Image Denoising Using Anisotropic Diffusion

Tom BENACIN

Theoretical approach :

I would like to reduce grain - or noise - in photographs while conserving edges. Rather than applying a global transformation to every image pixel, the idea is to exploit the local environment of each pixel. In an area where the color is very close, it will tend to uniformize this color. On both sides of an edge where the colors are very distinct, this uniformization will be weaker.

Our local approach hints at a diffusion phenomenon. Hence, formulating an energy and minimizing it is will be our goal. We will be relying on the Perona-Malik diffusion model.

$$E(I) = \int_{\Omega} c(\|\nabla I\|) d\Omega \qquad \text{where} \qquad c(\|\nabla I\|) = L(I, I_x, I_y, x, y)$$

First with a linear energy $c(x) = x$, then with a quadratic one $c(x) = x^2/2$. We use the gradient descent approach to determine local minima and obtain :

$$\nabla E = L_I - d(c'(I_x/\|\nabla I\|)/dx \; - \; d(c'(I_y/\|\nabla I\|)/dy$$

$$\nabla E = \; - \nabla (c'(\; \|\nabla I\|) \; I \; / \; \|\nabla I\| \;)$$

$$I_t = \; - \nabla E \; = \; \nabla (c'(\|\nabla I\|) \; I \; / \; \|\nabla I\|)$$

For the linear penalty c(x), to avoid divergence of It, we introduce an infinitesimal epsilon and get a rectified energy of the form :

$$E(I) = \int_{\Omega} \sqrt{I_x^2 + I_y^2 + \varepsilon} \, d\Omega$$

$$I_t = [I_x^2 I_{yy} \; - \; 2 I_x I_y I_{xy} \; + \; I_y^2 I_{xy} \; + \; \varepsilon(I_{xx} \; + \; I_{yy})]/(I_x^2 \; + \; I_y^2 \; + \varepsilon)^{3/2}$$

The CFL condition obtained is :

$$\Delta t / \sqrt{\varepsilon} \leq (\Delta x)^2 / 4$$

Now, for $c(x) = x^2/2$, the energy is of the form :

$$E(I) = \int_{\Omega} [\ (I_x^2 + I_y^2)/2\ ]\ d\Omega$$

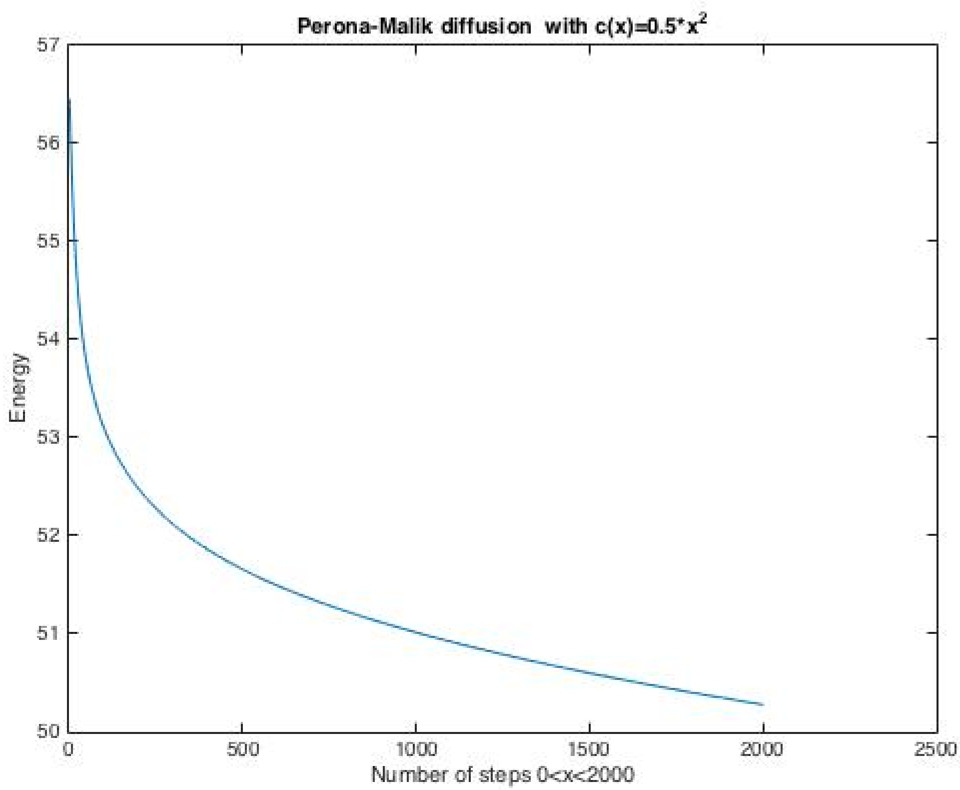
With gradient descent, we obtain a linear heat equation with diffusion coefficient one.

$$I_t = \Delta I$$

In both cases, we use a central difference spatially because we want the next state to depend on the local environment of each pixel. With a forward difference, there is a restriction on the number of pixel neighbors. For each pixel located at the edge of the image, we apply a central difference as if the same pixel is behind the edge.

Experimental results :

For the quadratic penalty $c(x) = x^2/2$, we get the following energy evolution. It is declining very steeply during the 200-300 first steps and then it's decreasing more slowly linearly.



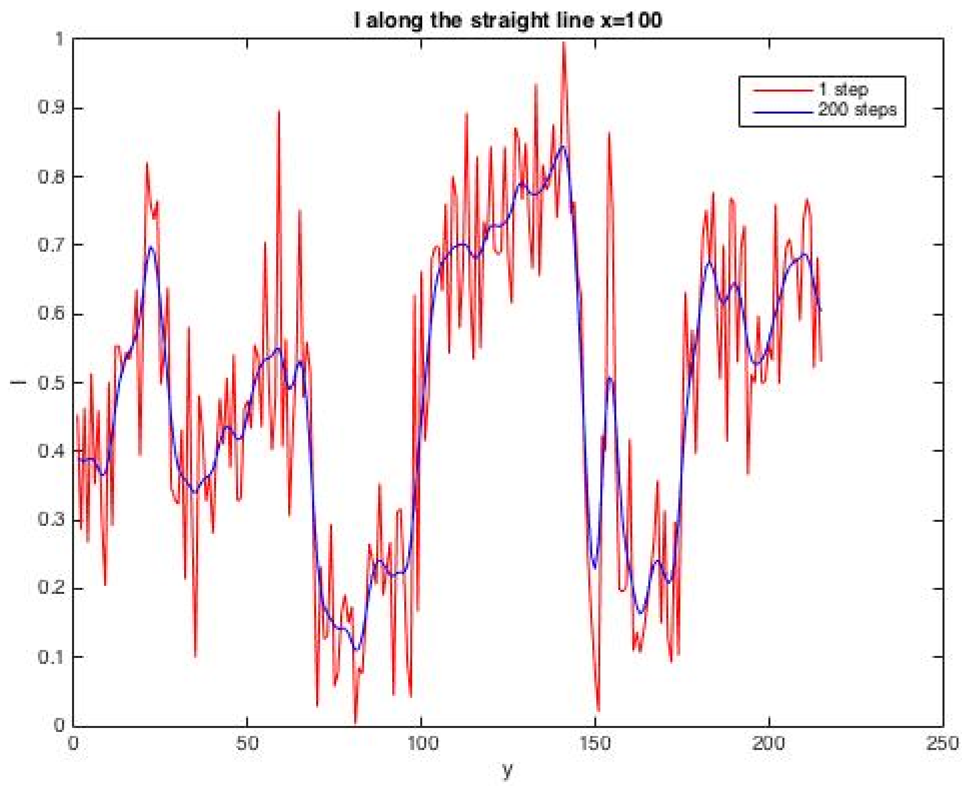
We present the following image for : 0 steps, 200 steps, 300 steps and 2000 steps.

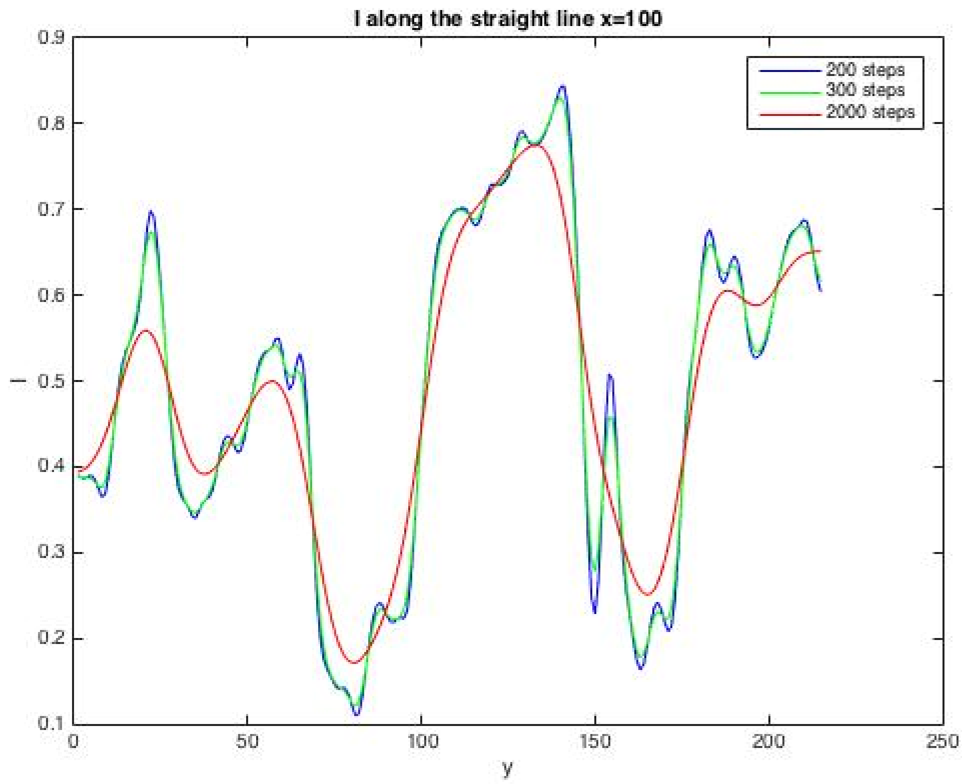
It is noticeable that the process seems to eliminate noise at first but that for a great number of iterations the diffusion process works against us and makes the image blurry.

Let's have a look at edges. Our approach is to follow a straight line along the picture to notice how edges are being conserved.

The initial edge relief is not shown because it is too noisy which makes the graph completely red. We choose $\Delta t = (\Delta x)^{2}/100$. It is a small step but since the algorithm runs quickly, it is worth getting a more precise result.

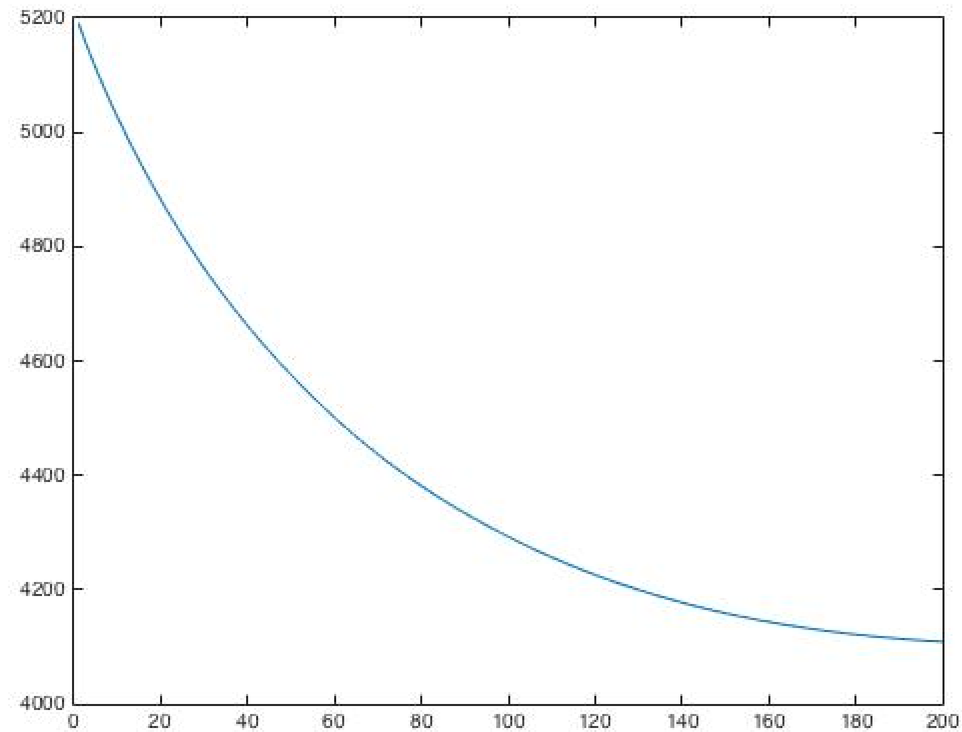
After applying 200 steps, it is as if the initial image has been averaged and distinct edges appear.





After adding another 100 steps, we notice that the edges are becoming less rough. After adding 1000 more steps, our process in smoothening the edges too much hence destroying them.

For the linear penalty $c(x) = x$, we obtain the following energy evolution depending on the number of steps. The (0x) axis is the number of steps and the (0y) axis is the image energy. I picked $\Delta t = \sqrt{\varepsilon}\,\Delta x / 10$ which is close to the CFL condition because the algorithm is very greedy. Hence, we won't be getting a very precise result As earlier, we notice a sharp drop at first put the second decline is even slower than earlier. Therefore, something efficient computationally would be around the inflexion point (around 80 steps).

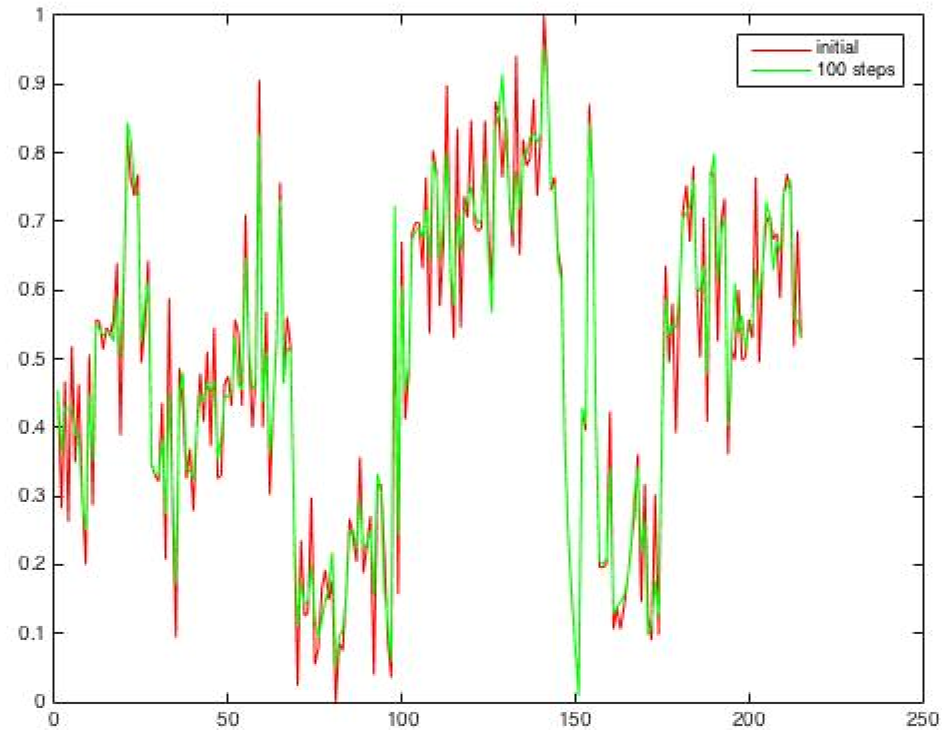


As a side note, I have noticed that an infinitesimal ε is very unpractical because it implies a very small time step and the results seems very similar from the original image.

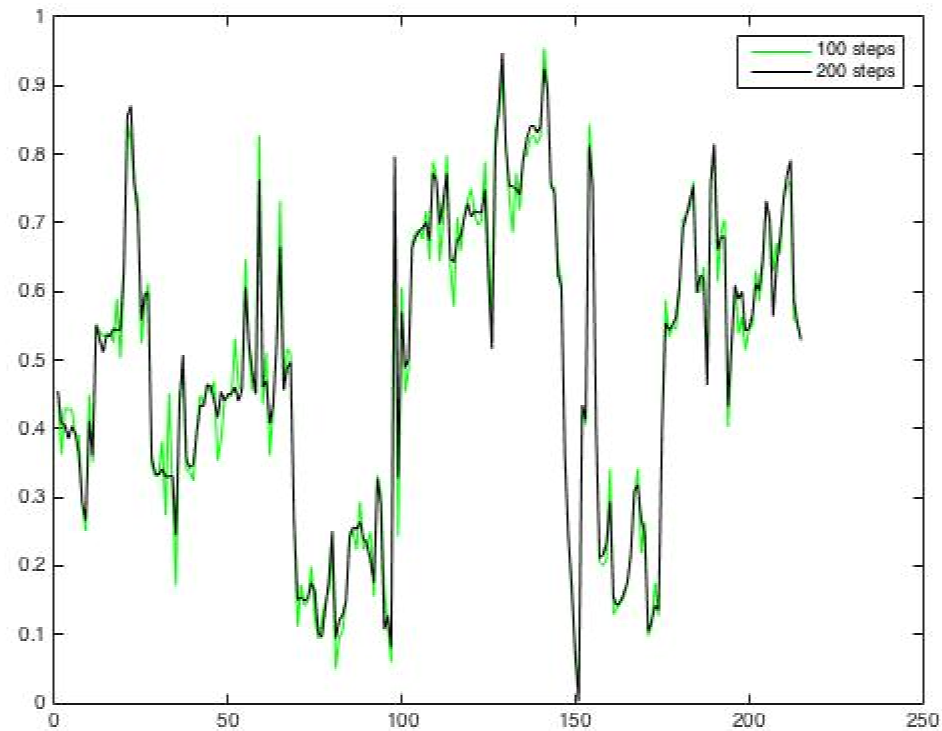
We present the image for : 0 steps, 100 steps, 200 steps, 400 steps.

It is noticeable that the grain of the image is being reduced, most particularly on the third picture. However, it seems that some sort of noise or blurriness is appearing for great numbers of steps. Therefore, determining what is the adapted number of steps seems to depend on the particularities of each image.

Let’s have a look at edge preservation. We notice the spikes after 100 steps are not as extreme as in the initial image, but the profile we obtain is clearly not as smooth as what we got with a quadratic penalty.



If we add another 100 steps, we get a similar treatment which gives us a smoother aspect overall but the curves are still pointy.



Conclusion :

This project was an opportunity to get new insights about image denoising. We confronted two models that derive from the Perona-Malik approach.

Computationally, the quadratic one gave very quick results but not the linear one. Therefore, it is the role of the experimentator to adapt the time step. It is clearly a trade-off, by reducing it we are getting a better precision but we are losing in speed.

It also seems that for an excessive number of steps, the result becomes unsatisfactory and we obtain blurriness or new forms of noise. Therefore, it is at the appreciation of the experimentator since it appears to depend on the image as well.

Regarding edge conservation, which is one of the goals of this study, the quadratic approach seems more destructive than the linear one. It is mainly because it smoothes the edges of the image whereas the linear penalty keeps a roughness in the edges.

From a practical standpoint, the quadratic penalty gives a quick approximation with only a few iterations. In the long run, for sharper edges and a more qualitative result, it seems smarter to opt for the linear penalty.