Isfahan University of Technology

Image Inpainting Using Sparse coding

Supervisor:

Dr. Mohammadali Khosravifard

Author:

Zahra sadat sajadifar

Sep.2019

Contents

Introduction

In signals and images processing to describe our data, we need a model that is important when implementing algorithms. The sparse land model is a powerful model for describing signals based on their sparseness and redundancy. According to this model, the signals and images can be depicted on non-orthogonal basis of a defined vector space, so that the obtained coefficients are sparse, i.e., the number of nonzero coefficients is few. The sparse representation is the same in terms of obtaining the corresponding coefficients of the signal based on specific bases with the conventional transforms such as Fourier transform, but unlike other transforms, the bases of sparse coding are not necessarily perpendicular to each other, and also the number of bases is not equal to the number of signal samples, in other words the defined vector space, known as the dictionary, has redundancy leading to sparse representation. There are various solving methods and algorithms such as matching pursuit and basis pursuit to obtain sparse representation of data that we will discuss in the following chapters. Also choosing appropriate dictionary is very important because sparse representation results from dictionary's redundancy. Different types of dictionaries and algorithms for learning it such as K-SVD are also discussed in the following chapters.

sparse coding is used in image processing such as denoising, compression, image resizing and inpainting. Image inpainting refers to filling-in missing pixels in known locations in the image. The relationship between this issue and the sparse representation is that due to the redundancy in image information, the distorted parts can be reconstructed by obtaining sparse representation of existing pixels in the image. In these problems we can estimate the missing values with respect to other image information and reconstruct the image. In this project, we concentrate on image inpainting and its implementation using sparse coding.

Chapter One: The sparse coding

In this chapter, we describe the sparse representation and dictionaries, the terms most commonly used in this case, how to solve this problem, and how to obtain the sparse representation of signals and images.

1.1 Sparse representation

sparse representation means that a signal can be represented as a linear combination of some bases vectors in a way that the coefficient vector become sparse. According to this model, each signal $x \in \mathbb{R}^n$ can be written as a linear combination of the specified matrix columns $D \in \mathbb{R}^{n \times k}$ as a dictionary. The dictionary matrix columns are called atoms, and in this matrix, $k \gg n$ which means that the matrix has redundancy. Mathematically we have:

$$arg \min_{\alpha} ||\alpha||_0 \quad subject\ to \quad x = D\alpha \tag{1.1}$$

Where the vector $\alpha \in \mathbb{R}^k$ is the vector of the coefficients and the notation $||\cdot||_0$ is the l_0-quasi-norm, which counts the number of non-zero element in the coefficients vector. In other words, to obtain the sparse representation of a signal, we need to solve an optimization problem in order to minimize the l_0-norm. But this is a NP-hard problem because if we consider the maximum number of nonzero coefficients L, we must obtain the choice $\binom{k}{L}$, and since $k \gg n$, this choice cannot be found in a finite time.

1.1.1 l_1-norm

To solve this problem, we can use a convex function in Equation 1.1. In general, the convex function is a function that, if we consider two arbitrary points on the function, the intersection of these two points is always equal to or above the function. Therefore, we use the following definitions to measure sparsity. The closest convex form to l_0, as illustrated in Figure 1.1, is l_1. For this reason, the l_1 is used empirically to solve the problem.

$$||\alpha||_p^p = \sum_{j=1}^{k} |\alpha_j|^p \tag{1.2}$$

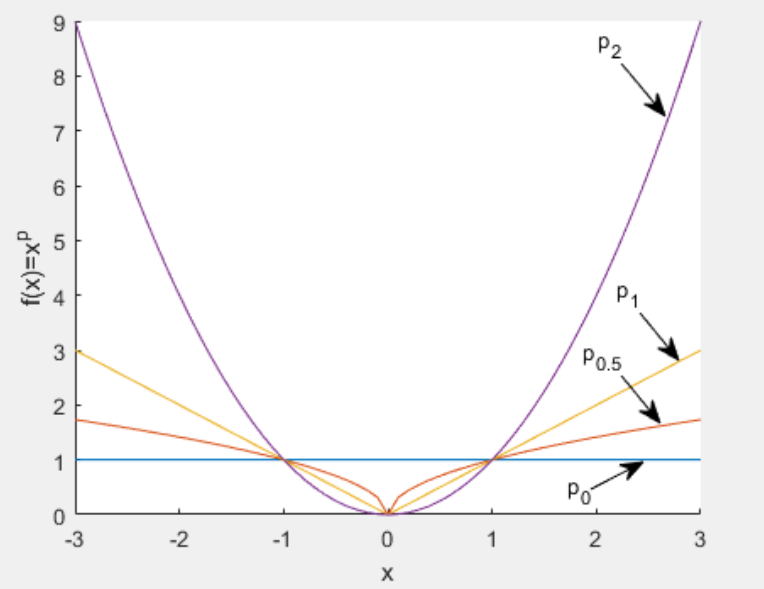


Fig. 1.1 The behavior of $||x||^p$ for various values of p.

1.1.2 Approximation

Instead of obtaining the signal $x \in \mathbb{R}^n$ exactly as a linear combination of dictionary matrix columns, we can take an error value as the threshold level and obtain the approximate sparse representation of the signal.

$$\arg\min_{\alpha} ||\alpha||_0 \quad subject\ to \quad ||x - D\alpha||_2^2 \leq \varepsilon^2 \tag{1.3}$$

1.1.3 Uniqueness

If the solution of Equation 1.3 is $\hat{\alpha}$, what is the guarantee that this solution is the same as the sparse representation of the x in $x = D\alpha$. In other words, is α = α̂?

To answer this question, we start with following definition. Assuming the dictionary $D \in \mathbb{R}^{n \times k}$, the spark of the given matrix D, δ = spark {D} is the smallest number of columns from D that are linearly-dependent. This is in contrast to the rank of matrix definition, which considers the largest number of dictionary columns to be linearly-independent. Now by definition of spark, the vectors in the null space of the dictionary matrix $Dx = 0$ must satisfy $||x||_0 \geq \delta$. In other words, if we consider the dictionary atoms as $d_j \,.\, j = 1. \cdots . k$ and the coefficient of each atom in the vector x, $x_j \,.\, j = 1. \cdots . k$, we want to solve the following equation:

$$d_1x_1 + d_2x_2 + \cdots + d_kx_k = 0 \quad (1.4)$$

We know that if d_j vectors are linearly-independent, all x_j coefficients are zero, and if they are linearly-dependent, the coefficients are nonzero. Here the number of linearly-dependent atoms is the spark of dictionary, so the number of non-zero elements of the coefficients vector must be greater than the spark. So determining the cause of $\left||x|\right|_0 \geq \delta$.

Now we assume that α_1 and α_2 are two representations of the vector x:

$$x = D\alpha_1 = D\alpha_2 \quad \rightarrow \quad D(\alpha_1 - \alpha_2) = 0 \quad (1.5)$$

As stated above, the $\left||\alpha_1 - \alpha_2|\right|_0 \geq \delta$ must be satisfied, and if α_1 satisfy $\left||\alpha_1|\right|_0 \leq \frac{\delta}{2}$, then for α_2 the equation $\left||\alpha_2|\right|_0 \geq \frac{\delta}{2}$ is set. Here's the uniqueness definition of sparse representation:

If we have a representation that satisfy $\left||\alpha|\right|_0 \leq \frac{\delta}{2}$, then necessarily this is the sparsest.

1.2 problem solving methods

There are several ways to solve the optimization problem and find sparse representation and in this section we consider two methods:

1.2.1 Relaxation methods (basis pursuit)

This method uses l_1-norm instead of l_0-norm as mentioned earlier and convex optimization and linear programming methods can be used to solve it.

$$arg \min_{\alpha} ||\alpha||_1 \quad subject\ to \quad ||x - D\alpha||_2^2 \leq \varepsilon^2 \quad (1.6)$$

1.2.2 Greedy methods (matching pursuit)

In this method, like other greedy algorithm, we consider the best possible choice at every step. In the MP method, we first find the best atom corresponding to the signal $x \in \mathbb{R}^n$. In other words, we choose the atom that minimize $||x - D\alpha||_2^2$. Next, subtract the atom's share from the signal and find the most convenient atom for the residual of the signal. The algorithm ends when either the number of atoms selected reaches a certain limit

(MP) or the error rate calculated at each step is less than stop criteria(MP error). If, at every step after selecting the atom that most closely matches the signal, we exclude it from the dictionary and do not use it in the next steps, we act in the orthogonal MP method. This algorithm is summarized as follows:

Task: approximate the solution of P_0 : $\arg\min_{\alpha} ||\alpha||_0 \quad subject\ to \quad x = D\alpha$

parameters: we are given the matrix $D \in \mathbb{R}^{n\times k}$, the vector x, and the error threshold ε_0

Initialization: initialize t=0, and set

- The initial solution α^0
- The initial residual $r^0 = x - D\alpha^0 = x$
- The initial solution support $S^0 = support\{\alpha^0\} = \emptyset$

Main iteration: increment *t* by 1

- Compute the errors $\epsilon(j) = \min_{z_j} \left\|d_j z_j - r^{t-1}\right\|_2^2$ for all $j = 1.2.\cdots.k$ using the optimal choice $z_j^* = d_j^T r^{t-1} / \left|\left|a_j\right|\right|_2^2$.
- Update support: find the minimizer, j_0 of $\epsilon(j)$: $\forall\, j \notin S^{t-1}\,.\epsilon(j_0) \leq \epsilon(j)$. And update $S^t = S^{t-1} \cup \{j_0\}$
- Update solution: compute α^t , the minimizer of $\|D\alpha - x\|_2^2$ subject to $support\{\alpha\} = S^t$
- Update residual: compute $r^t = x - D\alpha^t$
- Stopping rule: if $\left|\left|r^t\right|\right|_2 < \varepsilon_0$, stop. Otherwise, apply another iteration.

Output: the proposed solution is α^t obtained after k iteration.

Fig. 1.2 Orthogonal-Matching-Pursuit – a greedy algorithm for approximating the solution of (*P*0).

1.3 Dictionary and its training

Choosing a dictionary is very important in finding sparse representation. pre-constructed dictionaries can be used for this purpose, such as wavelets, DCTs, etc. some of these proposed dictionaries are often referred to also as transforms. Also we can use adapted dictionaries which are learned from the signal or image or from a data base.

1.3.1 DCT dictionary

Because of the use of the two-dimensional DCT Dictionary for images in later chapters, we explain how to construct it. If the investigated signals are in the space $\mathbb{R}^{64}$ and the number of dictionary atoms is 256, we first form a one-dimensional DCT matrix D_{1D} of size 8×16, where the k-th atom for k = 1,2, ..., 16 is given by $d_k^{1D} = \cos\left(\frac{(i-1)(k-1)\pi}{16}\right) \; . \quad i = 1.2.\cdots.8$. All atoms apart from the first are further processed by removing their mean. The final two-dimensional DCT dictionary of size 64×256 is obtained by a Kronecker-product of the D_{1D} matrices, i.e., $D_{2D} = D_{1D} \otimes D_{1D}$.

1.3.2 Dictionary learning and K-SVD algorithm

The main purpose of this algorithm is to find the dictionary matrix that fits the signal or image. Let's start with the symbols: $X \in \mathbb{R}^{n \times p}$ is the signal matrix which consists of p signals $x_m \in \mathbb{R}^n \; . m = 1.2.\cdots.p.$ $D \in \mathbb{R}^{n \times k}$ is a dictionary matrix whose number of atoms is less than the number of signals, i.e., $K < p$ and matrix $A \in \mathbb{R}^{k \times p}$ is a sparse coefficient matrix whose columns are $\alpha_m \in \mathbb{R}^n \; . m = 1.2.\cdots.p.$

In the first step, we need to initialize the matrix D. This initialization can be a DCT matrix or a random selection of k signals. Note that atoms must be normalized. Then with constant D we must obtain the matrix of the coefficients A by the methods of solving, for example, MP with the maximum number of L atoms (L is chosen to maintain the sparsity of the coefficients matrix):

$$arg \; \min_A \; \sum_{m=1}^{p} ||x_m - D\alpha_m||_2^2 \quad subject\ to \quad \forall\, m\,.\,\big|\big|\alpha_m\big|\big|_0 \leq L \tag{1.7}$$

Next we need to update the dictionary. At this point, matrices A and D are constant except one column of matrix D and the corresponding row in the matrix of coefficients that can be changed. To separate the j_0-th atom, we can write the Freubenius equation as follows:

$$||X - DA||_F^2 = \left\| X - \sum_{j=1}^{k} d_j a_j^T \right\|_F^2 = \left\| \left(X - \sum_{\substack{j=1 \\ j \neq j_0}}^{k} d_j a_j^T \right) - d_{j_0} a_{j_0}^T \right\|_F^2 \tag{1.8}$$

First, consider the atom d_{j_0} and its corresponding coefficient, j_0-th row of the matrix A represented by the symbol $a_{j_0}^T$. To obtain parts of the original signal affected by this atom, we have the following equation:

$$E_{j_0} = X - \sum_{\substack{j=1 \\ j \neq j_0}}^{k} d_j a_j^T \qquad 1.9$$

It should be noted that the vector $a_{j_0}^T$ with size of $p \times 1$ has zero value in some entries. To separate its nonzero components, we first define the following set of numbers, which denotes the location of the nonzero matrices of the matrix A in row j_0, or $a_{j_0}^T$:

$$\Omega_{j_0} = \{ i \mid 1 \leq i \leq p \,.\, A[j_0 \,.\, i] \neq 0 \} \qquad 1.10$$

Matrix P_{j_0} with p rows and $|\Omega_{j_0}|$ columns is defined as having one value in the elements $(i.\Omega_{j_0}(i))$ and zero in the other elements. The numbers i are the locations of the nonzero entries of $a_{j_0}^T$ so they include numbers between 1 and p, and the numbers $\Omega_{j_0}(i)$ are equal to the indices i, which means the numbers 1 to $|\Omega_{j_0}|$. Now we define $(a_{j_0}^R)^T = a_{j_0}^T P_{j_0}$ and $E_{j_0}^R = E_{j_0} P_{j_0}$. $(a_{j_0}^R)^T$ is the j_0-th row of the matrix A corresponding to j_0-th atom with separation of its nonzero components. $E_{j_0}^R$ with size of $n \times |\Omega_{j_0}|$, contains a portion of the signal that uses only the d_{j_0} atom. Now, by using the SVD algorithm or single value decomposition, we can find the optimal d_{j_0} and $a_{j_0}^R$ by minimizing the following:

$$\min_{d_{j_0} .a_{j_0}^T} \left\| E_{j_0} P_{j_0} - d_{j_0} a_{j_0}^T P_{j_0} \right\| = \min_{d_{j_0} .a_{j_0}^R} \left\| E_{j_0}^R - d_{j_0} (a_{j_0}^R)^T \right\| \qquad 1.11$$

According to this algorithm we decompose $E_{j_0}^R$ to $E_{j_0}^R = U \Delta V^T$, where d_{j_0} is the first column of U and $a_{j_0}^R$ equals the product of the first element of Δ and the first column of V, i.e., $a_{j_0}^R = \Delta[1.1] v_1 \,.\, d_{j_0} = u_1$. All the above steps are performed in a iterative loop and the modified dictionary and coefficients in each step are $D_{(t)}$ and $A_{(t)}$. The algorithm goes so far that $\left| \left| X - D_{(t)} A_{(t)} \right| \right|_F^2$ is small enough and the learned dictionary is $D_{(t)}$ in the final step. This algorithm is summarized as follows:

Task: train a dictionary D to sparsely represent the data $\{x_m\}_{m=1}^{p}$

Initialization: initialize t=0 and

- Initialize dictionary : build $D_{(0)} \in \mathbb{R}^{n\times k}$
- Normalization : normalize the columns of $D_{(0)}$

Main iteration: increment t by 1

- Sparse coding: sparse representation $\hat{a}_m \; for \; m = 1.2.\cdots.p$, these form $A_{(t)}$

$$\hat{a}_m = \arg\min_{a} \; ||x_m - D_{(t-1)}a||_2^2 \; subject \; to \; ||a||_0 \leq L$$

- K-SVD dictionary update: update the columns of dictionary and obtain $D_{(t)}$:

  Repeat for $j_0 = 1.\cdots.k$

  - Define the group of examples that use the atom d_{j_0}

$$\Omega_{j_0} = \{i \mid 1 \leq i \leq p \,.\, A[j_0\,.i] \neq 0\}$$

  - Compute the residual matrix

$$E_{j_0} = X - \sum_{\substack{j=1 \\ j\neq j_0}}^{k} d_j a_j^T$$

  - Restrict E_{j_0} by choosing only the columns corresponding to $\boldsymbol{\Omega_{j_0}}$ and obtain $E_{j_0}^R$.
  - Apply SVD decomposition $E_{j_0}^R = U\Delta V^T$
  - Stopping rule: if the change in $||X - D_{(t)}A_{(t)}||_F^2$ is small enough, stop. otherwise, apply another iteration.

- The desired result is $D_{(t)}$

Fig. 1.3 K-SVD algorithm for learning dictionary

1.4 image sparse representation

As we consider the applications of sparse representation in image processing in the following chapter, we will describe the sparse representation of images in this section. There are two approaches to image in sparse land; global modeling and local modeling. global modeling means that we consider the whole image as a signal, that is, if the size of image is $m \times n$, the size of signal is $(m \times n) \times 1$ and thus dictionary is $D \in \mathbb{R}^{(m\times n)\times k}$. This will require more computation and time to run the model. Instead of working on the whole image at once, we can consider small image patches x of size $\sqrt{n} \times \sqrt{n}$. This is called local modeling. Each patch represents a signal in the space $\mathbb{R}^n$ and, for example, if the total number of patches is p, image can be illustrated as a matrix with size of $n \times p$. This implies that there exists a dictionary $D \in \mathbb{R}^{n\times k}$. Every patch x has a corresponding representation $\alpha \in \mathbb{R}^k$ satisfying $\left||D\alpha - x|\right|_2 \leq \epsilon \,.\, \left||\alpha|\right|_0 = L$. This model can be used for image denoising. One natural option is to denoise each of those patches separately and tile the results. However, in doing so, visible artifacts may occur on block boundaries. One could also propose to work on overlapping patches and average the results in order to prevent such blockiness artifacts.

Chapter two: Sparse coding in image processing

2.1 image denoising

Consider a signal $y_0 \in \mathbb{R}^n$ as one of the $\sqrt{n} \times \sqrt{n}$ ideal image patches which is measured in presence of an additive zero-mean white and homogeneous Gaussian noise v with a known standard-deviation σ. The measured image, y, is thus:

$$y = y_0 + v \qquad 2.1$$

The goal here is to find an algorithm using sparse representation to eliminate noise and make the final image as close as possible to the original image y_0. First we need to get the sparse representation of the noisy signal. Then apply a hard threshold to the obtained coefficients vector to eliminate the coefficients whose value is low and due to noise. we were always led to some variant of the optimization problem:

$$(\mathrm{P}_0^{\epsilon}) \quad \arg\min_{\alpha} ||\alpha||_0 \quad subject\ to \quad ||D\alpha - y||_2^2 \leq \epsilon \qquad 2.2$$

The threshold ϵ is closely tied with the noise power, and a natural choice of it would be $C^2\sigma^2 n$, with 0.5< C <1.5. The solution to this problem, denoted by $\hat{\alpha}$, is the sparse representation that represents the desired clean image. Thus, the denoised output is $\hat{y} = D\hat{\alpha}$. Denoising the complete image can be done by the following stages:

1. Consider every pixel in y as a center of a patch of size $\sqrt{n} \times \sqrt{n}$ n (typical value of n is 64). We denote these patches as pi j.
2. Apply denoising of each such patch, pi j, using the thresholding algorithm:

    a) Compute $\tilde{q}_{ij} = D^T P_{ij}$ (the forward 2D-DCT transform).
    b) Pass the obtained vector $\tilde{q}_{ij}$ through a hard-thresholding operation with a pre-specified threshold (dependent on the noise power, and possibly different for each entry) and obtain $\hat{q}_{ij}$.
    c) Compute $\hat{P}_{ij} = D\hat{q}_{ij}$. This is the denoised version of the patch.

3. Merge the denoised patches by averaging them one on top of the other.

2.2 Inpainting sparse-land signals

Sometimes we lose some parts of the image in specific locations and need to fill those areas properly and reconstruct the image. Image inpainting refers to filling-in missing pixels in known locations in the image. In this section we consider this problem, and more specifically, we focus on ways to inpaint images using the *Sparse-Land* signal model. We start our discussion with the core ideas for inpainting sparsely represented signals, and then extend this discussion to images.

Consider the signal $y_0 \in \mathbb{R}^n$. This signal has a sparse representation as $y_0 = D\alpha_0$ which is $||\alpha_0||_0 = L$. The measured signal is $y = My_0$ and the matrix M is a degradation operator that removes p samples from the original signal. The matrix M is a rectangular matrix with size of $(n-p) \times n$ and is obtained by removing p rows corresponded to the dropped samples in a $n \times n$ homogeneous matrix. To find the original signal and fill the holes, just solve the following problem:

$$arg \min_{\alpha} ||\alpha||_0 \quad subject\ to \quad y = MD\alpha \tag{2.3}$$

We denote the solution of this problem by $\hat{\alpha}_0$ and the reconstructed signal is $\hat{y}_0 = D\hat{\alpha}_0$. From Equation 2.3, it can be understood that for inpainting, it is sufficient to consider the parts of the signal that are available and obtain their sparse representations with the new dictionary which is removed the rows of dictionary corresponded to the missing parts of the signal. Then the product of coefficients and the whole dictionary is obtained the reconstructed signal.

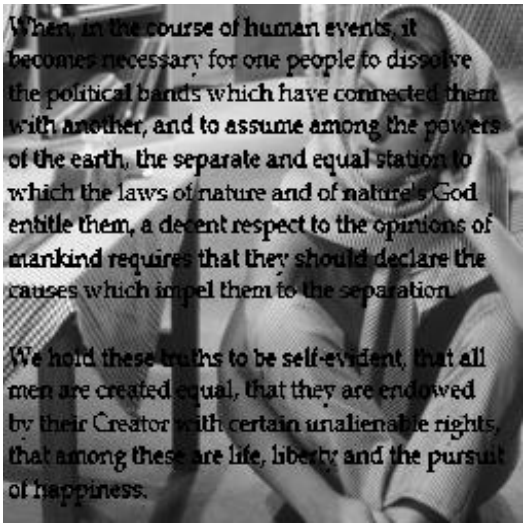
2.2.1 image inpainting

The generalization of this problem for images is just like before, so we must first consider the image as $\sqrt{n} \times \sqrt{n}$ patches and apply the steps mentioned for signal inpainting to the patches. We can express this algorithm for images as a formula and a problem. Suppose the image y is a measured image and contains only pixels whose values are known and the image z is the desired image at the output of the algorithm. The dictionary $D \in \mathbb{R}^{n \times k}$ and matrix M as masks are known. For image mask, we consider a binary matrix with dimensions equal to the dimensions of the image, with all pixels having a value of one except the pixels whose values are missed. M_k is the *k*-th patch in matrix M, which corresponds to *k*-th patch in image z. In the reconstructed image z every patch $p_k = R_k z$ of size $\sqrt{n} \times \sqrt{n}$ in every location k should have a sparse representation q_k with bounded error.

$$arg \min_{q_k} ||q_k||_0 \quad subject\ to \quad \left|\left|M_k(Dq_k - p_k)\right|\right|_2^2 < \varepsilon \tag{2.4}$$

2.2.2 M(mask)

In this project we assume that the locations of the missing holes and pixels in the image are known and the purpose is only to reconstruct the image. Some masks that are used are text, crack, 9 blocks of different dimensions, missing different percentages of image in random places and even painting on the image. As mentioned, these masks are binary matrices that are equal to zero in the missing locations of the image, and by multiplying them by the matric of original image, the image we want to run the algorithm on achieved. Here are some examples of these masks on Barbara images:



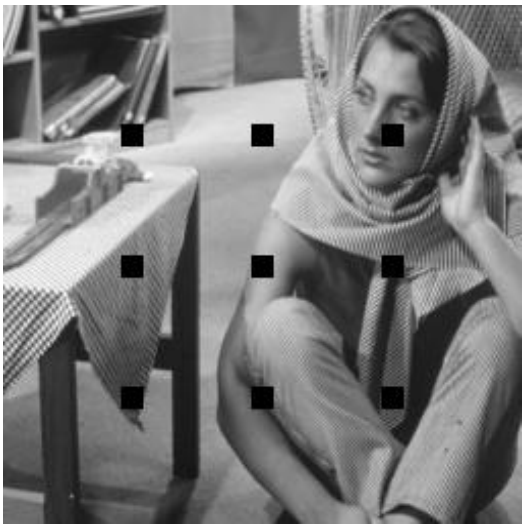
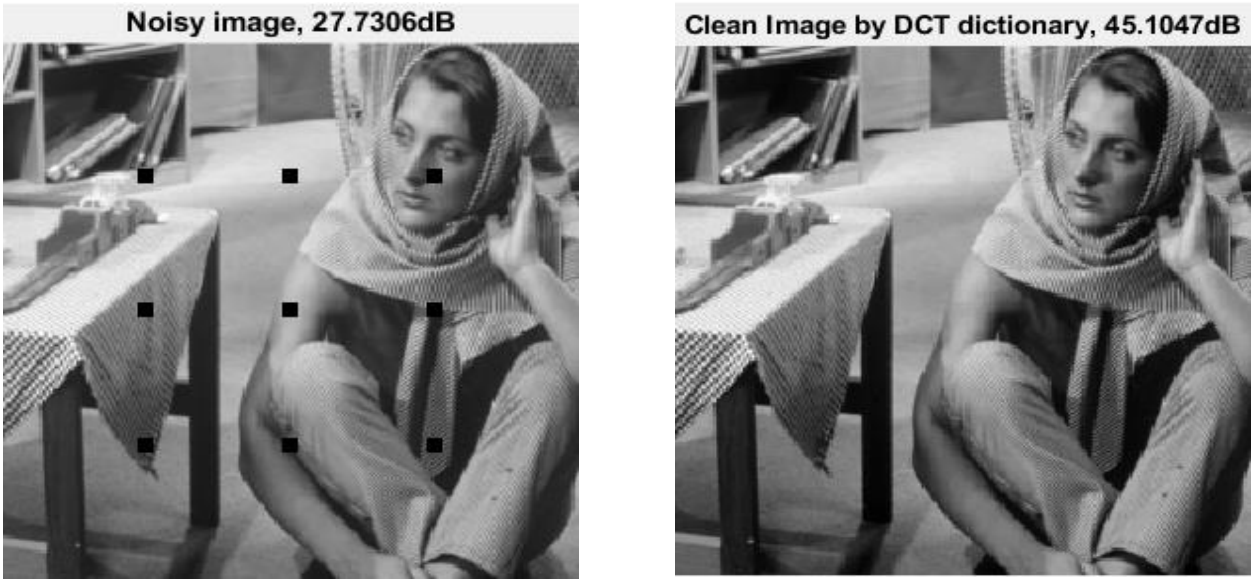


Fig. 2.1 Masks used in this project

The important thing about holes and masks is that if the holes are too large, for example, larger than 8×8 patches, they can no longer be rebuilt locally by separating 8×8 patches because the whole patch is corrupted and lost. Global modeling can be used to solve this problem, i.e., consider the whole image as a patch, and the other solution is to choose the larger size for the image patches according to the dimensions of the holes. We examine the effect of size of patches in Figure 2.2. These inpainted images were performed using a sparse representation method by the DCT Dictionary.



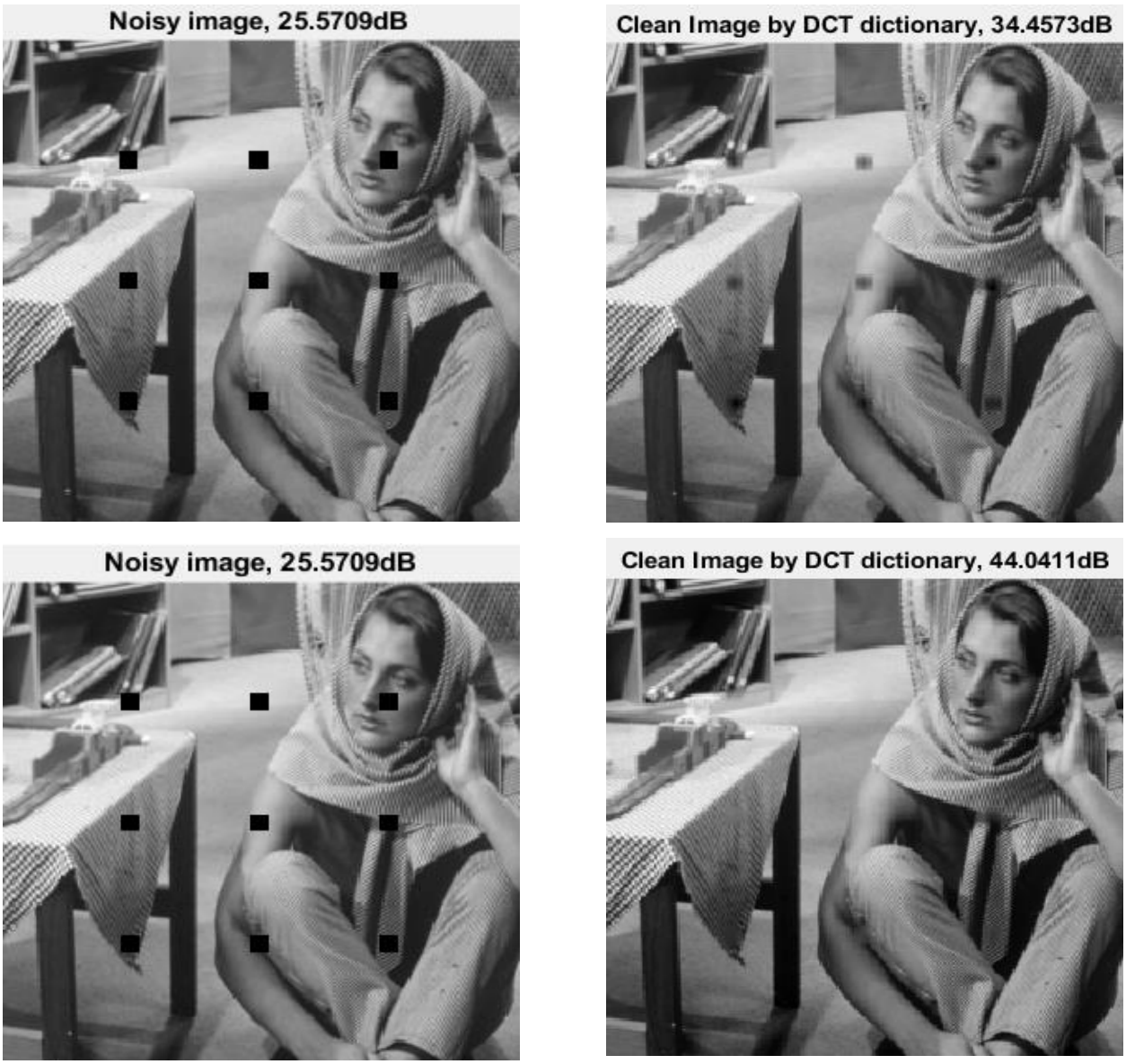


Fig. 2.2 Barbara images with two patterns of missing pixels – 9 blocks of size 7×7 with patches of size 8×8 (top) – 9 blocks of size 9×9 with patches of size 8×8 (middle) – 9 blocks of size 9×9 with patches of size 12×12 (bottom)

2.2.3 Dictionary for inpainting

The dictionary in inpainting is also obtained with the methods already mentioned, which means that we can use pre-constructed dictionaries such as DCT, or the dictionary learned from the image. In inpainting adaptive dictionary, learning is done using the weighted K-SVD algorithm. In this algorithm, as the title implies, not all pixels in the image are used and only parts of the image that are not missed, should be used to learn the dictionary. In other words, the weight of the pixels is the mask matrix which determines existing and missing pixels. The weighted K-SVD is thus the same algorithm as described in Chapter one and Figure 1.3 with two differences:

- When finding a sparse representation in the iterations of the algorithm, only the existing pixels of the image should be used. Equation 2.4 should be applied.

- When calculating the residuals E_{j0}^{R}, we also need to eliminate the effect of missing pixel at each iteration. To do this, simply consider the mask $M_{j_0}^{R}$ with the same definition. This matrix has the dimensions $n \times |\Omega_{j_0}|$ and contains the part of the mask corresponded to the parts of the signal that only use the d_{j_0} atom. By element-wise multiplication between this matrix and residual in the following equation, the missing parts of the matrix are zeroed and only the existing pixels contribute in singular value decomposition.

$$\min_{d_{j_0}.a_{j_0}^{R}} M_{j_0}^{R} \otimes \left\| E_{j_0}^{R} - d_{j_0}(a_{j_0}^{R})^{T} \right\| \qquad 2.5$$

Chapter three: Results

In this chapter, we present the results of the implementation of the methods described in inpainting. At first, we need to determine the criteria for evaluating the accuracy of the results. For this criterion we used the PSNR image, which is defined as follows:

$$PSNR = 10\,log_{10}\frac{MAX^2}{MSE} \tag{3.1}$$

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left(I(i.j) - I'(i.j)\right)^2 \tag{3.2}$$

The *MAX* value corresponds to the maximum value of pixels in the image, which is 255 in the simulation and the *MSE* is the mean square error. Also I is the original image and I' for the PSNR corresponding to the noise image equal to the noise image and for PSNR for the reconstructed image equal to the output image.

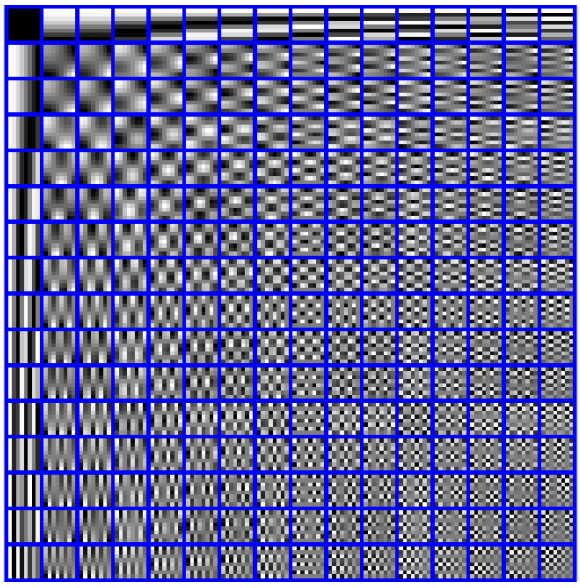


Fig. 3.1 DCT dictionary with 256 atoms and patches of size 8×8

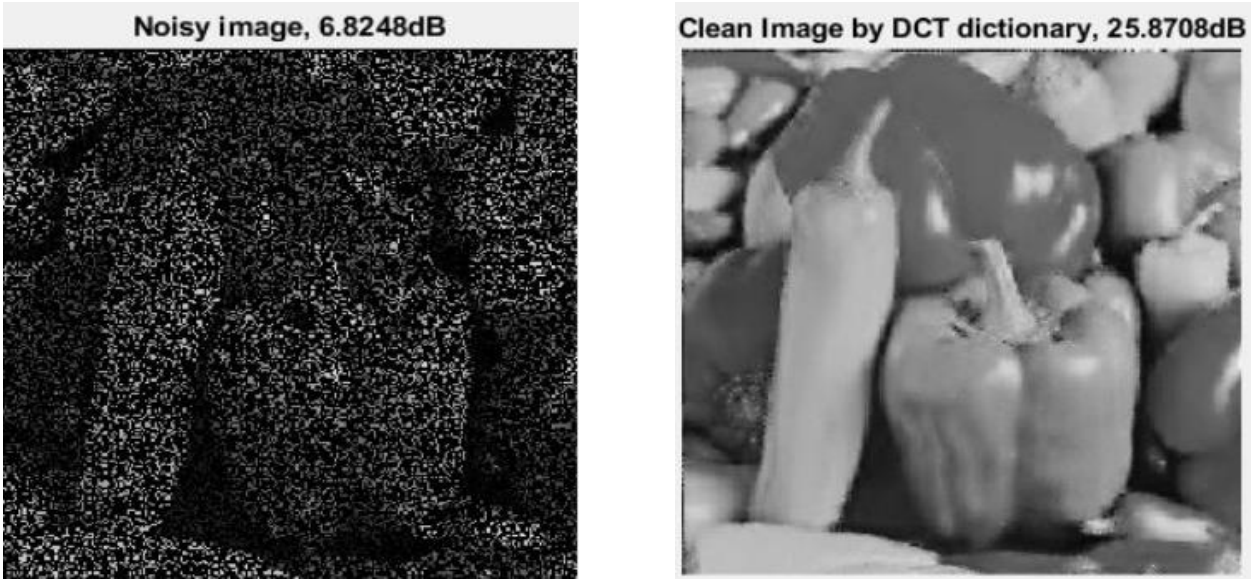


Fig. 3.2 Peppers image with 75% missing pixels

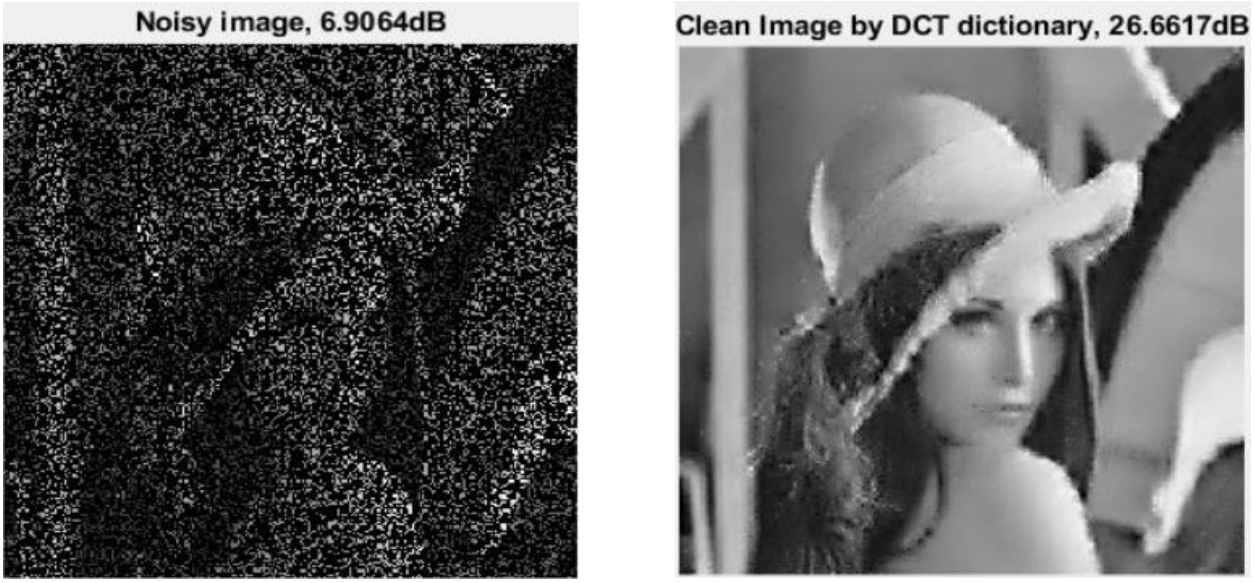


Fig. 3.3 Lena image with 75% missing pixels

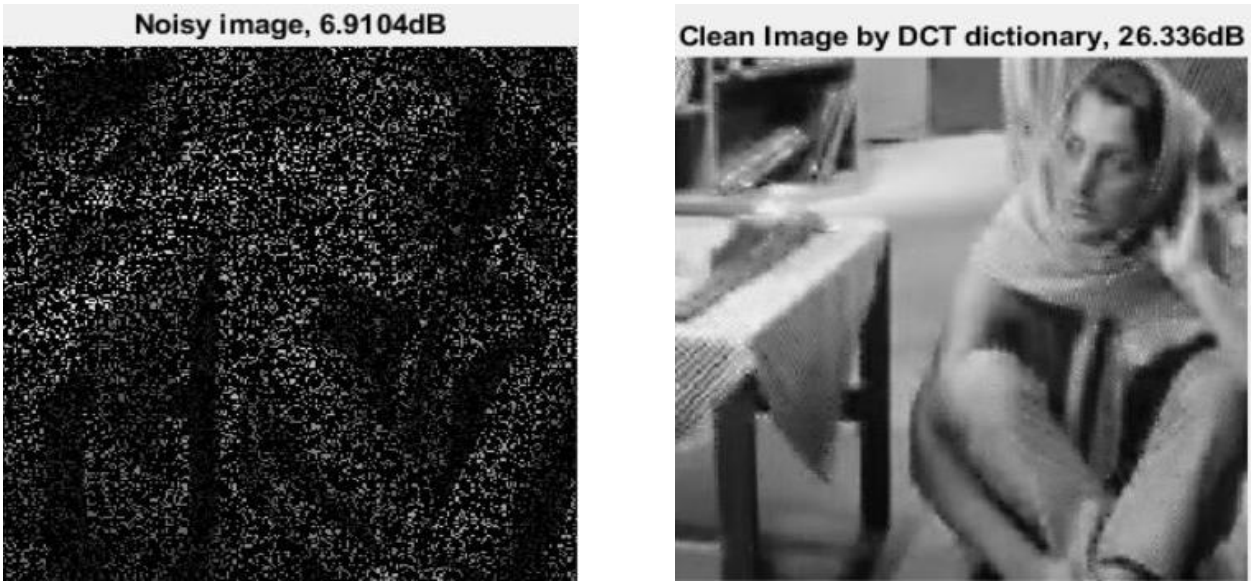


Fig. 3.4 Barbara image with 80% missing pixels

Fig. 3.5 House image with crack missing pixels' pattern

Fig. 3.6 Boats image with crack missing pixels' pattern

Fig. 3.7 Barbara image with text missing pixels' pattern

Fig. 3.8 Barbara image with text missing pixels' pattern

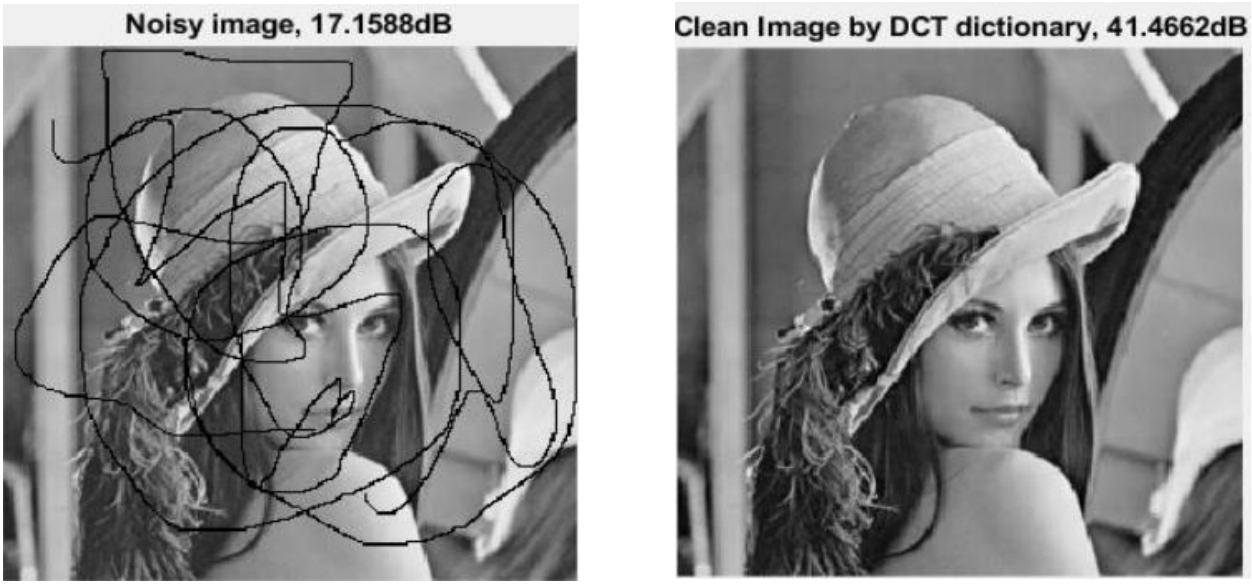


Fig. 3.9 Lena image with drawing missing pixels' pattern

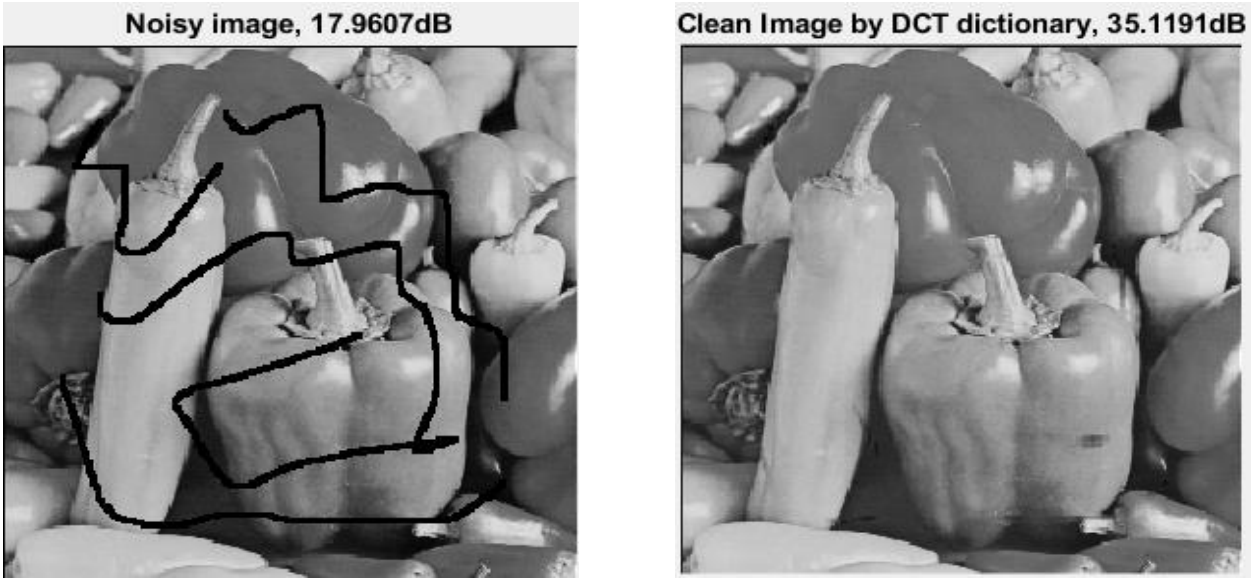


Fig. 3.10 Peppers image with drawing missing pixels' pattern

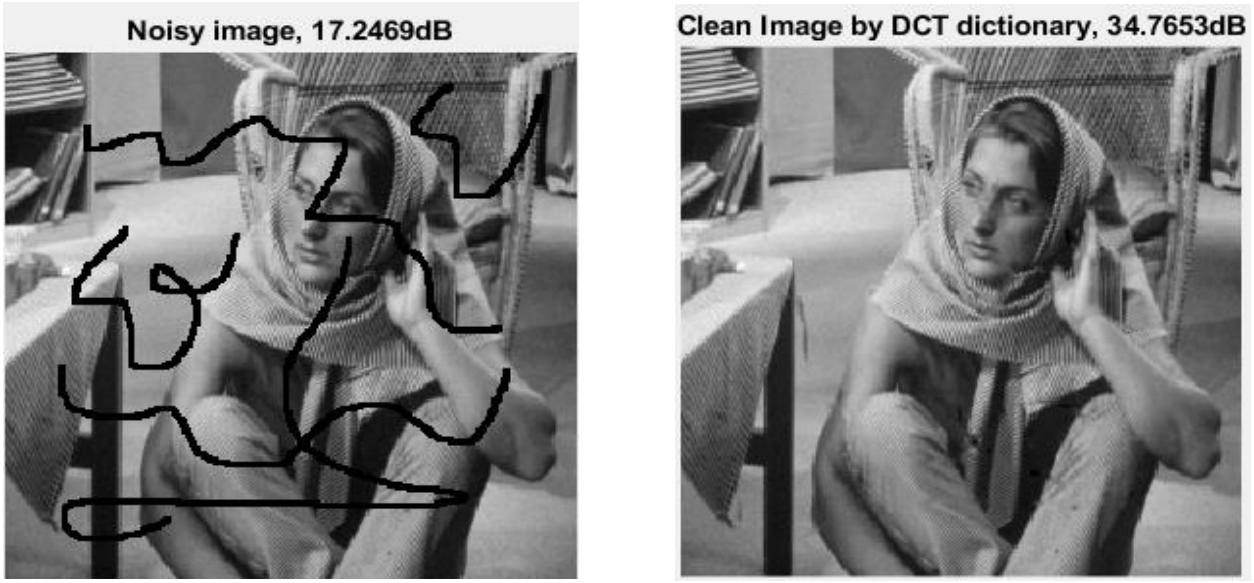
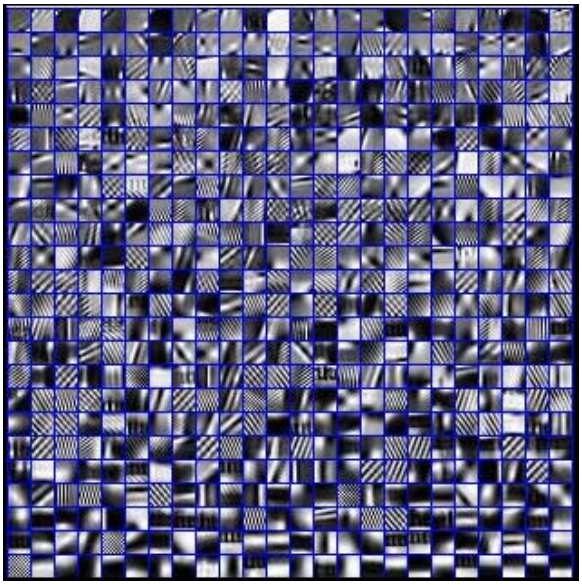


Fig. 3.11 Barbara image with drawing missing pixels' pattern



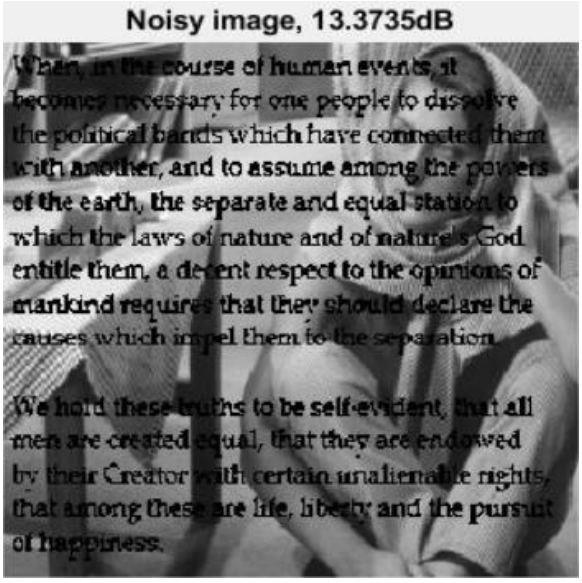


Fig. 3.12 Barbara image with text missing pixels' pattern – adaptive dictionary

References

1 . M. Elad, “Sparse and Redundant Representations from Theory to Applications in Signal and Image Processing,” Dec.2009.

2 . M. J.Mairal, G.Sapiro, “Sparse representation for color image restoration,” *IEEE Trans. Image Process,* vol.17, pp.53–69,2008.

3 . M. Aharon, M. Elad, and A. M. Bruckstein, “The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations,*’’ IEEE Trans. Image* Process., vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

4 . M. Elad and M. Aharon, “Image denoising via learned dictionaries and sparse representation,” presented at the IEEE ComputerVision and Pattern Recognition, New York, Jun. 2006.

5 . M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Process.,* vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

6 . J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Non-local sparse models for image restoration,” in Proc. International Conference on Computer Vision, 2009.