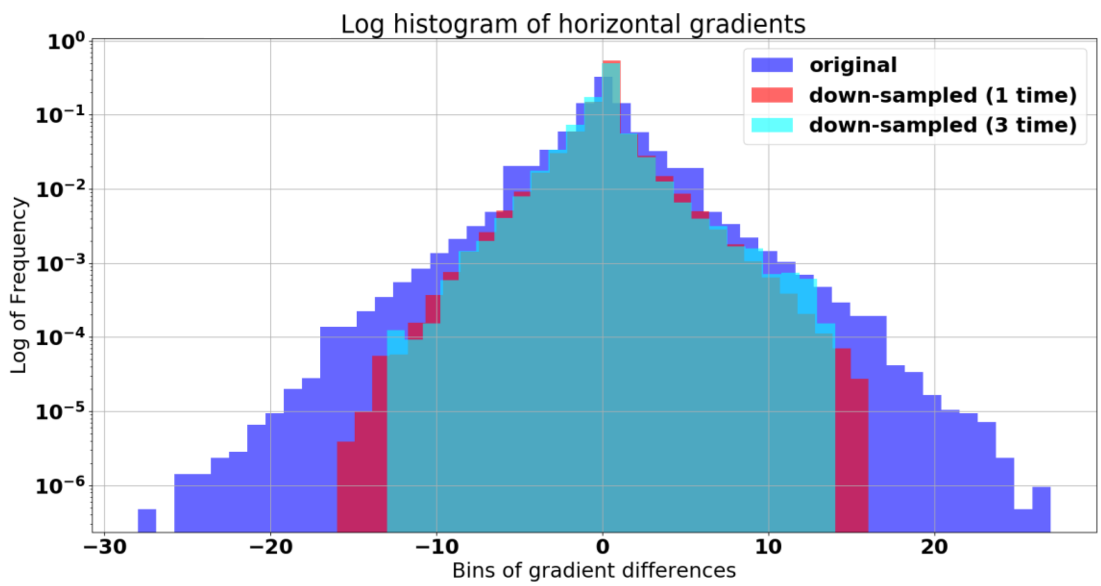

Figure 8: Histogram of horizontal gradients with down-sampled versions under log scale

As seen from Figure 7 and 8, the histograms of down-sampled horizontal gradients are very close to the original histogram. Under log scale, gradients difference tends to sit in a narrower range after down-sampling. The reason is because down-sampling keeps the low frequency components of the images almost the same while eliminates most high frequency components when the image resolution decreases.

My intuitive explanation is that when averaging a patch with almost the same pixel values, the averaged patch is almost the same to the original one. However, this is not true when averaging a patch whose pixel values vary sharply.

Question 6:

Synthesize a uniform noise image, i.e. each pixel is drawn independently from a uniform number in [0, 31]. Repeat 1-2-4 to compare the difference between a noise image and a natural image. Do a 2×2 average instead of sub-sampling.

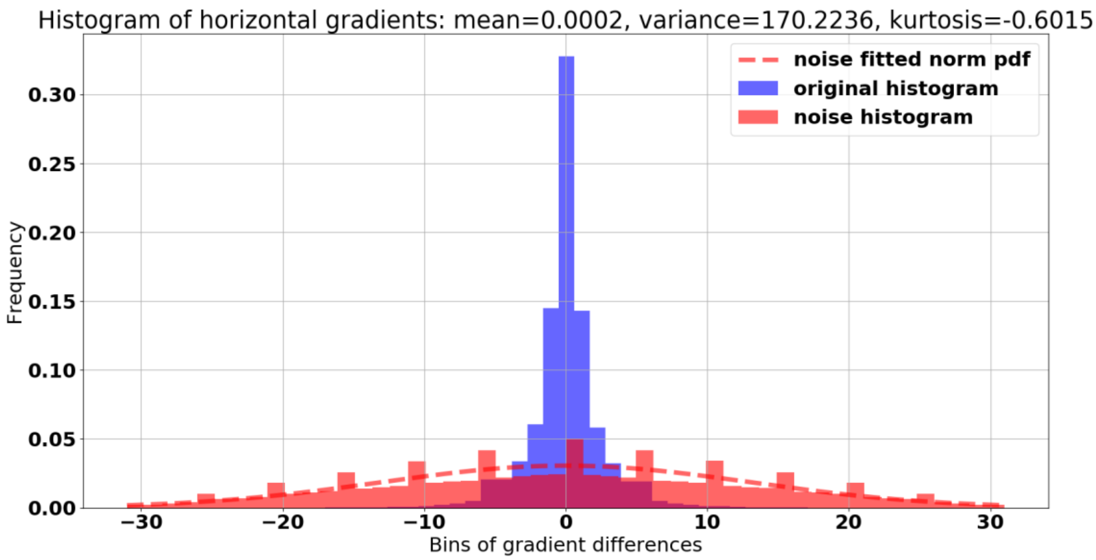

Figure 9: Histogram of horizontal gradients of the noise image

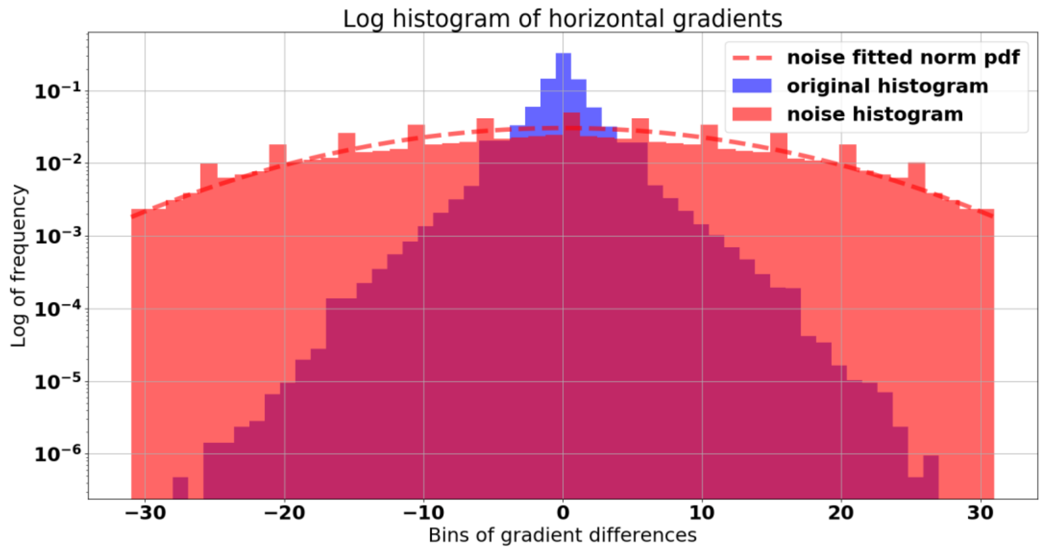

Figure 10: Histogram of horizontal gradients of the noise image under log scale

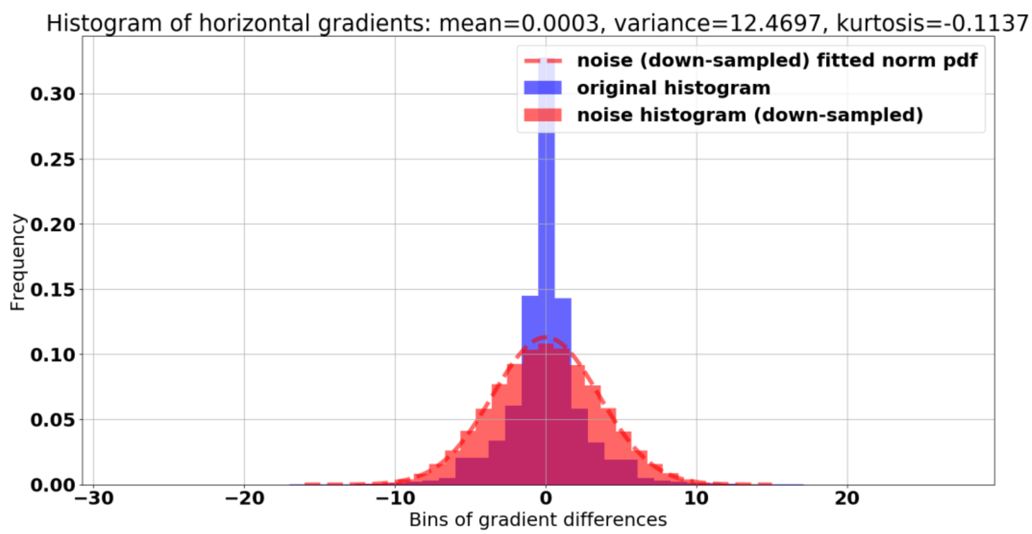

Figure 11: Histogram of horizontal gradients of the down-sampled noise image

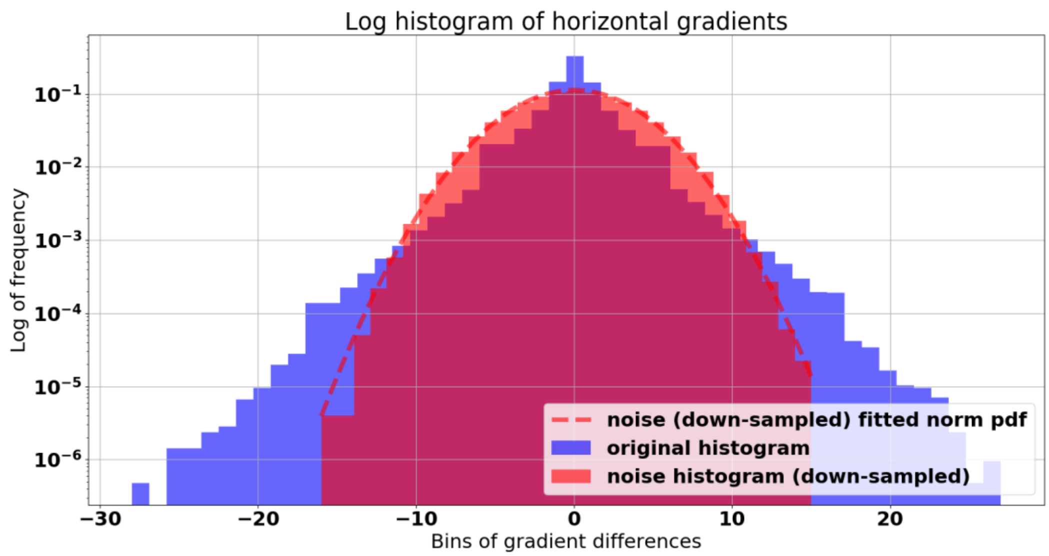

Figure 12: Histogram of horizontal gradients of the down-sampled noise image under log scale

According to Figure 9 and 10, the generated noise image has gradients distributed widely. The reason is because every pixel value is generated randomly following a uniform distribution.

However, after down-sampling the noise image, the gradients are distributed more narrowly. The reason is because averaging smoothens the original image by eliminating all sudden pixel changes.

The important statistics are listed in the table below:

Table 2: Mean, variance, and kurtosis of the noise image and the down-sampled noise image

	Noise image	**Down-sampled noise image**
Mean	0.0002	0.0003
Variance	170.2236	12.4697
Kurtosis	-0.6015	-0.1137

3 Verify the 1/f power law observation in natural images

Do an FFT (Fast Fourier Transform) on the grey image I which returns a Fourier image (ξ, η) which is complex number matrix indexed by (ξ, η) for its horizontal and vertical frequencies. Compute the amplitude (modulus) of each complex number $A(\xi, \eta)$. Denote the frequency $f = \sqrt{\xi^2 + \eta^2}$, and transfer to a polar coordinate, and we calculate the total Fourier power $A^2(\xi, \eta)$ for each frequency (i.e. you need to discretize f, and calculate the $A^2(\xi, \eta)$ averaged over the ring for each f, stop f when the circle hits the boundary of the Fourier image).

Question 1:

Plot $\log A(f)$ against $\log f$. This should be close to a straight line for each image. Plot the curves for the 4 images in one figure for comparison.

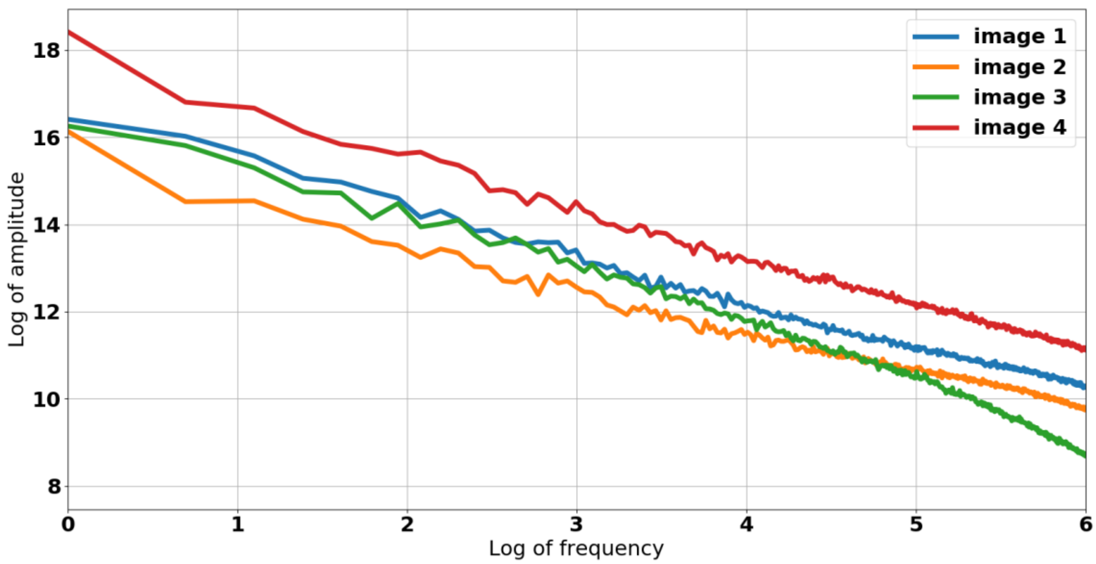

Figure 13: Log of amplitude over log of frequency

As seen in Figure 13, the log of amplitude of the frequency component over log of frequency is indeed close to a straight line. The reason is because in a natural image, high frequency component has relatively lower power than low frequency component. Besides, the linear relationship between them is interesting.

Question 2:

Compute the integration (summation in discrete case) of $S(f_0) = \int_\Omega A^2(\xi, \eta) d\xi d\eta$ over the domain:

$$\Omega(f_0) = \left\{ (\xi, \eta) : f_0 \le \sqrt{\xi^2 + \eta^2} \le 2 f_0 \right\}$$

Plot $S(f_0)$ over f_0, the plot should fit to a horizontal line (with fluctuation) as $S(f_0)$ is supposed to be a constant over f_0.

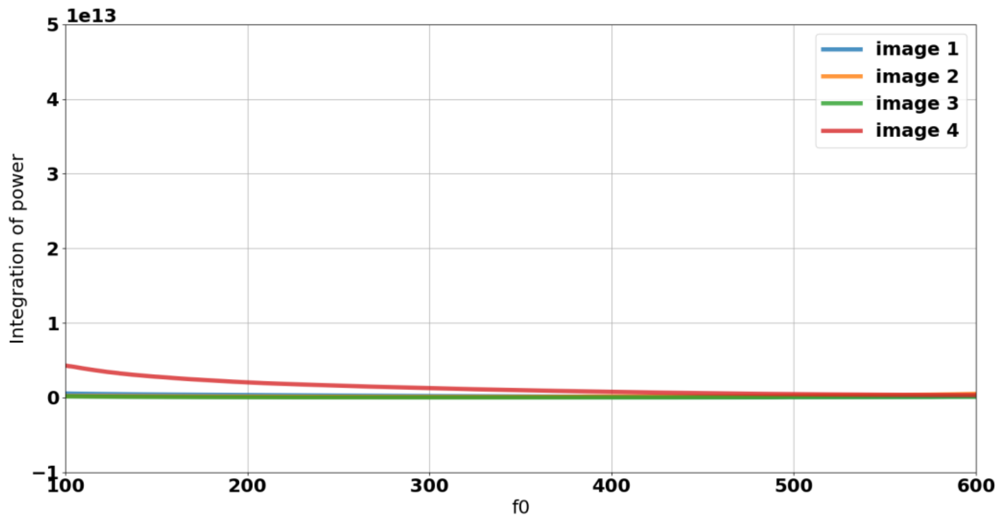

Figure 14: $S(f_0)$ over f_0

As seen from Figure 14, the integration of power is indeed a straight line under a rather large scale. However, after zooming in into a smaller region, the line is no longer very straight.

4 A 2D scale invariant world

Suppose we simulate a toy 2D world where the images consist of only 1D line segments. In an image, a line segment is represented by its center (x_i, y_i), orientation θ_i and length r_i. The line segments are independently distributed with uniform probability for their centers and orientations. The length follows a probability $p(r)$ follows $1/r^3$, i.e. a cubic power law. In each unit area, the number of line segments has a certain constant mean.

Question 1:

Simulate 1 image I_1 of size 1024×1024 pixels with a total N lines. (You need to record all the N lines whose centers are within a range of the image, truncate long lines and hide (discard) lines shorter than a pixel.)

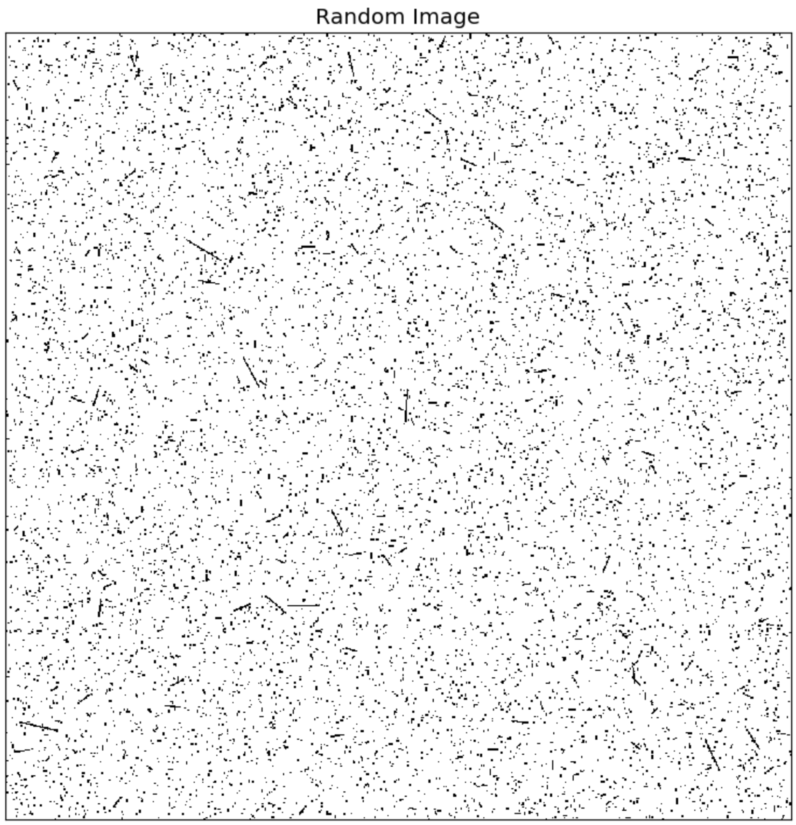

Figure 15: Synthesized random image

In Figure 15, the number of lines are 10,000.

Question 2:

Simulate 2 new images I2 and I3 of size 512×512 and 256×256 pixels respectively. I2 and I3 are down-sampled version of I1 and are generated by shortening the N line segments in I1 by 50% and 25% respectively (discard lines shorter than 1).

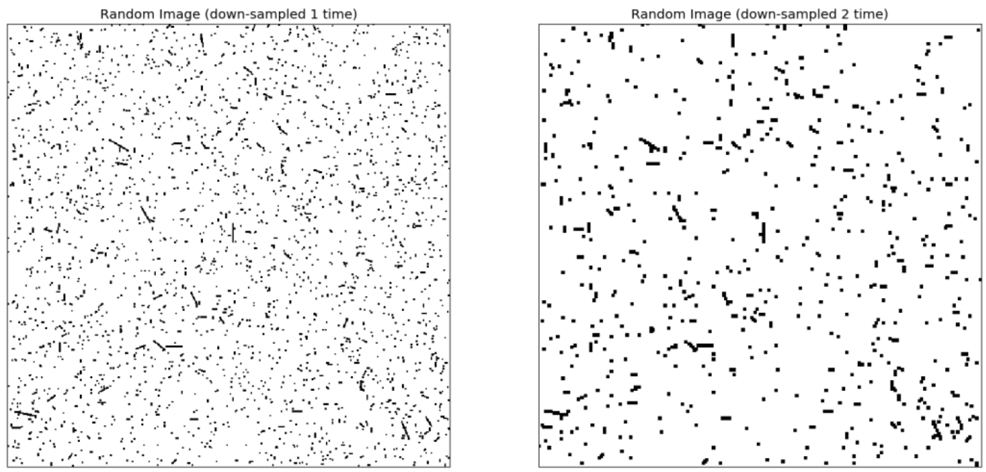

Figure 16: Down-sampled synthesized random images

Question 3:

Crop 2 image patches of size 128×128 pixels randomly from each of the three images I1, I2, I3 respectively. Plot these six images

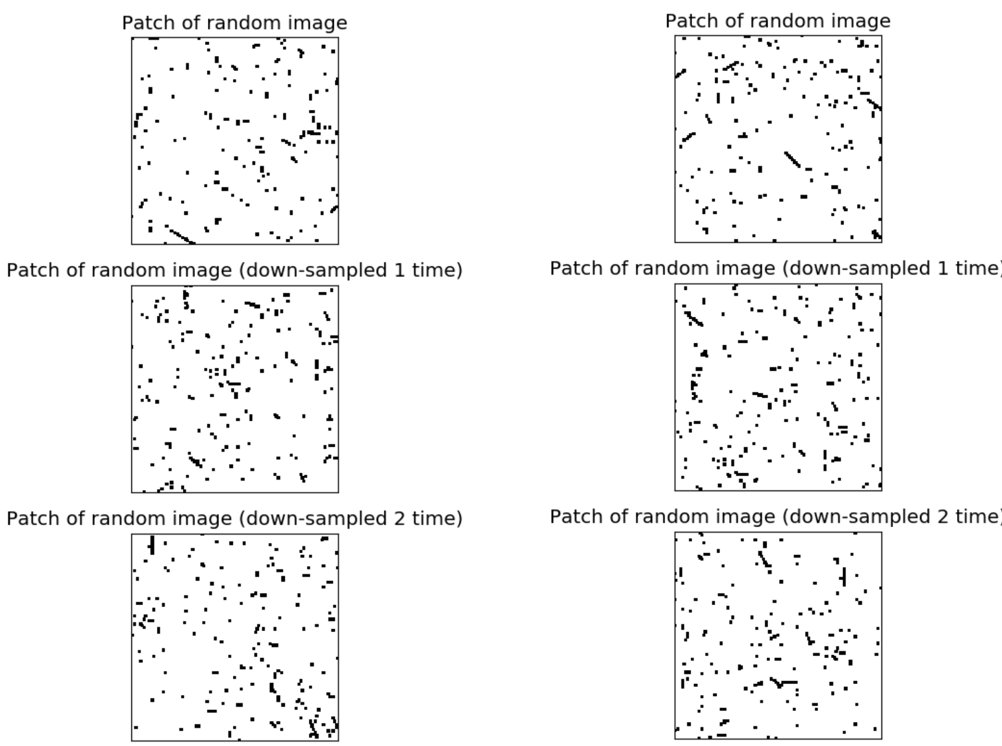

Figure 17: 6 patches randomly sampled from the previous 3 random images

It is indeed true that the source images of the 6 patches in Figure 17 cannot be told from one another. This perfectly proves that the original image is scale-invariant.