Non Local Image Denoising Using Image Adapted Neighborhoods

Álvaro Pardo

Department of Electrical Engineering, Faculty of Engineering and Technologies,
Universidad Católica del Uruguay
apardo@ucu.edu.uy

Abstract. In recent years several non-local image denoising methods were proposed in the literature. These methods compute the denoised image as a weighted average of pixels across the whole image (in practice across a large area around the pixel to be denoised). The algorithm non-local means (NLM) proposed by Buades, Morel and Coll showed excellent denoising capabilities. In this case the weight between pixels is based on the similarity between square neighborhoods around them. NLM was a clear breakthrough when it was proposed but then was outperformed by algorithms such as BM3D. The improvements of these algorithms are very clear with respect to NLM but the reasons for such differences are not completely understood. One of the differences between both algorithms is that they use adaptive regions to compute the denoised image. In this article we will study the performance of NLM while using image adapted neighborhoods.

1 Introduction

In this work we assume that the observed image, x , is the result of adding a random noise component n to the original noiseless image z . Therefore, the relationship at pixel i becomes: $x_i = z_i + n_i$. The problem of image denoising is to estimate z while preserving its features such as edges and texture. To preserve these features several non-linear or locally adapted methods have been developed. Non-local methods are an example of non-local and non-linear denoising methods. In [4] Buades, Morel and Coll presented the Non Local Means (NLM) denoising method. The underlying idea of this method is to estimate, z_i , using a weighted average of all pixels in the image. Given the pixel to be denoised, i , the weights w_{ij} measure the similarity between neighborhoods centered at i and j . The trick is that corresponding neighborhoods are found all over the image imposing a non-local nature to the method. In practice, this is not computationally efficient and similar neighborhoods are looked in a search windows around pixel i . For more details on similar non-local methods see [1,2,6,5,3].

In this work we explore our claim that locally adaptive neighborhoods play an important role in the performance of non-local methods. To prove this claim we will compare the performance of NLM against a NLM with image adapted neighborhood configurations. First, for synthetic images, based on the local structure of the image an optimal neighborhood is selected. For natural images we will use Principal Component Analysis (PCA) to extract image adapted neighborhoods.

2 Non Local Means Denoising

The NLM algorithm [4] estimates the denoised value at pixel i , $\hat{x}_i$, using a weighted average of all pixels in the search region R_i :

$$\hat{x}_i = \frac{\sum_{j \in R_i} w_{ij} x_j}{\sum_{j \in R_i} w_{ij}} \quad (1)$$

The weights w_{ij} reflect the similarity between pixels i and j based on the distance between neighborhoods N_i and N_j around them. In [4] the authors proposed to compute the weight as:

$$w_{ij} = \exp(-\|N_i - N_j\|_2^2 / \sigma^2). \quad (2)$$

N_i, N_j are image neighborhoods of size $(2K + 1) \times (2K + 1)$ centered at pixels i and j respectively and $\|N_i - N_j\|_2^2$ is de squared distance between them. The parameter σ controls the weights; with a small sigma only points with close neighborhoods receive weights close to one.

In some cases the weights w_{ij} are not able to discriminate between different neighborhood classes. This is critical along edges since pixels along them have less corresponding neighborhoods in the image. The main difficulty is that in several cases, for instance when i lies over a corner, it is very difficult to find similar neighborhoods in R_i . So, even if we could segment the pixels in R_i based on its neighborhoods configuration, it will not be enough. Therefore, in order to solve this situation we need to modify the neighborhood configuration.

2.1 Neighborhood Selection for Synthetic Images

The idea here is to select the neighborhood configuration that best suits the local structure around each pixel i . This a classical idea that dates back to [7]. In Figure 1 we show the nine neighborhoods considered. If i is over a smooth region the configuration (9) should be selected. On the other hand, if i lies along a vertical edge configuration (3) or (4) should be used. The selection is based on the distance between the mean inside and outside the pixels in neighborhood.

When we plug the procedure described above into NLM algorithm we found that the best MSE for traditional NLM is outperformed by the modified NLM here proposed. For the image in Figure 2 the MSE for NLM is 4.38 and for the modified NLM here proposed is 2.12. So, adapting the neighborhood during similarity computation seems a promising idea. Unfortunately, when we applied the same algorithm to real images we found that the modified NLM, for the images un Figure 3, was outperformed by the traditional NLM. Observing the map that showed the selected neighborhood configuration for each pixel we found that it was very noisy. That means that the selection of the neighborhood was unstable. On one hand the noise present in the image makes it very difficult to correctly select the true neighborhood configuration. On the other hand, natural images rarely have local configurations as the ones shown in Figure 1. Therefore, to apply the same idea to natural images we need another way to introduce the local configurations into the neighborhoods similarity computation. The requirements

Fig. 1. Neighborhood configurations. From left to right and top to bottom the neighborhoods are numbered from 1 to 9.

Fig. 2. Synthetic image corrupted with Gaussian noise

for this are stability against noise and the capability to extract the local configurations present in the image. In next section we will show that traditional Principal Component Analysis (PCA) fulfills both requirements and gives good results in terms of MSE.

3 PCA and NLM

The use of PCA in the context of non-local method was mainly used to reduce the computational complexity via dimensionality reduction. In [9] the author reviews the more relevant literature in this area. Also explores the use of PCA to compute what he calls the Principal Neighborhoods (the eigenvectors computed with PCA). We will also use PCA to compute image adapted neighborhoods. The main difference with his approach and ours is that in our case we use this idea to justify the need of image adapted neighborhoods. In this way we are looking the same problem from another

Fig. 3. Test images: Barbara, Couple, Einstein, GoldHill, House, Boat, Peppers, Baboon

point of view. Finally, the estimation of the parameters of NLM is done in a completely different way (see Section 4).

Our claim is that using PCA we can extract the local configuration of each neighborhood being processed. Using the correct neighborhood we compute the distance considering only the relevant information while being, at the same time, insensitive to noise. Instead of selecting the relevant information in the pixel space, as we did for synthetic images, for natural images we do it in the projected space obtained with PCA. PCA is well known for its stability in the computation of the principal components (eigenvectors) in the presence of noise in the data. Furthermore, the selection of the relevant information to compute the neighborhood similarity is image dependent (PCA is computed for each image being denoised). This is another advantage of PCA; although for every image we start with the same neighborhood configuration in the space of pixels the similarity is image dependent via the principal components. So far we considered that the original image z contains some structure and therefore the use of principal components will allow us to robustly compute the similarity between neighborhoods. As we will see in the Section 4 with textured images there is no clear benefit in using principal components to compute the similarity.

4 NLM Using PCA Neighborhoods

In this section we discuss the implementation of NLM using PCA image adapted neighborhoods.

The new algorithm has several parameters. First we have the parameters of NLM: the size of the neighborhoods, the size of the search windows and the width of the exponential kernel. In this work we fix the first two with values 3×3 and 7×7 respectively. For the setting of the width we follow the idea presented in [5,3] and set the width σ proportional to the noise variance. This can be justified using the following result. The expected distance for two identical neighborhoods with additive and independent gaussian noise with zero mean and variance σ_n results:

$$E \left\{ \sum_{k=1}^{(2K+1)^2} (X_i^k - X_j^k)^2 \right\} = \sum_{k=1}^{(2K+1)^2} E \{ (N_i^k - N_j^k)^2 \} = 2(2K+1)^2 \sigma_n^2. \quad (3)$$

Based on this, first we estimate the σ_n using [8] and then set $\sigma = 3\sqrt{2}\sigma_n$ (for $2K+1 = 3$). The last parameter is the dimension of the projection and will be discussed in the reminder of this section.

The computation of the image adapted neighborhoods is done using PCA. Given the set X containing the n neighborhoods for the image, PCA is applied to obtain the principal components P and the projected information Y . Since we are using neighborhoods of size 3×3 the size of X is $n \times 9$, the size of P is 9×9 and the size of Y is $n \times 9$. The distance between two points in X can be computed using the corresponding

Fig. 4. MSE for test images while varying the number of components d in PCA. The red (horizontal) line indicates the result of traditional NLM. Images: Barbara, Couple, Einstein, Goldhill, House, Boat, Peppers, Baboon.

projected samples in Y . If the coordinates in Y are sorted in descending order with respect to the eigenvalues obtained from PCA, the distance can be approximated with the first d coordinates. This idea was traditionally used to speed up the computational time of NLM. Here we argue that not only this speeds the computations of NLM but also give improvements in terms of MSE since the distance is computed in another space where noise is reduced and the local structure of the image is introduced.

The distance between two neighborhoods N_i and N_j is computed using its corresponding projections in Y : Y_i and Y_j . If we use only d coordinates the we have an approximation of the distance:

$$\|N_i - N_j\|_2^2 \approx \sum_{k=1}^d (Y_i^k - Y_j^k)^2.$$

As we will show the approximation at the end improves the results. This reinforces our idea that for the computation of the distances between neighborhoods we must use the local structure of the image. Using PCA components we do it so and also we are immune to the effects of the noise. In Figure 4 we show the evolution of the MSE for different values of d . The red line shows the result of NLM (using all the coordinates in Y). We can observe that in all cases but one, the optimal results are obtained with $d < 9$. Thus, the adaptation of the neighborhoods to the image statistics improves the results of NLM. The only case where this is not true is for baboon, an extremely textured image. Additionally, the MSE is stable with d between 5 and 8. In these experiments artificial Gaussian noise with $\sigma_n = 10$ was added to each image.

5 Discussion, Conclusions and Future Work

As we showed in Figure 4 the MSE using the image adapted neighborhoods improves with respect to traditional NLM. So, not only we obtain a computationally improved algorithm but also we gain in terms of MSE. Based on this data we confirm our claim that the adaptation to the local structure of the image is important in this kind of algorithms. For synthetic images the adaptation can be done using naive neighborhoods. On the other hand, natural images need neighborhoods that reflect the statistics of the image itself. The only image where our proposal fails is Baboon. This is an extremely textured image where it is difficult to extract the structure. This stresses that discovering the structure of the image neighborhoods it is important for this kind of algorithms.

We believe that with this evidence we add some light to the study of non-local denoising algorithms. Although PCA has been studied before, here we study the problem from another perspective. For the future, we want to study the optimal estimation of d and its influence on the results. It is interesting to note that consistently in all the experiments performed the MSE is stable between $d=5$ and $d=8$. The work [9] studied this problem and would be an starting point in our work. Also, we expect to explore the possibility of defining binary masks in the PCA space as we did in the pixel space in the case of synthetic images.

References

1. Awate, S.P., Whitaker, R.T.: Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(3), 364–376 (2006)
2. Boulanger, J., Kervrann, C., Bouthemy, P.: Adaptive space-time patch-based method for image sequence restoration. In: *Workshop on Statistical Methods in Multi-Image and Video Processing*, SMVP 2006 (May 2006)
3. Buades, A., Coll, B., Morel, J.M.: The staircasing effect in neighborhood filters and its solution. *IEEE Transactions on Image Processing* 15(6), 1499–1505 (2006)
4. Buades, A., Coll, B., Morel, J.M.: Denoising image sequences does not require motion estimation. In: *Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance*, pp. 70–74 (2005)
5. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. *SIAM Multiscale Modeling and Simulation* 4(2), 490–530 (2005)
6. Mahmoudi, M., Sapiro, G.: Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE Signal Processing Letters* 12(12), 839–842 (2005)
7. Nagao, M., Matsuyama, T.: *A structural analysis of complex aerial photographs*. Plenum Press, New York (1980)
8. Olsen, S.I.: Noise variance estimation in images. In: *Proc. 8th SCIA*, pp. 25–28 (1993)
9. Tasdizen, T.: Principal neighborhood dictionaries for nonlocal means image denoising. *IEEE Transactions on Image Processing* 18(12), 2649–2660 (2009)