On the Fly Image Denoising using Patch Ordering[★]

Ozden Colak[a,b,*], Ender M. Eksioglu[b]

[a]TUBITAK BILGEM Information Technologies Institute, Kocaeli, Turkiye
[b]Istanbul Technical University, Electronics and Communication Engineering, Istanbul, Turkiye

ARTICLE INFO

Keywords:
image denoising,
patch ordering,
online dictionary learning.

ABSTRACT

We introduce an image denoising algorithm which utilizes a novel online dictionary learning procedure together with patch ordering. The developed algorithm employs both the non-local image processing power of patch ordering and the sequential patch-based update of online dictionary learning. The patch ordering process exploits the similarities between patches of a given image which are extracted from different locations. Joint processing of the ordered set of image patches facilitates the non-local image processing ability of the algorithm. The algorithm starts with the extraction of a maximally overlapped set of patches from the given noisy image. Then, the extracted patches are reordered by using a distance measure, and the 3D ordered patch cube is formed. The ordered patch cube is used sequentially to update an overcomplete dictionary. In each iteration, firstly the present patch is denoised using sparse coding over the current overcomplete dictionary. Secondly, the overcomplete dictionary is updated using the current image patch, and the dictionary is passed to the next iteration. We call this process as "on the fly denoising", because each patch is individually denoised using an instantaneously updated overcomplete dictionary. Patch ordering together with online dictionary learning ensures that the dictionary is adapted to different neighborhoods of patches in the patch cube. This adaptation of the dictionary to specialized local patch structures in the patch cube promises improved denoising performance when compared to dictionary learning algorithms devoid of such adaptation. Simulation results indicate that the introduced online method presents improved denoising performance in comparison to both online and batch dictionary learning algorithms from the literature while maintaining similar computational complexity.

1. Introduction

In recent years, the patch based image processing which operate the small parts of the handled image has been a popular research area because of increasing the performance via utilizing the self-similarity property of the natural images. Plenty of patch based image processing approaches have been proposed (Tomasi and Manduchi, 1998; Yaroslavsky, 1985; Buades et al., 2005; Awate and Whitaker, 2006; Ram et al., 2013b; Dabov et al., 2007). Non-local patch based image processing methods have demonstrated significant performance improvements due to joint processing of spatially distant but similar image patches. In this regard there have been several attempts at nonlocal image processing and especially image denoising (Ram et al., 2013b; Dabov et al., 2007; Ram et al., 2013a; Buades et al., 2005; Muresan and Parks, 2003; Zhang et al., 2010). Since the most disturbing effect on images is noise, the image denoising is one of the most studied subfield of image processing. Numerous image denoising methods are proposed in the literature (Aharon et al., 2006; Rubinstein et al., 2013; Eksioglu and Bayir, 2014a; Milanfar, 2013; Colak and Eksioglu, 2019). Patch ordering is one of the remarkable approaches to non-local processing (Ram et al., 2013b). On the other hand, sparse signal representations and compressed sensing have formed another prominent front for image processing research. Sparse representation expresses a given signal as a linear combination from an overcomplete (redundant) set of "basis functions" which are known as atoms. The problem for the construction of the dictionary is known as dictionary learning. Plenty of methods have been suggested to learn overcomplete dictionaries (Rubinstein et al., 2013; Eksioglu and Bayir, 2014b,a; Aharon et al., 2006; Ravishankar and Bresler, 2013; Yaghoobi et al., 2013). The core aim of dictionary learning is to specifically train atoms which work well for sparse representation of a given data set. Predefined, non-adaptive dictionaries and basis sets such as redundant wavelet transforms were the norm for signal representations for a long time (Mallat, 2008). However, recent works have shown that better results can be achieved using learned dictionaries, since they are adapted to the specific data under consideration (Elad and Aharon, 2006). Learned sparse models allow flexibility to adjust the representation for different data sets.

Most of the previously listed methods which consider dictionary learning problem are batch algorithms. These methods use the entire data set as a batch for each iteration. On the other hand there are online dictionary learning methods which have been proposed more recently (Skretting and Engan, 2010; Mairal et al., 2009; Eksioglu, 2014). To process the data set sequentially instead of using the entire data set for each iteration is the main characteristic of online dictionary learning methods. This idea enables us to update the current dictionary estimate by processing only a small subset (or an incoming part) of the data set. Online algorithms are useful when the problem requires handling of very large training data sets such as learning a dictionary adapted

[★]This research is funded by TUBITAK (The Scientific and Technological Research Council of Turkey) under project no. 217E076.

*Corresponding author
ozden.colak@tubitak.gov.tr (Ozden Colak); eksioglue@itu.edu.tr (Ender M. Eksioglu)
ORCID(s): 0000-0002-0762-7266 (Ozden Colak); 0000-0002-7869-4159 (Ender M. Eksioglu)

to small image patches. Online techniques form a very effective alternative to batch methods especially if the data set contains a large number of images which corresponds to several millions of extracted patches. The Recursive Least Squares dictionary learning algorithm (RLS-DLA) Skretting and Engan (2010) is one of the popular online dictionary learning method that is derived from the of RLS algorithm as utilized in adaptive filtering. Similar to the RLS algorithm, the RLS-DLA iteratively updates the dictionary estimate without any explicit matrix inversion by using the innovation extracted from the current data. Hence, RLS-DLA provides reasonable computational complexity. RLS-DLA can be utilized in numerous image processing problems, e.g. image denoising as studied in Eksioglu (2014).

In this work we develop a new online image denoising algorithm which employs patch ordering. We start with the extraction of maximally overlapped patches of the given noisy image. The 3D patch cube is obtained after sequential placement of all extracted patches. By using a distance measure, these patches are reordered in the 3D cube using according to their similarities, in parallel with (Ram et al., 2013b). The patches in the 3D patch cube are processed one by one to update the current dictionary estimate. The current dictionary estimate is also utilized to adaptively denoise the particular patch under consideration. These two concurrent and iterative steps lead to the novel online dictionary learning based denoising algorithm. The patch ordering step employs nonlocal processing to bring similar patches into spatial proximity, leading to dictionaries adapted to particular patch similar neighborhoods. According to the best of our knowledge, this work presents a novel combination of patch ordering and online dictionary learning for sparse representations. The resulting algorithm is an original method for image denoising which benefits from both nonlocal processing and online sparse representations.

We can summarize the core steps of the proposed image denoising algorithm for a given noisy image as follows:

Step 1. Patches of a certain size are extracted from the image, and they are reordered to obtain an ordered 3D patch cube.

Step 2. In an online setting, for every incoming patch a suitable dictionary is formed from the current dictionary estimate by employing an online dictionary learning step. This dictionary update is separated into two substeps as given with Step 2a and Step 2b.

Step 2a. The sparse representation (or atom selection) problem is solved using the current dictionary by applying a sparse solver on the currently handled patch.

Step 2b. The current dictionary (for first patch this corresponds to the initial dictionary) is updated by only using the sparse representation for the current patch in an online manner.

Step 3. The denoising of the current patch is realized by using the sparse coefficients obtained in Step 2a and the current dictionary.

2. Proposed algorithm: On the Fly (OTF) Denoising

We present a novel image denoising algorithm that utilizes the patch ordering approach. In this work, the dictionary is continuously updated for the given sequence of patches, and the patches at hand are individually denoised with the current dictionary.

The algorithm starts with the patch extraction process. The patches of a given noisy image are extracted with maximally overlaps. The following step of the algorithm is the patch ordering process. In this step the extracted patches are reordered as a 3D cube according to a similarity metric which is measured with a method based on Euclidean distance. In the main iteration of the algorithm these ordered patches are handled one by one. Each patch in the 3D cube is used to update the currently formed dictionary. To get the innovation of the current patch a sparse coding method is used. In the dictionary update stage, the innovation that is gathered from the handled patch is inflicted to the current dictionary to obtain the updated dictionary. Before starting to the new iteration the final step is the denoising of the handled patch by utilizing the current dictionary. These steps are reiterated for each image patch in the 3D patch cube. At the end of the last patch's iteration the denoised versions of all the image patches will be obtained. By locating these image patches to the image canvas the entire denoised image is generated.

To mathematically describe the algorithm we can start with giving the corruption of an image with noise via a linear equation as follows.

$$\mathbf{Y} = \mathbf{X} + \mathbf{N} \tag{1}$$

In equation (1), $\mathbf{X} \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$ and $\mathbf{Y} \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$ shows the original image and the noisy image respectively. $\mathbf{N}$ is additive independent and identically distributed white Gaussian noise with zero mean and variance σ^2. The vectorized image vector can be defined as $\boldsymbol{y} = vec(\mathbf{Y}) \in \mathbb{R}^N$. Here $vec(\cdot)$ indicates an operator which transforms a given matrix to a column stacked vector.

We extract overlapped patches of size $\sqrt{n}\times\sqrt{n}$ from the noisy image as the initial step of our proposed algorithm. As it is used generally, we set the extent of the overlap as maximum in order to lead the best denoising results. Let we define an image patch which corresponds to ith pixel of the given image as $\tilde{\mathbf{Z}}_i \in \mathbb{R}^{\sqrt{n}\times\sqrt{n}}$. The vectorized form of the patch $\tilde{\mathbf{Z}}_i$ can be given as $\tilde{\boldsymbol{z}}_i = vec(\tilde{\mathbf{Z}}_i)$ where the size of this vectorized image patch $\tilde{\boldsymbol{z}}_i$ is $\mathbb{R}^n$. By concatenating of all the vectorized patches of the given image, we obtain the total patch vector $\tilde{\boldsymbol{z}}$.

By extending the edges of the given image and by setting the extent of overlap for patches to unity, we can assign a patch vector that is centered on each pixel of the given image. As the result of this process the number of generated patches is equalized to the total pixel number of the image. Therefore if the size of the patches is set to $\sqrt{n}\times\sqrt{n}$, the size of the generated total patch vector $\tilde{\boldsymbol{z}}$ will be $M = nN$.

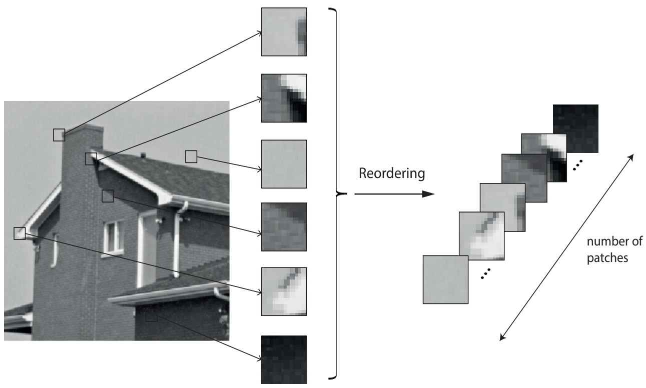


Figure 1: Patch extraction from an image and reordering of the extracted patches.

Once the maximally overlapped patches are extracted, the next step is forming a 3D patch cube from these extracted patches. We realize the reordering process by using a similarity metric. There exist various measures to calculate similarity of vectors in the literature. Among these we employed the squared Euclidean distance method as also preferred in (Ram et al., 2013b). Reordering the patches according to their similarity allows us to employ the non-local processing. Let $\tilde{\mathbf{Z}}_i$ and $\tilde{\mathbf{Z}}_j$ indicate the patch matrices of ith and jth pixels respectively, then two patch vectors $\tilde{\boldsymbol{z}}_i$ and $\tilde{\boldsymbol{z}}_j$ correspond to the vectorized versions of these patch matrices. By using above notations the squared Euclidean distance is given as follows.

$$w(\tilde{\boldsymbol{z}}_i, \tilde{\boldsymbol{z}}_j) = \|\tilde{\boldsymbol{z}}_i - \tilde{\boldsymbol{z}}_j\|^2 \quad (2)$$

By reordering the patches according to the obtained order from (2), we rearranged the patches with respect to their similarity. We can define this new rearrangement as a special permutation matrix $\mathbf{P}_{\boldsymbol{y}} \in \mathbb{R}^{M\times M}$, where $\mathbf{P}_{\boldsymbol{y}}$ has unit values occurring in diagonal blocks as described in (Ram et al., 2013b). These permutation matrix includes the ordering information of the particular $\boldsymbol{y}$ image vector, hence the $(\cdot)_{\boldsymbol{y}}$ subscript is added to denote this case. We can define the reordered patch vector as $\boldsymbol{z} = \mathbf{P}_{\boldsymbol{y}}\tilde{\boldsymbol{z}}$. On the other hand, we can specify the total patch vector as the concatenation of the individual patch vectors $\boldsymbol{z}_i$'s:

$$\boldsymbol{z} = [\boldsymbol{z}_1^T, \boldsymbol{z}_2^T, ..., \boldsymbol{z}_N^T]^T \quad (3)$$

Finally we can define the patch extraction and reordering processes together using a single operator as in (4), which converts an input image to the reordered patch vector.

$$\boldsymbol{z} = \mathcal{T}(\mathbf{Y}) \quad (4)$$

The graphical expression of these patch extraction and reordering can be given with Fig. 1.

As it is declared in the previous section the following step of the proposed algorithm is the denoising of each patch by employing a dictionary. For this purpose we initially use a dictionary for denoising the first patch of the reordered 3D patch cube and for every patch we sequentially update this dictionary. This problem can be solved by using various methods. In this work we employ the online dictionary learning paradigm of RLS-DLA algorithm.

2.1. RLS-DLA Algorithm

Various online dictionary learning based denoising methods have already been proposed in the literature. The recursive least squares dictionary learning algorithm (RLS-DLA) is employed in most of these methods. The RLS-DLA which deals with the following problem (Skretting and Engan, 2010):

$$\min_{\mathbf{D},\boldsymbol{w}}\{\|\mathbf{Y} - \mathbf{D}\mathbf{W}\|_F^2 + \gamma \sum_{i=1}^{N} \|\boldsymbol{w}_i\|_0\} \quad (5)$$

As in many work, RLS-DLA algorithm solves the above minimization problem by splitting it into two parts which starts with finding a suitable $\mathbf{W}$ by realizing the optimization over the fixed $\mathbf{D}$, and follows with updating $\mathbf{D}$ by utilizing fixed $\mathbf{W}$. The first term of the solution of equation (5) is sparse coding term. In this term, $\boldsymbol{w}_i \in \mathbb{R}^K$ which is the sparse coefficient vector of $\tilde{\boldsymbol{z}}_i \in \mathbb{R}^n$ is found using the dictionary $\mathbf{D}_{i-1} \in \mathbb{R}^{n\times K}$ according to a sparseness constraint. This problem can be solved by using various greedy algorithms from the literature such as Basic Matching Pursuit (BMP) (Mallat and Zhifeng Zhang, 1993), Matching Pursuit (MP) (Cotter et al., 1999) and Orthogonal Matching Pursuit (OMP) (Pati et al., 1993). These algorithms give an optimal solution but not the exact solution. In this work we employed the OMP method of Pati et al. (1993) which is one of the most known greedy vector selection methods. The sparse coding problem can be given as in (6).

$$\boldsymbol{w}_i = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\tilde{\boldsymbol{z}}_i - \mathbf{D}_{i-1}\boldsymbol{w}\|_2^2 + \gamma\|\boldsymbol{w}\|_0 \quad (6)$$

The instantaneous sparse coefficient matrix $\mathbf{W}_i$ is formed by concatenating the obtained sparse coefficient vector $\boldsymbol{w}_i$ with the previous sparse coefficient matrix $\mathbf{W}_{i-1}$ as in (7):

$$\mathbf{W}_i = [\mathbf{W}_{i-1}|\boldsymbol{w}_i] \quad (7)$$

The update approach of the RLS-DLA algorithm which calculates $\mathbf{W}_i$ from $\mathbf{W}_{i-1}$, is generated by utilizing the relation between the successive sparse coefficient matrices without any explicit matrix inversion. After the $\mathbf{W}_i$ is obtained, the next step is to update the dictionary which is given as an optimization problem in (8).

$$\mathbf{D}_i = \underset{\mathbf{D}}{\operatorname{argmin}} \|\tilde{\mathbf{Z}}_{(i)} - \mathbf{D}\mathbf{W}_i\|_F^2 \quad (8)$$

The solution to the above problem can be given as follows:

$$\mathbf{D}_i = \tilde{\mathbf{Z}}_{(i)}\mathbf{W}_i^{\dagger} \quad (9)$$

Here, $\tilde{\mathbf{Z}}_{(i)}$ indicates the concatenated data vectors up to ith patch as $\tilde{\mathbf{Z}}_{(i)} = [\tilde{\boldsymbol{z}}_1, \tilde{\boldsymbol{z}}_2, ..., \tilde{\boldsymbol{z}}_i]$. In RLS-DLA algorithm this process is repeated as an epoch iteration (a complete run over the available training set).

2.2. Proposed Algorithm : OTF

In our proposed image denoising algorithm we utilized the RLS-DLA algorithm to update the initialized dictionary which will be used for denoising of each patch of the ordered 3D patch cube. In RLS-DLA algorithm an initial dictionary is updated by processing the patch vectors sequentially in a nonordered alignment. We suggest that to employ the patch ordering manner to the nonordered patches forces the dictionary to be learned the local sections of the patch structure. By considering $\boldsymbol{z}_i \in \mathbb{R}^n$ represents the individual patches for $i = \{1, 2, ..., N\}$ in the reordered total patch vector $\boldsymbol{z} \in \mathbb{R}^M$, the sparse coding and the dictionary update process can be formalized as similar with the RLS-DLA:

$$\boldsymbol{w}_i = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{z}_i - \mathbf{D}_{i-1}\boldsymbol{w}\|_2^2 + \gamma\|\boldsymbol{w}\|_0 \tag{10}$$

The second term of the solution is the dictionary update step. After the sparse coefficient vector $\boldsymbol{w}_i$ is obtained, the current dictionary $\mathbf{D}_{i-1}$ is updated using the multistep approach of the RLS algorithm and as a result the updated dictionary $\mathbf{D}_i$ is obtained. The details of this dictionary update step can be given as follows (Skretting and Engan, 2010):

$$\begin{aligned}
&\text{Representation error computation } : \boldsymbol{r} = \boldsymbol{z}_i - \mathbf{D}_{i-1}\boldsymbol{w}_i \\
&\text{Forgetting factor } : \mathbf{C}_{i-1} = \lambda^{-1}\mathbf{C}_{i-1} \\
&\text{Vector update } : \boldsymbol{u} = \mathbf{C}_{i-1}\boldsymbol{w}_i \\
&\text{Scalar update } : \alpha = \frac{1}{1 + \boldsymbol{w}_i^T\boldsymbol{u}} \\
&\text{Update } \mathbf{C} \text{ matrix } : \mathbf{C}_i = \mathbf{C}_{i-1} - \alpha\boldsymbol{u}\boldsymbol{u}^T \\
&\text{Update the dictionary } : \mathbf{D}_i = \mathbf{D}_{i-1} + \alpha\boldsymbol{r}\boldsymbol{u}^T
\end{aligned} \tag{11}$$

The steps given in (10) and (11) are similar with the sparse coding step and the dictionary update step of the RLS-DLA algorithm which are given with (6) and (8) respectively. The denoised version of a patch is obtained by the sparse representation of it namely utilizing the dictionary and the sparse coefficient vector belongs to this patch.

In the proposed algorithm the reordered patches are processed sequentially hence the learned dictionary is evolved into a specific section of the patch structure where the standard algorithms that process the non-ordered data have to sequentially deal with the patches which are not similar.

In the literature there are various methods to solve this two-step problem. Batch methods such as Olshausen and Field (1997); Aharon et al. (2006); Engan et al. (2007); Kreutz-Delgado et al. (2003), find the sparse coefficient vector of each patch using the same dictionary $\mathbf{D}$. Hence, the total sparse coefficient matrix $\mathbf{W}$ is obtained by concatenating the individual sparse coefficient vectors together as given below.

$$\mathbf{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2 \dots \boldsymbol{w}_N] \tag{12}$$

After that this process is repeated as an epoch iteration (a complete run over the available training set). Therefore the

Algorithm 1 On the Fly Denoising using Patch Ordering - OTF

Input: The noisy image $\mathbf{Y}$, initial dictionary $\mathbf{D}_0$, $\mathbf{C}_0 = \mathbf{I}_{(N\times N)}$ and forgetting factor λ.

Obtain the image and total patch vectors $\boldsymbol{y}$ and $\tilde{\boldsymbol{z}}$.

Reorder the patch vector: $\boldsymbol{z} = \mathbf{P}_{\boldsymbol{y}}\tilde{\boldsymbol{z}}$.
(The total operation: $\boldsymbol{z} = \mathcal{T}(\mathbf{Y})$).

for $i := 1, 2, \dots, N$ **do** ▷ patch iteration

$\boldsymbol{w}_i = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{z}_i - \mathbf{D}_{i-1}\boldsymbol{w}\|_2^2 + \gamma\|\boldsymbol{w}\|_0$ ▷ step 1

$\boldsymbol{r} = \boldsymbol{z}_i - \mathbf{D}_{i-1}\boldsymbol{w}_i$, $\mathbf{C}_{i-1} = \lambda^{-1}\mathbf{C}_{i-1}$ ▷ step 2
$\boldsymbol{u} = \mathbf{C}_{i-1}\boldsymbol{w}_i$, $\alpha = \frac{1}{1+\boldsymbol{w}_i^T\boldsymbol{u}}$
$\mathbf{C}_i = \mathbf{C}_{i-1} - \alpha\boldsymbol{u}\boldsymbol{u}^T$
$\mathbf{D}_i = \mathbf{D}_{i-1} + \alpha\boldsymbol{r}\boldsymbol{u}^T$ ▷ dictionary update
$\hat{\boldsymbol{z}}_i = \mathbf{D}_{i-1}\boldsymbol{w}_i$ ▷ denoising the handled patch

end for

Obtain the denoised patch vector: $\hat{\boldsymbol{z}} = [\hat{\boldsymbol{z}}_1^T, \hat{\boldsymbol{z}}_2^T, ..., \hat{\boldsymbol{z}}_N^T]^T$.

Obtain the denoised image: $\hat{\mathbf{Y}} = \mathcal{T}'(\hat{\boldsymbol{z}})$.

Output: The denoised image $\hat{\mathbf{Y}}$.

common outline of batch methods can be summarized as: after all sparse coefficient vectors are found using $\mathbf{D}_{i-1}$, the new dictionary $\mathbf{D}_i$ is obtained by utilizing $\mathbf{W}_{i-1}$. On the contrary of batch methods, online methods such as Skretting and Engan (2010); Mairal et al. (2009); Eksioglu (2014), firstly obtain the sparse coefficient vector $\boldsymbol{w}_i$ of the handled patch using an entire dictionary matrix $\mathbf{D}_{i-1}$ and recalculate the previous sparse coefficient vector $\mathbf{W}_{i-1}$ by concatenating the vector $\boldsymbol{w}_i$ and obtain $\mathbf{W}_i$. Then the dictionary matrix is updated by using the recalculated sparse coefficient vector $\mathbf{W}_i$. This process is repeated for every patch hence the final dictionary is obtained after the last patch is processed. In some works such as RLS-DLA Skretting and Engan (2010) this process is repeated as an epoch iteration. In online methods the better convergence for the dictionary update is obtained via evaluating the sparse coefficient vectors individually rather than a batch approach. In the online method of Eksioglu (2014) a similar approach is utilized. In this work we propose a novel online image denoising algorithm that employs a different dictionary matrix and a different sparse coefficient vector for denoising of each patch. We utilize the obtained individual sparse coefficient vector $\boldsymbol{w}_i$ to update the dictionary $\mathbf{D}_{i-1}$. And concurrently we denoise the related handled patch by employing this individual sparse coefficient vector $\boldsymbol{w}_i$ and the dictionary $\mathbf{D}_{i-1}$. As seen from Algorithm 1 the sparse coefficient vectors of each patch in the generated 3D patch cube are obtained from different dictionary matrices. These dictionary matrices are sequentially updated versions of an initial dictionary $\mathbf{D}_0$. As a result, we

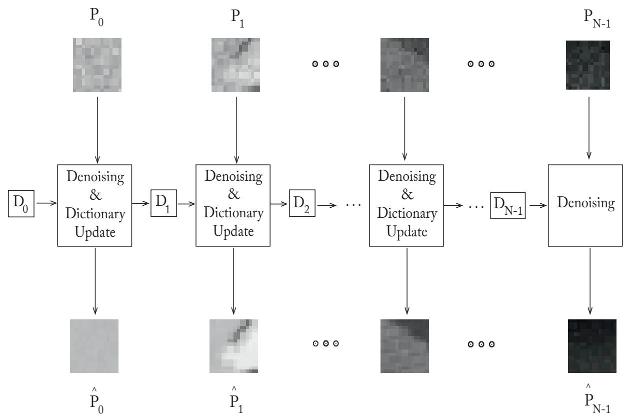


Figure 2: The concurrent dictionary update and patch denoising process.

are continuously updating the dictionary for the given sequence of patches, and we individually denoise the patches with the current dictionary. The denoising of the current patch could be done by utilizing the newly updated dictionary with the corresponding sparse coefficient vector which can be obtained by employing an OMP algorithm once more. However we have not seen any additional advantage of this operation to the denoising performance at our experiments.

In the Algorithm 1 the main structure of our proposed approach is summarized with pseudo-codes. The main novelty of this work is to use the patch ordering idea to strenghten the learning performance of the dictionary via processing the similar patches in a sequential order. On the other hand it can be seen that the proposed algorithm has a remarkable difference according to the exist image denoising methods in the context of the structure of the denoising scheme. We can summarize this structure as follows: the denoising of each individual patch in the 3D patch cube is accomplished with an adaptive dictionary that is updated in accordance with the streaming data. In Fig. 2 this original structure of the algorithm is portrayed.

All of the mentioned processes that starts with the given image and ends up with the denoised image are given sequentially with block diagrams in Fig. 3.

3. Simulations

We give the natural image denoising performance of the proposed "On the Fly (OTF) denoising" algorithm in this section. We also study the effect of reordering the extracted patches on the image denoising performance for this framework, where the patches for the given image are reordered according to a certain similarity metric. To emphasize the importance of reordering, we give the image denoising performance of the same setup without patch ordering. The setup without patch ordering inherently corresponds to the image denoising application of the RLS-DLA algorithm (Skretting and Engan, 2010). Hence, we evaluate the effect of employing patch ordering by comparing the results of the newly introduced OTF denoising with RLS-DLA denoising.

We performed the implementations in Matlab on a system which has a 2.4 GHz Intel Core i7 CPU, 8 GB RAM and 64-bit Windows 7 operating system.

We obtained the denoising results by realizing the algorithms on the test images of the Set12 dataset (Zuo et al., 2018) for various noise standard deviation levels. The used natural images which are the members of the Set12 dataset can be seen in Fig. 4.

To measure the image denoising performance of the different methods we utilize the peak-signal-to-noise ratio (PSNR [dB]) and the structural similarity index measure (SSIM). For a reference image $\mathbf{X} \in \mathbb{R}^{M \times N}$ and a test image $\mathbf{Y} \in \mathbb{R}^{M \times N}$, the PSNR metric is given as follows (Hore and Ziou, 2010).

$$\mathrm{PSNR}(\mathbf{X}, \mathbf{Y}) = 10 \log_{10} \left(\frac{255^2}{\mathrm{MSE}(\mathbf{X}, \mathbf{Y})} \right) \tag{13}$$

Here, $\mathrm{MSE}(\mathbf{X}, \mathbf{Y}) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (\mathbf{X}_{ij} - \mathbf{Y}_{ij})^2$, and 255 denotes the maximum intensity for the images.

Similarly the definition of the SSIM metric is given as follows.

$$\mathrm{SSIM}(\mathbf{X}, \mathbf{Y}) = l(\mathbf{X}, \mathbf{Y}) c(\mathbf{X}, \mathbf{Y}) s(\mathbf{X}, \mathbf{Y}) \tag{14}$$

Here, $l(\mathbf{X}, \mathbf{Y})$, $c(\mathbf{X}, \mathbf{Y})$ and $s(\mathbf{X}, \mathbf{Y})$ functions measure the luminance, contrast and structure closeness of the two images respectively. One can find the detailed expressions for these two image quality metrics in (Hore and Ziou, 2010).

In Fig. 5 we visualize the evolution of various atoms of the dictionary. As it is denoted in the previous section, the dictionary is updated by using the sparse coefficient vector of the processed patches. Therefore the dictionary gets closer to a structure that can construct the given image more accurately by combining its atoms. We see the advantage of the initial reordering of the patches by analyzing the atoms of the dictionary which are locally adapted while the denoising process is going on.

The image denoising results for the proposed OTF algorithm and RLS-DLA algorithm are given with Table 1 and Table 2. As we can see from these denoising results employing patch ordering idea to the extracted patches before the denoising process increases the denoising performance. All of the individual denoising results and the average results show that OTF algorithm performs better than RLS-DLA algorithm.

In addition Fig. 6, Fig. 7 and Fig. 8 depict the noisy and original "butterfly", "house" and "peppers" images together with the denoised versions for the methods of OTF and RLS-DLA at $\sigma = 10$ noise level. Although there exist obvious differences between the results of the two methods with respect to the PSNR metric, the resulting images are visually quite similar. Additionally we give the original, noisy and denoised versions of these images for $\sigma = 25$ noise level with the zoomed local sections in Fig. 9, Fig. 10 and Fig. 11.

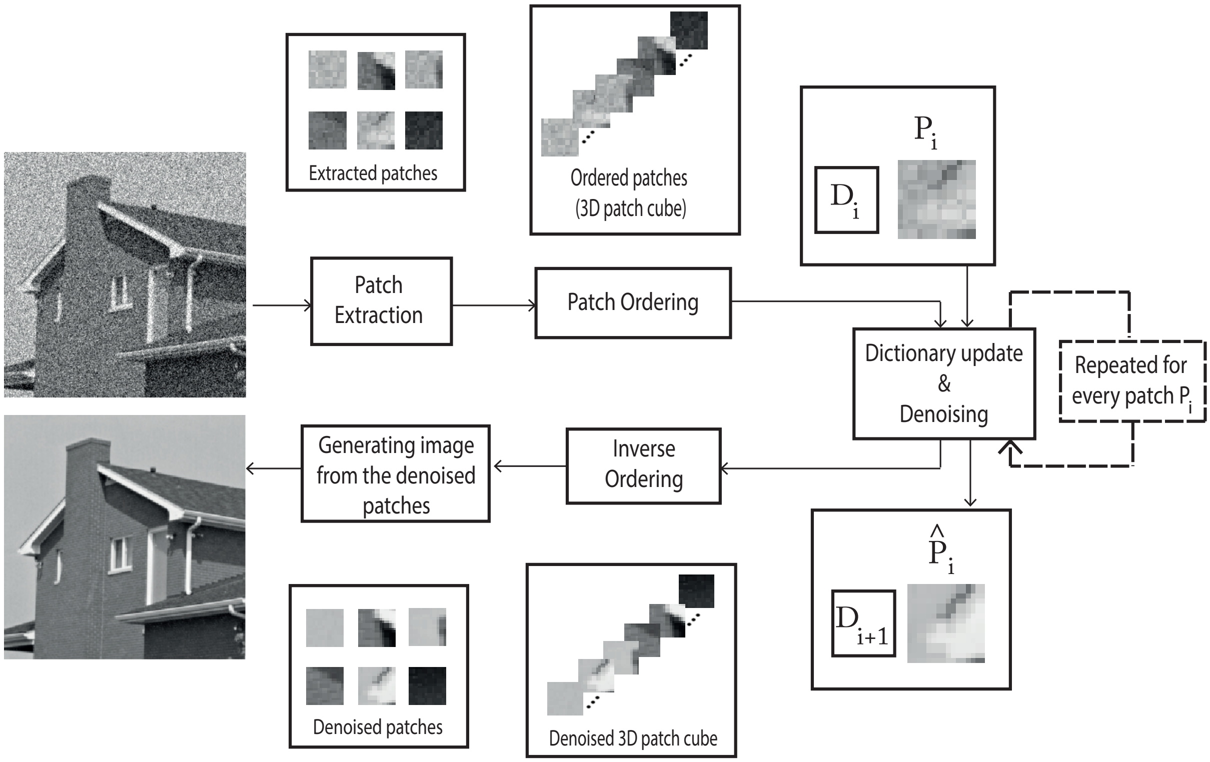


Figure 3: Block diagram of the proposed method.

Table 1
PSNR results of different images and noise levels for the proposed method OTF and RLS-DLA Skretting and Engan (2010).

σ [dB] \ image	method	1	2	3	4	5	6	7	8	9	10	11	12	Avgs.
10	RLS-DLA	33.21	35.67	33.74	32.72	33.18	32.69	33.02	35.51	34.27	33.60	33.52	33.53	33.72
	OTF	33.45	36.06	34.09	32.95	33.52	32.83	33.06	35.60	34.36	33.69	33.60	33.59	**33.90**
15	RLS-DLA	31.09	34.04	31.80	30.52	31.03	30.50	30.79	33.73	32.31	31.72	31.49	31.48	31.71
	OTF	31.34	34.28	32.10	30.73	31.40	30.64	30.94	33.75	32.41	31.75	31.51	31.52	**31.86**
20	RLS-DLA	29.72	32.85	30.41	29.02	29.54	29.04	29.33	32.39	30.76	30.33	30.10	30.04	30.29
	OTF	29.88	33.12	30.62	29.16	29.87	29.20	29.50	32.37	30.83	30.34	30.10	30.06	**30.42**
25	RLS-DLA	28.64	31.77	29.34	27.81	28.38	27.94	28.29	31.31	29.52	29.26	29.05	28.88	29.18
	OTF	28.79	32.00	29.46	27.96	28.65	28.07	28.40	31.28	29.51	29.28	29.05	28.91	**29.28**
35	RLS-DLA	27.04	30.01	27.71	26.05	26.63	26.30	26.79	29.65	27.58	27.65	27.53	27.09	27.50
	OTF	27.18	30.10	27.84	26.11	26.85	26.40	26.81	29.60	27.53	27.67	27.55	27.10	**27.56**
50	RLS-DLA	25.35	27.88	25.83	24.11	24.78	24.41	25.13	27.75	25.33	25.93	25.95	25.24	25.64
	OTF	25.49	27.95	25.90	24.16	24.88	24.53	25.14	27.71	25.28	25.94	25.96	25.26	**25.68**
75	RLS-DLA	23.15	25.17	23.37	21.97	22.31	22.02	23.16	25.73	22.90	23.96	24.29	23.51	23.46
	OTF	23.13	25.27	23.48	21.99	22.42	22.05	23.18	25.72	22.82	24.00	24.29	23.55	**23.49**
Avgs.	RLS-DLA	28.31	31.06	28.89	27.46	27.98	27.56	28.07	30.86	28.95	28.92	28.85	28.54	28.78
	OTF	**28.47**	**31.25**	**29.07**	**27.58**	**28.23**	**27.67**	**28.15**	**30.86**	**28.96**	**28.95**	**28.87**	**28.57**	**28.88**

4. Conclusions

In this work, a new image denoising algorithm which continuously updates the dictionary for the given sequence of patches and individually denoises the patches with the instantaneously updated dictionary is presented. The 'patch ordering' approach of non-local image processing literature is combined with an online dictionary learning algorithm, and as a result a new image denoising algorithm is developed. The natural image denoising performances of the RLS-DLA and the proposed method are compared. The results show that the proposed algorithm increases the denoising performance by processing the image patches nonlocally by using a similarity-based ordering rather than the natural spatial local ordering.

References

Aharon, M., Elad, M., Bruckstein, A., 2006. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing 54, 4311–4322. doi:10.1109/TSP.2006.881199.

Table 2
SSIM results of different images and noise levels for the proposed method OTF and RLS-DLA Skretting and Engan (2010).

image / σ [dB]	method	1	2	3	4	5	6	7	8	9	10	11	12	Avgs.
10	RLS-DLA	0.5484	0.5013	0.7512	0.8275	0.8168	0.6460	0.7125	0.6197	0.7600	0.6768	0.7079	0.7404	0.6924
	OTF	0.5535	0.5119	0.7545	0.8301	0.8227	0.6485	0.7147	0.6199	0.7611	0.6798	0.7093	0.7414	**0.6956**
15	RLS-DLA	0.4864	0.4240	0.6993	0.7625	0.7751	0.5906	0.6466	0.5571	0.7091	0.5928	0.6181	0.6506	0.6260
	OTF	0.4891	0.4289	0.7020	0.7655	0.7825	0.5916	0.6497	0.5555	0.7109	0.5929	0.6177	0.6518	**0.6282**
20	RLS-DLA	0.4371	0.3881	0.6589	0.7099	0.7392	0.5499	0.5950	0.5142	0.6631	0.5264	0.5467	0.5835	0.5760
	OTF	0.4404	0.3892	0.6607	0.7126	0.7460	0.5528	0.5977	0.5122	0.6643	0.5262	0.5464	0.5843	**0.5777**
25	RLS-DLA	0.3961	0.3585	0.3262	0.6651	0.7072	0.5167	0.5540	0.4804	0.6196	0.4757	0.4915	0.5282	0.5349
	OTF	0.3962	0.3632	0.6273	0.6674	0.7128	0.5189	0.5551	0.4783	0.6178	0.4753	0.4911	0.5293	**0.5360**
35	RLS-DLA	0.3379	0.3152	0.5758	0.5935	0.6491	0.4606	0.4937	0.4287	0.5422	0.4034	0.4131	0.4371	0.4709
	OTF	0.3408	0.3175	0.5771	0.5944	0.6550	0.4621	0.4929	0.4269	0.5409	0.4036	0.4145	0.4387	**0.4721**
50	RLS-DLA	0.2894	0.2645	0.5122	0.5091	0.5812	0.3930	0.4254	0.3675	0.4443	0.3313	0.3387	0.3392	0.3985
	OTF	0.2862	0.2647	0.5105	0.5075	0.5773	0.3898	0.4270	0.3678	0.4458	0.3306	0.3373	0.3374	**0.3997**
75	RLS-DLA	0.2184	0.1976	0.4182	0.4007	0.4742	0.2833	0.3515	0.3001	0.3261	0.2516	0.2650	0.2490	0.3113
	OTF	0.2186	0.1998	0.4247	0.4020	0.4803	0.2845	0.3517	0.3002	0.3229	0.2541	0.2666	0.2529	**0.3132**
Avgs.	RLS-DLA	0.3872	0.3499	0.6057	0.6381	0.6770	0.4910	0.5400	**0.4669**	**0.5809**	0.4653	0.4828	0.5038	0.5157
	OTF	**0.3897**	**0.3536**	**0.6084**	**0.6402**	**0.6829**	**0.4931**	**0.5410**	0.4658	0.5803	**0.4662**	**0.4835**	**0.5054**	**0.5175**

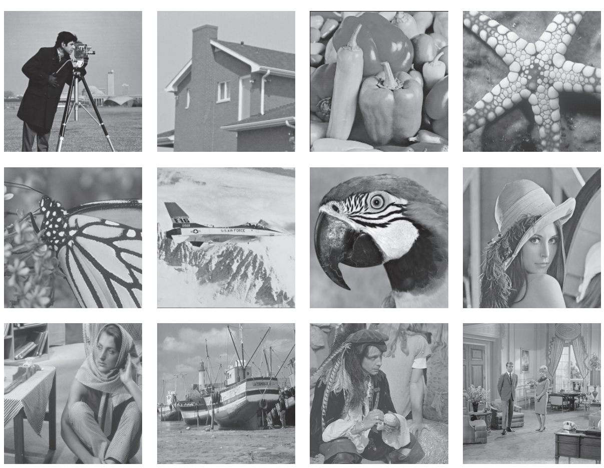


Figure 4: Images in the Set12 dataset.

Awate, S., Whitaker, R., 2006. Unsupervised, information-theoretic, adaptive image filtering for image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 364–376. doi:10.1109/TPAMI.2006.64.

Buades, A., Coll, B., Morel, J.M., 2005. A review of image denoising algorithms, with a new one. Multiscale Modeling & Simulation 4, 490–530. URL: http://dx.doi.org/10.1137/040616024, doi:10.1137/040616024, arXiv:http://dx.doi.org/10.1137/040616024.

Colak, O., Eksioglu, E.M., 2019. Image denoising using patch ordering and 3d transformation of patches. IET Image Processing 13, 2636–2646. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-ipr.2018.6431, doi:https://doi.org/10.1049/iet-ipr.2018.6431, arXiv:https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-ipr.2018.6431.

Cotter, S.F., Adler, R., Rao, R.D., Kreutz-Delgado, K., 1999. Forward sequential algorithms for best basis selection. IEE Proceedings - Vision, Image and Signal Processing 146, 235–244. doi:10.1049/ip-vis:19990445.

Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K., 2007. Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on Image Processing 16, 2080–2095. doi:10.1109/TIP.2007.901238.

Eksioglu, E.M., 2014. Online dictionary learning algorithm with periodic updates and its application to image denoising. Expert Systems with Applications 41, 3682 – 3690. URL: http://www.sciencedirect.com/science/article/pii/S0957417413009561, doi:https://doi.org/10.1016/j.eswa.2013.11.036.

Eksioglu, E.M., Bayir, O., 2014a. K-SVD meets transform learning: Transform k-svd. IEEE Signal Processing Letters 21, 347–351. doi:10.1109/LSP.2014.2303076.

Eksioglu, E.M., Bayir, O., 2014b. Overcomplete sparsifying transform learning algorithm using a constrained least squares approach, in: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7158–7162. doi:10.1109/ICASSP.2014.6854989.

Elad, M., Aharon, M., 2006. Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image Processing 15, 3736–3745. doi:10.1109/TIP.2006.881969.

Engan, K., Skretting, K., Husøy, J.H., 2007. Family of iterative ls-based dictionary learning algorithms, ils-dla, for sparse signal representation. Digital Signal Processing 17, 32 – 49. URL: http://www.sciencedirect.com/science/article/pii/S105120040600025X, doi:https://doi.org/10.1016/j.dsp.2006.02.002.

Hore, A., Ziou, D., 2010. Image quality metrics: PSNR vs. SSIM, in: 20th International Conference on Pattern Recognition, pp. 2366–2369. doi:10.1109/ICPR.2010.579.

Kreutz-Delgado, K., Murray, J.F., Rao, B.D., Engan, K., Lee, T., Sejnowski, T.J., 2003. Dictionary learning algorithms for sparse representation. Neural Computation 15, 349–396. doi:10.1162/089976603762552951.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., 2009. Online dictionary learning for sparse coding, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, New York, NY, USA. pp. 689–696. URL: http://doi.acm.org/10.1145/1553374.1553463, doi:10.1145/1553374.1553463.

Mallat, S., 2008. A Wavelet Tour of Signal Processing. Elsevier.

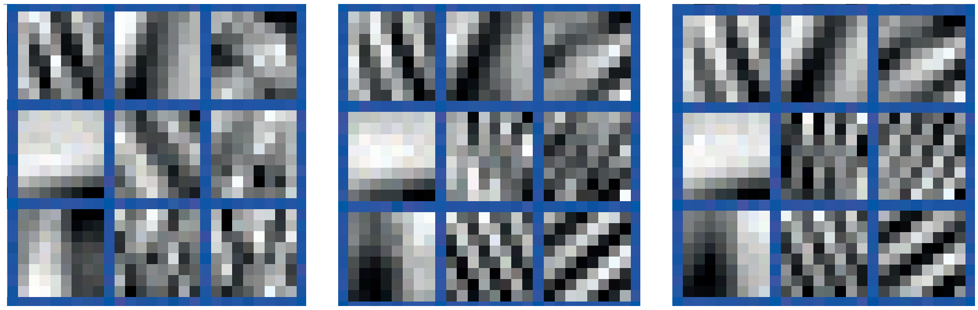
Mallat, S.G., Zhifeng Zhang, 1993. Matching pursuits with time-frequency dictionaries. IEEE Transactions on Signal Processing 41, 3397–3415. doi:10.1109/78.258082.

Milanfar, P., 2013. A tour of modern image filtering: New insights and methods, both practical and theoretical. IEEE Signal Processing Magazine 30, 106–128. doi:10.1109/MSP.2011.2179329.

Muresan, D.D., Parks, T.W., 2003. Adaptive principal components and image denoising, in: Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429), pp. I–101–4 vol.1. doi:10.1109/ICIP.2003.1246908.

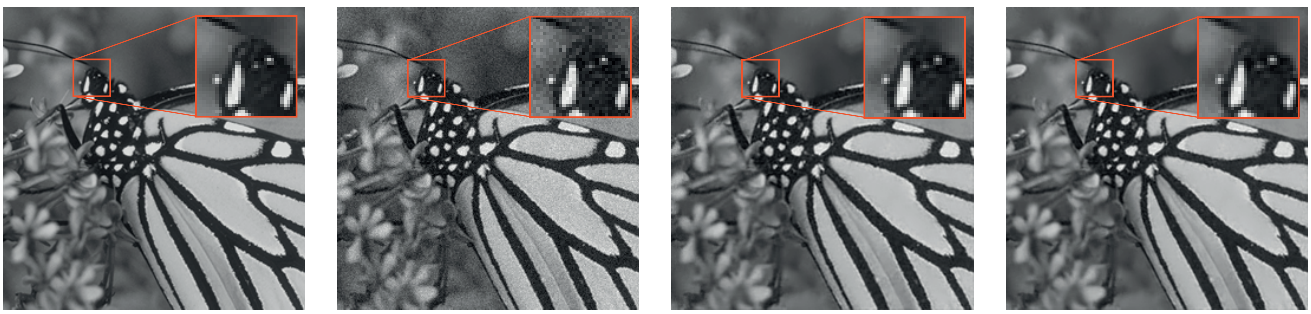
Olshausen, B.A., Field, D.J., 1997. Sparse coding with an overcomplete basis set: A strategy employed by v1? Vision Research 37, 3311 – 3325. URL: http://www.sciencedirect.com/science/article/pii/S0042698997001697, doi:https://doi.org/10.1016/S0042-6989(97)00169-7.

Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S., 1993. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition, in: Proceedings of 27th Asilomar Conference on Signals, Systems and Computers, pp. 40–44 vol.1. doi:10.1109/ACSSC.1993.



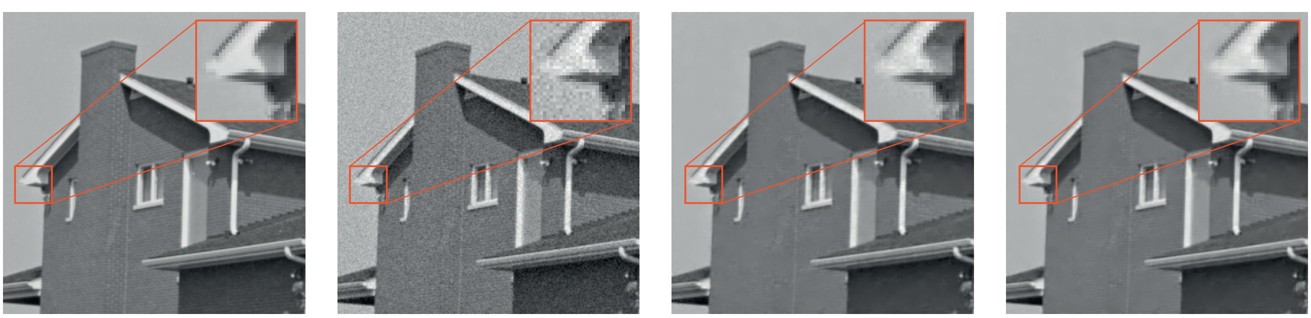
a) b) c)

Figure 5: Changes of some specific atoms at different stages. a) After 12×10^4 patches. b) After 20×10^4 patches. c) After 26×10^4 patches.



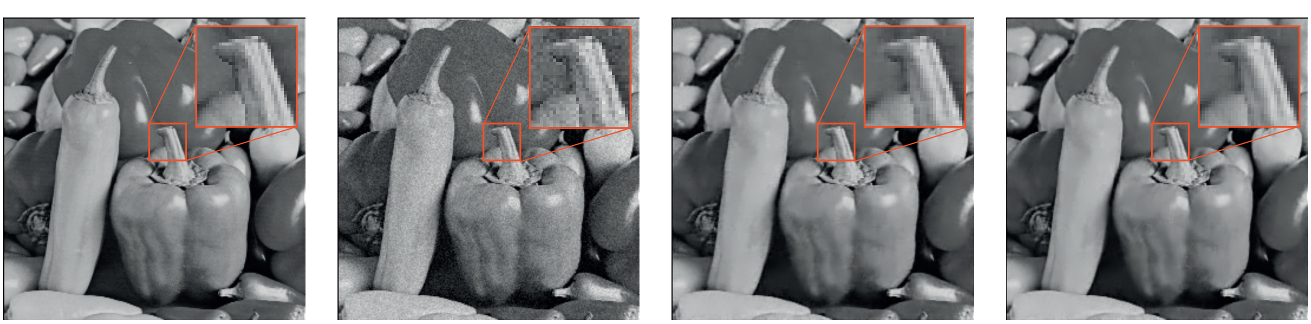
a) b) c) d)

Figure 6: Image denoising results for 'Butterfly',at $\sigma = 10$ noise level. a) Original image. b) Noisy image. c) Denoised image using RLS-DLA (PSNR : 33.18 dB). d) Denoised image using OTF (PSNR : 33.52 dB).



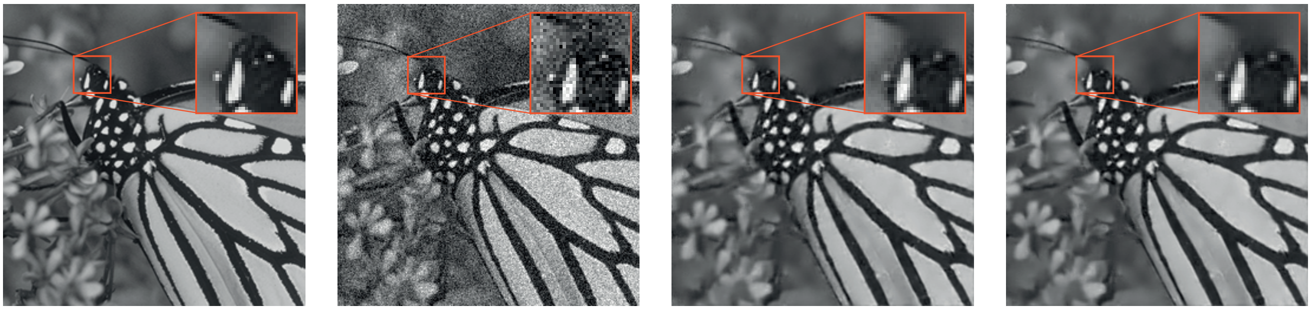
a) b) c) d)

Figure 7: Image denoising results for 'House',at $\sigma = 10$ noise level. a) Original image. b) Noisy image. c) Denoised image using RLS-DLA (PSNR : 35.67 dB). d) Denoised image using OTF (PSNR : 36.06 dB).



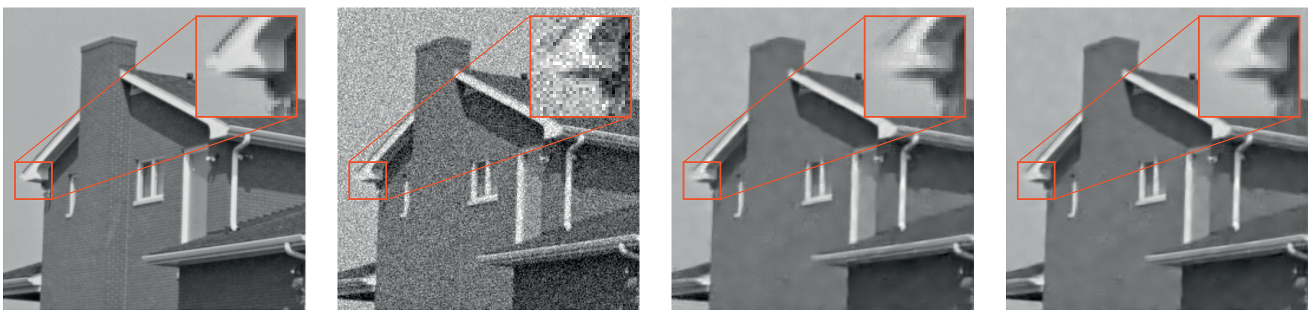
a) b) c) d)

Figure 8: Image denoising results for 'Peppers',at $\sigma = 10$ noise level. a) Original image. b) Noisy image. c) Denoised image using RLS-DLA (PSNR : 33.74 dB). d) Denoised image using OTF (PSNR : 34.09 dB).



a) b) c) d)

Figure 9: Image denoising results for 'Butterfly',at $\sigma = 25$ noise level. a) Original image. b) Noisy image. c) Denoised image using RLS-DLA (PSNR : 28.38 dB). d) Denoised image using OTF (PSNR : 28.65 dB).



a) b) c) d)

Figure 10: Image denoising results for 'House',at $\sigma = 25$ noise level. a) Original image. b) Noisy image. c) Denoised image using RLS-DLA (PSNR : 31.77 dB). d) Denoised image using OTF (PSNR : 32.00 dB).

342465.

Ram, I., Elad, M., Cohen, I., 2013a. Image denoising using nl-means via smooth patch ordering, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1350–1354. doi:10.1109/ICASSP.2013.6637871.

Ram, I., Elad, M., Cohen, I., 2013b. Image processing using smooth ordering of its patches. IEEE Transactions on Image Processing 22, 2764–2774. doi:10.1109/TIP.2013.2257813.

Ravishankar, S., Bresler, Y., 2013. Learning sparsifying transforms. IEEE Transactions on Signal Processing 61, 1072–1086. doi:10.1109/TSP.2012.2226449.

Rubinstein, R., Peleg, T., Elad, M., 2013. Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model. IEEE Transactions on Signal Processing 61, 661–677. doi:10.1109/TSP.2012.2226445.

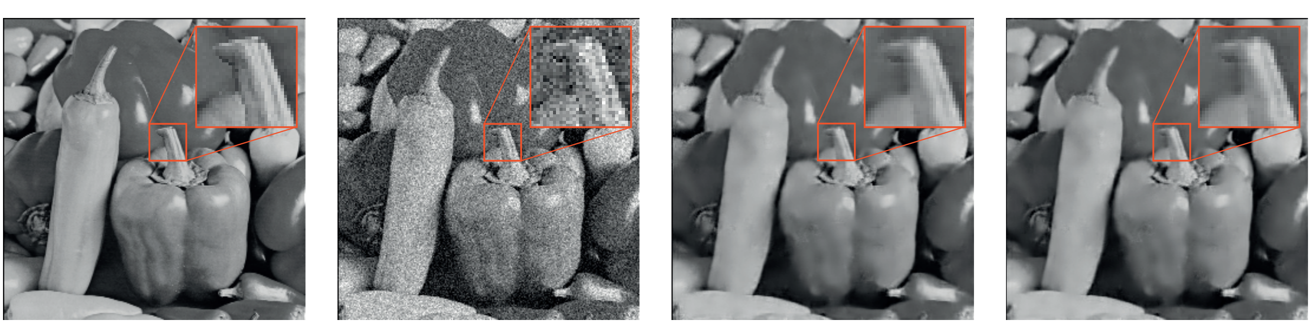
Skretting, K., Engan, K., 2010. Recursive least squares dictionary learning algorithm. IEEE Transactions on Signal Processing 58, 2121–2130. doi:10.1109/TSP.2010.2040671.

Tomasi, C., Manduchi, R., 1998. Bilateral filtering for gray and color images, in: Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), pp. 839–846. doi:10.1109/ICCV.1998.710815.

Yaghoobi, M., Nam, S., Gribonval, R., Davies, M.E., 2013. Constrained overcomplete analysis operator learning for cosparse signal modelling. IEEE Transactions on Signal Processing 61, 2341–2355. doi:10.1109/TSP.2013.2250968.

Yaroslavsky, L., 1985. Digital Picture Processing : An Introduction. Springer-Verlag.

Zhang, L., Dong, W., Zhang, D., Shi, G., 2010. Two-stage image denoising by principal component analysis with local pixel grouping. Pattern Recogn. 43, 1531–1549. URL: http://dx.doi.org/10.1016/j.patcog.2009.09.023, doi:10.1016/j.patcog.2009.09.023.



a) b) c) d)

Figure 11: Image denoising results for 'Peppers',at $\sigma = 25$ noise level. a) Original image. b) Noisy image. c) Denoised image using RLS-DLA (PSNR : 29.34 dB). d) Denoised image using OTF (PSNR : 29.46 dB).

Zuo, W., Zhang, K., Zhang, L., 2018. Convolutional Neural Networks for Image Denoising and Restoration. Springer International Publishing, Cham. pp. 93–123. URL: https://doi.org/10.1007/978-3-319-96029-64, doi:10.1007/978-3-319-96029-6_4.