中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-14-001

A Proximal Method for Dictionary Updating in Sparse Representations

Guan-Ju Peng and Wen-Liang Hwang

Feb. 24, 2014 || Technical Report No. TR-IIS-14-001

http://www.iis.sinica.edu.tw/page/library/TechReport/tr2014/tr14.html

A Proximal Method for Dictionary Updating in Sparse Representations

Guan-Ju Peng and Wen-Liang Hwang
Institute of Information Science, Academia Sinica, Taiwan

Abstract

In this paper, we propose a dictionary updating method, called the PMK-SVD method, and show numerically that it can stabilize the dictionary updating process, increase the convergence speed, and converge to a dictionary that outperforms the dictionary derived by the K-SVD method. The proposed method is based on the proximal point approach, which imposes a constraint on the distance of the dictionary modifications in the dictionary updating process. Specifically, we incorporate the approach into the well-known MOD and K-SVD dictionary updating algorithms and combine the results to obtain the PMK-SVD method. We analyze the complexity of the proposed method and compare it with that of the K-SVD method. The results of experiments demonstrate that our method outperforms K-SVD.

1 Introduction

Sparse approximation means that a signal, as a vector, can be well approximated in a low-dimensional column subspace of a matrix. If a signal is sparsely approximated by some basis elements (atoms), then its structure can be derived from that of the atoms. The discrete Fourier and wavelet transforms, represented as square matrices, are sparse approximations of the stationary oscillatory signals and the piecewise smooth signals respectively [1]. This is the reason that the Fourier and wavelet transforms have been widely applied in various domains. For a signal rich in structure that cannot be sparsely approximated by the Fourier or wavelet atoms (i.e., the Fourier or wavelet representations are less effective on the signal), a recently proposed approach [2] derives a sparse approximation of the signal with an over-complement dictionary. Basic linear algebra rules out the unique representation of the signal with the dictionary. Thus, to derive a sparse solution, a constraint is usually imposed on the number of non-zero coefficients. The sparse coding problem and the dictionary learning problem are always encountered when deriving sparse representations with a dictionary. The sparse coding problem tries to derive the sparse solution (where the dictionary is supposed to be known and fixed); while the dictionary

learning problem involves designing a dictionary that can simultaneously approximate a large class of signals sparsely. A plenty of algorithms have been proposed to solve the sparse coding problem [3, 4, 5, 6, 7, 8]. In this paper, we focus on the dictionary learning problem.

Let $Y = [y_1, \cdots, y_L]$ be a matrix of L training signals with $y_i \in R^N$. The objective of the dictionary learning problem is to design a proper dictionary $D \in R^{N \times K}$ (with the normalized column norm) that can derive a sparse representation of the training signals simultaneously. The problem can be formulated as the following minimization problem:

$$\min_{W,D} \|Y - DW\|_F^2 \qquad \text{subject to} \qquad \|w_i\|_0 \leq S, \quad \text{for } i = 1, \cdots, L, \tag{1}$$

where w_i is the ith-column of the matrix W, and S is the maximal number of non-zero elements in each column of W. Usually, iterative algorithms are employed to solve the problem. Each iteration is comprised of two steps: a sparse-coding step and a dictionary-updating step. In the first step, the dictionary is fixed and the elements in W are updated; and in the second step, the dictionary is updated and used in the sparse-coding step of the next iteration. The only difference between the various dictionary learning algorithms is the way they execute the dictionary-updating step. Conceptually, most of the algorithms are based on the Method of Optimal Directions (MOD) algorithm [9] and the K-SVD algorithm [10]. The dictionary-updating step in the MOD approach is based on the least square solution of a system of equations; and in the K-SVD approach, the step is based on the K-means algorithm. The K-SVD approach extends the code words of the K-means algorithm from 1-sparse (at most 1 non-zero coefficient) to S-sparse (at most S non-zero coefficients). The state-of-the-art algorithms based on both batched approaches achieve almost the same convergence rates and yield optimal dictionaries for various sets of experimental data [11].

In this article, we propose a novel dictionary updating approach that exploits the proximal point method, a well-known convex optimization algorithm [12, 13], to solve the dictionary learning problem. The proximal point method has been used to solve the sparse coding problem [13, 8, 14, 15, 16]; however, to the best of our knowledge, it has not been applied to solve the dictionary updating problem. For the sparse coding problem, the proximal point method is used to solve variations of the problem where the $L0$ or $L1$ norm is replaced with another non-smooth regularization term [17, 18], such as the trace norm [19] or the hierarchical norm [20, 21, 22]. Other works that utilize the proximal point approach to solve the sparse coding problem are discussed in [23, 24].

To solve the dictionary updating problem, the MOD algorithm updates the dictionary D in Equation (3) with fixed coefficients W in each iteration; while the K-SVD algorithm updates the dictionary and the coefficients simultaneously. K-SVD also places a constraint on the support of the derived coefficients, but not on the derived dictionary; thus, the variation in the

sequence of dictionaries derived by the K-SVD approach can be large. In numerical analysis, a large variation in the sequence of the dictionaries may overshoot the optimal dictionary and slow down the convergence process. The overshooting problem can be solved by imposing a regularization term on dictionary modifications to control the variations in the derived dictionaries. Specifically, we extend K-SVD by using the proximal point method to regularize the distance between the dictionaries obtained in two consecutive iterations. Therefore, the dictionary updating process derives a new dictionary that takes account of the current estimated dictionary and the estimation method. The approach is similar to the successive over-relaxation approach[1] which is normally used to stabilize and increase the convergence of Newton's method. The over-relaxation approach resolves the overshooting problem in cases where the first derivative of a function does not behave well in the neighborhood of a root or several roots are closely aggregated. Moreover, we show that the proposed approach (1) preserves the characteristics of the K-SVD method; and (2) integrates the MOD algorithm and the K-SVD dictionary updating method in a single framework.

Normally, the optimal point method is exploited when the original objective function of an optimization problem is not differentiable, or it is difficult to derive an optimal solution directly. When the proximal point method is applied to an optimization problem, an extra function $d(x, x_i)$ is added to the original objective function $f(x)$ to form the proximal (or surrogate) function $q(x, x_i)$, where x_i is derived from the previous iteration. Under the proximal point approach, $q(x, x_i)$ must have the following properties: (1) $q(x_i, x_i) = f(x_i)$; and (2) $q(x, x_i) \geq f(x)$ for all x and x_i in the feasible domain so that the sequence generated by the recurrent equation

$$x_{i+1} = \arg\min_{x} q(x, x_i) \tag{2}$$

converges to a local minimum of the original function $f(x)$ as $i \to \infty$. Under the proximal point approach, the dictionary learning problem can be defined so as to solve the following optimization problem:

$$\min_{W,D} \|Y - DW\|_F^2 + \lambda||D - D_{old}||_F^2 \quad \text{subject to} \quad \|w_i\|_0 \leq S, \quad \text{for } i = 1, \cdots, L, \tag{3}$$

where $\|Y-DW\|_F^2+\lambda||D-D_{old}||_F^2$ is the proximal function, w_i is the ith-column of the matrix W, S is the maximal number of non-zero elements in each column of W, and D_{old} is the dictionary derived in the previous iteration.

The optimization problem in Equation (3) can be solved by a number of techniques. In this paper, we use a combination of matrix calculus and singular value decomposition (SVD). As the

[1]The formulation of a successive over-relaxation depends on the problem. The simplest formulation may be $x_{n+1} = \lambda x_n + (1 - \lambda) f(x_n)$, where $f(x_n)$ is the root estimation function, and λ is the relaxation parameter.

MOD method is based on matrix calculus and the K-SVD method is based on singular value decomposition, the proposed method is called PMK-SVD. To evaluate the method's performance, we conduct experiments on dictionary recovery, sparse approximations, filling in the missing pixels of an image, and compression of an image. The results demonstrate that the PMK-SVD method outperforms the K-SVD method.

The reminder of this paper is organized as follows. In Section 2, we review the MOD and the K-SVD approaches. In Section 3, we present two proximal point-based dictionary learning approaches for the MOD method and the K-SVD method, and show that the approaches can be combined to derive a new learning algorithm. We also analyze the complexity of different learning methods. In Section 4, we consider a number of implementation issues, and discuss the results of experiments conducted to compare the performance of the dictionaries learned by different methods. Section 5 contains some concluding remarks.

2 The MOD and K-SVD Approaches

Many dictionary learning algorithms are conceptually similar to the original algorithms in the MOD approach and the K-SVD approach.

2.1 The MOD approach

The MOD approach tries to update a dictionary D based on the current coefficients W, derived by a sparse coding method, to minimize the least square error of

$$\min_{D} ||Y - DW||_F^2. \tag{4}$$

The optimal dictionary is obtained by solving the following equation:

$$\frac{\partial \|Y - DW\|_F^2}{\partial D} = 0, \tag{5}$$

and the analytical solution is

$$D = YW^T(WW^T)^{-1}. \tag{6}$$

Because WW^T is not always a full-rank matrix, the iterative steepest descent update rule is proposed in place of Equation (6), and the dictionary is updated by

$$D = D_{old} + \eta(DW - Y)W^T, \tag{7}$$

where D_{old} is the dictionary before the updating step in the current iteration, and η is a coefficient that determines the speed of convergence. The iterative LS-based dictionary learning algorithm

(ILS-DLA) is a state-of-the-art algorithm that is conceptually similar to MOD method. It extends the MOD method by designing signal dependent block-based dictionaries and overlapping dictionaries.

The MOD method was originally designed for the batch processing. Recently, an on-line dictionary learning algorithm and a recursive least squares dictionary learning algorithm (RLS-DLA) were proposed by Mairal et al. [25] and Skretting and Engan [11] respectively. The algorithms use on-line dictionary updating rules that perform dictionary learning by processing one signal at a time in sequence. They are suitable for dictionary learning in on-line applications.

2.2 The K-SVD Approach

Next, we consider the K-SVD algorithm. In contrast to the algorithms in the MOD category, which are based on the linear regression approach, the K-SVD algorithm is conceptually generalized from the K-means algorithm. Let D_{old} be the current dictionary and W_{old} be the current coefficients derived by the sparse coding step for D_{old}. The K-SVD algorithm tries to solve the following optimization problem, using D_{old} and W_{old} as the initial inputs:

$$\min_{D,W} ||Y - DW||_F^2 \quad \text{with } supp(W) \subseteq supp(W_{old}), \tag{8}$$

where the support of $W(supp(W))$ represents the locations of the non-zero coefficients in W. In the dictionary updating step, the K-SVD approach imposes a stronger constraint than the number of non-zero coefficients, because it requires that the locations of the non-zero coefficients of W are a subset of those of W_{old}. Aharon et al. [10] show that the solutions, D and W, of the above optimization problem can be derived in an iterative manner by updating one column d_k at a time in the dictionary along with the corresponding k-th row coefficients, denoted as r_k^T (the transpose of r_k). To do this, we let $\|r_k^T\|_0$ denote the number of non-zero coefficients in r_k^T, and $p_k(i)$ denote the index of the i-th non-zero coefficient in r_k^T. In addition, we let Ω_k be the $L \times \|r_k^T\|_0$ matrix, with the entries $(p_k(i), i)$ set at one and other entries set at zero. As a result, the i-th column of the matrix $W\Omega_k$ becomes the $p_k(i)$-th column of the matrix W. To modify the column d_k and the corresponding coefficients r_k^T simultaneously, we rewrite Equation (8) as follows:

$$\begin{aligned} ||Y\Omega_k - DW\Omega_k||_F^2 &= ||Y\Omega_k - \sum_{j=1}^{K} d_j r_j^T \Omega_k||_F^2 & (9) \\ &= ||E_k\Omega_k - d_k r_k^T \Omega_k||_F^2, & (10) \end{aligned}$$

where $E_k = Y - \sum_{j \neq k} d_j r_j^T$. Let $E_k\Omega_k = U_k \Delta V_k^T$ be the singular value decomposition of $E_k\Omega_k$. The K-SVD algorithm updates d_k by setting it as the first column of U_k, updates $r_k^T\Omega_k$ by setting it as the first row of V_k, and then multiplies the result by $\Delta(1,1)$. The modification of one column and the corresponding coefficients reduces the value of Equation (10).

Skretting [11, 26] observed that, in practice, the K-SVD and the ILS-DLA algorithms converge to a dictionary in almost the same number of iterations. However, the computation time of K-SVD is longer due to the higher computational complexity (SVD calculations) of the dictionary updating rule.

3 The Proximal Point Approach

The proposed proximal point approach updates the dictionary by solving the following optimization problem with $\lambda \geq 0$:

$$\min_{W,D} \|Y - DW\|_F^2 + \lambda||D - D_{old}||_F^2 \quad \text{subject to} \quad \|w_i\|_0 \leq S, \quad \text{for } i = 1, \cdots, L, \tag{11}$$

where $\|Y - DW\|_F^2 + \lambda||D - D_{old}||_F^2$ is the proximal function, D_{old} is obtained from the previous iteration, and w_i is the ith-column of W. Next, we show that the optimization sequence in Equation (11) converges. Let (D_n, W_n) and (D_{n-1}, W_{n-1}) be the solutions derived by the equation in any two consecutive iterations. The dictionary updating step of the proximal point approach converges because

$$\begin{aligned} ||Y - D_n W_n||_F^2 &\leq ||Y - D_n W_n||_F^2 + \lambda||D_n - D_{n-1}||_F^2 &(12)\\ &\leq ||Y - D_{n-1} W_{n-1}||_F^2 + \lambda||D_{n-1} - D_{n-1}||_F^2 &(13)\\ &= ||Y - D_{n-1} W_{n-1}||_F^2. &(14) \end{aligned}$$

The inequality in Equation (13) occurs because (D_n, W_n) minimizes Equation (11) with $D_{old} = D_{n-1}$. The proximal function in (11) can be solved by applying matrix calculus (the MOD style) or singular value decomposition (the K-SVD style), which we discuss in the next two subsections.

3.1 Proximal MOD (P-MOD) Method

In the MOD method, the dictionary is updated with a fixed coefficient matrix. The proximal point approach can be incorporated into the MOD method by rewriting Equation (4) as follows:

$$\min_{D} \|Y - DW\|_F^2 + \lambda\|D - D_{old}\|_F^2. \tag{15}$$

Let $g(D, D_{old}) = \|Y - DW\|_F^2 + \lambda\|D - D_{old}\|_F^2$ be the proximal function. Then, the optimal dictionary of Equation (15) can be derived by taking the partial derivative of $g(D, D_{old})$ with respect to D and setting the result to 0. Let $tr(A)$ represent the trace of the matrix A. Given the fact that $tr(A^T A) = \|A\|_F^2$, after some straightforward calculations, we obtain

$$\begin{aligned} \frac{\partial g(D, D_{old})}{\partial D} &= \frac{\partial \|Y - DW\|_F^2}{\partial D} + \lambda \frac{\partial \|D - D_{old}\|_F^2}{\partial D}\\ &= 2(-YW^T + DWW^T + \lambda(D - D_{old}))\\ &= 0. &(16) \end{aligned}$$

Thus, the dictionary updating rule (called the P-MOD method) is formulated as

$$D = (D_{old} + \frac{1}{\lambda} Y W^T)(I + \frac{1}{\lambda} W W^T)^{-1}, \tag{17}$$

where I is the identity matrix. Because we can select the value of the coefficient λ to ensure that the matrix $(I + \frac{1}{\lambda} W W^T)$ is a full rank matrix, and therefore invertible, the derived optimal dictionary is reliable.

We use $\tilde{D}_{n-1}$ and $\tilde{W}_{n-1}$ to denote, respectively, the dictionary and the coefficient matrix derived in the previous iteration of the P-MOD method. The current dictionary is obtained by

$$\tilde{D}_n = (\tilde{D}_{n-1} + \frac{1}{\lambda} Y \tilde{W}_{n-1}^T)(I + \frac{1}{\lambda} \tilde{W}_{n-1} \tilde{W}_{n-1}^T)^{-1}; \tag{18}$$

and the current coefficient matrix $\tilde{W}_n$ can be derived by solving the following sparse coding problem:

$$\min_{\tilde{W}} \|Y - \tilde{D}_n \tilde{W}\| \qquad \text{subject to} \qquad \|\tilde{w}_i\|_0 \leq S. \tag{19}$$

Because of the dictionary and the coefficient updating rules in Equations (18) and (19), respectively, the P-MOD method converges since

$$\begin{aligned} ||Y - \tilde{D}_n \tilde{W}_n||_F^2 &\leq ||Y - \tilde{D}_n \tilde{W}_{n-1}||_F^2 & (20) \\ &\leq ||Y - \tilde{D}_n \tilde{W}_{n-1}||_F^2 + \lambda ||\tilde{D}_n - \tilde{D}_{n-1}||_F^2 & (21) \\ &\leq ||Y - \tilde{D}_{n-1} \tilde{W}_{n-1}||_F^2. & (22) \end{aligned}$$

Equation (20) is obtained because $\tilde{W}_n$ is the solution of Equation (19); and Equation (22) is obtained because $\tilde{D}_n$ solves Equation (15) with $D_{old} = \tilde{D}_{n-1}$ and $W = \tilde{W}_{n-1}$.

3.2 Proximal K-SVD (PK-SVD) Method

The proximal point approach can be incorporated into the K-SVD method, which minimizes the following proximal point surrogate function:

$$\min_{D,W} ||Y - DW||_F^2 + \lambda ||D - D_{old}||_F^2 \quad \text{with } supp(W) \subseteq supp(W_{old}). \tag{23}$$

If $\lambda = 0$, this optimization problem becomes the K-SVD problem. To solve it, we use the same approach as K-SVD by iteratively updating one column at a time in the dictionary along with the corresponding non-zero row coefficients. Let d_k and r_k^T denote the k-th column in D and the k-th row in W respectively. In addition, let $p_k(i)$ be the i-th non-zero index in r_k^T, and let $||r_k^T||_0$ be the number of non-zero coefficients. If we let Ω_k be an $L \times \|r_k^T\|_0$ matrix in which the entries $(p_k(i), i)$ are set at one and other entries are set at zero, $r_k^T \Omega_k$ is a $1 \times ||r_k^T||_0$ row vector with non-zero entries; for example, if $r_k^T = [0\ 1\ 0\ 0\ 2]$, then $r_k^T \Omega_k$ is $[1\ 2]$.

Next, we consider the procedure used to update the k-th column d_k in the dictionary and the corresponding non-zero coefficients $r_k^T\Omega_k$. First, we have

$$\begin{aligned} ||Y\Omega_k - DW\Omega_k||_F^2 &= ||(Y - \sum_{j\neq k} d_j r_j^T)\Omega_k - d_k r_k^T\Omega_k||_F^2 && (24)\\ &= ||E_k\Omega_k - d_k r_k^T\Omega_k||_F^2, && (25)\\ &= ||\tilde{E}_k - d_k x_k^T||_F^2, && (26)\end{aligned}$$

where $E_k = Y - \sum_{j\neq k} d_j r_j^T$, $\tilde{E}_k = E_k\Omega$, and $x_k^T = r_k^T\Omega_k$. Equation (26) is the K-SVD updating rule, which uses a rank 1 matrix, $d_k x_k^T$, to approximate the error matrix $\tilde{E}_k$. Let d_k^{old} denote the k-th column in the dictionary D_{old}; then, we have

$$||Y\Omega_k - DW\Omega_k||_F^2 + \lambda||D - D_{old}||_F^2 = ||\tilde{E}_k - d_k x_k^T||_F^2 + \lambda||d_k - d_k^{old}||_F^2 + C_k, \quad (27)$$

where C_k is a constant that is independent of d_k and x_k^T. Now, to obtain the optimal solution of the problem in Equation (23), we have to solve the following subproblem:

$$\min_{d_k, x_k^T} ||\tilde{E}_k - d_k x_k^T||_F^2 + \lambda||d_k - d_k^{old}||_F^2. \quad (28)$$

Comparison of Equations (26) and (28) shows that our approach is a rank 1 approximation of $\tilde{E}_k$ like the K-SVD approach; however, our d_k is penalized because of its distance from d_k^{old}. Therefore, the optimal solution cannot be obtained by the K-SVD method, which takes the SVD of $\tilde{E}_k = U\Delta V^T$, and then assigns $d_k = u_1$ (the first column of U) and $x_k^T = \Delta(1,1)v_1^T$ (by multiplying the first row of V^T by the $(1,1)$ element of Δ).

Let $f(d_k, x_k; \tilde{E}_k, d_k^{old}) = ||\tilde{E}_k - d_k x_k^T||_F^2 + \lambda||d_k - d_k^{old}||_F^2$. After taking the partial derivative of $f(d_k, x_k; \tilde{E}_k, d_k^{old})$ with respect to d_k and x_k and setting the results to zero, we obtain the following equations for the optimal solution of Equation (28):

$$\begin{aligned} (||x_k||^2 + \lambda)d_k &= \tilde{E}_k x_k + \lambda d_k^{old} && (29)\\ ||d_k||^2 x_k &= \tilde{E}_k^T d_k, && (30)\end{aligned}$$

where $||d_k|| = 1$ because each column of the dictionary is normalized. Next, we derive the solutions of d_k and x_k in (29) and (30). For convenience, we omit the subscript indices in the equations. We can rewrite Equation (29) as

$$||x||^2 d = \tilde{E}x + \lambda(d^{old} - d), \quad (31)$$

and multiply $||x||^2$ on both sides of Equation (30) to obtain

$$||x||^2 x = \tilde{E}^T||x||^2 d. \quad (32)$$

Then, we substitute $||x||^2 d$ obtained in Equation (31) into Equation (32) to obtain

$$||x||^2 x = \tilde{E}^T \tilde{E} x + \lambda \tilde{E}(d^{old} - d). \tag{33}$$

Let us assume that x is in the direction of an eigenvector of $\tilde{E}^T \tilde{E}$; that is, $x = ||x||\hat{x}$ and

$$\mu \hat{x} = \tilde{E}^T \tilde{E} \hat{x}; \tag{34}$$

then, Equation (33) becomes

$$\|x\|^3 \hat{x} = \mu \|x\| \hat{x} + \lambda \tilde{E}^T (d^{old} - d). \tag{35}$$

If $\lambda = 0$ and if $\hat{x}$ and μ are chosen as the first eigenvector and the corresponding largest eigenvalue of $\tilde{E}^T \tilde{E}$, we can derive the K-SVD solutions from the above equation. For $\lambda > 0$, if we multiply both sides of Equation (35) by $\tilde{E}^{T+} = (\tilde{E}\tilde{E}^T)^{-1}\tilde{E}$ (the pseudo inverse matrix of $\tilde{E}^T$), we have

$$\|x\|^3 \tilde{E}^{T+} \hat{x} = \mu \|x\| \tilde{E}^{T+} \hat{x} + \lambda P(d^{old} - d), \tag{36}$$

where $P = \tilde{E}^{T+} \tilde{E}^T$ is the orthogonal projection onto the range of $\tilde{E}$. Re-arranging the terms in Equation (36), we obtain

$$Pd = \frac{1}{\lambda}(\mu \|x\| - \|x\|^3) \tilde{E}^{T+} \hat{x} + P d^{old}. \tag{37}$$

If we let $Q = I - P$ be the orthogonal complement of P, we have

$$(P + Q)(d - d^{old}) = P(d - d^{old}) + Q(d - d^{old}) = d - d^{old}. \tag{38}$$

By imposing $Q(d - d^{old}) = 0$ (that is, the dictionary columns d and d^{old} have the same projection in the orthogonal complement space of P), we have

$$d = (P + Q)d = Pd + Qd = Pd + Qd^{old}. \tag{39}$$

To derive the norm of x, we could substitute Equation (37) into Equation (39) to obtain d as a function of $||x||$; and then solve $||x||$ by exploiting the fact that $d^T d = 1$. However, the approach would derive the roots of a polynomial of $||x||$ with degree 6, which would be computationally inefficient. Instead, we use the following approach to find the norm of x.

From Equation (30), we have $x^T x = \|x\|^2 = d^T \tilde{E} \tilde{E}^T d$. In addition, we let the SVD of $\tilde{E}\tilde{E}^T$ be

$$\tilde{E}\tilde{E}^T = \sum_{i=1}^{l} \sigma_i^2 v_i v_i^T. \tag{40}$$

The norm of x can be written as

$$\|x\|^2 = \sum_{i=1}^{l} \sigma_i^2 < d, v_i >^2, \tag{41}$$

where l is the rank of $\tilde{E}$ and $\sigma_1 \geq \cdots \geq \sigma_l$. This equation gives the bounds of the norm by

$$\sigma_l \leq ||x|| \leq \sigma_1. \tag{42}$$

Because of Equation (30), we can substitute x for $\tilde{E}^T d$ in Equation (35) and obtain

$$(\|x\|^2 + \lambda - \mu)x = \lambda \tilde{E}^T d^{old}. \tag{43}$$

Taking the norm on both sides of the above equation, we obtain

$$(\|x\|^2 + \lambda - \mu)\|x\| = \lambda \|E^T d^{old}\|. \tag{44}$$

Let d_o be the current estimate of d. Then, to update the dictionary column (atom) d^{new} and the corresponding coefficient vector x^{new} and use the following procedure: (1) if $\lambda = 0$, we return the result of the K-SVD method; or (2) if $\lambda > 0$, we substitute the current estimate d_o into Equation (41) to derive $||x_o||$, which is used in the term $\|x\|^2 + \lambda - \mu$ in Equation (44), to update the norm by

$$\|x^{new}\| \leftarrow \begin{cases} \frac{\lambda \|E^T d^{old}\|}{\|x_o\|^2 + \lambda - \mu}, \text{ if } \|x_o\|^2 + \lambda - \mu \neq 0, \\ \|x_o\|, \text{ otherwise.} \end{cases} \tag{45}$$

The term $||x^{new}||$ replaces $||x_o||$ in subsequent iterations until the norm does not change[2]. At that point, we substitute $||x^{new}||$ for $||x||$ in Equation (37) and derive Pd^{new}. Then, the dictionary column is updated by

$$d^{new} \leftarrow Pd^{new} + Qd^{old} \tag{46}$$

and used in place of d_o to repeat the process until some stopping condition is reached.

The updating rule in Equation (46) has a special case where d^{new} is not updated. This occurs when P is an identify matrix and therefore $Q = 0$. In this case, we can re-formulate Equation (31) as a root-finding problem and use Newton's method to update the vector d^{new}. Let the root-finding problem be $g(d) := (||x^{new}||^2 + \lambda)d - \tilde{E}x^{new} + \lambda d^{old}$, where the root of $g(d)$ is the solution of Equation (31). Then, based on Newton's method, the vector d^{new} is updated by

$$d^{new} \leftarrow d_o - \frac{g(d_o)}{g'(d_o)} = d_o - \frac{(||x^{new}||^2 + \lambda)d_o - \tilde{E}x^{new} + \lambda d^{old}}{||x^{new}||^2 + \lambda}. \tag{47}$$

This procedure yields the solutions of Equations (29) and (30), and thereby solves the optimal problem in Equation (28). Table 1 summarizes the steps of the proposed one column dictionary and coefficient updating method, called the One-Atom-PK-SVD method. Note that, in Step 1, the initial d_o and $\hat{x}$ are derived by the K-SVD method; that is, d_o is the first eigenvector of

[2]In practice, we found that one iteration is sufficient to derive the norm of x.

$\tilde{E}\tilde{E}^T$ by default and μ is therefore the largest eigenvalue. Moreover, in Step 6, we can use an alternative condition, such as the maximum number of iterations, to exit the algorithm. In Table 2, based on the One-Atom-PK-SVD, we present the proximal point K-SVD dictionary updating method, called the PK-SVD method; and Table 3 shows the **PK-SVD-Learning** algorithm, which utilizes the sparse coding step and the PK-SVD dictionary updating step alternately to learn an optimal dictionary for the training signals.

3.3 Combining the Proximal MOD and Proximal K-SVD Approaches

We combine the proximal point approaches of P-MOD and PK-SVD as follows.

Let D^{n-1} and W^{n-1} be, respectively, the dictionary and the coefficient matrix derived in the $(n-1)$-th iteration. First, we update the dictionary in the n-th iteration by using the P-MOD method, which modifies D^{n-1} by solving

$$\min_{D} ||Y - DW^{n-1}||_F^2 + \lambda_p ||D - D^{n-1}||_F^2, \tag{48}$$

and letting D_{pmod} be the solution of the above equation. Then, we modify the dictionary by solving

$$\min_{D,W} ||Y - DW||_F^2 + \lambda_s ||D - D_{pmod}||_F^2 \quad \text{with } supp(W) \subseteq supp(W^{n-1}). \tag{49}$$

The above optimization procedure is convergent. Let D^n and W^n be, respectively, the dictionary and the coefficient matrix after the n-th iteration. Then, we have

$$\begin{aligned}
||Y - D^n W^n||_F^2 &\leq ||Y - D^n W^n||_F^2 + \lambda_s ||D^n - D_{pmod}||_F^2 && (50)\\
&\leq ||Y - D_{pmod} W^{n-1}||_F^2 + \lambda_s ||D_{pmod} - D_{pmod}||_F^2 && (51)\\
&\leq ||Y - D_{pmod} W^{n-1}||_F^2 + \lambda_p ||D_{pmod} - D_{pmod}||_F^2 && (52)\\
&\leq ||Y - D_{pmod} W^{n-1}||_F^2 + \lambda_p ||D_{pmod} - D^{n-1}||_F^2 && (53)\\
&\leq ||Y - D^{n-1} W^{n-1}||_F^2 && (54)
\end{aligned}$$

In the above derivations, Equation (51) is derived because D^n and W^n are the solutions of Equation (49); and Equation (54) is obtained because D_{pmod} is the solution of Equation (48). Table 4 shows the combined dictionary learning approach. Because the method is a combination of the P-MOD and PK-SVD dictionary updating methods, we call it the **PMK-SVD Learning** method.

3.4 Computational Complexity of Dictionary Updating Methods

We compare the computational complexity of the following dictionary updating methods: K-SVD, P-MOD, PK-SVD, and PMK-SVD. The complexity is represented by the parameters: S, N, K, L, which correspond to the number of non-zero elements, the dimensions of signals, the

number of columns in a dictionary, and the number of training signals respectively. Generally, we use $L \geq K \geq N \geq S$ for dictionary learning. According to [27], the complexity of the SVD calculation of a matrix of size $R^{m \times n}$ is $O(mn^2 + m^2 n + n^3)$. Because all the dictionary updating methods are iterative algorithms, it is difficult to compare their complexity without knowing the number of iterations that each algorithm requires to converge. Thus, our comparison is based on one iteration of each method.

A. The K-SVD Method:

This method's complexity is bounded by calculating the matrix $E_k\Omega_k$ and performing SVD on it. Calculating $E_k\Omega_k$ (Equation (10)) takes $O(KLN)$ steps because E_k can be obtained by KNL steps and Ω_k is used to to select some columns from E_k, therefore, the calculation takes at most NL steps.

The average complexity of SVD is bounded by the size of the matrix $E_k\Omega_k$. The number of rows in $E_k\Omega_k$ is N; and the number of columns is the number of non-zero coefficients in the k-th row of the coefficient matrix W of size $K \times L$. Because each column of W contains at most S non-zero coefficients, the number of non-zero coefficients in W is bounded by LS. If the non-zero coefficients are distributed uniformly in W, the average non-zero coefficient in each row is $\frac{SL}{K}$; thus, the size of $E_k\Omega_k$ is $N \times \frac{SL}{K}$. As a dictionary has K columns, each iteration of the K-SVD dictionary's update procedure takes $O(K(KLN + N^2(\frac{SL}{K}) + N(\frac{SL}{K})^2 + (\frac{SL}{K})^3))$ time.

B. The P-MOD Method:

This method is based on Equation (17). Its complexity is bounded by the inverse of the $K \times K$ matrix, which costs $O(K^3)$, and three matrix multiplications take $O(NLK)$ (YW^T), $O(K^2L)$ (WW^T), and $O(NK^2)$ (the multiplications of the matrices $(D_{old} + \frac{1}{\lambda}YW^T)$ and $(I + \frac{1}{\lambda}WW^T)^{-1}$). The complexity of the P-MOD method is $O(K^2L) + O(K^3) + O(NLK) + O(NK^2)$.

C. The PK-SVD Method:

The step-wise complexity of **PK-SVD** is shown in the last column of Table 2. Step 2.1 calculates E_k and Ω_k (the cost was given in part **A** of this section). The complexity is bounded by Step 2.2, which updates one column of the current dictionary and its corresponding coefficients at a time. The step-wise complexity of One-Atom-PK-SVD is detailed in the last column of Table 1. The complexity is bounded by the SVD decomposition in Step 1. As a result, the PK-SVD and the K-SVD methods have the same order of complexity, which is $O(K(KLN + N^2(\frac{SL}{K}) + N(\frac{SL}{K})^2 + (\frac{SL}{K})^3))$.

D. The PMK-SVD Method:

The complexity of the P-MOD+PK-SVD method is a combination of the complexities of the P-MOD and the PK-SVD methods. When the number of training signals, L, is the dominant parameter, the complexities of the K-SVD, P-MOD, PK-SVD, and PMK-SVD methods are $O(L^3)$, $O(L)$, $O(L^3)$, and $O(L^3)$ respectively.

4 Implementation Issues and Performance Evaluation

In this section, to distinguish between a dictionary updating method and a dictionary learning method, we use boldface letters to denote the latter. We consider a number of implementation issues and compare the performances of the dictionaries learned by the **K-SVD** learning method, the **PK-SVD** learning method (see Table 3, the algorithm **PK-SVD-Learning**), and the **PMK-SVD** learning method (See Table 4 the algorithm **PMK-SVD-Learning**).

4.1 Implementation Issues

We consider three implementation issues: (1) the sparse coding algorithm; (2) the values of the Lagrangian multiplier (λ_s and λ_p); and (3) the balance between the actual running time and the performance. For the first issue, we conducted experiments on the dictionary learning methods by using the Order Recursive Matching Pursuit (ORMP) method proposed in [28, 29] to implement the sparse coding stage. The ORMP method is more efficient than the OMP method [5, 11, 26], and it achieves a better performance. For the second issue, the values of the Lagrangian multipliers λ_s (in the PK-SVD updating step) and λ_p (in the P-MOD updating step) are derived experimentally. We suggest setting the value of λ_s at $\frac{\sigma^2}{100}$, where σ is the 2-norm of x_o in Equation (45), and setting the value of λ_p in the range $[3-10]$. Meanwhile, in the **PMK-SVD** learning method, shown in Table 4, the original value of λ_p is set at 10, and the value is reduced to $0.8\lambda_p$ in Step 4 after every 3 iterations.

For the third issue, the proximal point-based dictionary updating method employs a loop that modifies the dictionary derived in the previous iteration. The modification is necessary to ensure that the proximal point method converges to an optimal solution in each dictionary updating stage. However, we found that, in practice, convergence is not necessary in each stage for the proximal point method to achieve a good dictionary learning performance. In our implementation, the **PK-SVD** learning algorithm performs the PK-SVD dictionary updating step in one iteration; and the **PMK-SVD** learning algorithm performs the PK-SVD dictionary updating step as well as the P-MOD dictionary updating step in one iteration.

4.2 Performance Evaluation

We evaluate the performance of the dictionary learning algorithms in terms of (1) recovery of the original dictionary from the training data; (2) deriving sparse approximations; and (3) supporting image processing applications.

4.2.1 Recovery of the Original Dictionary

In this experiment, the original random dictionary D of size 20×50 is generated with i.i.d. uniformly distributed entries, where each column of the dictionary is normalized to 1 by the l^2 form. Then, $1,500$ items of training data are generated from D. Each synthetic training signal is a linear combination of the three columns in D, where the coefficients and the column locations in D are selected in a uniformly distributed i.i.d. random distribution. Finally, white Gaussian noise is added to the training signals to obtain noisy signals with various signal-to-noise ratios ($SNRs$).

The initial dictionary for the dictionary learning algorithm is comprised of the first 50 training signals. The sparsity level, s, is set at 3; and each learning algorithm performs 30 iterations.

To evaluate the performance of the learned dictionary $\hat{D}$ in recovering the original dictionary, we compare it with the performance of the original dictionary D that generates the training data. Because the order of the columns in D and $\hat{D}$ may be different, the comparison is made by measuring the distances between their columns via

$$1 - \|d_i^T \hat{d}_j\|, \tag{55}$$

where d_i is the i-th column of D, and $\hat{d}_j$ is the j-th column of $\hat{D}$. If the distance is less than 0.01, the column d_i is regarded as successfully recovered. The recovery ratio is calculated by dividing the number of recovered columns by the total number of columns, which is 50 in this case.

Table 5 compares the recovery ratios of **K-SVD**, **ODL** (Online Dictionary Learning) [30], and **RLS** (Recursive Least Square dictionary learning) [11]. The table also shows the ratio gains of the proposed methods over **K-SVD**. The experiments were conducted on training data with the following noise levels: 5, 10, 15, and 20 dBs as well as on the noiseless case. The experiment for each noise level involved 100 trials that used different training data (each trial had $1,500$ items of data).

4.2.2 Sparse Approximations

The goal of this experiment is to determine whether a dictionary can represent the original signals as sparse approximations. The experiment models the original signals by their sparse representations whose dictionaries are obtained by the learning algorithms. The performance is measured by the signal-to-noise ratio, where the noise term is derived from the error obtained by subtracting the original signals from their sparse approximations. Thus, the higher the signal-to-noise ratio, the better will be dictionary learning algorithm.

Let D, X and W be the learned dictionary, the training signals, and the coefficients respec-

tively. Then the *SNR* is calculated as follows:

$$\log_{10}(\frac{tr(XX^T)}{tr((X-DW)(X-DW)^T)}), \tag{56}$$

where tr is the trace operation. In this experiment, we also measure and compare the actual execution times of the different dictionary updating methods. The experiments for sparse approximation conducted on two sets of data as follows.

A. Random signals

The training data set contains $4,000$ vectors of size 16 generated by the normal or uniform random processes. The size of the dictionary is 16×32, and the initial dictionary is selected from the first 32 training signals. Table 6 shows the resulting *SNR*s of the dictionary learning algorithms.

B. Image blocks

The training data set contains $12,288$ blocks of size 8×8 randomly selected from some generic 512×512 grey scale images. In the experiments, the 8×8 blocks are mapped to 1D vectors of size 64 after their DC values have been removed. The dictionary size is set at 64×128. The initial dictionary is obtained by concatenating the DCT basis and the Haar basis. In Figure 1, we compare the *SNR*s after each iteration of the five dictionary learning algorithms on the training data set. The results show that the proximal point-based learning methods yield better *SNR*s than the **K-SVD** learning method in a smaller number of iterations. They also converge to better dictionaries if more iterations are performed.

4.2.3 Image Processing Applications

The experiment results in the previous two subsections show that the dictionaries learned by the **PMK-SVD** learning method outperform those learned by the **PK-SVD** method. Thus, in the following, we only compare the performance of the **PMK-SVD** dictionary and **K-SVD** dictionary (a state-of-the-art algorithm) on two image processing applications: (1) filling in the missing pixels of a corrupted image; and (2) image compression.

For both applications, the dictionaries are learned by the following procedure. The training data consists of $53,629$ blocks of size 8×8, selected randomly from 23 images. The DC values of the training vectors are removed so that the mean of each resultant training vector is zero. In the dictionary learning phase, we set the sparsity level at 8 and let the size of the dictionary be 64×513. The first column of the dictionary is a constant-value vector that represents the DC, so it is not updated during the dictionary learning phase. The other 512 atoms of the original dictionary are selected at random from the training data. We performed 80 iterations for each dictionary learning method.

A. Filling in the Missing Pixels

The objective is to recover the values of the missing pixels with the help of the learned dictionary. To this end, we remove a fraction of each training vector's pixels (between 0.2 and 0.7) by setting their values at zero. The pixels to be removed are selected at random. Let $\hat{D}$ be a learned dictionary. To reconstruct the image from the corrupted one, the missing pixels of each block are filled by the following process

Let x denote a block of pixels, and let a matrix Q indicate the positions of the non-corrupted pixels in x. Then, their product, Qx, is a vector that only includes the non-corrupted coefficients in x; and $Q\hat{D}$ denotes the matrix in which each column only includes the positions of the non-corrupted coefficients in x in the corresponding column of D. Then, for each corrupted item of training data x, we solve the following sparse coding problem:

$$\min_{w_Q} \|w_Q\|_0 \qquad \text{subject to} \qquad \|Qx - Q\hat{D}w_Q\| \leq \epsilon. \tag{57}$$

The value of ϵ is set at 0.001, and we allow w_Q to have at most 32 non-zero coefficients by the ORMP algorithm. The reconstructed block $\hat{x}$ can be obtained from

$$\hat{x} = \hat{D}\hat{w}_Q, \tag{58}$$

where $\hat{w}_Q$ is the solution of Equation (57).

The performance of the learned dictionary $\hat{D}$ is measured by calculating the distortion, in terms of the *SNR*, between the original and the reconstructed images. In Figure 4, the original test image Hepburn (which is not one of the training images) is shown in the top row. The second row of Figure 4 shows the images that correspond to a percentage of corrupted pixels. From left to right, the percentages of missing pixels are 20%, 30%, 40%, 50%, 60%, and 70%. Figure 2 compares the *SNR*s. We observe that SNR gain of the **PMK-SVD** dictionary over the **K-SVD** dictionary ranges from 0.7 to 2.0 dB.

B. Image Compression

An image is encoded by dividing it into 8×8 disjoint blocks, which are represented by the columns of a matrix X. Then, the ORMP algorithm and the learned dictionary $\hat{D}$ are used to perform sparse approximation of each column in X. The derived coefficients form the coefficient matrix W.

An image is compressed by storing only the locations and values of the non-zero coefficients in W. Therefore, a compressed block can be represented by the following run-length structure:

$$\{Blocks\} = \{NumCoeff, Coeff_1, Run_1, Coeff_2, Run_2, ...\}, \tag{59}$$

where $NumCoeff$ represents the number of non-zero coefficients, $Coeff_i$ denotes the value of the i-th non-zero coefficient, and Run_i is the number of zeros between the i-th and the $i+1$-th non-zero coefficients. The non-zero coefficients are quantized by scalar quantization. Note that

the amount of quantization applied to the DC values is much smaller than that used for the other coefficients.

After the sparse representation and the run-length coding of a block, entropy coding is applied to the *NumCoeff*, *Coeff*, and *Run* in the coding structure in Equation (59). Huffman Coding is used for entropy coding, and three coding tables are constructed to represent the three kinds of coding symbols. The coding bitstream of the whole image is formulated as follows:

$$\{Image\} = \{ImageSize, NumBlocks, Block_1, Block_2, ...\}, \tag{60}$$

where *ImageSize* records the size of the image, *NumBlocks* is the total number of blocks, and $Block_i$ is the coded stream of the i-th block. The *rate* is the number of the bits used to represent the image with the above structure. We assume that the dictionary $\hat{D}$ used in the coding process is known to both the encoder and the decoder, so it does not occupy any bits in the coded stream.

At the receiver site, if we use $\hat{W}$ to denote the sparse coefficients after re-scaling the quantized coefficients, the reconstructed image, denoted by $\hat{X}$, can be obtained by

$$\hat{X} = \hat{D}\hat{W}. \tag{61}$$

The coding performance is measured by the *Peak Signal-to-Noise Ratio (PSNR)* where the *Mean-Squared-Error (MSE)* is derived from the difference between the original X and the reconstructed images $\hat{X}$.

By applying different quantization steps to encode the sparse coefficients of the blocks, we can obtain the Rate-Distortion curve (R-D curve). Figure 3 shows the R-D curve for coding the image Hepburn by using different dictionaries. Compared to the K-SVD dictionary, the PMK-SVD dictionary can reduce each pixel by 0.01 to 0.1 bits with approximately the same *PSNR* value.

5 Concluding Remarks

The state-of-the-art dictionary updating algorithms do not impose a constraint on the derived dictionary; thus, the variation in the sequence of derived dictionaries can be large. This may create the overshoot problem in the neighborhood of the optimal dictionary and slow down the convergence. We impose a regularization term on dictionary modifications to overcome the problem. We formulate the approach as the proximal point method and have successfully incorporated the method into the MOD and K-SVD dictionary updating methods. The proximal point-based MOD and K-SVD updating algorithms are called P-MOD and PK-SVD respectively. We also show that the algorithms can be combined to form a hybrid dictionary updating method called PMK-SVD. Theoretically, the derived method can converge and obtain an optimal dictionary. In the experiments, we compared the performance of the dictionaries on the following

applications: sparse approximation of signals and images, filling the missing pixels in an image, and image compression. In all cases, the dictionary derived by the PMK-SVD updating method outperformed those constructed by the compared methods. In our future work, we will extend the proposed method to learn the analytical dictionary proposed in [31].

References

[1] S. Mallat, *A wavelet tour of signal processing.* Academic Press, 1998.

[2] M. Elad, *Sparse and Redundant Representations - From Theory to Applications in Signal and Image Processing.* Springer, 2010.

[3] T. Blumensath and M. E. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 298–309, April 2010.

[4] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, December 1993.

[5] Y. C. Pati, R. Rezaiifar, Y. C. P. R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers*, 1993, pp. 40–44.

[6] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, December 2005.

[7] M. Elad, "Why simple shrinkage is still relevant for redundant representations?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5559–5569, December 2006.

[8] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, November 2004.

[9] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *Proceedings of the 24 th IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, 1999, pp. 2443–2446.

[10] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, November 2006.

[11] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2121–2130, April 2010.

[12] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 14, no. 5, pp. 877–898, May 1976.

[13] I. Y. Kenneth Lange, David R. Hunter, "Optimization transfer using surrogate objective functions," *Journal of Computational and Graphical Statistics*, vol. 9, no. 1, pp. 1–20, Janurary 2000.

[14] M. A. T. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak, "Majorization-minimization algorithms for wavelet-based image restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2980–2991, December 2007.

[15] M. Elad, B. Matalon, and M. Zibulevsky, "Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization," *Applied and Computational Harmonic Analysis*, vol. 23, no. 3, pp. 346 – 367, March 2007.

[16] T. Honkela, W. Duch, M. A. Girolami, and S. Kaski, "Artificial neural networks and machine learning - icann 2011 - 21st international conference on artificial neural networks, espoo, finland, june 14-17, 2011, proceedings, part i," in *ICANN (1)*, ser. Lecture Notes in Computer Science, vol. 6791. Springer, 2011.

[17] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, Janurary 2009.

[18] Y. Nesterov, "Gradient methods for minimizing composite functions," *Mathematical Programming*, vol. 140, no. 1, pp. 125–161, Janurary 2013.

[19] S. Ji and J. Ye, "An accelerated gradient method for trace norm minimization," in *Proceedings of the 26 th International Conference on Machine Learning*, 2009, pp. 457–464.

[20] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for sparse hierarchical dictionary learning," in *Proceedings of the 27 th International Conference on Machine Learning*, 2010, pp. 487–494.

[21] ——, "Proximal methods for hierarchical sparse coding," *Journal of Machine Learning Research*, vol. 12, pp. 2297–2334, 2011.

[22] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *Annals of Statistics*, vol. 37, no. 6A, pp. 3468–3497, June 2009.

[23] R. Baraniuk, E. Candes, M. Elad, and Y. Ma, "Applications of sparse representation and compressive sensing," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 906–909, June 2010.

[24] J. C. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *Journal of Machine Learning Research*, vol. 10, pp. 2899–2934, 2009.

[25] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, Janurary 2008.

[26] K. Skretting and K. Engan, "Image compression using learned dictionaries by RLS-DLA and compared with K-SVD," in *International Conference on Acoustics, Speech, and Signal Processing*, 2011, pp. 1517–1520.

[27] G. H. Golub and C. F. Van Loan, *Matrix computations (3rd ed.)*. Johns Hopkins University Press, 1996.

[28] S. Chen and J. Wigger, "Fast orthogonal least squares algorithm for efficient subset model selection," *IEEE Transactions on Signal Processing*, vol. 43, no. 7, pp. 1713–1715, July 1995.

[29] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, February 1995.

[30] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 689–696.

[31] T. P. R. Rubinsten and M. Elad, "Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model," *IEEE Trans. on Signal Processing*, pp. 661–677, February 2013.

Table 1: The One-Atom-PK-SVD dictionary updating method

Algorithm	One-Atom-PK-SVD($\tilde{E}$,d^{old},λ_s)	
Input	$\tilde{E} \in R^{N \times \frac{SL}{K}}$: see Equation (26) $d^{old} \in R^N$: the atom from the previous proximal iteration $\lambda_s \in R$: the Lagrangian multiplier	
Output	$d^{new} \in R^N$: the atom for the current proximal iteration $x^{new} \in R^{N \times K}$: the coefficients for the current proximal iteration	
Step	1. Perform SVD on $\tilde{E} := U\Delta V^T$ (Equation (40)).	**Complexity**
	$d_o \leftarrow v_1$ (v_1 is the first column of U).	$O(N^2(\frac{SL}{K})$
	$\hat{x} \leftarrow v_1$ (v_1 is the first column of U).	$+N(\frac{SL}{K})^2$
	$\mu \leftarrow \sigma_1^2$ (σ_1^2 denotes $\Delta(1,1)^2$).	$+(\frac{SL}{K})^3)$
	$d^{new} \leftarrow d_o$	
	$x^{new} \leftarrow \mu\hat{x}$	
	If $\lambda = 0$, return d^{new} and x^{new};	
	2. Estimate the norm of x (Equation(41)):	$O(N^2)$
	$\|x_o\|^2 \leftarrow \sum \sigma_i^2 (d_o^T v_i)^2$,	
	where σ_i are singular values of $\tilde{E}$, and v_i are the columns of U.	
	3. Update the norm of x (Equation (45)):	$O(N(\frac{SL}{K}))$
	$\|x^{new}\| \leftarrow \begin{cases} \|x_o\|, & \text{if } \|x_o\|^2 + \lambda_s - \mu = 0, \text{ or} \\ \frac{\lambda_s \|\tilde{E}^T d^{old}\|}{\|x_o\|^2 + \lambda_s - \mu} & \text{otherwise.} \end{cases}$	
	4. Derive Pd^{new} (Equation (37)) and x^{new}:	
	Perform SVD to obtain the pseudo inverse $\tilde{E}^{T+}$.	$O(N(\frac{SL}{K})^2$
		$+N^2(\frac{SL}{K})$
		$+N^3)$
	$P \leftarrow \tilde{E}^{T+}\tilde{E}^T$.	$O(N^2(\frac{SL}{K}))$
	$Pd^{new} \leftarrow \frac{1}{\lambda s}(\mu\|x^{new}\| - \|x^{new}\|^3)\hat{E}^{T+}\hat{x} + Pd^{old}$;	$O(N^2)$
	$x^{new} \leftarrow \|x^{new}\|\hat{x}$.	$O(\frac{SL}{K})$
	5. Update the atom d^{new} as follows:	
	$d^{new} \leftarrow \begin{cases} d^{new} - \frac{(\|x^{new}\|^2 + \lambda)d^{new} - \tilde{E}x^{new} + \lambda d^{old}}{\|x^{new}\|^2 + \lambda} & \text{if } P = I, \text{ or} \\ Pd^{new} + (I - P)d^{old} & \text{otherwise.} \end{cases}$	$O(N^2)$
	6. If $\|d^{new} - d_o\| > \epsilon_2$, let $d_o \leftarrow d^{new}$ and go to Step 2;	
	else return d^{new} and x^{new}.	

Table 2: The PK-SVD dictionary updating method

Algorithm	PK-SVD(D^0,W^0,Y,λ_s)	
Input	$D^0 \in R^{N\times K}$: the original dictionary $W^0 \in R^{K\times L}$: coefficients from the sparse coding step $Y \in R^{N\times L}$: training signals $\lambda_s \in R$: Lagrangian multiplier	
Output	$D \in R^{N\times K}$: the updated dictionary $W \in R^{N\times K}$: the updated coefficients	
Step	1. $n \leftarrow 1$;	**Complexity**
	2. **Loop** : For $k \leftarrow 1$ to K, (update d_k^{n-1}, the k-th column of D^{n-1}).	
	2.1. Obtain Ω_k^{n-1} and $\tilde{E}_k^{n-1}$ (Equations (25) and (26)).	$O(KLN)$
	2.2. $[d_k^n, x_k^n] \leftarrow$One-Atom-PK-SVD($\tilde{E}_k^{n-1}, d_k^{n-1}, \lambda_s$).	$O(N^2(\frac{SL}{K})$ $+N(\frac{SL}{K})^2 +(\frac{SL}{K})^3)$
	2.3. Let the k-th column of D^n be d_k^n and let the coefficients in W^n with support on Ω_k^{n-1} be x_k^n.	$O(N + (\frac{SL}{K}))$
	2.4. **End of Loop**	
	3. If $\|\|D^n - D^{n-1}\|\|_F \leq \epsilon$ or $n \geq maxiter$, return D^n and W^n ($maxiter$ is an integer specified by the users).	
	4. $n \leftarrow n + 1$; go to Step 2.	

Table 3: The **PK-SVD** dictionary learning algorithm

Algorithm	**PK-SVD-Learning**
Input	$D \in R^{N\times K}$: the original dictionary $Y \in R^{N\times L}$: the training signals $\lambda_s \in R$: the Lagrangian multiplier
Output	$D \in R^{N\times K}$: the updated dictionary $W \in R^{N\times K}$: the updated coefficients
Step	**Loop**: until some stopping conditions are satisfied. 1. Sparse coding : use Y and D to derive sparse solution W. 2. Perform dictionary updating with the PK-SVD method $[D, W] \leftarrow$ PK-SVD(D,W,Y,λ_s). **End of Loop** 3. Return D and W.

Table 4: The hybrid dictionary learning algorithm derived by combine the P-MOD and PK-SVD dictionary updating methods.

Algorithm	**PMK-SVD-Learning**
Input	$D \in R^{N\times K}$: the original dictionary $\lambda_p \in R$: Lagrangian multiplier of the P-MOD method $\lambda_s \in R$: Lagrangian multiplier of the PK-SVD method $Y \in R^{N\times L}$: the training signals
Output	$D \in R^{N\times K}$: the learned dictionary $W \in R^{N\times K}$: the learned coefficients
Step	**Loop**: until some stopping conditions are satisfied. 1. Sparse coding : use Y and D to derive the sparse solution W. 2. Dictionary updating by the P-MOD method (Equation (18) and λ_p) : perform $maxiter1$ times of dictionary updating step with D as the initial dictionary ($maxiter$ is an integer specified by the users). Let D_{pmod} be the resultant dictionary. 3. Perform dictionary updating by the PK-SVD method $[D, W] \leftarrow$ PK-SVD(D_{pmod},W,Y,λ_s). 4. Reduce the dictionary search region of the P-MOD method : reduce the value of λ_p; **End of Loop** 5. Return D and W.

Table 5: Comparison of the average dictionary recovery ratios of different methods. Columns 5 and 6 show, respectively, the average recovery ratio gains of **PK-SVD** and **PMK-SVD** over **K-SVD**.

Noise Level (dB)	**ODL** Recovery Ratio (%)	**RLS** Recovery Ratio (%)	**K-SVD** Recovery Ratio (%)	**PK-SVD** Recovery Ratio Gain (%)	**PMK-SVD** Recovery Ratio Gain (%)
5	15.20	14.14	60.20	+1.90	+3.10
10	56.82	56.82	83.20	+0.48	+1.94
15	69.48	69.16	88.04	+0.50	+2.36
20	73.50	73.48	89.78	+0.88	+1.72
Noise-less	72.34	73.20	90.76	+0.52	+2.40

Table 6: The SNR gains of **PK-SVD** and **PMK-SVD** over **K-SVD** on signals generated by the uniform or normal distribution; the number of training signals is 2000; S is the sparsity level; and 30 iterations are performed to learn a dictionary.

Type	S	**KSVD** SNR(dB)	**PK-SVD** SNR Gain	**PMK-SVD** SNR Gain	**RLS** SNR Gain	**ODL** SNR Gain
Uniform	4	13.186	0.004	**0.020**	-0.078	-0.080
Uniform	6	17.327	0.012	**0.051**	-0.190	-0.189
Uniform	8	22.151	0.121	**0.301**	-0.276	-0.456
Normal	4	7.822	0.013	**0.047**	-0.133	-0.154
Normal	6	12.107	0.007	**0.090**	-0.153	-0.227
Normal	8	16.980	0.198	**0.223**	-0.430	-0.577

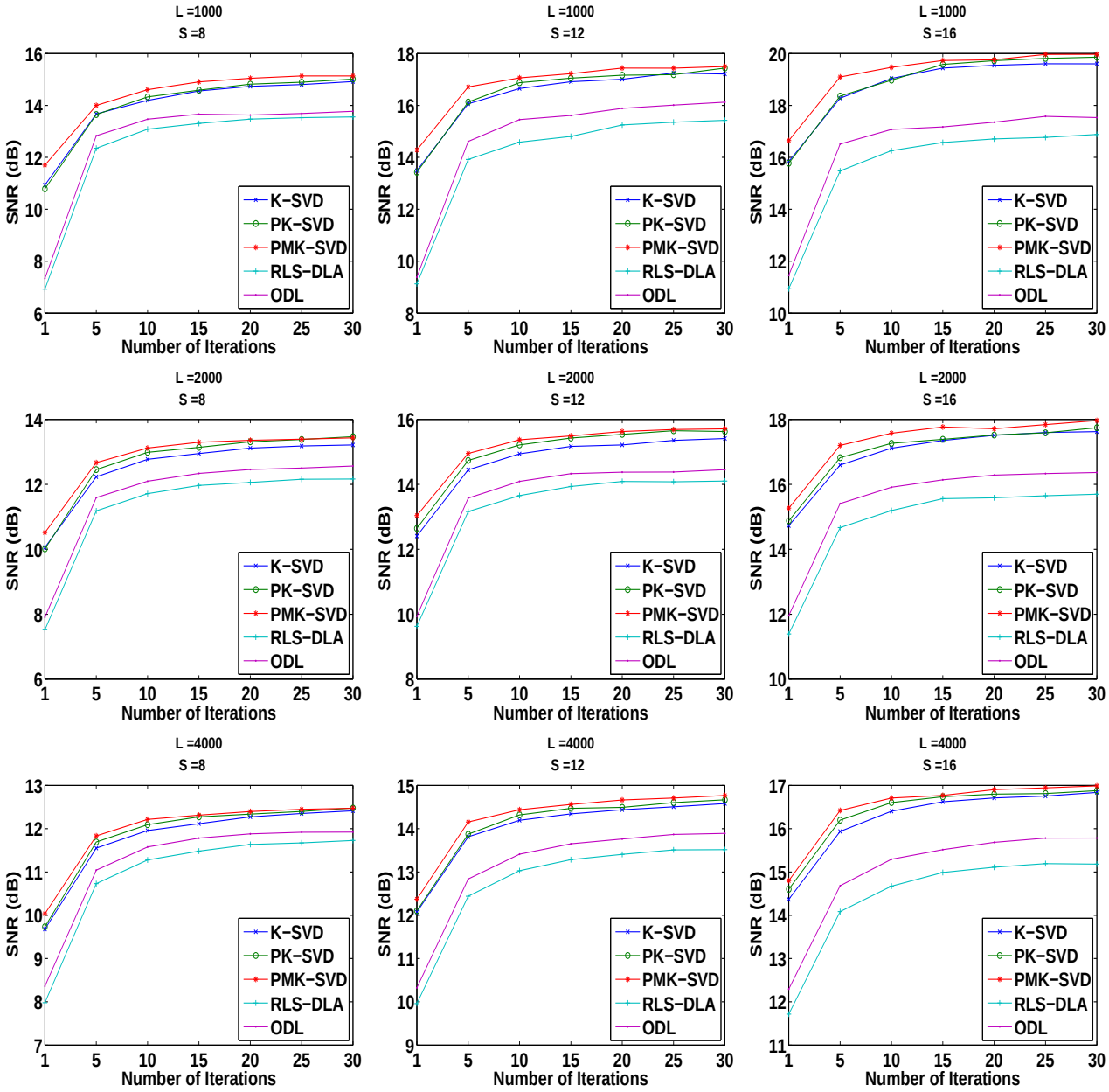

Figure 1: Comparison of the performance of sparse approximation of image blocks. The training data comprises 1000 blocks (top row), 2000 blocks (middle row), and 4000 blocks (bottom row) selected at random from the images in Figure **??**. The length of each signal is 64, and the number of iterations is set at 30. The sparsity levels are set at 8(left-hand column), 12(middle column), and 16(right-hand column). The **PMK-SVD** method achieves the best performance in all iterations.

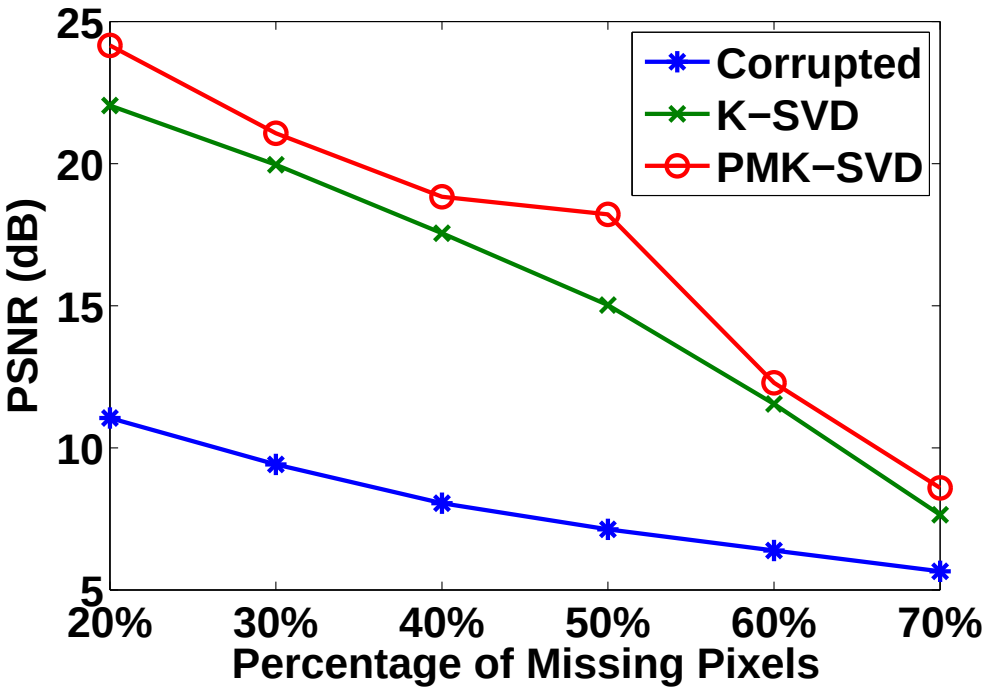

Figure 2: Comparison of the PSNRs when the dictionaries learned by K-SVD and PMK-SVD are used to recover the missing pixels of the Hepburn image (Figure 4). The curve labeled "Corrupted" corresponds to the PSNR of the image with the missing pixels.

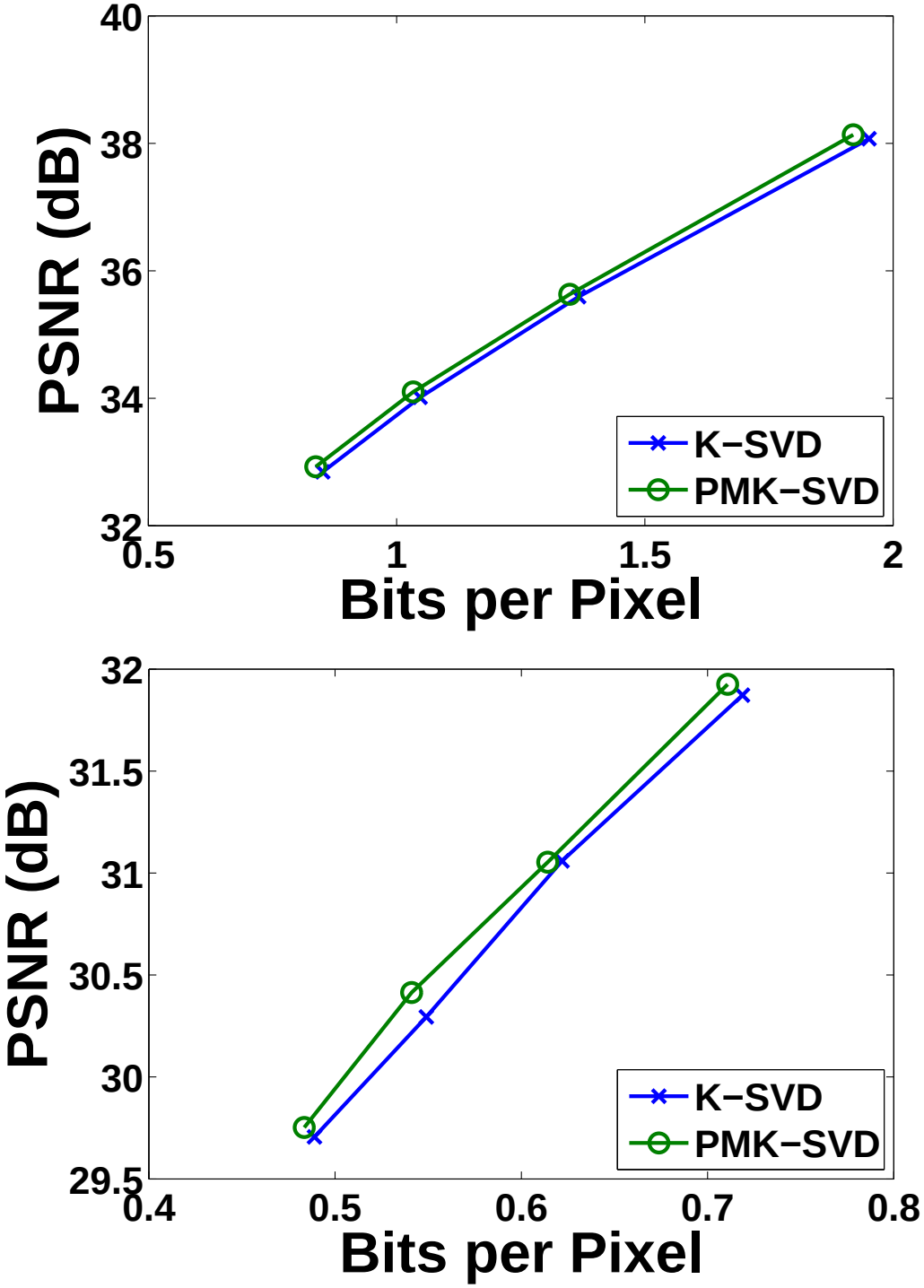

Figure 3: Comparison of the R-D curves obtained by the dictionaries learned by K-SVD and PMK-SVD for the Hepburn image. Top: at a low bit rate between 0.7 and 2 bits per pixel; Bottom: at a very low bit rate between 0.5 and 0.7 bits per pixel.

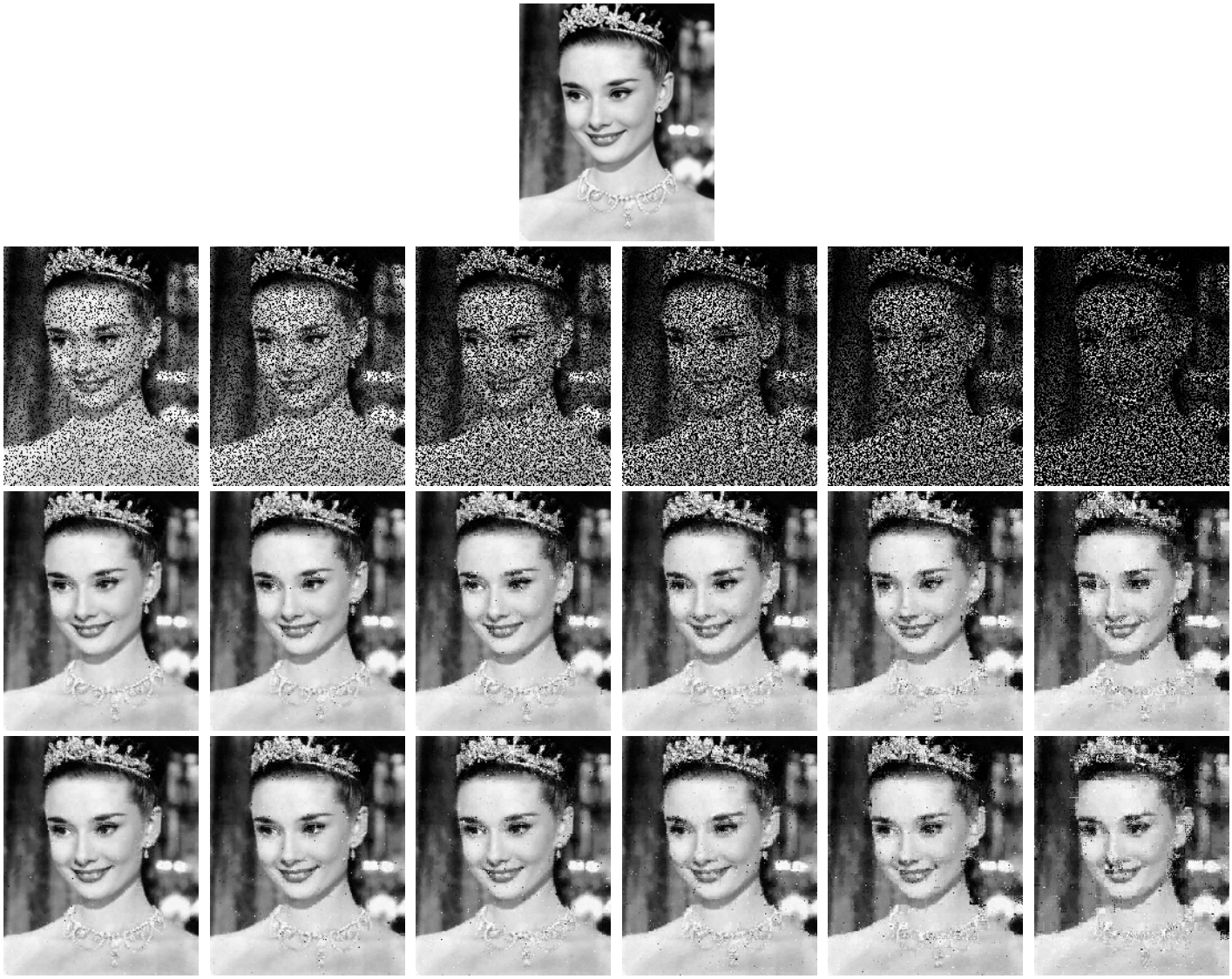

Figure 4: Filling the missing pixels of the Hepburn Image. Top: the original image. Second row from left to right: the percentages of the missing pixels are 20%, 30%, 40%, 50%, 60%, and 70%. The images in the third and fourth rows of each column are reconstructed by using the **K-SVD** and the **PMK-SVD** learning methods respectively for the corrupted image in the same column.