Non-Local Means Denoising Using a Content-Based Search Region and Dissimilarity Kernel

Hila Berkovich, David Malah, and Meir Barzohar
Department of Electrical Engineering
Technion - Israel Institute of Technology
Haifa 32000, Israel
hilaber@tx.technion.ac.il, malah@ee.technion.ac.il, meirb@visionsense.com

***Abstract*— The Non-Local Means (NLM) denoising algorithm uses a weighted average of pixels, within a defined search region in an image, to estimate the noise-free pixel value. The search region is usually a rectangular neighborhood, centered at the pixel of interest, which may include pixels whose original gray value do not match the value of the original central pixel. Consequently, their participation in the averaging process degrades denoising performance. To eliminate their effect, researchers suggest creating an adaptive search region which excludes those dissimilar pixels. In this paper, we present a novel model-based method which defines a set of similar pixels, from the initial search region, using the statistical distribution of the dissimilarity measure. Moreover, to enhance the denoising, our method also adaptively assigns one of two dissimilarity kernels to each pixel, based on its local features. Experimental results show that the proposed algorithm has better performance than the original one in terms of PSNR, SSIM, and visual quality and is found to be more efficient than other examined approaches.**

Keywords—Image Denoising; Non-Local Means

I. INTRODUCTION

Image denoising is used to find the best estimate of the original image given its noisy version. Many methods for image denoising have been suggested, and a comprehensive review of them can be found in [3]. Among the proposed denoising schemes, patch-based methods have drawn much attention in the image processing community. In 2005, Buades et al. [3] introduced the Non-Local Means (NLM) denoising algorithm which takes advantage of image redundancy by comparing pixel neighborhoods within an extended search region. Each pixel value is estimated as a weighted average of all other pixels in this search region. These pixels are each assigned a weight that is proportional to the similarity between the local neighborhood of the reference pixel and their local neighborhood, such that pixels whose neighborhood is the most similar to the neighborhood of the reference pixel are given the largest weights. Moreover, the weights are controlled by a weight smoothing parameter (h), which steers their decay. It increases with the noise variance in the image and it is usually set constant for the entire image. Since image pixels are highly correlated while noise is typically independent and identically distributed (*i.i.d.*), weighted averaging of these pixels results in noise reduction. Consequently, the uniqueness of the NLM approach lies in its ability to exploit spatial correlation in a defined neighborhood (search region) for noise removal. The search region is usually a rectangular neighborhood, centered at the pixel of interest (POI), which may include pixels whose original gray value do not match the value of the original central pixel. Therefore, their participation in the averaging process may degrade the de-noising performance. The NLM algorithm is sensitive to the selection of the weight smoothing parameter (h). This parameter plays the same role as kernel support, such that the larger the support, the smoother the image becomes. Hence, for textural regions a smaller value of h should be used than for smooth regions, for the same given noise STD. As a result, there are NLM modifications that suggest using an adaptive h [5], which is matched to local structure. An alternative to using an adaptive h is to replace the isotropic square search region in the original NLM by an adaptive anisotropic region in which the most similar pixels to the POI are selected (based on comparing local neighborhoods). This anisotropic neighborhood can better exploit the local image structure. Hence, denoising performance is improved, especially for pixels that belong to textural regions.

Mahmoudi et al. [8] propose to pre-classify the neighborhoods based on local average and gradients, but the calculation of the gradient is affected by noise. Coupe et al. [4] and Kervrann et al. [7] use patch average and patch variance to rule out dissimilar patches. Dinesh at al. [5] suggest a correlation-based patch classification method. The correlation is computed using an inner product between two normalized patches. Only patches, within the search region, whose correlation (with respect to the central patch) is higher than a pre-defined threshold, are considered during the averaging process. The problem with these methods is that the measures which affect pre-selection of pixels use global thresholds which are chosen somewhat heuristically and may vary based on image characteristics, hence imply lack of robustness.

Azzabou et al. [1] suggest partitioning the image into two classes: noisy smooth zones and noisy texture/edge zones. Each pixel is characterized with a statistical model that defines a membership degree to each class. The method relies on a prior, which does not necessarily satisfy all explored images, and involves a Gaussian Mixture Model as well as EM optimization that are computationally expensive. Orchard et al. [9] propose an alternative strategy that uses the SVD to more efficiently eliminate dissimilar pixel pairs. The method relies on dimensionality reduction of the image patches by setting a global dimension for all patches. This dimension value may cause over-smoothing since texture and edge patches should be characterized with a higher dimension than smooth patches. In [2], Brox et al. suggest to classify pixels by using a cluster tree approach and K-Means (K=2). This is equivalent to image segmentation based on an iterative binary classification and hence is not necessarily robust. Sun et al. [12] present a method that determines a pixel-wise adaptively-shaped search region, within which the image is homogeneous. The method is subjected to a contiguous search region shape that is not necessarily the best shape within the pre-defined search region.

In this work, we too suggest an adaptive approach which classifies the search region into two groups: similar pixels (with respect to the central pixel) and dissimilar pixels. The novelty of our approach is that pixel association with any of these two groups is based on a probabilistic model that characterizes the dissimilarity measure between two compared patches. Consequently, each pixel is characterized by a content-based search region constructed from the similar pixels group. Moreover, in order to enhance the denoising effect, the dissimilarity kernel is also adjusted to the local structure, instead of using a single kernel type (Uniform or Box), as usually done. The suggested approach improves NLM performance both PSNR-wise and SSIM-wise [11], as well as visually.

The remainder of this paper is as follows. Section II introduces a brief description of the NLM method. The proposed adaptive method is detailed in Section III. Section IV introduces experimental results with comparison to the common NLM method and the method described in [12]. Finally, Section V presents a summary and concluding remarks.

II. NON-LOCAL MEANS IMAGE DENOISING

In this section, a brief overview of the non-local means method introduced in [3] is presented. Let X and Y be the original and the observed noisy images, respectively. It is assumed that the original image is corrupted by a Gaussian noise N with a zero mean and a known standard deviation σ_n, such that

$$Y = X + N, \quad N \sim \mathrm{N}(0,\sigma_n^2) \tag{1}$$

Each pixel in the restored image is derived as the weighted average of all gray values within a defined search region:

$$\hat{X}_i = \sum_{j\in S_i} w_{i,j} Y_j \tag{2}$$

where i represents a pixel index, S_i refers to a rectangular search region of size MxM centered at pixel i. The normalized weights, which can be referred to as similarity probabilities, are defined as:

$$w_{i,j} = \frac{1}{W_i}\cdot \exp\left(-\frac{d_p(i,j)}{h^2}\right),\ i \in Y, j \in S_i \tag{3}$$

such that $W_i = \sum_{j\in S_i} w_{i,j}$ is a weight normalization factor, $d_p(i,j)$ is the dissimilarity measure, and h is the weight smoothing parameter, which is typically controlled manually in the algorithm. Choosing a very small h leads to noisy results almost identical to the input, while a very large h gives an overly-smoothed image. The dissimilarity measure $d_p(i,j)$ is defined over the corresponding similarity patches as follows:

$$d_p(i,j) = \left\| Y(A_i) - Y(A_j) \right\|_{2,a}^2 \tag{4}$$

where $Y(A_i)$ defines a vector of neighborhood pixel values and A_i represents a square similarity patch of size $p \times p$ centered at pixel i *(p<M)*. The similarity patches overlap within a given S_i. The vector norm is simply the Euclidean difference, weighted by a Gaussian kernel of zero mean and variance a that is used to smooth out the neighborhood while calculating the weights (i.e., give decaying weight to pixels away from similarity patch center). This filter reduces the effect of differences in pixel intensities as they get spatially further away from the center of the patch. In practice, instead of a Gaussian kernel, simpler kernels are used; a Uniform kernel (which assigns the same weights to all the pixels of the similarity patch), whose corresponding dissimilarity measure is denoted $d_p^U(i,j)$, and a Box kernel, illustrated in Fig. 1, whose corresponding dissimilarity measure is denoted $d_p^B(i,j)$.

The weights computation is a bottle neck of the NLM algorithm. Therefore, restricting the size of the search region (M^2) is important for practical implementation.

In this paper, the similarity patch is set to be 5x5 (*p*=5) and the search region is set to 11x11 (*M*=11), as suggested in [10].

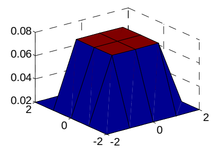


Fig. 1: The Box kernel used for dissimilarity computation

III. PROPOSED ADAPTIVE METHOD

This section describes the two main innovations of our suggested method.

A. *Search Region Pixel- Classification*

NLM uses a weighted average of pixels, over a defined region, for noise reduction. In the traditional NLM algorithm, for each pixel value to be recovered, all pixels in its defined search region are assigned an adequate weight, i.e., the search region is isotropic. The problem is that there usually exist some pixels whose local neighborhood is very different from that of the POI, referred as dissimilar pixels. Based on eqn. (3), the weights assigned to the dissimilar pixels will always take positive values. Even though they will be assigned small weights, their participation in the averaging process could degrade the overall denoising performance. Moreover, since edge preserving property is very important for a good denoising scheme, the use of dissimilar pixels in the weighted averaging results in an inevitable blurring effect.

To solve the above mentioned problem, researchers have proposed to partition the isotropic search region (S_i) into a set of dissimilar pixels, denoted here S_i^D, and a set of similar pixels, denoted S_i^S. The weighted averaging is then applied only on the S_i^S set.

We propose to determine the above partition by using an approximated distribution of the normalized dissimilarity measure within the S_i^S set, as follows: at first, we refer to the normalized dissimilarity measure that applies a Uniform kernel, which is defined as:

$$\frac{d_p^U(i,j)}{2\sigma_n^2} = \frac{1}{p^2}\frac{\left\|Y(A_i)-Y(A_j)\right\|_2^2}{2\sigma_n^2} = \frac{1}{p^2}\sum_{\substack{m\in A_i\\ l\in A_j}}\left(\frac{Y_m-Y_l}{\sqrt{2}\sigma_n}\right)^2 \qquad (5)$$

Assuming that the original gray values of all the pixels in S_i^S are the same as pixel *i*, i.e., $X_j = X_i \quad \forall j \in S_i^S$, the following applies:

$$\sum_{\substack{m\in A_i\\ l\in A_j}}\left(\frac{Y_m-Y_l}{\sqrt{2}\sigma_n}\right)^2 = \sum_{\substack{m\in A_i\\ l\in A_j}}\left(\frac{N_m-N_l}{\sqrt{2}\sigma_n}\right)^2 \quad , \; A_i, A_j \subset S_i^S \qquad (6)$$

The normalized variable $(N_m - N_l)/\sqrt{2}\sigma_n$ is distributed $\mathrm{N}(0,1)$ as a linear combination of two *i.i.d* variables. Under the simplifying assumption that the dissimilarity values related to the same reference patch over the search region are uncorrelated (in spite of patches overlaps and the use of the same reference patch), the sum (in eqn. (6)) of p^2 squared independent standard normal variables has a Chi-Square distribution with p^2 degrees of freedom. For large p^2, the Chi-Square distribution converges to a Normal distribution with the following first two moments:

$$\chi_{p^2}^2 \to \mathrm{N}\left(p^2, 2p^2\right) \qquad (7)$$

Fig. 2 illustrates the goodness of fit between the two distributions, for $p^2 = 25$.

Combining eqns. (5) and (7) produces the approximated distribution of the dissimilarity measure, defined for the Uniform kernel:

$$\frac{p^2 \cdot d_p^U(i,j)}{2\sigma_n^2} \sim \mathrm{N}\left(p^2, 2p^2\right) \qquad (8)$$

And the distribution of the *normalized* dissimilarity measure:

$$\tilde{d}_p^U(i,j) = \frac{d_p^U(i,j)}{2\sigma_n^2} \sim \mathrm{N}\left(1, \frac{2}{p^2}\right) \qquad (9)$$

In the same manner, for a general kernel the dissimilarity measure is defined as follows:

$$\tilde{d}_p^G(i,j) = \frac{d_p^G(i,j)}{2\sigma_n^2} = \sum_{\substack{m\in A_i\\ l\in A_j}}\alpha_m\frac{(N_m-N_l)^2}{2\sigma_n^2}, \quad s.t. \sum_{m\in A_i}\alpha_m = 1 \qquad (10)$$

where α_m represents the kernel's weights within a given $p\times p$ similarity patch ($\alpha_m = 1/p^2$ for the Uniform kernel). This measure is distributed (in approximation) [6] as:

$$\tilde{d}_p^G(i,j) \sim \mathrm{N}\left(1, 2\sum_{m\in A_i}\alpha_m^{\,2}\right) \qquad (11)$$

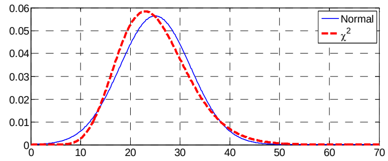


Fig. 2: Normal and Chi-Square distrbutions for $p^2 = 25$ degrees of freedom

We begin with the description of the proposed method when the Uniform kernel is used.

Fig. 3 illustrates a simple case in which S_i^D represents pixels whose original gray level difference with respect to the POI is C, i.e., $Y_m - Y_l = C + N_m - N_l \quad \forall m \in A_i \subset S_i^S, l \in A_j \subset S_i^D$.

Fig. 3: Description of a search region divided into two sets. S_i^D is a group of pixels whose original gray value difference with respect to S_i^S is C.

In a more general case, S_i is divided into multiple sub-regions, i.e., $Y_m - Y_l = C_{m,l} + N_m - N_l \quad \forall m \in A_i \subset S_i^S, l \in A_j \subset S_i^D$, such that the following applies:

$$\sum_{\substack{m\in A_i\\ l\in A_j}}\left(\frac{Y_m-Y_l}{\sqrt{2}\sigma_n}\right)^2 = \sum_{\substack{m\in A_i\\ l\in A_j}}\left(\frac{C_{m,l}+N_m-N_l}{\sqrt{2}\sigma_n}\right)^2, \; A_i \subset S_i^S, \; A_j \subset S_i^D \qquad (12)$$

The term on the r.h.s. of eqn. (12) has a Non-Central Chi-Square distribution with p^2 degrees of freedom and Non-Centrality parameter

$$\lambda = \frac{\sum_{m\in A_i,\, l\in A_j} C_{m,l}^2}{2\sigma_n^2} \qquad (13)$$

Finally, the dissimilarity measure in S_i is characterized by the following first two moments (assuming *N* is *i.i.d*):

$$\mathrm{E}\left[\tilde{d}_p^U(i,j)\right] = 1 + \frac{\lambda}{p^2} \qquad (14)$$
$$Var\left[\tilde{d}_p^U(i,j)\right] = \frac{2}{p^2} + \frac{4\lambda}{p^4}$$

Therefore, the variance of the *normalized* dissimilarity measure vector of the entire search region is equal or larger than $2/p^2$.

The search region can be divided into two sets on the basis of the aforementioned statistical analysis. For example, one can apply a mean-based threshold on the elements $\left\{\tilde{d}_p^U(i,j)\right\}_{j\in S_i}$, like $1+3\sqrt{2}/p$ (corresponding to three STD from the mean, see eqn. (9)). However, this kind of threshold is not robust enough due to a potential cross-talk between the distributions of the two sets of pixels. Setting a threshold on the accumulated variance of the dissimilarity elements in S_i turned out to be more robust and is described in Table I. This scheme is based on the assumption that pixels associated with S_i^S are characterized by a smaller $\tilde{d}_p^U(i,j)$; hence a hard thresholding can be applied.

TABLE I. SEARCH REGION PIXEL CLASSIFICATION

1. For a given pixel *i*, calculate the normalized vector whose elements are $\tilde{d}_p^U(i,j)\ \forall j\in S_i$.
2. Sort the elements of $\tilde{d}_p^U(i,j)$ in an ascending order of values.
3. Compute an accumulated variance of the sorted elements of $\tilde{d}_p^U(i,j)$, by starting with the first two elements and adding one element at a time.
4. Stop accumulating elements when the computed variance exceeds $2/p^2$ and set a dissimilarity threshold which corresponds to the dissimilarity value of the last pixel which participates in the vector accumulation, denoted TH_d.
5. Partition the search region S_i into the following two sets:

$$S_i^S=\left\{j\in S_i \mid \tilde{d}_p^U(i,j)\le TH_d\right\}$$
$$S_i^D=\left\{j\in S_i \setminus S_i^S\right\}$$

Refer to Fig. 4 for an illustration of an accumulated variance vector for a smooth patch and a textural patch. Pay attention that for the smooth patch; all the search region pixels are included in S_i^S since the accumulated variance of all these pixels is smaller than $2/p^2$, whereas for the textural patch only a small fraction of its search region pixels is included.

Hence, each pixel in the image is associated with its individual S_i^S set that defines which pixels will be included in the averaging process. Moreover, each pixel is associated with a normalized cardinality value, $r_i=\left|S_i^S\right|/M^2$, which is defined as the number of pixels used to denoise it, normalized by the number of pixels in the original search region. Fig. 5(b) displays the *normalized cardinality matrix* R, whose elements are r_i, of the image Lena with $\sigma_n=20$. As expected, smooth patches are characterized by a large value (closer to 1), whereas structural patches are associated with a smaller value (usually smaller than 0.5).

We have described above the search region pixel-classification method for the Uniform kernel. It can easily be extended to a general kernel, e.g., the Box kernel, by replacing the $2/p^2$ threshold by $2\sum_{m\in A_i}\alpha_m{}^2$ (refer to eqn. (11)).

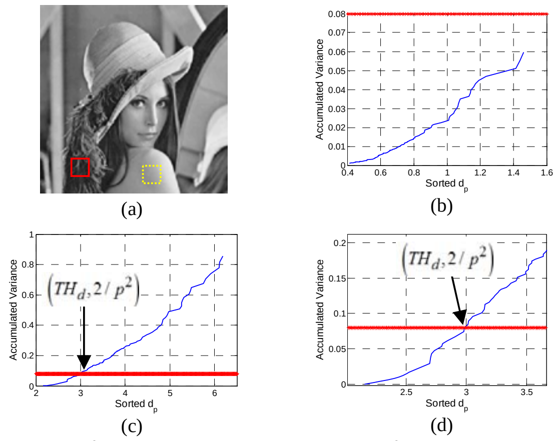


Fig. 4: Graphs of accumulated dissimilarity variance for noisy image Lena with $\sigma_n=20$. (a) Lena (256x256) with two marked search regions, the red one corresponds to texture and the dashed yellow corresponds to a smooth region. (b) Accumulated dissimilarity variance for the smooth search region. (c) Accumulated dissimilarity variance for the textured search region. (d) Zoom-in view of the plot in (c). The red line represents the $2/p^2$ value (for *p*=5). All pixels in S_i whose dissimilarity is smaller or equal to TH_d (see Table I) are associated with S_i^S.

B. *Dissimilarity Kernel-Type Adaptation*

Simulations suggest that the Uniform kernel is more adequate for smooth regions, whereas the Box kernel is more adequate for texture or edges. Consequently, one can use the information embedded in the R matrix in order to decide which pixels will be assigned with which kernel for their corresponding weight computation. The suggested scheme, which is described in Table II, requires the determination of two sets of similar pixels to pixel *i*: S_i^{SU}, S_i^{SB}, which are computed with the Uniform and the Box kernels, respectively, resulting in two normalized cardinality matrices R^U, R^B for each image. Their corresponding elements are denoted r_i^U, r_i^B. The data in each of these matrices is clustered, using K-Means with K=2, into two classes with two corresponding centroids. The cluster, denoted $C_{\max}$, is associated with the larger centroid and represents pixels with a larger r_i value than the set of pixels, denoted $C_{\min}$, associated with the smaller centroid, and hence typically relate to a smooth search region. The cluster $C_{\min}$ represents pixels that are typically considered structural.

Since the computation of two normalized cardinality matrices is time consuming, we suggest a simplified scheme that is described in Table III. This scheme may create an inconsistency in the pixel classification into Smooth or Non-Smooth classes, which characterizes, on average, less than 5% of the pixels in the entire image. This inconsistency stems from the fact that the S_i^S set of pixels whose $r_i<1$, is computed using the Box kernel. However, a pixel having a large r_i value may be assigned to $C_{\max}$ in the K-Means classification. Therefore, the corresponding weights will be calculated using the Uniform kernel although the respective S_i^S set was calculated using the Box kernel. Since the fraction of such pixels is relatively small, the two schemes provide very similar results. Hence, the less time consuming scheme in Table III is recommended.

Moreover, in order to relate to images which are characterized entirely by very small details (both centroids are small) or smooth regions (both centroids are large), the two centroids are compared to 0.5. If they are both larger than 0.5, all pixel weights are computed using the Uniform kernel. Similarly, if they are both smaller than 0.5, all pixel weights are computed using the Box kernel.

TABLE II. DISSIMILARITY KERNEL-TYPE ADAPTATION

1. For each pixel i in the noisy image, do the following:
   a. Compute the elements $\tilde{d}_p^U(i,j)$ and $\tilde{d}_p^B(i,j)$, $\forall j \in S_i$ of the normalized dissimilarity vectors.
   b. Compute the variance of *all* of the elements of each of the vectors.
      i. If the variance of all the elements of $\tilde{d}_p^U(i,j), \forall j \in S_i$ is smaller or equal to $2/p^2$ (see eqn. (9)), then pixel i is considered smooth under this test. Hence, $S_i^{SU} = S_i$ and $r_i^U = 1$.
      ii. If the variance of all the elements of $\tilde{d}_p^B(i,j), \forall j \in S_i$ is smaller or equal to $2\sum_{m \in A_i} \alpha_m^2$ (see eqn. (11)), then pixel i is considered smooth under this test. Hence, $S_i^{SB} = S_i$ and $r_i^B = 1$.
   c. Otherwise, compute the sets S_i^{SU}, S_i^{SB} using Table I and the respective kernels. Then, compute the corresponding r_i^U, r_i^B.
2. Cluster the data in each of R^U, R^B into two clusters using K-Means with K=2, such that each matrix is divided into a set of smooth pixels and a set of structural pixels.
3. Set pixel weights (before normalization) according to the following rule:

$$\forall j \in S_i^S : w_{i,j} = \begin{cases} f\left(d_p^U(i,j)\right) & r_i^U \in C_{\max}^U \text{ and } r_i^B \in C_{\max}^B \\ f\left(d_p^B(i,j)\right) & \text{Otherwise} \end{cases} \quad (15)$$

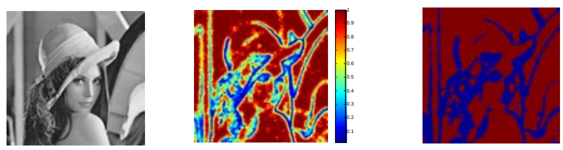
   where $C_{\max}^B, C_{\max}^U$ are the sets associated with the centroids of the larger cluster in R^B, R^U matrices, respectively, and $f\left(d_p^U(i,j)\right)$ represents a function of $d_p^U(i,j)$, like the exponential term in eqn. (3).
4. Normalize the computed weights by their sum (see eqn. (3)).

IV. EXPERIMENTAL RESULTS

To evaluate the performance of our method, we have used several natural images corrupted by synthetic Gaussian noise (with $\sigma_n = 20, 30$). We compared our approach to the conventional NLM algorithm with both the Box kernel and the Uniform kernel. The same parameters were used for all the methods examined, i.e., a similarity patch of size 5x5 (p=5), a search region of size 11x11 (M=11) and $h = \sigma_n$. Two types of performance evaluation were conducted: an objective evaluation using the common measures PSNR and SSIM [11] and a visual evaluation based on the perceived quality by a human observer.

Table IV summarizes the quantitative denoising results (objective evaluation) for different images with different noise levels. From its analysis, we can conclude that the proposed approach obtains somewhat higher PSNR and SSIM values than the conventional NLM algorithm for both the Box and the Uniform dissimilarity kernels. This tendency is preserved both for textural images (e.g., Baboon) and for smoother images (e.g., Lena, Pepper). When using only similar pixels with a set kernel, the PSNR improvement is smaller. Average improvement based on three selected images with $\sigma_n = 20$ provides an increase of 0.07 dB and 0.38 dB for using the Box kernel and the Uniform kernel respectively, compared to an average increase of 0.15dB and 0.42 dB when adding the adaptive kernels.

A visual comparison is given in Fig. 6, which compares the denoised images produced by the conventional NLM (for both the Box and the Uniform dissimilarity kernels) and our proposed NLM, for the image Lena with $\sigma_n = 20$. Fig. 6(b), which represents a conventional NLM denoising using a Box kernel, preserves structure but is characterized with granularity in smooth regions. Fig. 6(c), which presents the denoising results of the conventional NLM with Uniform kernel, reduces the granularity effect, but over-smoothes texture and edges. On the other hand, Fig. 6(d), which presents the proposed method, is characterized by both preservation of structural information and no granularity in smooth regions. Moreover, since the proposed approach chooses only similar pixels from the search region, structures are sharper than in the conventional NLM method even when it uses the Box kernel.



(a) (b) (c)

Fig. 5: (a) Lena (256x256). (b) Normalized cardinality matrix (R) for noisy image with $\sigma_n = 20$. (c) Clustered elements of R using K-Means, with K=2. Red pixels correspond to smooth regions.

Sun et al. [12] also present a combination of the NLM method with an adaptive search region. We tried to compare our proposed method to their approach by using their parameter values for the search region (M=21) and the similarity patch (p=5), as well as adjusting h to produce the best results for both methods. The comparison is based on level of improvement with respect to the conventional NLM since [12] does not report image sizes and dissimilarity kernels. For example, the image Pepper with $\sigma_n = 20$ is reported to have an improvement of 0.23 dB, whereas our proposed approach provides an improvement of 0.3 and 0.63 dB with respect to NLM using a Box kernel and NLM with a Uniform kernel, respectively. The method presented in [12] uses a contiguous adaptive search region based on Local Polynomial Approximation (LPA) directional kernels that are selected using a predefined set of directions and scales. This selection is subjected to a manual choice of parameters, thus making the algorithm less robust. Our method, on the other hand, does not enforce a contiguous search region, and additionally involves a locally adaptive dissimilarity kernel which enhances the denoising effect.

TABLE III. SIMPLIFIED SCHEME FOR DISSIMILARITY KERNEL-TYPE ADAPTATION

1. For each pixel i in the noisy image, do the following:
   a. Compute the variance of *all* the elements of $\tilde{d}_p^U(i,j),\ \forall j \in S_i$. If the resultant variance is smaller or equal to $2/p^2$ (see eqn. (9)), pixel i is considered smooth. Hence, $S_i^{SU} = S_i$ and $r_i = 1$.
   b. Otherwise, determine S_i^{SB} by computing the elements of the vector $\tilde{d}_p^B(i,j),\ \forall j \in S_i$ using the scheme in Table I, but with the Box kernel. Then, Compute the corresponding r_i.
2. Cluster the data in the normalized cardinality matrix, whose elements are r_i, into two clusters ($C_{\max}^B$, $C_{\min}^B$) using K-Means with K=2.
3. Set pixel weights (before normalization) according to the following rule:

$$\forall j \in S_i^S : w_{i,j} = \begin{cases} f\left(d_p^U(i,j)\right) & r_i \in C_{\max}^B \\ f\left(d_p^B(i,j)\right) & \textit{Otherwise} \end{cases} \quad (16)$$

4. Normalize the computed weights by their sum (see eqn. (3)).

TABLE IV. QUANTITAVE EVALUATION

Image Name/ Noise STD	NLM with Box Kernel		NLM with Uniform Kernel		Proposed Approach	
	PSNR [dB]	***SSIM***	***PSNR [dB]***	***SSIM***	***PSNR [dB]***	***SSIM***
Lena/20	30.27	0.86	30.11	0.87	**30.48**	**0.88**
Lena/30	28.03	0.78	28.03	0.81	**28.32**	**0.82**
Barbara/ 20	29.19	0.87	29.11	0.87	**29.33**	**0.88**
Baboon/ 20	25.54	0.74	24.78	0.69	**25.62**	**0.75**
Pepper/ 20	30.39	0.87	30.28	0.88	**30.55**	**0.89**
Pepper/ 30	28.06	0.81	28.03	0.83	**28.39**	**0.84**

V. CONCLUSIONS

A modified Non-Local Means denoising algorithm that uses an adaptive search region matched to each pixel, and a content-based dissimilarity kernel, is developed in this paper. The adaptive search region creates an anisotropic neighborhood adapted to image local structure by excluding dissimilar pixels from the weighted averaging. The exclusion is based on a statistical model which characterizes the dissimilarity measure of the pixels in the search region. Moreover, the dissimilarity kernel is also adapted to image structure in order to achieve smoothness in flat regions, as well as edge and texture preservation in structural regions. Consequently, the denoising strength of the NLM algorithm is enhanced. Experiments show that the proposed method is better than the conventional NLM denoising scheme, for the two dissimilarity kernels used, both objectively (PSNR-wise and SSIM-wise) and visually and is found to be more efficient than other approaches examined.

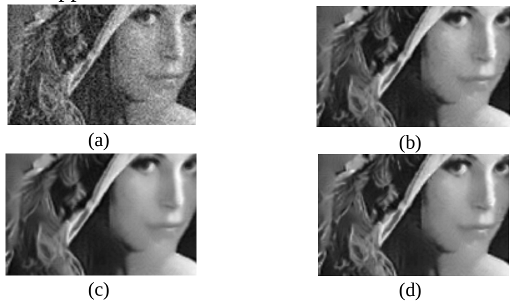


Fig. 6: Denoising variations of the image Lena (256x256) with $\sigma_n = 20$, a zoom-in view of the shoulder and face (flat regions) and of the feathers (texture). (a) Noisy image. (b) Denoised image using conventional NLM with a Box dissimilarity kernel. (c) Denoised image using conventional NLM with a Uniform dissimilarity kernel. (d) Denoised image using proposed approach.

REFERENCES

[1] Noura Azzabou, Nikos Paragios, Frederic Guichard, "Uniform and textured Regions Separation in Natural images towards MPM Adaptive Denoising", SSVM Proceedings of the 1st international conference of Scale space and variational methods in computer vision, pp. 418-429.

[2] Thomas Brox, Oliver Kleinschmidt, and Daniel Cremers, "Efficient Nonlocal Means for Denoising of Textural Patterns", IEEE Transactions on Image Processing, pp. 1083 – 1092, July 2008.

[3] A. Buades, B. Coll, and J. M. Morel, "A Review Of Image Denoising Algorithms, With a New One", Multiscale Model. Simul. Vol. 4, No. 2, pp. 490–530, 2005.

[4] Pierrick Coupé, Pierre Yger, Sylvain Prima, Pierre Hellier, Charles Kervrann, and Christian Barillot, "An Optimized Blockwise Nonlocal Means Denoising Filter for 3-D Magnetic Resonance Images", IEEE Trans Med Imaging, April 2008; 27(4): pp. 425–441.

[5] Dinesh peter. J, Govindan. V. K, Abraham T. Mathew, "Robust Estimation Approach for Nonlocal-means Denoising Based On Structurally Similar Patches", Int. J. Open Problems Compt. Math., Vol. 2, No. 2, June 2009, pp. 311-331.

[6] Siegfried Gabler and Christian Wolff, "A quick and easy approximation to the distribution of a sum of weighted chi-square variables", Statistische Hefte 28, pp. 317-325 (1987).

[7] C. Kervrann, J. Boulanger, and P. Coupé, “Bayesian Non-Local Means Filter, "Image Redundnacy and Adaptive Dictionaries for Noise Removal", in Proc. of SSVM, F. Sgallari, A. Murli, and N. Paragios, Eds., Berlin, 2007, vol. LNCS 4485, pp. 520–532, Springer-Verlag.

[8] M. Mahmoudi and G. Sapiro, “Fast image and video denoising via nonlocal means of similar neighborhoods,” IEEE Signal processing letters, vol.12, no.12, pp. 839-842, 2005.

[9] Jeff Orchard, Mehran Ebrahimi, Alexander Wong, "Efficient Nonlocal-Means Denoising Using The SVD", Image Processing, 2008. ICIP 2008. 15th IEEE International, pp. 1732 – 1735, October 2008.

[10] Joseph Salmon, "On two parameters for denoising with Non-Local Means", Signal Processing Letters IEEE, Vol. 17, Issue 3, pp. 269-272, March 2010.

[11] Z. Wang, A. C Bovik, H. R Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity", IEEE Transactions on Image Processing, Vol. 13, No. 4, pp. 600-612, April 2004.

[12] Weifeng Sun, Min Han, "Adaptive Search Based Non-Local Means Image De-noising", Image and Signal Processing, 2009, CISP '09, pp. 1-4.