Digital Image Processing Assignment

NLM Implementation

S Sai Girish
16CO244

The Three different methods of NLM methods chosen are : Non-Local Means Filter (NLmeansfilter), Probabilistic Non-Local Means (PNLM) and Fast Nonlocal Means (fast_nl_means).

Non-Local Means Filter (NLmeansfilter) algorithm and description:

Given a discrete noisy image v = {v(i) | i $\in$ I}, the estimated value NL[v](i), for a pixel i, is computed as a weighted average of all the pixels in the image, NL[v](i) = j$\in$I w(i, j)v(j), where the family of weights {w(i, j)}j depend on the similarity between the pixels i and j, and satisfy the usual conditions 0 $\leq$ w(i, j) $\leq$ 1 and j w(i, j)=1.

The similarity between two pixels i and j depends on the similarity of the intensity gray level vectors v(Ni) and v(Nj), where Nk denotes a square neighborhood of fixed size and centered at a pixel k. This similarity is measured as a decreasing function of the weighted Euclidean distance, v(Ni) − v(Nj)2 2,a, where a > 0 is the standard deviation of the Gaussian kernel. The application of the Euclidean distance to the noisy neighborhoods raises the following equality E‖v(Ni) − v(Nj)‖2 2,a = ‖u(Ni) − u(Nj)‖2 2,a + 2σ2.

This equality shows the robustness of the algorithm since in expectation the Euclidean distance conserves the order of similarity between pixels.

The pixels with a similar grey level neighborhood to v(Ni) have larger weights in the average.

These weights are defined as, w(i, j) = 1 Z(i) e− ‖v(Ni)−v(Nj)‖2 2,a h2 , where Z(i) is the normalizing constant Z(i) = j e− ‖v(Ni)−v(Ni)‖2 2,a h2 and the parameter h acts as a degree of filtering. It controls the decay of the exponential function and therefore the decay of the weights as a function of the Euclidean distances.

The NL-means not only compares the grey level in a single point but the geometrical configuration in a whole neighborhood. This fact allows a more robust comparison than neighborhood filters.

Probabilistic Non-Local Means (PNLM) brief description:

Instead of including the exponential function in weighting pixels like NLM, PNLM proposes the following probabilistic weight

$$w_{l,k} = f_{l,k}(\widehat{D}_{l,k}/\rho^2)$$

Where $f_{t,k}$ is the theoretical probability density function (p.d.f.) of the r.v. $D_{t,k}$, $D^{\wedge}{}_{t,k}$ is the estimated distance with estimated variance, and is a tuning parameter. This weight function can be interpreted as the probability of seeing a noisy patch difference when two clean patches match perfectly. Since it is clear it's given a smaller weight in the more typical case when this perfectly matching condition fails, the equation then gives the largest similarity weight we shall consider.

Fast Nonlocal Means (fast_nl_means) brief description:

Although the nonlocal means (NLM) algorithm takes a significant step forward in the image filtering field, it suffers from a high computational complexity. To deal with this drawback, this method proposes an acceleration strategy based on a correlation operation. Instead of per-pixel processing, this approach performs a simultaneous calculation of all the image pixels with the help of correlation operators.

This method filters the whole image simultaneously instead of filtering pixel by pixel as the original NLM one. This new approach shows that both the computational complexity and the running time are greatly reduced when compared to the original NLM method. The acceleration method has to be applied using the simplified Euclidean distance but there are methods using which we can overcome this limit by effectively calculating the Gaussian kernel convolution.

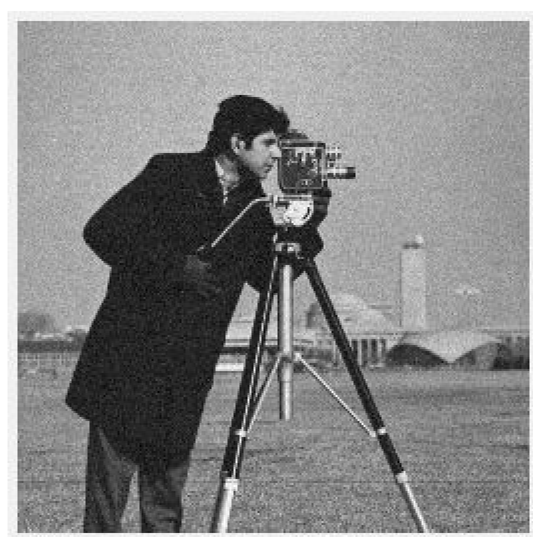
Analysis of NLM methods with Experimental Results

Let's take the image of a cameraman as shown below.

(Original Image)

Let's Create noisy images by adding Gaussian noise of standard deviations 5, 10, 15, 20 and 25.

(Noise of SD 5)



(Noise of SD 10)

(Noise of SD 15)

(Noise of SD 20)

(Noise of SD 25)

Now, **For an Image with Gaussian noise of standard deviation 5, let's compare the results of the three NLM methods using PSNR as a similarity metric.**

(NLM, PSNR = 34.2604dB)

(Fast NLM, PSNR = 34.2818dB)

(Probabilistic NLM, PSNR = 37.3817dB)

From the PSNR values obtained from the denoised images using the three different NLM methods, we can say that Probabilistic NLM performs better than the other two and Fast NLM performs slightly better than the NLM method.

For an Image with Gaussian noise of standard deviation 10, let’s compare the results of the three NLM methods using PSNR as a similarity metric.


(NLM, PSNR = 32.031dB) (Fast NLM, PSNR = 32.0432dB)

(Probabilistic NLM, PSNR = 33.0513dB)

Again from the PSNR values obtained from the denoised images using the three different NLM methods, we can say that Probabilistic NLM performs little better than the other two and Fast NLM performs slightly better than the NLM method.

For an Image with Gaussian noise of standard deviation 15, let's compare the results of the three NLM methods using PSNR as a similarity metric.

(NLM, PSNR = 30.5117dB)

(Fast NLM, PSNR = 30.5108dB)

(PNLM, PSNR = 30.814dB)

From the PSNR values obtained from the denoised images using the three different NLM methods, we can say that Probabilistic NLM performs slightly better than the other two and NLM performs slightly better than the Fast NLM method.

For an Image with Gaussian noise of standard deviation 20, let's compare the results of the three NLM methods using PSNR as a similarity metric.

(NLM, PSNR = 29.2255dB)

(Fast NLM, PSNR = 29.2226dB)

(Probabilistic NLM, PSNR = 29.3868dB)

From the PSNR values obtained from the denoised images using the three different NLM methods, we can say that Probabilistic NLM performs little better than the other two and NLM performs slightly better than the Fast NLM method.

For an Image with Gaussian noise of standard deviation 25, let's compare the results of the three NLM methods using PSNR as a similarity metric.

(NLM, PSNR = 28.1023dB)

(Fast NLM, PSNR = 28.1065dB)

(Probabilistic NLM, PSNR = 28.3633dB)

From the PSNR values obtained from the denoised images using the three different NLM methods, we can say that Probabilistic NLM performs little better than the other two and Fast NLM performs slightly better than the NLM method.

CONCLUSION

From the PSNR values obtained from the denoised images using the three different NLM methods for the Images with Gaussian noise of standard deviation 5, 10, 15, 20, 25, we can conclude that the probabilistic NLM method performs better than Fast NLM and NLM methods for all the values of standard deviation. Whereas there is no conclusive evidence of one of NLM and Fast NLM performing better than the other. For standard deviations 5, 10, 25 Fast NLM performs slightly better than the NLM and for standard deviations 15 and 20, NLM performs better than Fast NLM.

RESOURCES

Link for the code is available at
https://github.com/saigirishgilly98/NLM_Implementation/blob/main/Code.m

Link for the generated images which are used in this document are available at
https://github.com/saigirishgilly98/NLM_Implementation/tree/main/Output_Images