Practice 2
Image processing with variational approaches

Meaza Eyakem and Baglan Aitu

Task 1. Checking the convexity and computation of gradients.

Gradient:

1, CHECK THE CONVEXITY AND COMPUTATION OF

Gradients of the above functions J_H and J_{Pm}.

$$J_H(u) = \frac{1}{2} \sum_{x \in \Omega} \|\nabla u(x)\|^2$$

$$J_H(u) = 1/2 \, \|\nabla u\|_Y^2$$

$$\nabla J_H(u) = ?$$

$$\nabla J_H(u) = J_H(u+h) - J_H(u).$$

$$J_H(u+h) = \frac{1}{2}\|\nabla u, + \nabla h\|_Y^2$$ → using second order approximation.

$$J_H(u+h) = \frac{1}{2}\|\nabla u\|^2 + \langle \nabla u, \nabla h \rangle + \frac{1}{2}\|\nabla h\|_Y^2$$

$$J_H(u+h) = J_H(u) + \langle \nabla u, \nabla h \rangle + o(h)$$

$$\langle \nabla u, \nabla h \rangle = \langle \nabla^* \nabla u, h \rangle$$

$$\langle -\mathrm{div}\, \nabla u, h \rangle$$

$$\langle -\Delta u, h \rangle$$

$$\langle \nabla u, \nabla h \rangle = -\Delta u$$ → By substiting this term

$$J_H(u+h) - J_H(u) = -\Delta u + O(h)$$

$$\nabla J_H(u) = -\Delta u$$

Convexity:

CONVEXITY OF $J_H(u) = 1/2 \sum_{x \in \Omega} \|\nabla u(x)\|^2$

J Is Convex If, $\nabla^2 J$ is posetive.

$\nabla^2 J$ is POSETIVE WHEN

$$\langle y, \nabla^2 J \, y \rangle \geq 0 \quad \ldots 1.$$

So, we will proof If J satsfies the above equation.

$$\nabla J(u) = -\Delta u$$

$$\nabla J(u+h) = -\Delta u - \Delta h$$

$$\nabla J(u+h) = \nabla J(u) + \nabla^2 J(u) h$$

$$-\Delta u - \Delta h = \nabla J(u) + \nabla^2 J(u) h$$

$$-\cancel{\Delta u} - \Delta h = -\cancel{\Delta u} + \nabla^2 J(u) h$$

$$\nabla^2 J(u) = -\Delta$$

- Proof If 1 is satsfied

$$\langle y . \nabla^2 J(u) . y \rangle \geq 0$$

$$\langle y . -\Delta y \rangle = \langle \nabla y, \nabla y \rangle$$

$$= \|\nabla y\|^2 \geq 0$$

- Therefore J_H is convex function which qurentees That there is a minimizer.

Gradient of Perona-Malik Model:

PERONA MALIK PDE

$$J_{pm} = \sum_{x \in \Omega} \sqrt{\|\nabla u(x)\|^2 + 1}$$

$$J_{pm}(u+h) = \sqrt{\|\nabla u(x) + \nabla h\|^2 + 1}$$

$$= \sqrt{(\|\nabla u\|^2 + 1)} \cdot \sqrt{\left(1 + \frac{2 \nabla u \cdot \nabla h}{\|\nabla u\|^2 + 1} + o(|h|)\right)}$$

$$J_{pm}(u+h) = J_{pm}(u) + \sum \frac{\nabla u}{\sqrt{\|\nabla u\|^2 + 1}} + o(|h|)$$

∇J

$$\nabla J_{pm}(u) = \left\langle \frac{\nabla u}{\sqrt{\|\nabla u\|^2 + 1}}, \nabla h \right\rangle$$

$$= \left\langle \frac{\nabla : \nabla u}{\sqrt{\|\nabla u\|^2 + 1}}, h \right\rangle$$

$$\nabla J_{pm}(u) = -\mathrm{div}\left(\frac{\nabla u}{\sqrt{\|\nabla u\|^2 + 1}}\right.$$

Task 2. ***Computation of gradient J***

$$J_D(u) = 1/2 \sum_{x \in \Omega} \| u(x) - f(x) \|^2$$

$$= 1/2 \| u - f \|_X^2$$

$$J_D(u+h) = 1/2 \| (u+h) - f \|_X^2$$

$$= 1/2 \| u - f \|^2 + \frac{1}{2} \cdot 2 \langle u - f, h \rangle + 1/2 \| h \|^2$$

$$= J_D(u) + \langle u - f, h \rangle + o(|h|)$$

$$\nabla J_D(u) = \langle u - f, h \rangle$$

$$\underline{\nabla J_D(u) = u - f}$$

Checking Convexity of the function:

Convexity

$$J_D(tu+(1-t)v) \leq t\,J_D(u) + (1-t)\,J_D(v)$$

$$\Rightarrow \frac{1}{2}\|(tu+(1-t)v) - tf - (1-t)f\|_X^2$$

$$= \frac{1}{2}\|(t(u-f) + (1-t)(v-f))\|_X^2$$

$$\frac{1}{2}\|(t(u-f)+(1-t)(v-f))\|_X^2 \leq \frac{1}{2}t\|u-f\|_X^2 + \frac{1}{2}(1-t)\|v-f\|_X^2$$

$p = u-f$; $\rho = v-f$.

$\|x+y\| \leq \|x\| + \|y\|$ - Triangle inequality

$$\frac{1}{2}\|\nabla(tp+(1-t)\rho)\|_X^2 \leq \frac{1}{2}t^2\|\nabla p\|_X^2 + \frac{1}{2}(1-t)^2\|\nabla \rho\|_X^2$$

$$\frac{1}{2}t^2\|p\|_X^2 + \frac{1}{2}(1-t)^2\|\rho\|_X^2 \leq \frac{1}{2}t\|p\|_X^2 + \frac{1}{2}(1-t)\|\rho\|_X^2$$

$$\frac{1}{2}\|(tp+(1-t)v\|_X^2 \leq \frac{1}{2}t^2\|p\|_X^2 + \frac{1}{2}(1-t)^2\|\rho\|_X^2$$

$$\leq \frac{1}{2}t\|p\|_X^2 + \frac{1}{2}(1-t)\|\rho\|_X^2$$

$$\frac{1}{2}\|tp+(1-t)v\|_X^2 \leq \frac{1}{2}t\|p\|_X^2 + \frac{1}{2}(1-t)\|\rho\|_X^2$$

*** Noisy images:**

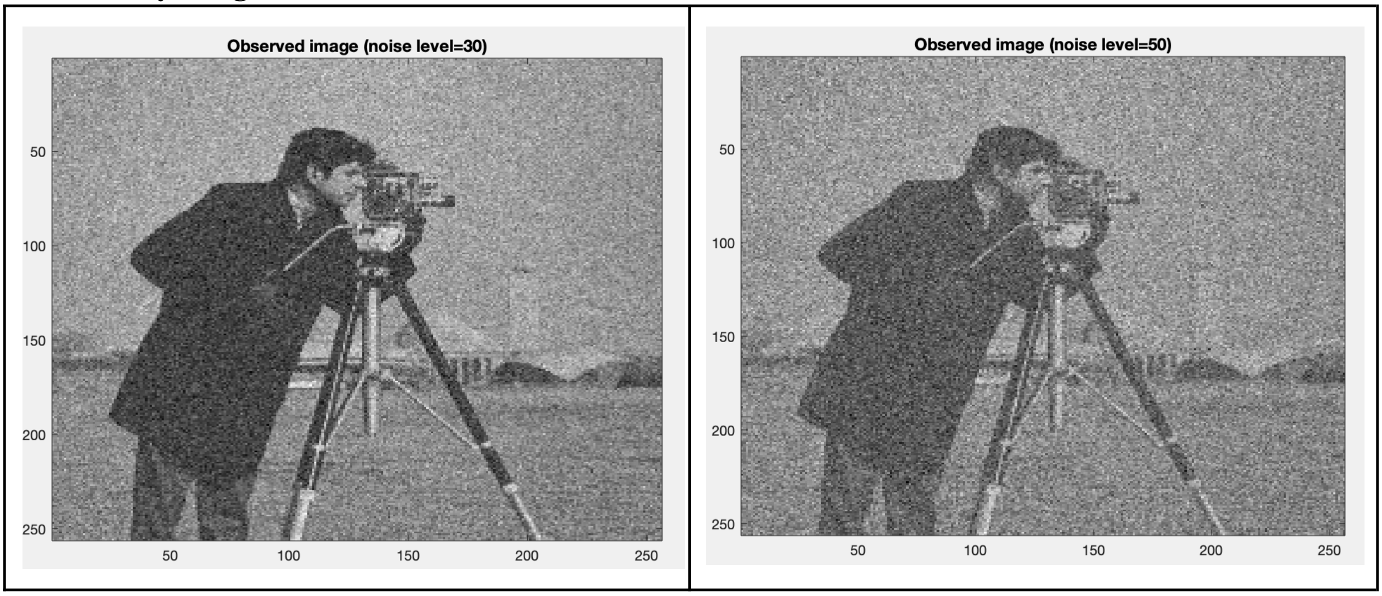
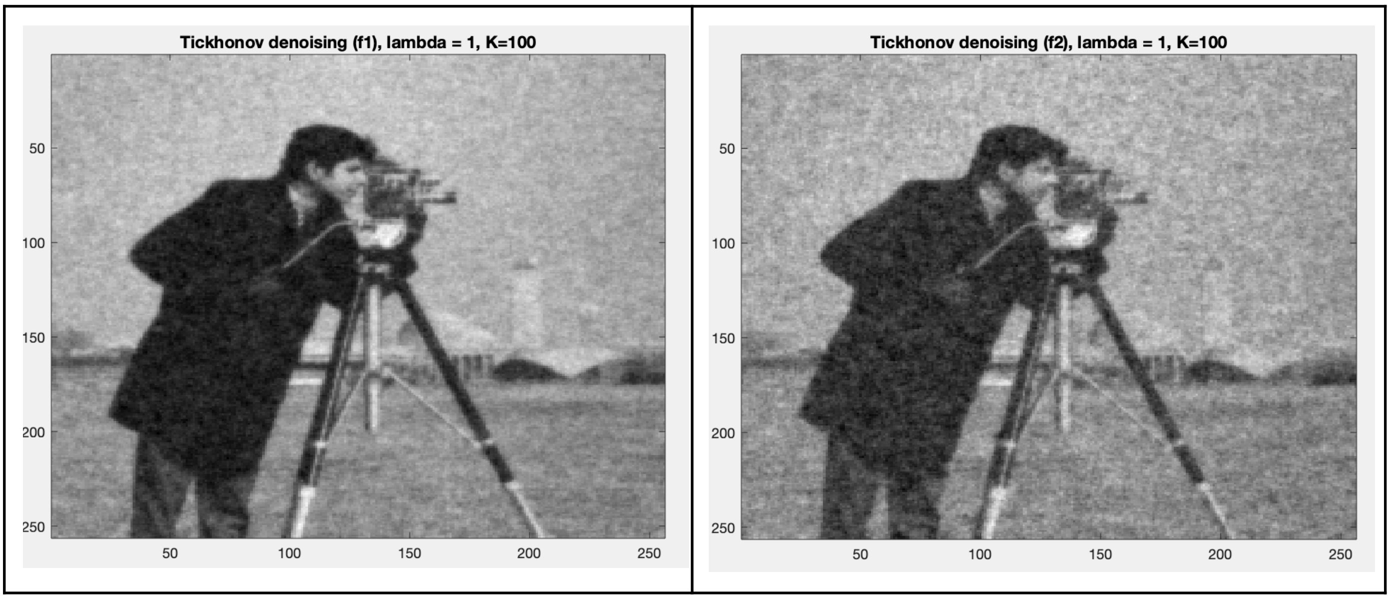


Fig 1. Noisy images: f1 (left) and f2 (right)

We obtained two images f1 with noise levels 30, and f2 with 50 (Fig 1) in order to see how denoising algorithms behave in different noise levels in following tasks.

Task 3,5. Tikhonov regularization



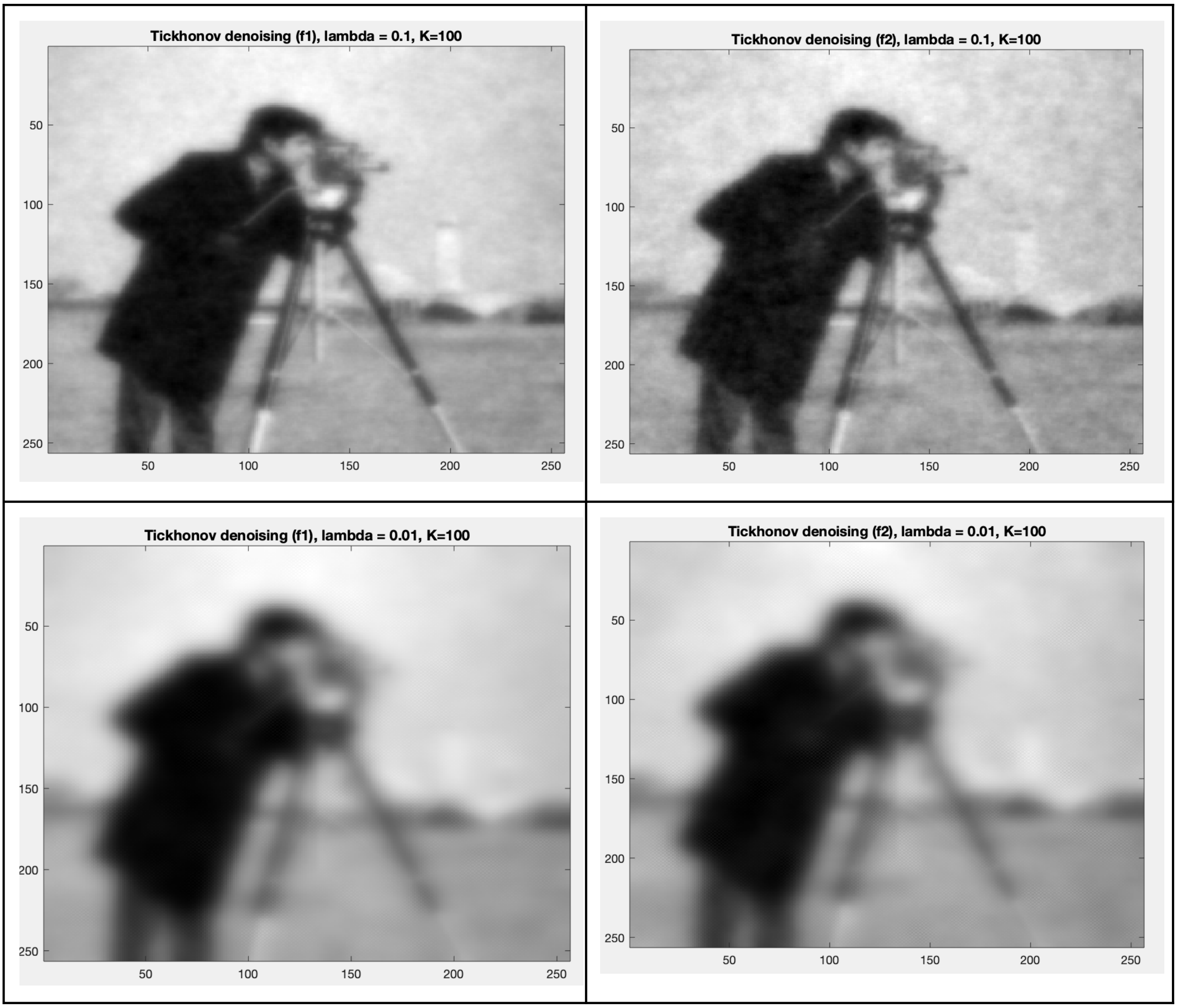
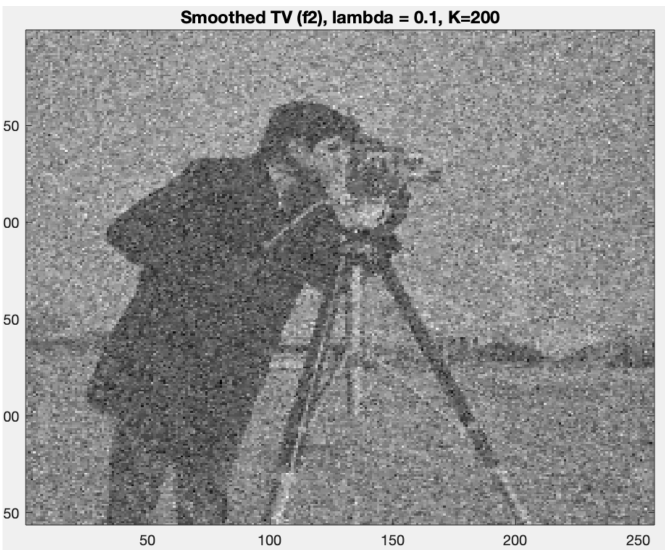


Fig 2. Tikhonov denoising

As we can see in Fig 2, Tikhonov regularization produces smoothed images by reducing the noise of the observed image. The penalty term lambda determined the level of blurriness of the denoised image. Decreasing the value of lambda yields a more smoothed image. This can be inferred from the penalty term of the Ticknov, As the timestep increases with the decrease of the lambda, which indicates the gradient is penalized highly.

If we look to the last row of Fig 2, images with different noise levels look the same. Theoretically, the image with less noise which means with less high frequencies should be more blurred. However, in our case both images look the same with *lambda = 0.01.* According to that we can conclude, the more noise (high frequencies) we have in the image, the faster it will be blurred. In general, the algorithm is not practical since it doesn't preserve the edges. In order to solve this issue, we can use the Smoothed Total Variation regularization method. The smoothing is also dependent on the number of iterations. The number of iterations to find the optimal results depends on the value of lambda used.

Task 4,5. Smoothed Total Variation



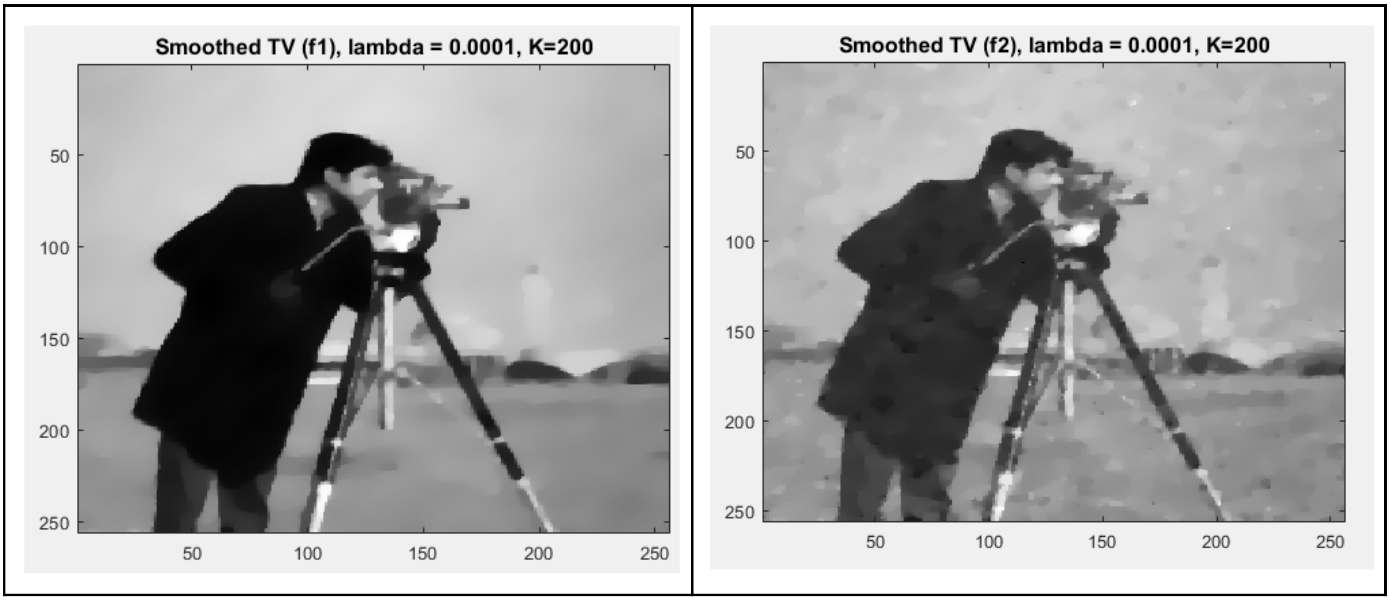


Fig 3. Smoothed Total Variance

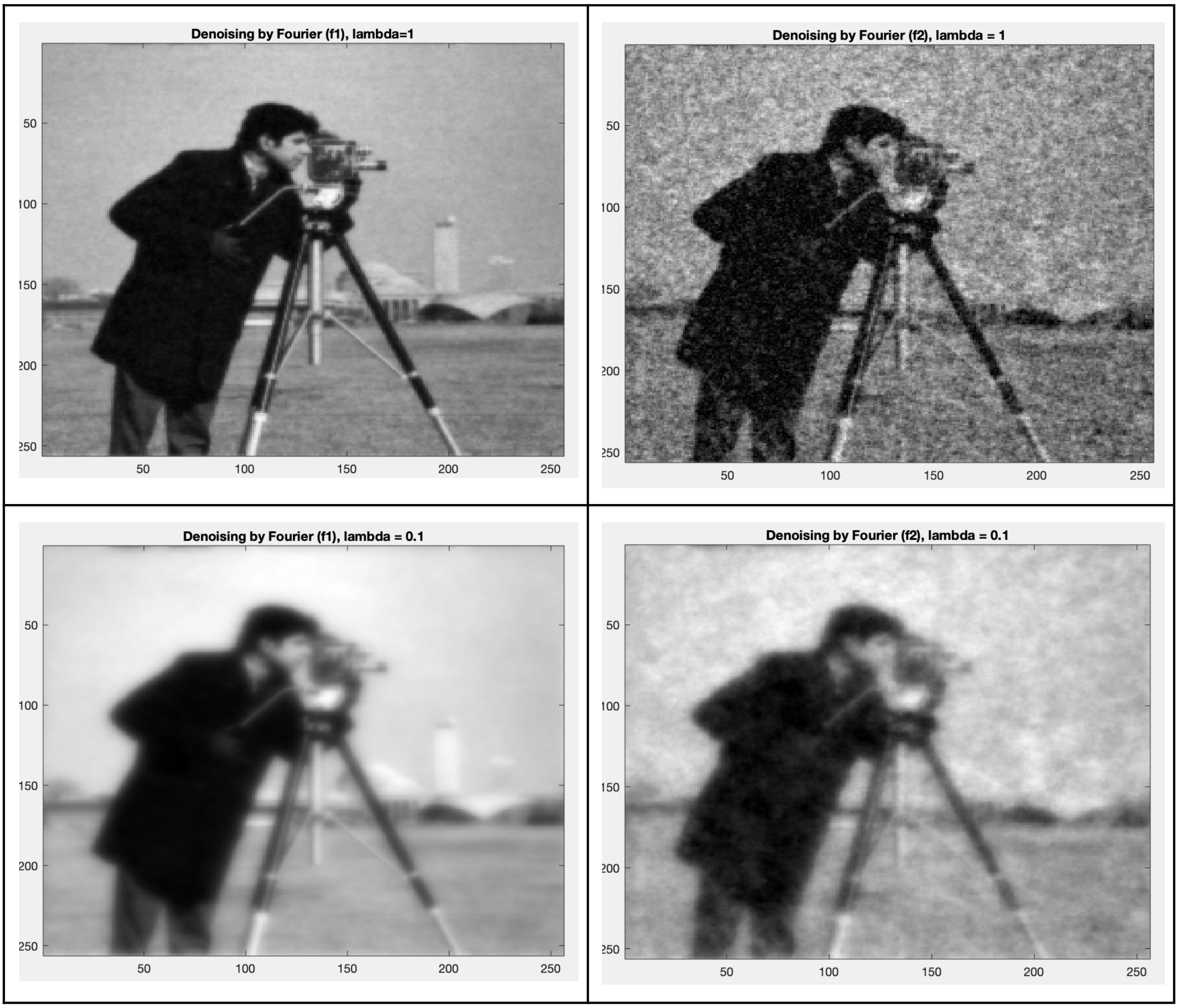
The results of *Total Variance Regularization* (Fig 3) are much better than the previous method. Comparatively with Tikhonov, edges are preserved in low numbers of *lambda* which is responsible for the smoothness. This improvement was achieved by introducing the variation regularization function which enforces the image to be piecewise constant. It diffuses the smoothness anisotropically by preserving the edges unlike Tikhonov.

However, if we compare the results of f1 and f2, we can see that noises still exist in f2 (Fig 3, last row, right column), while f1 is clean. Based on this, we conclude that the TVR method works well for images with less noise. If the image was more noisy (>50), it wouldn't be restored. In this scenario, smoothness is indirectly proportional to the level of noise

By observing the figures in Figure 3, one can dictate that the effects of the parameter lambda are different for different noise levels of the image. The image with small noise level was restored with a good smoothing level, with the edges preserved, with a relatively larger amount of lambda(0.01) in the third row of Fig 3. While the noisy image was still not recovered in that case. This is due to the fact that larger lambda means small timestep or jumps of the noisy image. Therefore, Image with high noise requires a large amount of jumps/timesteps (small amount of lambda) or large amount of iterations in the large lambda case.

Due to this reason, one can see the image with high level of noise was restored by preserving the edges and small artifacts in the last row of the figures with very small lambda, while the image with low noise are very smoothed.

Task 7. Deconvolution by Fourier Transform



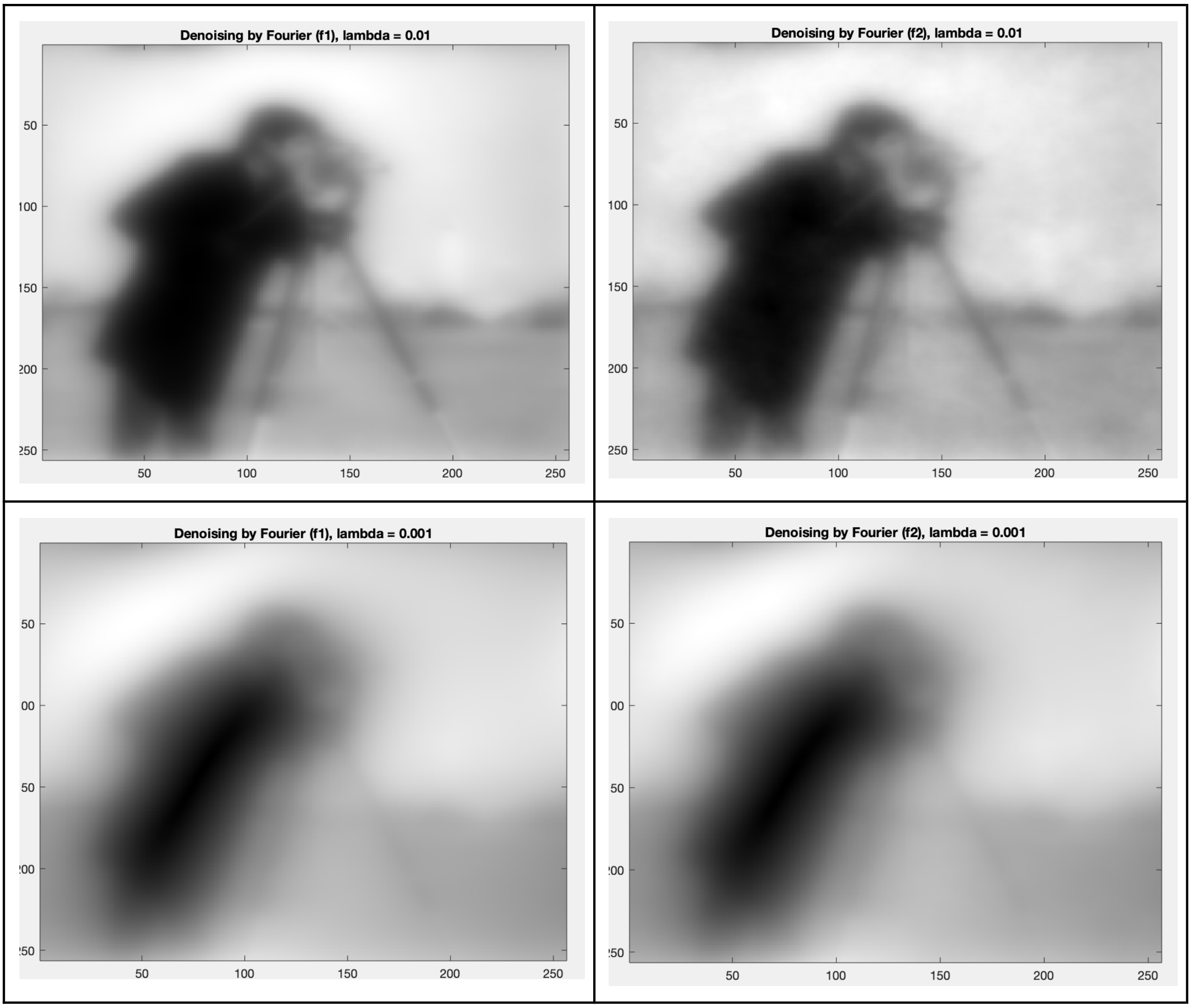


Fig 4. Deconvolution by Fourier transform

By comparing the results of Fourier transform denoising (Fig 4) and Tikhonov denoising (Fig 2), one can notice that they are almost the same. The Fourier algorithm also smoothes the image when the lambda value is decreased. Moreover, in the small number of lambda, the results for images f1 and f2 (Fig 4 last row) look the same. According to these observations, we can conclude they have the same properties. The advantage of Fourier is that it is fast, which might be more suitable for real-time tasks, and replace the Tikhonov approach.

The lambda value of 0.01gives better result by preserving the edges during smoothing, while the decreased values of lambda provides very smoothed results.

Task 8. ***Math***

$$J_A(u) = \frac{1}{2} \sum \|Au(x) - f(x)\|^2$$

$$= 1/2 \, \|Au - f\|_X^2$$

$$J(u+h) = 1/2 \, \|A(u+h) - f\|_X^2 = \frac{1}{2} \|Au - f + Ah\|^2$$

$$= \frac{1}{2} \left[\|Au - f\|^2 + 2 \langle Au - f, Ah \rangle + o(|h|) \right]$$

$$= 1/2 \left[\|Au - f\|^2 + 2 \langle A^*(Au - f), h \rangle + o(|h|) \right]$$

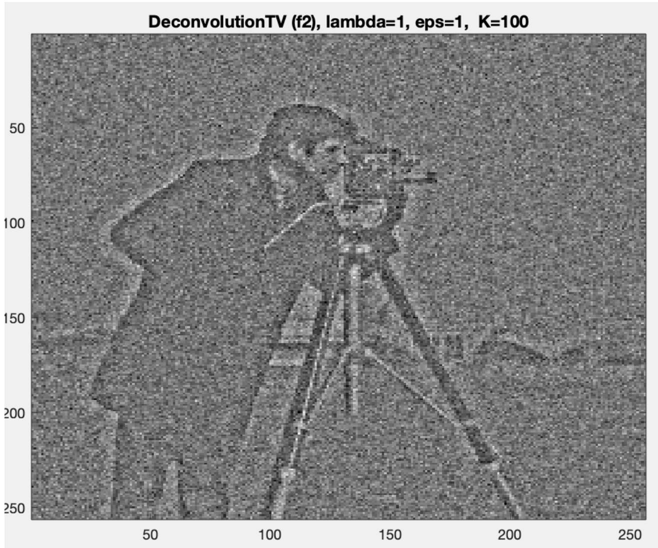
$$A^* = A^T$$

$$J_A(u+h) = J_A(u) + \langle A^T(Au - f), h \rangle + o(|h|)$$

$$\nabla J_A(u) = \langle A^T(Au - f), h \rangle + o(|h|)$$

$$\underline{\nabla J_A(u) = A^T(Au - f)}$$

Task 9, 12. Total variation deconvolution



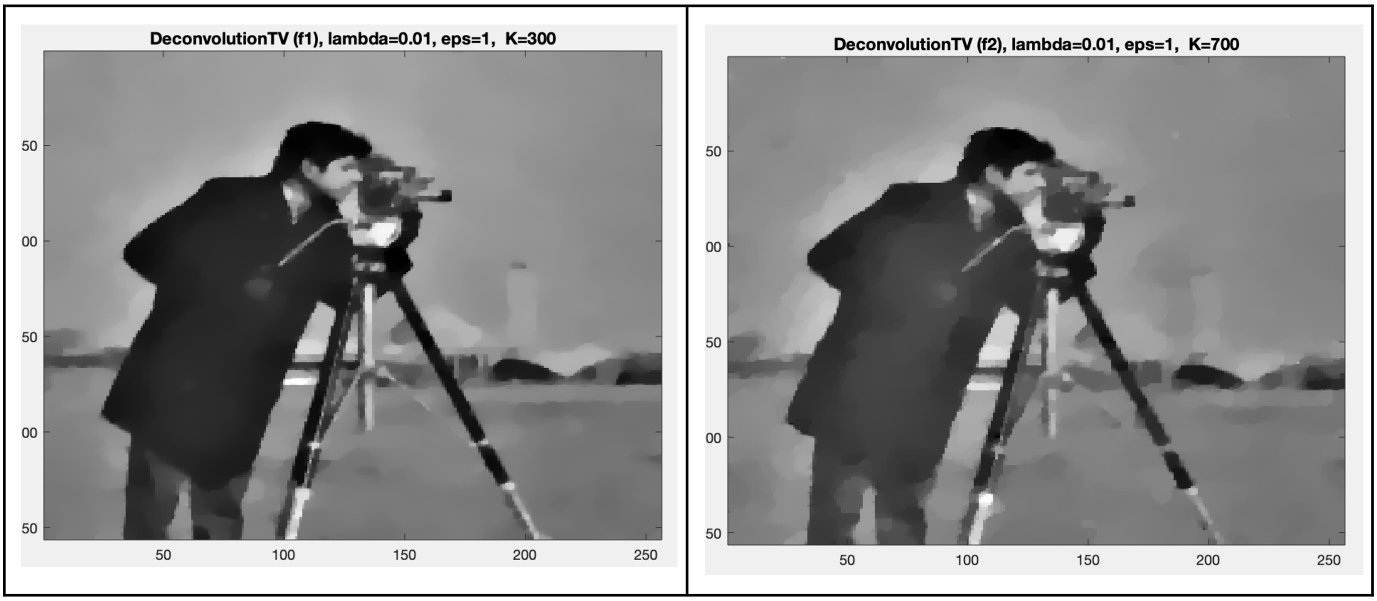


Fig 5. Total variation deconvolution (effect of lambda, K)

Total variation deconvolution approach smoothes the image to reduce the image and most importantly it preserves the edges (Fig 5). It is clear that by decreasing the *lambda,* the noise gets smooth like in previous approaches. In the last row of Fig 5, the result of f2 is clear with sharp edges which shows that clear denoising can be achieved by the increase of iteration number. However, higher values of lambda (>1) give opposite effects to denoising (Fig 5, first row).

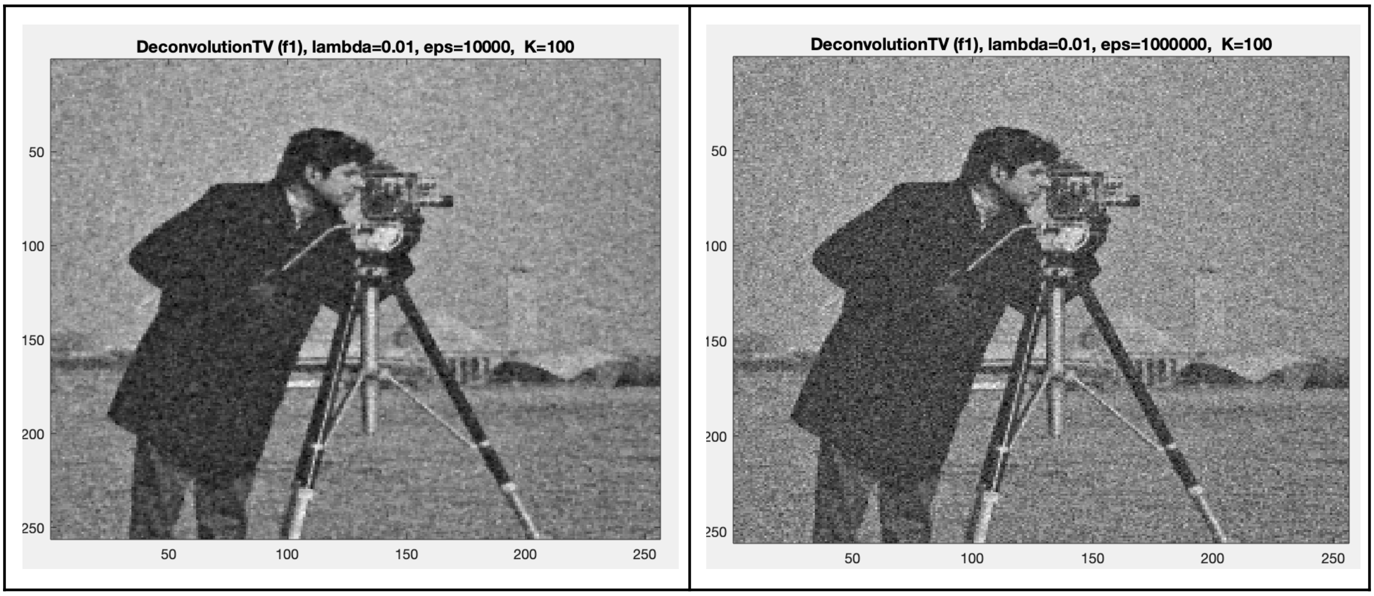
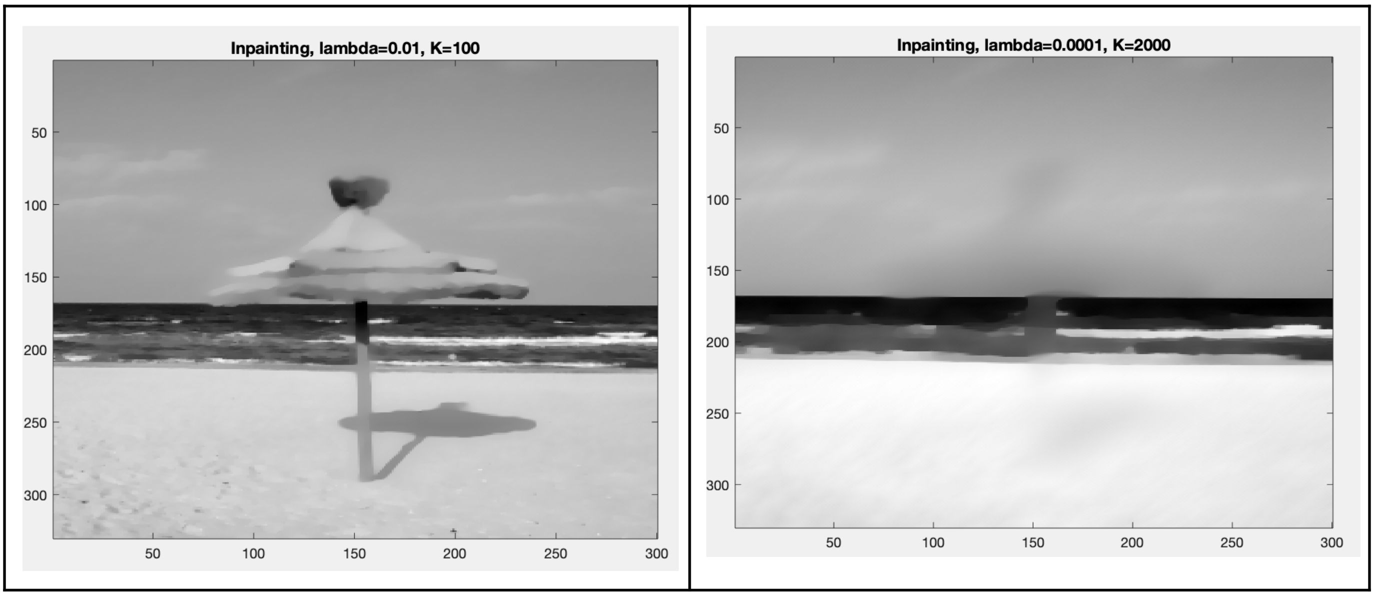


Fig 6. Total variation deconvolution (effect of eps on f1)

In Fig 6, we can see that *eps* value doesn't have much influence on the results. The reason for that in the equation, its square is just arithmetically added to the norm of the image. According to Fig 6, the increase of its value lowers the strength of smoothing.

Task 10. Total variation inpainting



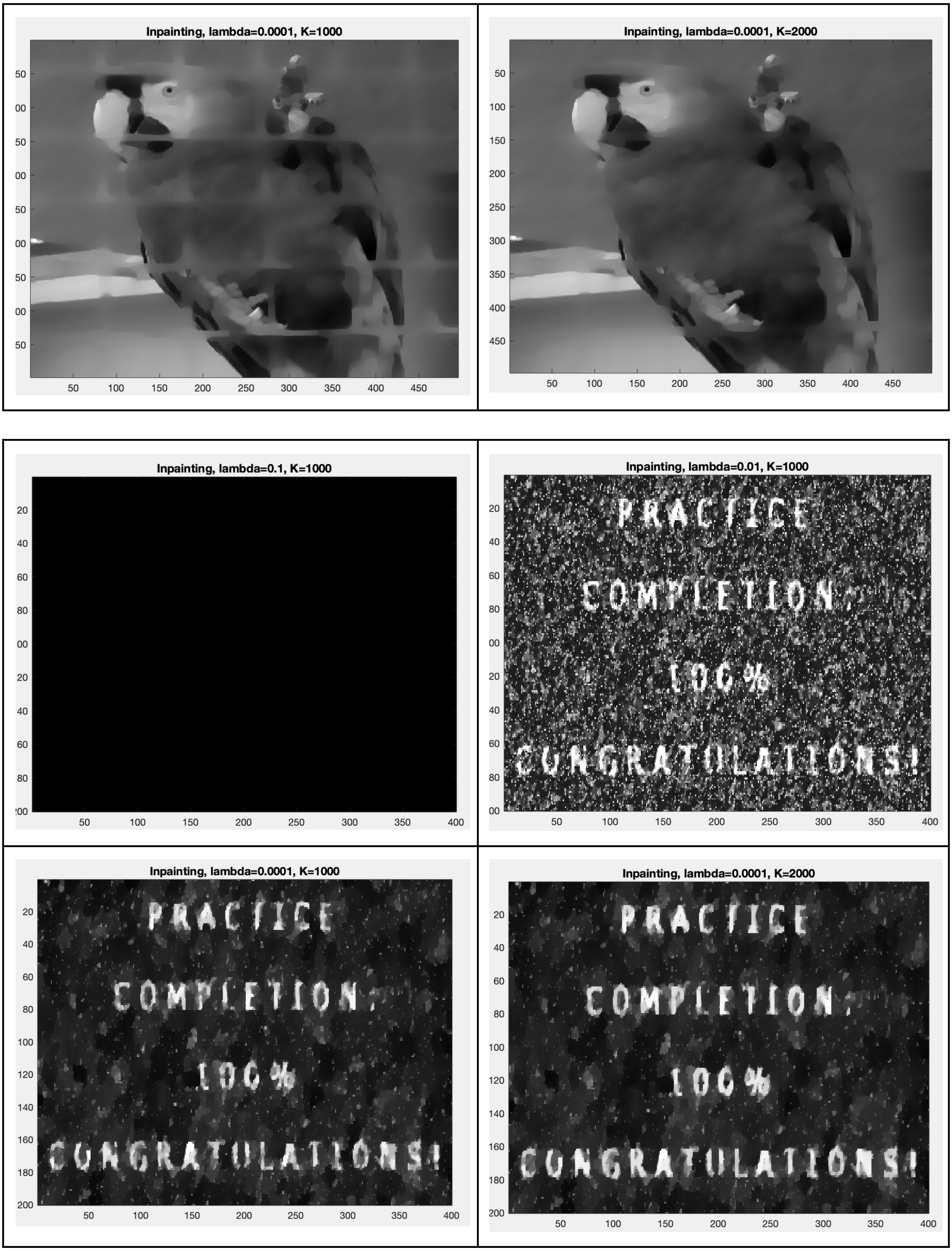
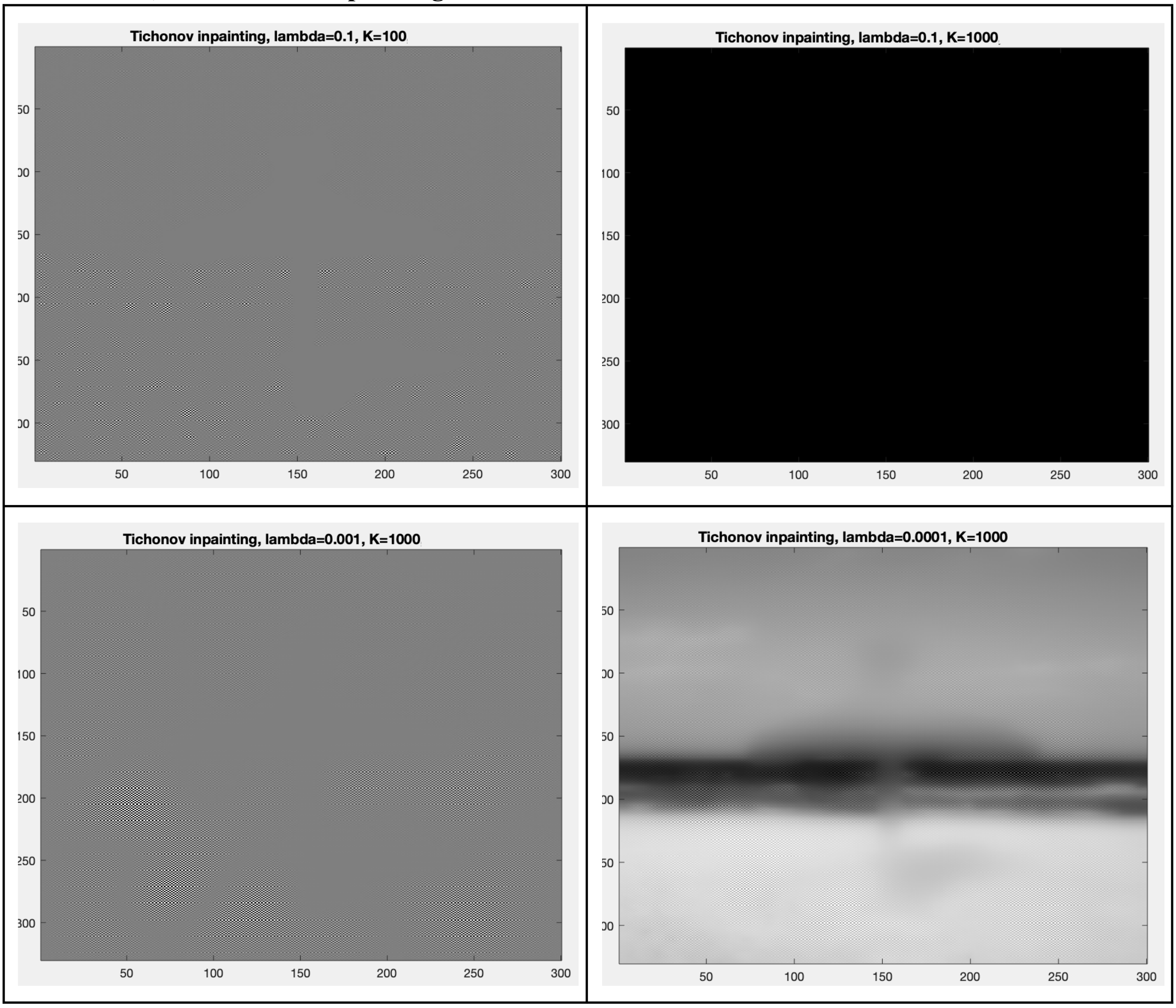


Fig 7. Total variation inpainting

Inpainting by total variation, the elimination of objects can be achieved by the decrease of *lambda* and increase of *iteration number*. It blurs the part of the image depending on the applied mask. Need to note, there is a threshold for the *lambda* value. For example, in the third row of Fig 7, lambda=0.1 nullified the pixel values of the image.

Task 11, 13. Tikhonov inpainting



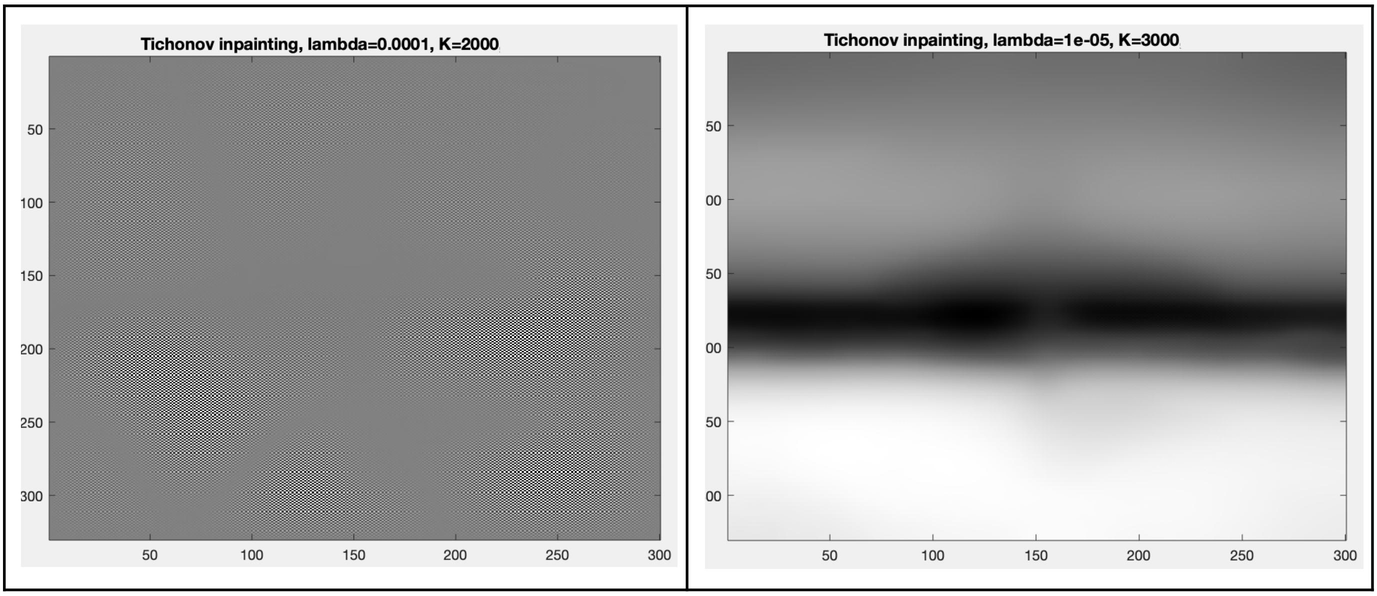


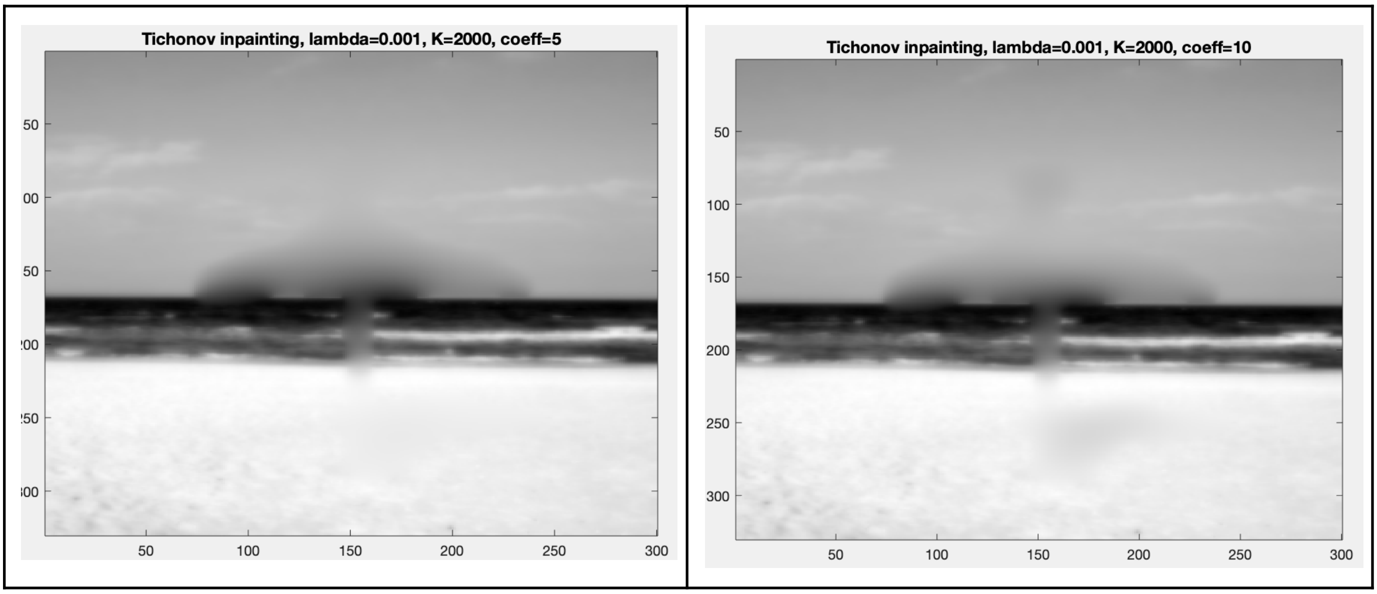
Fig 8. Tikhonov inpainting (lambda, K effect)

In Fig 8, we can see inpainting with the Tikhonov approach gives chaotic results. All of them are unacceptable. Even the results with a visible beach (Fig 8, 2 columns, last two rows) are too blurry and there is no influence of the mask (the whole image was blurred).

To solve this issue, we decided to change the given number *coefficient* in the equation of time step (1). By default, it was 4 in previous tasks.

$$t = 1/(lambda + coeff) \tag{1}$$

For this task, we set a higher number of values to get the smoothing only in the mask region. By experiments, we observed that from *coeff=5* acceptable results can be obtained and tuning of *coeff* gives different paintings (Fig 9). For example, *Coeff=5* has a better painting in the bottom part of the image (sand), and *coeff=10* has a better painting in the top part of the image (sky).



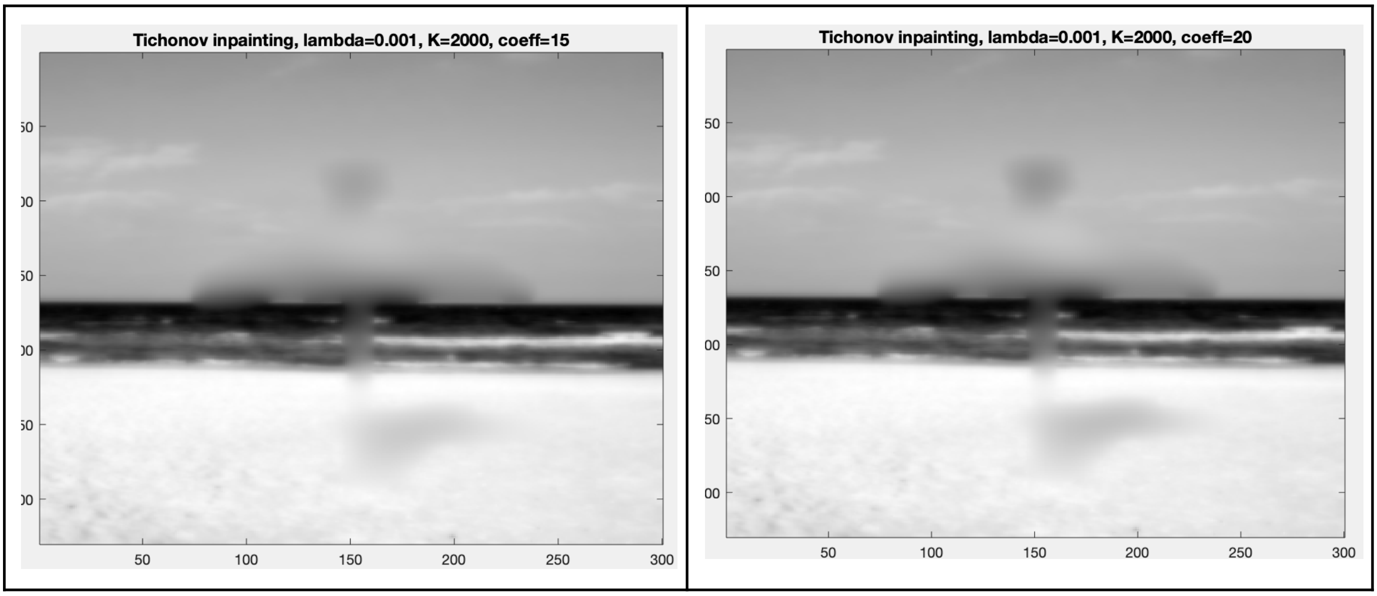
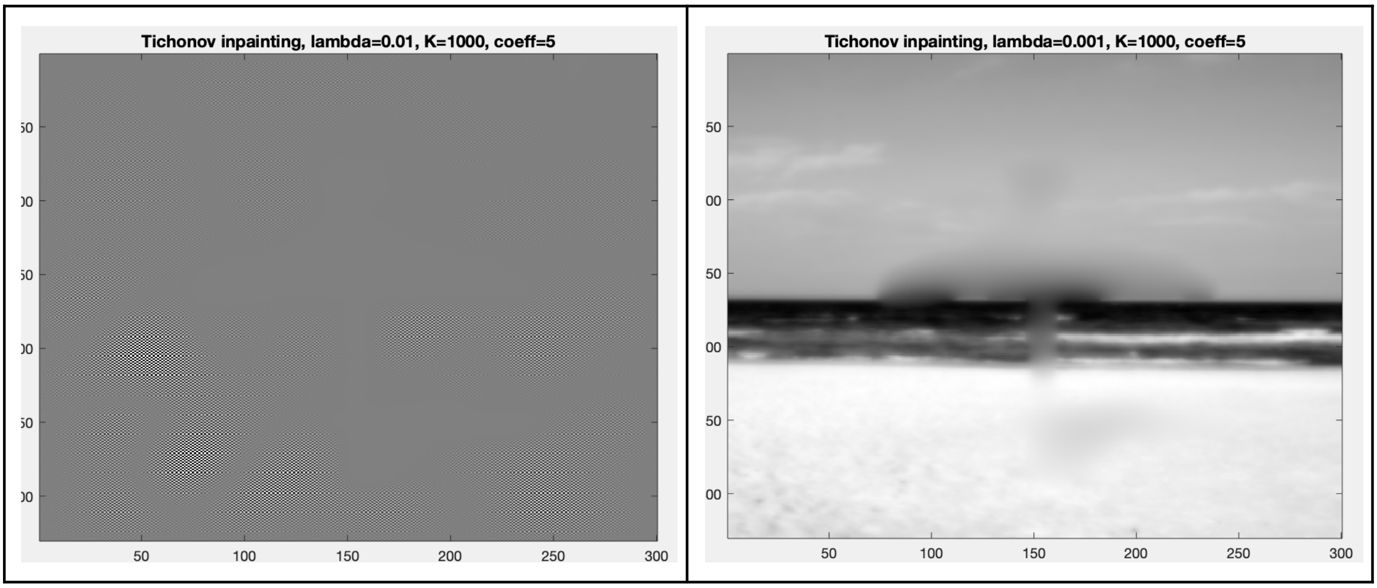


Fig 9. Effect of *coeff* in Tikhonov painting



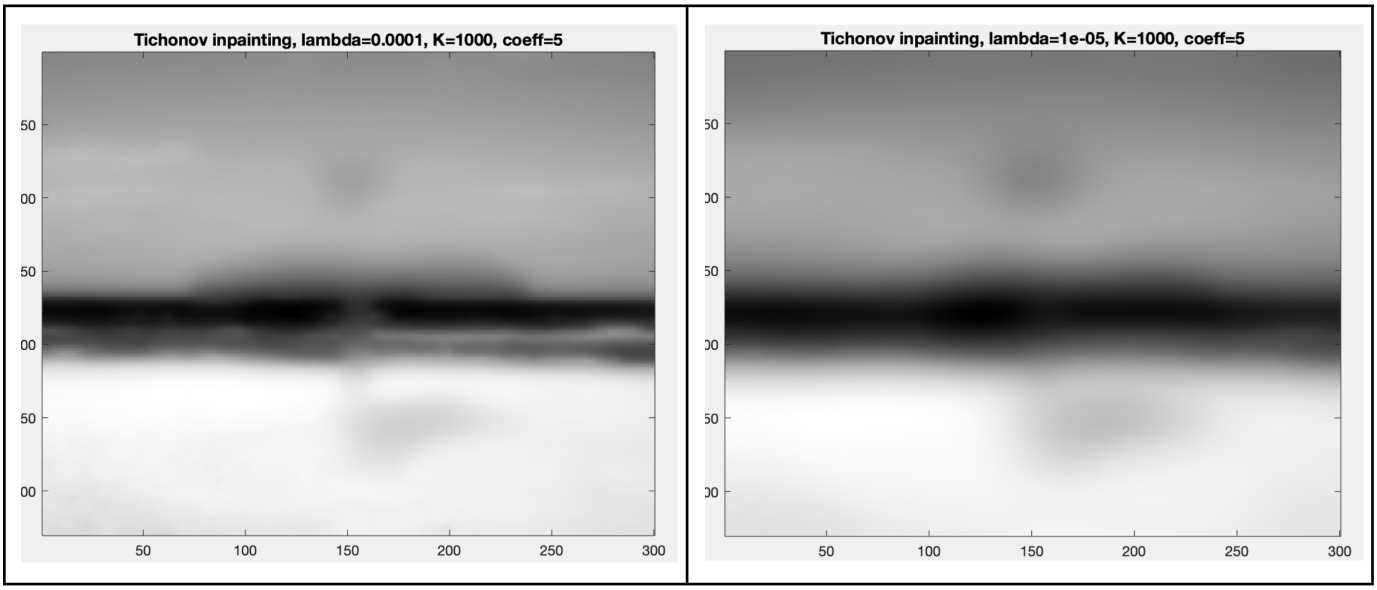
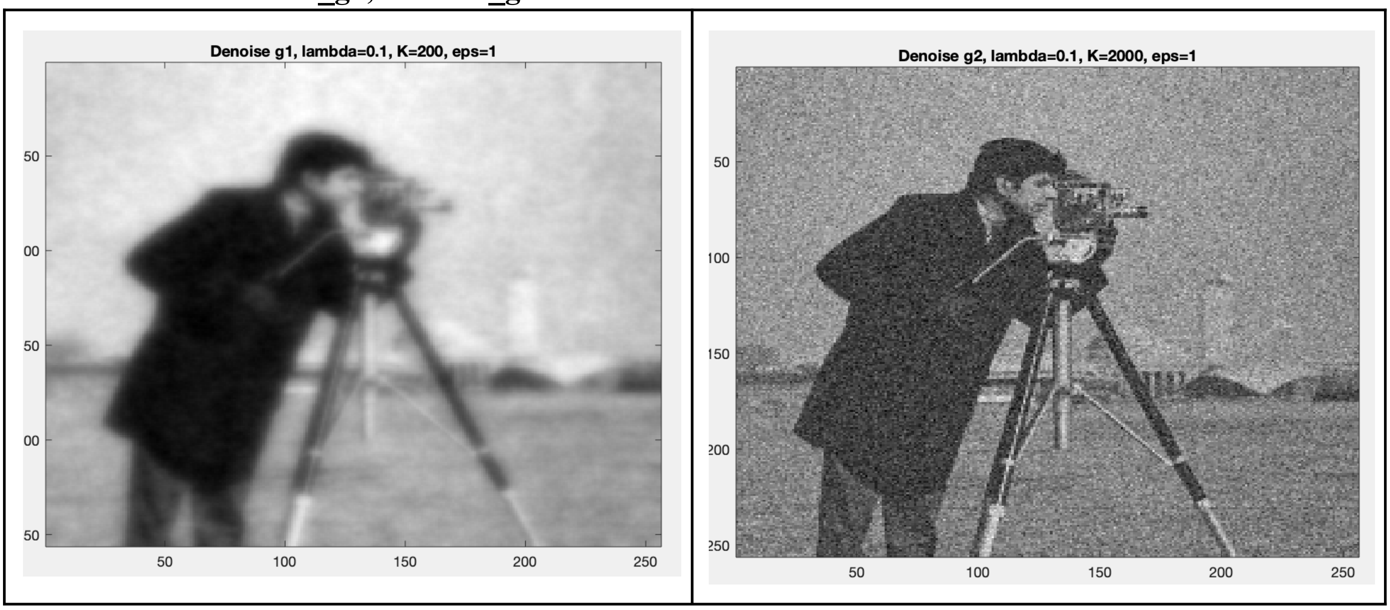


Fig 10. Effect of *lambda* in Tikhonov painting

As it is shown in Fig 10, by decreasing the lambda value image gets more smooth. However, in *lambda=0.01,* there was no acceptable result. Based on these experiments, we say it has the following disadvantages: 1) the losing sharpness in denoising 2) tuning the parameters.

Task 14. Denoise_g1, Denoise_g2



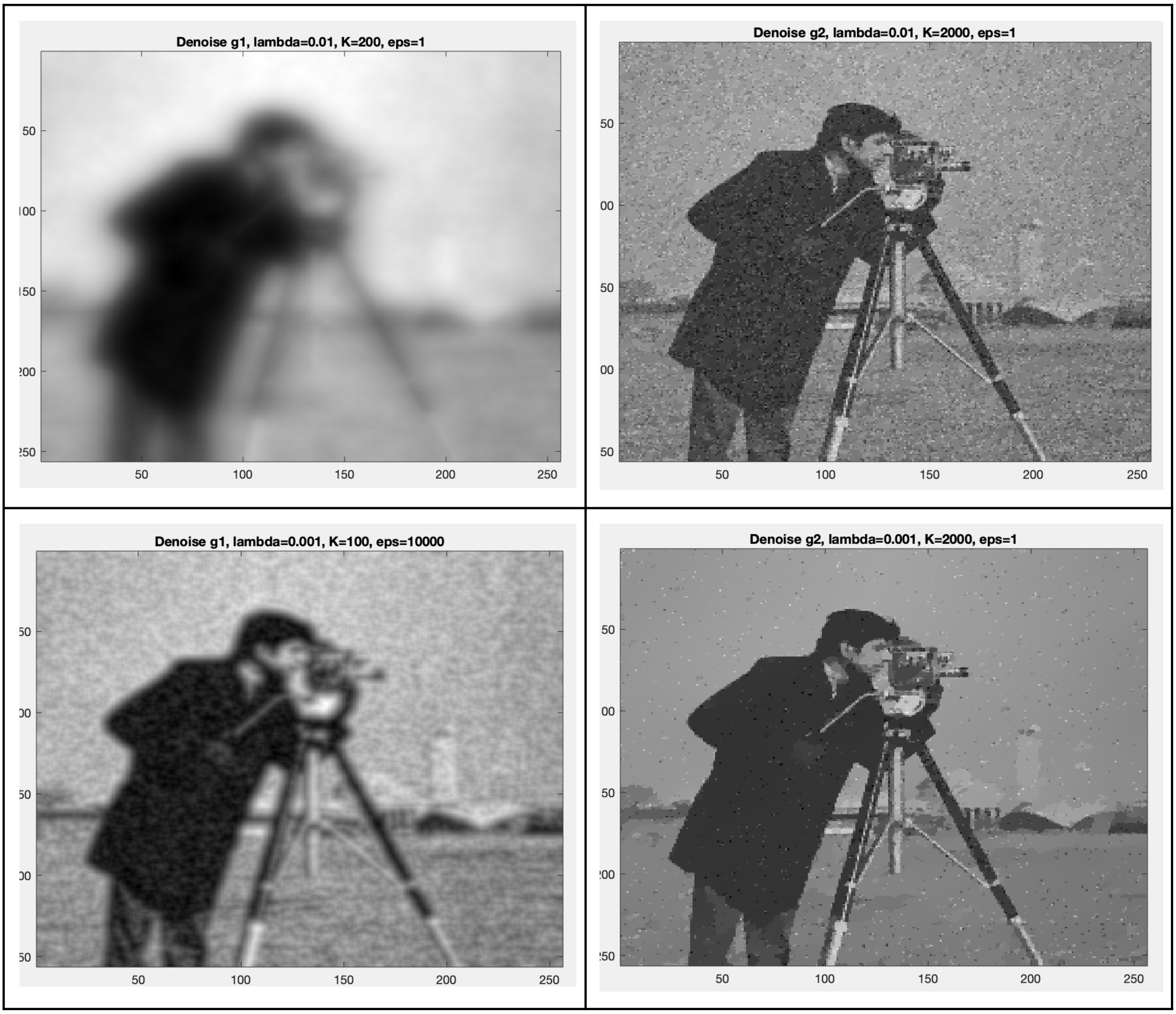


Fig 11. Denoising by g1 and g2

1 $\varphi_1(\xi) = \dfrac{\xi^2}{1+\xi^2}$

$$\nabla \varphi_1(\xi) = \frac{2\xi \cdot (1+\xi^2) - \xi^2 \cdot \xi \cdot 2}{(1+\xi^2)^2} =$$

$$= \frac{2\xi + \cancel{2\xi^3} - \cancel{2\xi^3}}{(1+\xi^2)^2} = \frac{2}{(1+\xi^2)^2}$$

2 $\varphi_2(\xi) = \log(1+\xi^2)$

$$\nabla \varphi_2(\xi) = \frac{2\xi}{1+\xi^2}$$

Denoising by g1 (Fig 11) gave worse results comparatively with the Total Variation approach (Fig 3). It is more similar to the Tikhonov (Fig 2) approach rather than TV. Its result can be improved by tuning the parameters. For example, increasing the *eps* value. If we look at g2 results (Fig 11, second column) has better results than g1 in a high number of iterations(Fig 11, first column).