Supporting Information
Edge detection with Mie-resonant dielectric metasurfaces

Andrei Komar[1], Rifat Ahmmed Aoni[1], Lei Xu[1,2,3], Mohsen Rahmani[1,3], Andrey Miroshnichenko[2], and Dragomir Neshev[1]

[1]ARC Centre of Excellence for Transformative Meta-Optical Systems (TMOS), Research School of Physics, The Australian National University, Canberra, ACT 2601, Australia
[2]School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia
[3]Advanced Optics and Photonics Laboratory, Department of Engineering, School of Science and Technology, Nottingham Trent University, Nottingham, NG11 8NS, UK

The Supporting Information document has 8 pages, 5 figures, and 1 table.

1 Kernel convolution

Kernel convolution is one of the computational methods to implement edge detection [1]. In this approach a digital filter is applied to an image. Kernel convolution is a process where we take a small grid of numbers and pass them over the whole image,

$$g(x.y) = \omega * f(x,y) = \sum_{\mathrm{d}x=-a}^{a} \sum_{\mathrm{d}y=-b}^{b} \omega(\mathrm{d}x, \mathrm{d}y) f(x+\mathrm{d}x, y+\mathrm{d}y), \tag{1}$$

where $g(x,y)$ is the filtered image, $f(x,y)$ is the original image, and ω is the filter kernel. The operation is being implemented through the whole image for all pixels $-a \leq \mathrm{d}x \leq a$ and $-b \leq \mathrm{d}y \leq b$. Depending on the kernel, different effects can be achieved like blurring, sharpening, embossing, edge detection, and others. For the edge detection, the kernels also can be different. For example two kernels that both implement edge detection but with more and less prominent edges are

$$\omega_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \omega_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}. \tag{2}$$

2 Fourier filtering

Here, we present the example, exhibiting, that by applying amplitude high-pass filter in Fourier space, the edge detection effect can be achieved. We are going to demonstrate the effect using only mathematical

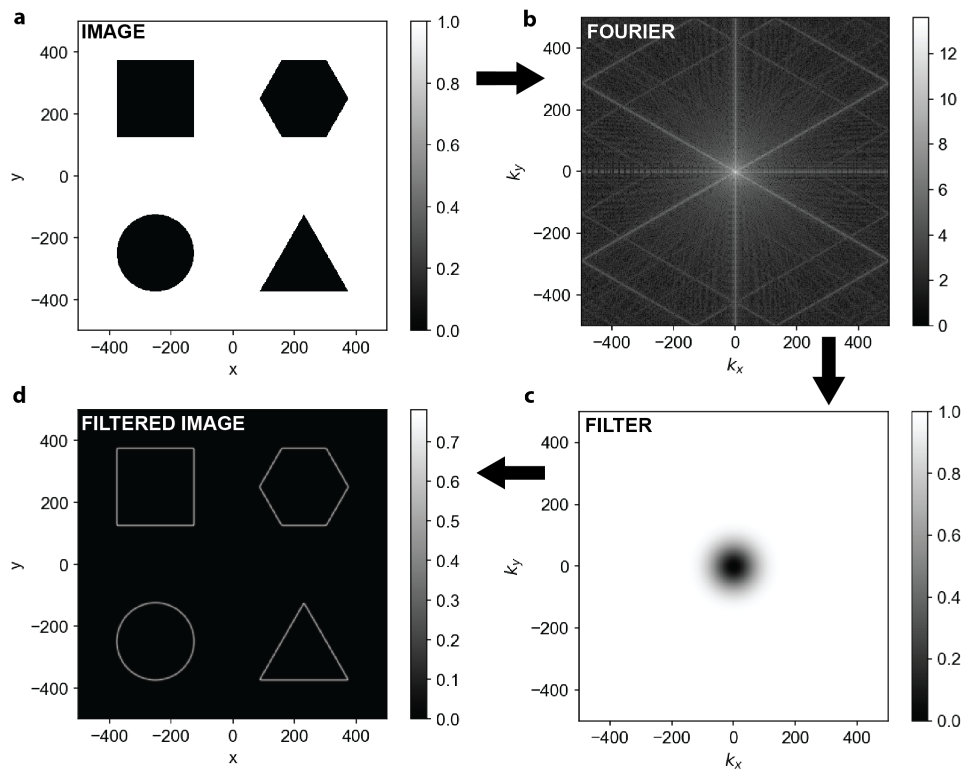

Figure S1: Filtering in Fourier space. (a) The initial image. (b) Fourier spectrum of the image. (c) Inverse Gaussian filter in Fourier space. (d) The resulting signal after filtering.

approach, applying particularly Fourier transform, inverse Fourier transform, and multiplication operation. All our calculations were done in Python utilizing the Numpy library.

First, we start with an image that has several objects with clear, sharp edges. The image, which we use, is presented in Fig. S1a. Mathematically, our image is a 2D array of amplitudes a_{mn} from 0 to 1. In our case the image has a size 1000×1000 pixels. The number of pixels, discretization, can be arbitrary, it defines the discrete frequency of our numerical experiment. Next step is to implement a discrete Fourier transform (DTF) of our image to obtain its Fourier spectrum. The formula for 2D DTF is

$$A_{kl} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} a_{mn} \exp\left(-2\pi i \left(\frac{mk}{M} + \frac{nl}{N}\right)\right) \qquad k = 0, \cdots, M-1; \quad l = 0, \cdots, N-1, \tag{3}$$

where, the elements of an array A_{kl} are complex amplitudes of the Fourier spectrum. The spectrum is shown in Fig. S1b. It is worth mentioning that by only applying the equation (3) to our initial amplitudes a_{mn}, the zero frequency, which corresponds to the straight propagation of light, or DC (direct current) component of the signal, is in the first position of the Fourier spectrum array. To present the spectrum in more convenient form, we apply the shift function to move the zero frequency to the center of the graph. We also plot the natural logarithm of the absolute values of the complex amplitudes

$$\ln\left(1 + |A_{kl}|\right). \tag{4}$$

Now, we want to cut the low frequencies of the spectrum by an amplitude filter. We have chosen the inverse Gaussian shape for our filter

$$U_{kl} = 1 - \exp\left(-\frac{k^2 + l^2}{2\sigma^2}\right), \tag{5}$$

shown in Fig. S1c, where σ is the standard deviation or variance of the Gaussian function. The filter cuts the low frequencies, so it is a high-pass filter. To apply the filter, we multiply the amplitudes of the Fourier spectrum A_{kl} with the filter values U_{kl}. The phase of the spectrum remains the same, that is the reason why we mentioned in the beginning that we would use the amplitude filter.

The last step is to implement inverse Fourier transform of our filtered spectrum to obtain the resulting image. The formula for the inverse Fourier transform is

$$a_{mn} = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} A_{kl} \exp\left(2\pi i \left(\frac{km}{M} + \frac{ln}{N}\right)\right) \qquad m = 0, \cdots, M-1; \quad n = 0, \cdots, N-1. \tag{6}$$

As you may see the resulting image presented in Fig. S1d contains only edges, or silhouettes, of the initial objects. The presented example illustrates that the high-pass filtering of a Fourier spectrum of an image produces the edge detection effect on the image, that is only sharp changes in the initial amplitude remain visible.

3 Mathematics of second derivative

Fourier filtering of sharp edges with inverse-Gaussian shape transmittance results in the edge detection effect, the intensity peaks remain at the position of edges. However, if an initial image does not have sharp features, it means its intensity function is smooth, can we predict the result of Fourier filtering? The short answer is yes, the resulting intensity function can be approximated by the second derivative of the initial function. Let's prove this mathematically in one dimension. In two dimensions we should just add the second coordinate, but all other logic is the same. As it was mentioned, our filter is an inverse-Gaussian function in Fourier space

$$g(k) = 1 - e^{-\frac{k^2}{c^2}}, \tag{7}$$

where c is a constant, the width of this function, and k is a space coordinate in Fourier space for one dimension case. For simplicity we consider $c = 1$.

We denote our two fundamental functions, the initial image intensity function as $f_{\mathrm{im}}(x)$ and the resulted edge detection function as $f_{\mathrm{ed}}(x)$. The Fourier transform of image intensity is

$$F_{\mathrm{im}}(k) = \mathscr{F}\{f_{\mathrm{im}}(x)\}. \tag{8}$$

When we apply our inverse-Gaussian filter in Fourier space, it means we multiply the filter function $g(k)$ with the Fourier transform of the image function $F_{\mathrm{im}}(k)$

$$g(k) \cdot F_{\mathrm{im}}(k) = \left(1 - e^{-k^2}\right) \cdot F_{\mathrm{im}}(k). \tag{9}$$

For the small values of k, we consider only the first member of the Taylor series of the filter function, which is

$$g(k) = 1 - e^{-k^2} = k^2 - \frac{k^4}{2!} + \frac{k^6}{3!} - \frac{k^8}{4!} + \cdots \approx k^2. \tag{10}$$

Now, we take into account one of the fundamental properties of the Fourier transform, which states: The Fourier transform of a second derivative of a function is equivalent to multiplying the Fourier transform of

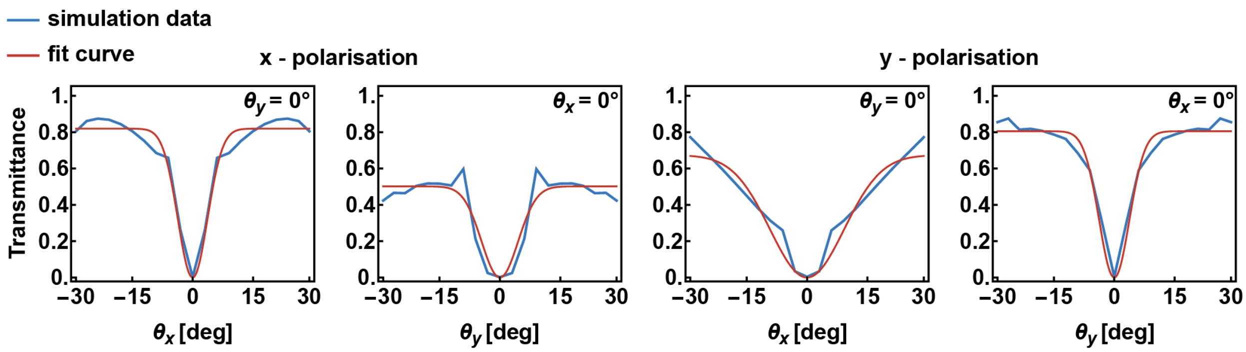

Figure S2: Simulation transmittance for magnetic dipole mode at 1400 nm for x- and y-polarization states along x and y directions (blue curves) and inverse-Gaussian fit of these data (red curves).

the function itself to the square of the frequency

$$\mathscr{F}\left\{\frac{\mathrm{d}^2 f_{\mathrm{im}}(x)}{\mathrm{d}x^2}\right\} \stackrel{\mathrm{def}}{=} k^2 \cdot F_{\mathrm{im}}(k). \tag{11}$$

Therefore, the result of the filtering process is the inverse Fourier transform of the multiplication of the first member of the Taylor expansion eq. (10) with the Fourier transform of the image function eq. (8)

$$f_{\mathrm{ed}}(x) = \mathscr{F}^{-1}\left\{k^2 \cdot F_{\mathrm{im}}(k)\right\} = \frac{\mathrm{d}^2 f_{\mathrm{im}}(x)}{\mathrm{d}x^2}. \tag{12}$$

As you may see for the small values of k, that is why we used the word "approximated" above, the Fourier filtering with inverse-Gaussian shaped filter is exactly equal to the second derivative of the initial function, that, for its part, enhances the sharp features of the image. In Fig. S2, we demonstrate the simulated transmittance results for our metasurface for magnetic mode at 1400 nm. We can claim, that the shape of the angular transmittance of our metasurface resembles well the inverse-Gaussian function.

4 Resolution of edge detection

We regard theoretically how to determine the resolution of edge detection for metasurface approach and what is the value of the resolution for our experimental metasurface. We define the resolution as a distance between two edges that can be clearly resolved after Fourier filtering with inverse-Gaussian filter in k-space. We consider single squared pulse function with the width $2T$ shown in Fig. S3a and find the width of inverse-Gaussian filter when two edges start blurring after high-pass filtering in Fourier space. The Fourier transform of our pulse signal is shown as a blue curve in Fig. S3b. Now, we apply three filters with different widths in the frequency space. The inverse-Gaussian shaped filter has frequency transmittance determined by formula

$$g(\omega) = 1 - e^{-\frac{\omega^2}{2\sigma^2}}, \tag{13}$$

where σ is a standard deviation of the Gaussian function or in our case the width of the filter.

We use filters with the widths $\sigma = \pi/(2T)$, $\sigma = \pi/T$, and $\sigma = 2\pi/T$, presented in Fig. S3b. To apply a filter means that we should multiply the Fourier spectrum of our initial signal to the filter function. After the inverse Fourier transform of the filtered signals, we receive an edge detected result in the real space. Fig. S3c demonstrates the obtained edges for each filter. As we can see, when the width of a filter equals to π/T, the edges start overlapping. When the width equals to π/W, where W is the distance between two

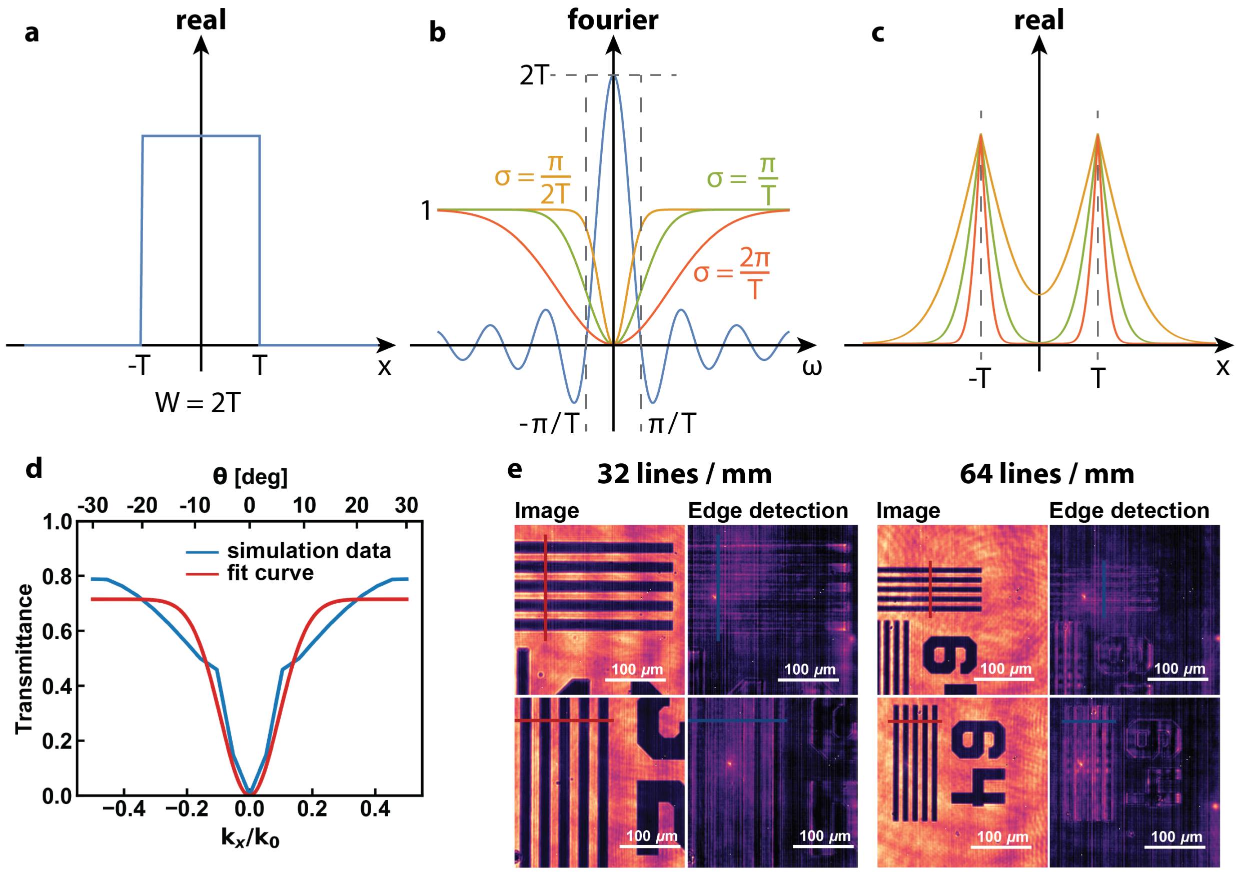

Figure S3: Edge detection resolution. (a) Initial signal – signal squared pulse. (b) Fourier transform of the initial signal and filters with different widths. (c) Resulting edge detection after filtering. (d) Simulated angular transmittance of our sample at the magnetic mode resonance (1400 nm) and fitting curve for the inverse-Gaussian function. (e) Experimental edge detection for targets with 32 and 64 lines per mm.

edges, the edges are already almost half overlapped, but still can be resolved. This value of W, we define as the resolution of edge detection for inverse-Gaussian filtering in frequency space.

Now, we will find the resolution of edge detection for our sample. Fig. S3d shows the simulated angular transmittance of our sample at the magnetic mode resonance at the wavelength of 1400 nm. The red curve is the inverse-Gaussian fitting of our simulated transmittance. The width of this curve is equal to $\sigma = 0.1k_0$, where $k_0 = 2\pi/\lambda$. Therefore, the resolution, as we defined it above, is

$$W = \frac{\pi}{\sigma} = \frac{\pi\,\lambda}{0.1\,2\pi} = \frac{\lambda}{0.2}. \tag{14}$$

The wavelength of the magnetic mode resonance is $\lambda = 1.4$ μm, the minimum distance between two edges that can be resolved after filtering with our metasurface is hence $W = 7$ μm.

In the main text of our article, we presented the edge detection results for the stripes with the width 45 μm that is far above the resolution. However, we also measured edge detection with our metasurface for smaller stripes, 32 lines per mm that gives us the stripe width equal to approximately 15 μm and 64 lines per mm, the stripe width is about 8 μm. The experimental results are presented in Fig. S3e. For the target 64 lines per mm, we can observe that the edges are overlapping and they look blurred that matches the resolution value for our metasurface sample.

5 Histogram approach

Fig. S4 and Fig. S5 present the result of color adjust for the experimental edge detection images. The infrared camera, we used in experiments, has a 14 bits ADC. It means that the whole camera range of counts is from 0 to 16383. However, even the completely dark condition produces about 4000 counts of ADC. To remove this level and to better adjust the presentation of the images in the color code, we have scaled every image based on a histogram approach. The histograms demonstrate the number of pixel for each ADC count. We have cut the counts that are not presented in the images. The values of the cutoff counts remained the same for the sample result (image) and for the edge detection result (edge detection) to properly compare the edge detection effect and the original image. Fig. S4 and Fig. S5 contain the histograms, the raw images from the camera, and the color scaled images after the adjustment procedure. The cutoff counts are summarized in Table S1.

Resonance	Stripes	Min cutoff count	Max cutoff count
Magnetic mode 1400 nm	horizontal	4500	14000
	vertical	4500	15000
Electric mode 1570 nm	horizontal	3500	14000
	vertical	3500	13500

Table S1: Cutoff counts for each experimental image.

References

[1] Wikipedia contributors. Kernel (image processing) — Wikipedia, the free encyclopedia, 2020. [Online; accessed 8-April-2020].

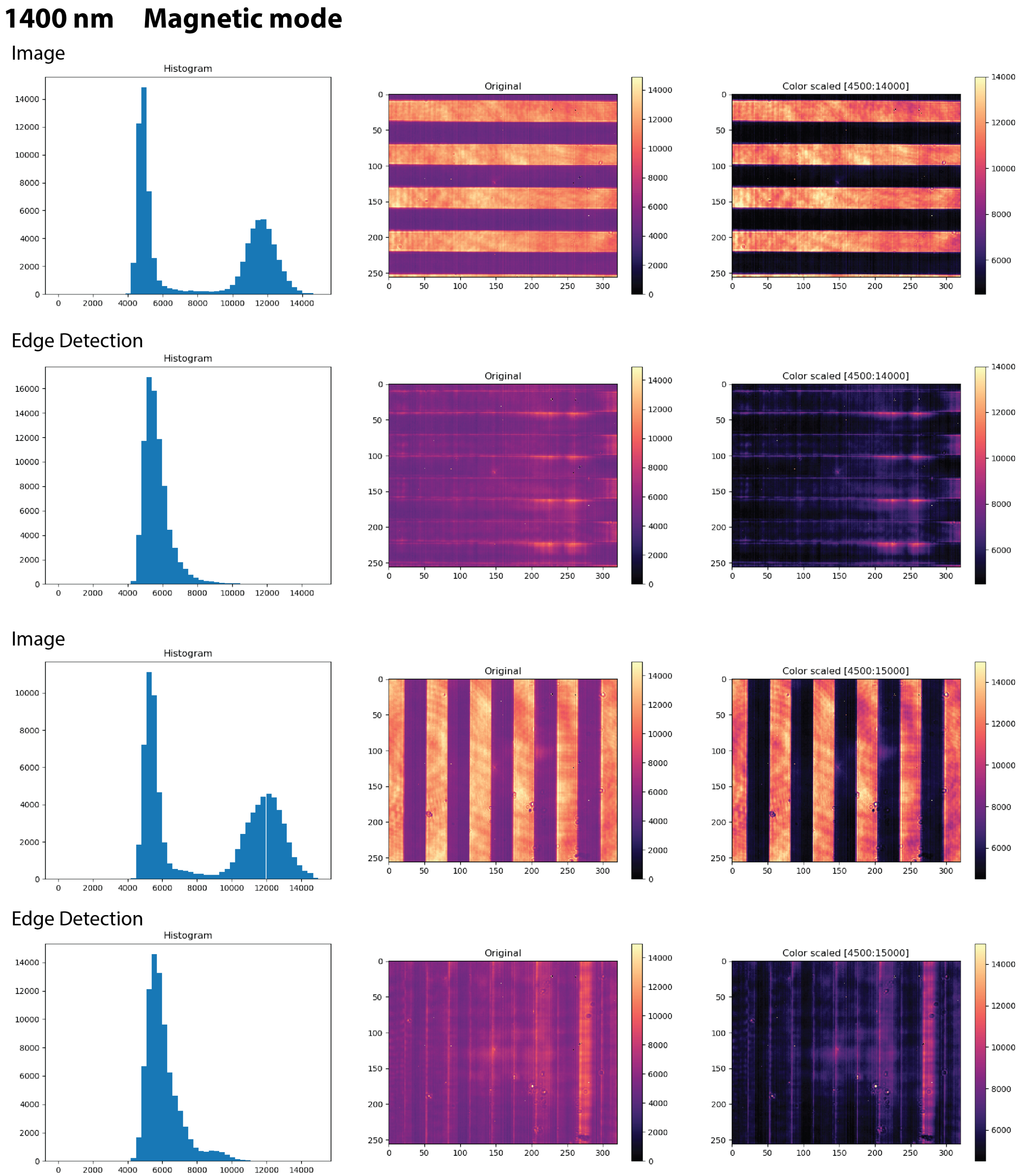

Figure S4: Results of color adjust for magnetic dipole resonance mode.

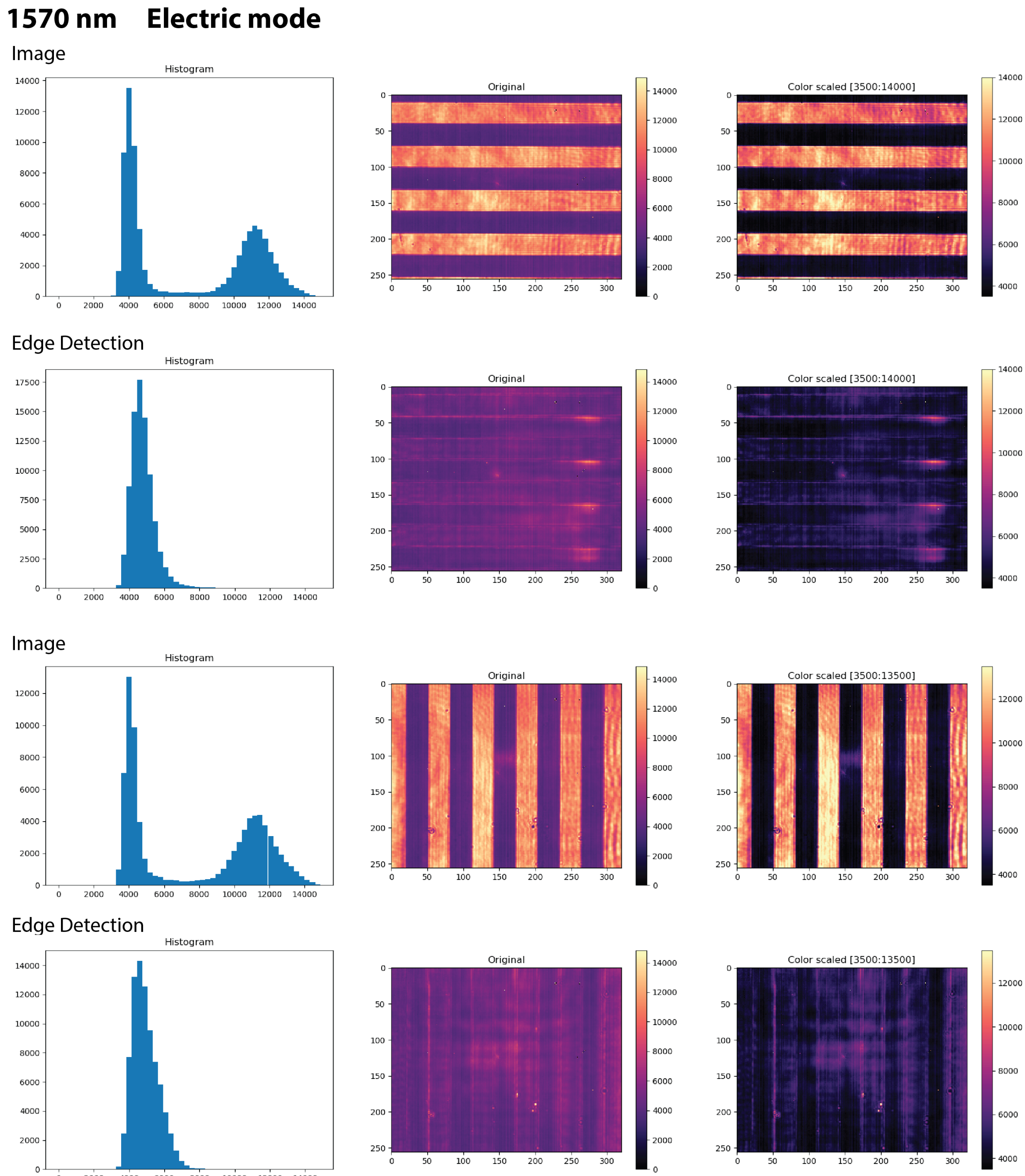

Figure S5: Results of color adjust for electric dipole resonance mode.