Supervised Online Dictionary Learning for Image Separation Using OMP

Yuxin Zhang and Bo Yuan

Intelligent Computing Lab, Division of Informatics
Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, P.R. China
zhang_yuxin@foxmail.com,yuanb@sz.tsinghua.edu.cn

Abstract. In this paper, we propose a new algorithm to perform single image separation based on online dictionary learning and orthogonal matching pursuit (OMP). This method consists of two separate processes: dictionary training for representing morphologically different components and the separation stage. The training process takes advantage of the prior knowledge of the components by adding component recovery error control penalties. The learned dictionaries have lower coherence with each other and better separation ability, which can benefit the separation process in two ways. Firstly, simple sparse coding methods such as OMP can be used to efficiently obtain superior performance. Secondly, well trained dictionaries can lead to satisfactory separation results even when the components are similar. The dictionaries obtained can also serve as good initial inputs for other models using dictionary learning and sparse representation. Experiments on complex images confirm that better results can be achieved efficiently by our method compared to other state-of-the-art algorithms.

Keywords: Online Dictionary Learning, Image Separation, Morphological Component Analysis, OMP

1 Introduction

Data separation is a kind of fundamental transformation with wide applications in fields such as cosmology, geography and biomedical engineering. As a branch of data separation, the purpose of image separation or decomposition is to identify different components in a single image. A typical objective is to efficiently extract and separate texture and cartoon components mixed in the same image. Single image decomposition has been successfully applied in image processing for tasks such as rain moving from a picture [1] and reflection separation [2].

There are a number of existing methods in the literature that can find a proper separation in different ways. Some of them formulate the problem as matrix factorization and resort to techniques in the field of linear algebra for solutions [3]. Others use algorithms based on numerical methods and variational models [4, 5]. There are also prior-based models such as morphological components analysis（MCA) [6]. Most of these algorithms can get reasonable results, but there is still room for improvement with respect to both the speed and performance of the decomposition.

Studies have shown that the sparse representation model can be applied to data separation [7]. The sparsity model has been widely used in the field of signal processing due to its elegant mathematical foundation and ability to depict the essence of natural signals [8]. The core idea of sparse representation is to transform the original signal into a linear combination of a small number of representation atoms. The set of such atoms is called a dictionary. A famous example is the morphological components analysis method [6] for decomposing signals into their building atoms. The success of decomposing mixed signals into desired components relies heavily on the proper choice of dictionaries used in the separation step. Dictionaries must be highly effective in representing their respective components, which means that the signal can be decomposed into a linear combination of a small number of atoms from the dictionary. Traditional dictionaries such as DCT transform, wavelet and shearlet [9] can be very efficient for the separation of certain components from a single signal mixture.

However, the ability of these designed dictionaries to transform signals into sparse representation and extract them from mixed images is limited. In order to separate morphologically distinct components, more flexible dictionaries are in need. To solve this problem, the idea of dictionary learning was introduced. Using dictionary learning to accomplish different signal processing tasks is very popular nowadays. Common dictionary learning methods include MOD [10], K-SVD [11] and online dictionary learning [12]. There are mainly two approaches to incorporating dictionary learning into the process of separation. The first one is to learn dictionaries for each component dataset and then use MCA or sparse coding methods to reconstruct the parts of images containing multiple features [13]. The other one is to combine dictionary learning and the separation process together [14, 15]. These two approaches have their own advantages and disadvantages, which make them suitable for different scenarios. Our method follows the idea of the first strategy and improves upon the performance of the K-SVD based separation method [13] by training better dictionaries.

Note that, in the second method, since the update of the dictionaries and the separation of data are achieved at the same time, the learned dictionaries are of little use for other separation tasks, even if the components are of similar patterns. In real-world applications, this feature can cause unnecessary computational redundancy. By contrast, employing two independent processes can improve the applicability of the dictionaries and reduce the separation time. However, using fixed dictionaries in the separation phase means that dictionaries need to be effective in representing their own components and have minimum correlations with each other. To tackle this challenge, we proposed a supervised dictionary learning method, which aims at learning dictionaries that are suitable for quick and successful separation using sparse coding.

Another issue when applying the second method is the requirement for proper initial dictionaries. Well initialized dictionaries can lead to faster convergence and better final results while poor initialization may cause total failure of the algorithm. In the context of data separation, not only should the initial dictionary have strong representation ability for its aiming component, it is also of great importance that dictionaries for different components are as incoherent as possible with each other. To achieve this objective, some researchers proposed to add incoherence based penalties to the loss function [16] which may be useful in some cases but will compromise the representation ability of the dictionaries. The algorithm proposed in this paper is able to obtain dictionaries that meet all the above requirements to serve as quality initial dictionaries.

2 Algorithm Outline

In this paper, we proposed a novel dictionary learning method for single image separation based on sparse representation. One important assumption of our algorithm is that there is a training dataset for each component. Without loss of generality, we only discuss the situation where the image is composed of two components.

The algorithm consists of two relatively separate processes. Firstly, we use supervised dictionary learning methods to train localized dictionaries for the components. Then the separation of images containing similar components is performed using the learned dictionaries via sparse coding algorithms. In this paper, we used OMP [17] to obtain sparse representations. The problem of decomposing an image into different components can be formulated as:

$$\mathrm{f} = u_1 + u_2 \tag{1}$$

where f represents the mixture of component vectors $u_1,\ u_2 \in \mathbb{R}^m$. In image processing, signals are represented by vectors rearranged from small patches extracted from the original image by either distinct or sliding style. In sparse representation, we assume that each component can be represented sparsely using dictionaries. This leads to the following equation:

$$\mathrm{f} = D_1\alpha_1 + D_2\alpha_2 = [D_1\ D_2]\begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix} \tag{2}$$

where $D_1, D_2\ \in \mathbb{R}^{m*n}$ are dictionaries and $\alpha_1, \alpha_2 \in \mathbb{R}^n$ are sparse representations of components u_1, u_2.

In the learning phase, we use the online dictionary learning method to find the dictionaries in eq. (2). The main contribution of our algorithm is the introduction of the component recovery error control into the original objective function (i.e., use supervision to improve the separation ability of the dictionaries). Suppose β_1 and β_2 are the solutions of the following problems:

$$\{\beta_1, \beta_2\} \triangleq \underset{\beta_1,\beta_2}{\operatorname{argmin}} \parallel \mathrm{f} - [D_1\ D_2]\begin{bmatrix}\beta_1\\ \beta_2\end{bmatrix} \parallel_2^2, \mathrm{s.t.} \parallel \beta_1 \parallel_0 + \parallel \beta_2 \parallel_0 < \mu \tag{3}$$

where $\mathrm{f} = u_1 + u_2 + e$, with e representing the noise. Ideally, the reconstructed components should satisfy the following constraints:

$$\parallel u_1 - D_1\beta_1 \parallel_2^2 < \varepsilon_1 \tag{4}$$

$$\parallel u_2 - D_2\beta_2 \parallel_2^2 < \varepsilon_2 \tag{5}$$

where $\varepsilon_1, \varepsilon_2$ are error controlling parameters. To achieve this goal, the objective function of the algorithm has to be modified as follows:

$$\{\widehat{D^1}, \widehat{D^2}, \widehat{\alpha^1}, \widehat{\alpha^2}\} = \underset{D^1, D^2, \alpha^1 \alpha^2}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} (\| f^i - D^1 \alpha_i^1 - D^2 \alpha_i^2 \|_2^2 + \lambda_1 \| u_i^1 - D^1 \alpha_i^1 \|_2^2 + \lambda_2 \| u_i^2 - D^2 \alpha_i^2 \|_2^2), \text{s.t.} \| \alpha^1 \|_0 + \| \alpha^2 \|_0 < \mu. \quad (6)$$

In eq. (6), λ_1 and λ_2 are the supervision factors, which control the recovery errors of different components. They can be set to different values or kept identical, depending on specific applications. With the ℓ_0 norm constraint, this problem is non-convex. The classical solution is to alternately update between the dictionaries and sparse representations. Here, we solve the above problem by alternately updating the dictionaries and sparse coefficients based on the online dictionary learning method [12] and OMP [17]. The framework of the training process is shown by Algorithm 1.

In the separation stage, the input is the mixed image and learned dictionaries for components of the image. Simple sparse coding algorithms such as OMP can be applied to find the sparse coefficients of the mixture using the combined dictionary. To reconstruct the original components, one just needs to multiply the local dictionaries with their corresponding coefficients and then transform the vectors back to patches. The details of the proposed algorithm are demonstrated in the next section.

Algorithm 1: Supervised online dictionary learning for image decomposition

Input: data set $X_1 \in \mathcal{R}^{n*m}$, $X_2 \in \mathcal{R}^{n*m}$, parameters $\lambda_1, \lambda_2, \mu_1, \mu_2$

Initialization: randomly select data samples from X_1, X_2 to initialize D^1, D^2

$A_0^1 \leftarrow \mathcal{O}$, $B_0^1 \leftarrow \mathcal{O}$, $C_0^1 \leftarrow \mathcal{O}$, $D_0^1 \leftarrow \mathcal{O}$, $A_0^2 \leftarrow \mathcal{O}$, $B_0^2 \leftarrow \mathcal{O}$, $C_0^2 \leftarrow \mathcal{O}$, $E_0^2 \leftarrow \mathcal{O}$

For t=1 to T do:

Update the coefficients:

Draw x_t^1 from $p(X_1)$ and x_t^2 from $p(X_2)$

Create the mixed signal: $f_t = x_t^1 + x_t^2$

Sparse coding using OMP:

$$\{\alpha_t^1, \alpha_t^2\} \triangleq \underset{\alpha^1, \alpha^2}{\operatorname{argmin}} \frac{1}{2} \| f_t - D^1 \alpha^1 - D^2 \alpha^2 \|_2^2, s.t. \| \alpha^1 \|_0 + \| \alpha^2 \|_0 < \mu$$

$A_t^1 \leftarrow A_{t-1}^1 + \alpha_t^1 {\alpha_t^1}^T$, $B_t^1 \leftarrow B_{t-1}^1 + x_t^1 {\alpha_t^1}^T$,

$C_t^1 \leftarrow C_{t-1}^1 + \alpha_t^2 {\alpha_t^1}^T$, $E_t^1 \leftarrow E_{t-1}^1 + f_t {\alpha_t^1}^T$,

$A_t^2 \leftarrow A_{t-1}^2 + \alpha_t^2 {\alpha_t^2}^T$, $B_t^2 \leftarrow B_{t-1}^2 + x_t^2 {\alpha_t^2}^T$,

$C_t^2 \leftarrow C_{t-1}^2 + \alpha_t^1 {\alpha_t^2}^T$, $E_t^2 \leftarrow E_{t-1}^2 + f_t {\alpha_t^2}^T$

Update D_t^1, D_t^2 by Algorithm 2, using D_{t-1}^1, D_{t-1}^2 as warm start:

$$\{D_t^1, D_t^2\} \triangleq \underset{D^1, D^2}{\operatorname{argmin}} \frac{1}{t} \sum_{i=1}^{t} \frac{1}{2} \| f_i - D^1 \alpha_i^1 - D^2 \alpha_i^2 \|_2^2 + \lambda_1 \| x_i^1 - D^1 \alpha_i^1 \|_2^2 + \lambda_2 \| x_i^2 - D^2 \alpha_i^2 \|_2^2$$

End for

Output: dictionaries D^1, D^2.

3 Implementation Details

3.1 Data Separation Using Sparse Coding Algorithm

In data separation using sparse representation, it is critical that the signal can be represented sparsely. If no limit is put on the number of atoms used by the representation, the process may fail to find parts that belong to different sources. In consideration of this, we need to precisely control the sparsity, leading to the choice of OMP. Also， it has been proved [13] that OMP is better than MCA for separation. The idea of OMP is, in each iteration, to find the atom that is most correlated to the residual and project the residual to the space spanned by the chosen atoms while making the remaining part orthogonal to that space. Details of the algorithm can be found in [17]. In both the training and the separation stages, OMP is applied to find the best representation of the mixture against the combined dictionary:

$$\{\alpha_t^1, \alpha_t^2\} \triangleq \underset{\alpha_t^1, \alpha_t^2}{\operatorname{argmin}} \frac{1}{2} \parallel f_t - [D^1 \; D^2] \begin{bmatrix} \alpha_t^1 \\ \alpha_t^2 \end{bmatrix} \parallel_2^2, s.t. \parallel \alpha_t^1 \parallel_0 + \parallel \alpha_t^2 \parallel_0 < \mu \tag{7}$$

After the representation is retrieved, it is used to update the dictionary atoms in the training stage. In the separation phase, it is for reconstructing the morphologically distinct components $\widehat{u_1}, \widehat{u_2}$.

For different situations, the parameter μ must be chosen accordingly to obtain good performance. Some separation related theories have been proposed to reveal the relation among the incoherence of dictionaries, the sparsity level and the separation error. Generally, the lower the coherence and the higher the sparsity, the better the separation results [18, 19]. Since coherence is more difficult to control, a relatively small μ as the sparsity controller is recommended to get good results.

3.2 Dictionary Training

Proper datasets are essential for training a pair of dictionaries for the two components. In image decomposition, it means that we need some images containing a single pattern from which patches can be drawn to form vectors as data samples. The training requires three datasets, two for the two components and one for mixed signals. The latter one is always generated during the training using the former two datasets.

During each training session, the first step is to find the sparse representation of the mixture using OMP. Then, we use an online dictionary learning method similar to [12] to update the dictionaries. The objective function is as follows:

$$\{\widehat{D^1}, \widehat{D^2}\} \triangleq \underset{D^1, D^2}{\operatorname{argmin}} \frac{1}{t} \sum_{i=1}^{t} (\frac{1}{2} \parallel f_i - D^1 \alpha_i^1 - D^2 \alpha_i^2 \parallel_2^2 + \lambda_1 \parallel x_i^1 - D_1 \alpha_i^1 \parallel_2^2 + \\ \lambda_2 \parallel x_{\mathrm{i}}^2 - D_2 \alpha_{\mathrm{i}}^2 \parallel_2^2) \tag{8}$$

Adopting the update process in [12], we use block-coordinate descent with warm starts to update each column of the dictionaries alternately. For example, when updating D^1, with the other dictionary fixed, the objective function becomes:

$$\{\widehat{D^1}\} \triangleq \underset{D^1}{\operatorname{argmin}} \frac{1}{t} \sum_{i=1}^{t} (\frac{1}{2} \parallel r^2 - D^1 \alpha_i^1 \parallel_2^2 + \lambda_1 \parallel x_i^1 - D^1 \alpha_i^1 \parallel_2^2) \tag{9}$$

where $\mathrm{r}^2 = f_i - D^2 \alpha_i^2$ is the representation residual. Using simple algebraic calculation we have:

$$\{\widehat{D^1}\} \triangleq \underset{D^1}{\operatorname{argmin}} \frac{1}{t} ((\frac{1}{2} + \frac{\lambda_1}{2}) Tr(D^{1^T} D^1 A_t^1) - Tr(D^{1^T} E_t^1) + Tr(D^{1^T} D^2 C_t^1) - \lambda_1 Tr(D^{1^T} B_t^1)) \tag{10}$$

where $A_t^1 = \sum_{i=1}^{t} \alpha_i^1 \alpha_i^{1^T}, B_t^1 = \sum_{i=1}^{t} x_i^1 \alpha_i^{1^T}, C_t^1 = \sum_{i=1}^{t} \alpha_i^2 \alpha_i^{1^T}, E_t^1 = \sum_{i=1}^{t} f_i \alpha_i^{1^T}$. By calculating the first-order derivatives of eq. (10), it is clear that the j^{th} atom d_j of dictionary D^1 can be updated by:

$$\mathrm{u}_j \leftarrow \frac{1}{\theta} (\lambda_1 b_j + e_j - (1 + \lambda_1) D^1 a_j - D^2 c_j) + d_j \tag{11}$$

where a_j, b_j, c_j, e_j represent the j_{th} column of auxiliary matrices $A_t^1, B_t^1, C_t^1, E_t^1$, respectively. The same update procedure can be applied to D_2. Thus, the dictionaries can be updated by the algorithm shown below:

Algorithm 2: Dictionary Update

Input: $A_t^1, B_t^1, C_t^1, E_t^1, A_t^2, B_t^2, C_t^2, E_t^2, D_{t-1}^1, D_{t-1}^2, \lambda_1, \lambda_2, \mu_1, \mu_2$,θ.
Repeat:
  For j=1 to k do:
    Update the j_{th} atom of D^1 using eq. (11):
    $\mathrm{u}_j \leftarrow \frac{1}{\theta} (\lambda_1 b_j^1 + e_j^1 - (1 + \lambda_1) D_{t-1}^1 a_j^1 - D_{t-1}^2 c_j^1) + d_{t-1,j}^1$
    $d_{t,j}^1 \leftarrow \frac{1}{\max(\|u_j\|_2, 1)} u_j$.
  End for
  For
    Update the j_{th} atom of D^2 using eq. (11):
    $\mathrm{u}_j \leftarrow \frac{1}{\theta} (\lambda_2 b_j^2 + e_j^2 - (1 + \lambda_2) D_{t-1}^2 a_j^2 - D_{t-1}^1 c_j^2) + d_{t-1,j}^2$
    $d_{t,j}^2 \leftarrow \frac{1}{\max(\|u_j\|_2, 1)} u_j$.
  End for
Until convergence
Return D_t^1, D_t^2.

3.3 Parameter Selection

The success of the algorithm relies on choosing proper parameter values. One of the classical challenges frequently encountered in dictionary learning is how to choose the size of the dictionary and the sparsity constraint parameter. Unfortunately, there is no existing theoretical guidance for parameter selection in data separation using dictionary methods. Although for different separation cases, the optimal parameter values may vary a lot, some general rules can still be drawn from our systematic empirical studies for the selection of patch size, dictionary size, sparsity level, and the weight of the error control penalty.

1. Patch size: normally, smaller patches lead to better results in image denoising, restoration and so on. However, in image decomposition, the size of the patches should be adjusted according to the image components. In our experiments, we compared the results using patch sizes of 10×10 and 20×20.
2. Dictionary size: in sparse representation, over-complete dictionaries are used for better sparse representation. If *m* is the size of the dictionary and *n* is the dimension of the signal, then *m*/*n* is called the redundancy factor, which describes the over-completeness of the dictionary. Usually, the greater the redundancy factor, the higher the sparsity level of the representation, which in image compression tasks can be very appealing. However, in data separation, higher redundancy may cause higher correlation of the dictionaries. In our experiments, we used twice the dimension of the signals as the number of atoms for one dictionary. This setting of the redundancy factor has been shown to be suitable for most separation cases.
3. Sparsity level: as mentioned earlier, in the proposed algorithm, the sparsity of the transformation to the space of the learned dictionary has to be set in advance. It cannot be too high because the dictionary needs to be specialized for each component. In our experiments, the sparsity level was set to 10 and 20 for dictionaries with size of 100×200 and 400×800 respectively.
4. Error control coefficients (λ_1, λ_2): if these parameter values are too high, the learning of the dictionaries will be difficult to converge to a stable point. This is mainly because when the values of λ are too large, in the update stage, it will cause over-learning. That is, the dictionary atoms may be changed dramatically so that next time when the same signal is chosen, the atoms selected by OMP will be significantly different and the separation error cannot be reduced as expected. Using the line search method, we found that good dictionaries were learned with λ_1, λ_2 equal to 0.7 when the patch size was 10×10.

4 Experiments

We implemented the proposed algorithm using MATLAB 2014 with its core programs coded in C++ and tested it on Windows Server 2012 (64-bit version) with Intel Xeon CPU and 128GB RAM. We used adaptive MCA [14] and separation algorithms via dictionary learning algorithms including K-SVD [13] and online dictionary learning (ODL) [12] for comparison purpose. Three sets of experiments were conducted to show the effectiveness of our proposed algorithm from different perspectives.

4.1 Experiment 1

In this experiment, we show the general performance of our algorithm on separating two types of textures. We tested our algorithm on ten pairs of textures. Fully overlapping patches (10×10) were extracted from 300×300 images for training dictionaries with 200 atoms. During the separation process, 10 different mixtures of the same size consisting of similar patterns were used as the inputs. Experiment results measured by PSNR and FSIM are shown in Table 1 and Table 2, respectively and some of the sample images are shown in Figure 1. FSIM was a recently proposed powerful perceptual quality metric [20] for visual quality assessment. Each of the value was averaged over ten individual trials.

Table 1. Separation performance measured by PSNR

PSNR(dB)	Test-1	Test-2	Test-3	Test-4	Test-5	Test-6	Test-7	Test-8	Test-9	Test-10	Avg.
K-SVD	17.61	16.83	16.20	13.47	18.23	17.49	16.36	16.09	13.72	13.78	15.98
ODL	18.18	16.67	16.29	14.53	18.44	17.84	16.43	15.91	14.30	14.63	16.32
Adaptive MCA	16.50	14.75	14.73	13.54	15.55	16.51	15.14	14.18	13.71	14.09	14.87
Ours	18.38	17.00	16.69	15.03	18.44	18.73	16.85	16.02	14.23	15.39	16.68

Table 2. Separation performance measured by FSIM

FSIM	Test-1	Test-2	Test-3	Test-4	Test-5	Test-6	Test-7	Test-8	Test-9	Test-10	Avg.
K-SVD	0.7626	0.7767	0.7935	0.7654	0.7978	0.8179	0.7669	0.8183	0.7864	0.8209	0.7906
ODL	0.7542	0.7843	0.8075	0.7810	0.7958	0.8081	0.7641	0.8182	0.7849	0.8269	0.7925
Adaptive MCA	0.7209	0.7498	0.7800	0.7743	0.7355	0.7398	0.7472	0.7744	0.7672	0.8256	0.7615
Ours	0.7892	0.7891	0.8092	0.7949	0.7953	0.8398	0.7982	0.8125	0.7981	0.8179	0.8044

Table 3. Separation time using different methods with patch size 10×10

Time(s)	Test-1	Test-2	Test-3	Test-4	Test-5	Test-6	Test-7	Test-8	Test-9	Test-10	Avg.
Adaptive MCA	148.92	147.08	148.45	147.99	172.13	174.15	176.30	174.10	172.69	172.90	163.47
Ours	38.35	38.18	37.66	41.44	40.70	41.71	40.55	39.89	38.60	40.98	39.81

Table 4. Separation time using different methods with patch size 20×20

Time(s)	Test-1	Test-2	Test-3	Test-4	Test-5	Test-6	Test-7	Test-8	Test-9	Test-10	Avg.
Adaptive MCA	363.80	356.99	357.50	357.76	471.41	478.63	486.36	484.44	484.89	484.72	432.65
Ours	182.24	184.31	180.57	177.17	177.97	176.46	178.09	178.39	177.74	178.55	179.15

Table 1 and Table 2 show that the PSNR and FSIM values produced by our methods were higher than those by other algorithms in most of the cases and our algorithm achieved the highest average values. From Figure 1, we can see that the separated components using our algorithm contained less residual from the opposite components and provided better visual effect. Also, the dictionaries learned by the proposed algorithm had lower coherence levels compared to those learned without supervision.

Since the separation methods used in our algorithm, K-SVD and online dictionary learning based separation are identical, their separation times had no significant difference. In Table 3 and Table 4 we compared the time consumption by our algorithm and adaptive MCA with different patch sizes. It is clear that our algorithm took much less time compared to adaptive MCA. Although bigger patch means lower separation speed, the quality of the separated images is dramatically improved with the increase of patch size, which can be observed in Figure 2.

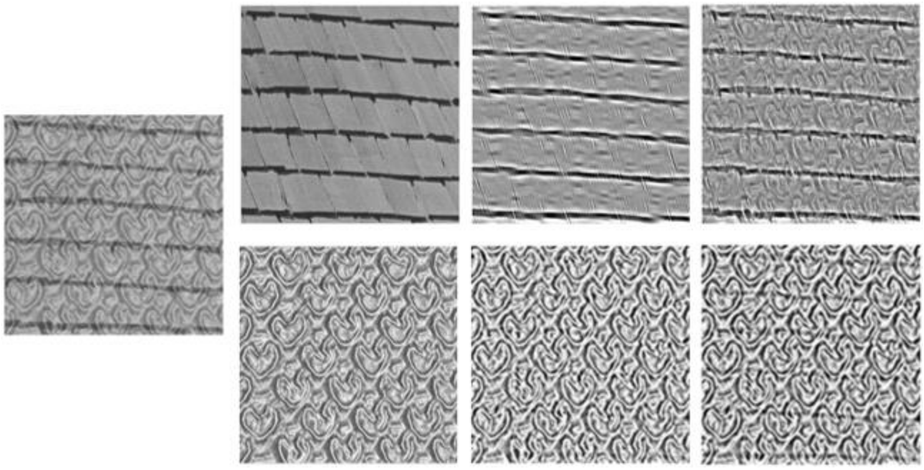

Fig. 1. Separation results on tile and heart-shape textures using our algorithm and K-SVD separation. From left to right: the mixture, the original components, the separated components using the proposed algorithm and K-SVD algorithm, respectively.

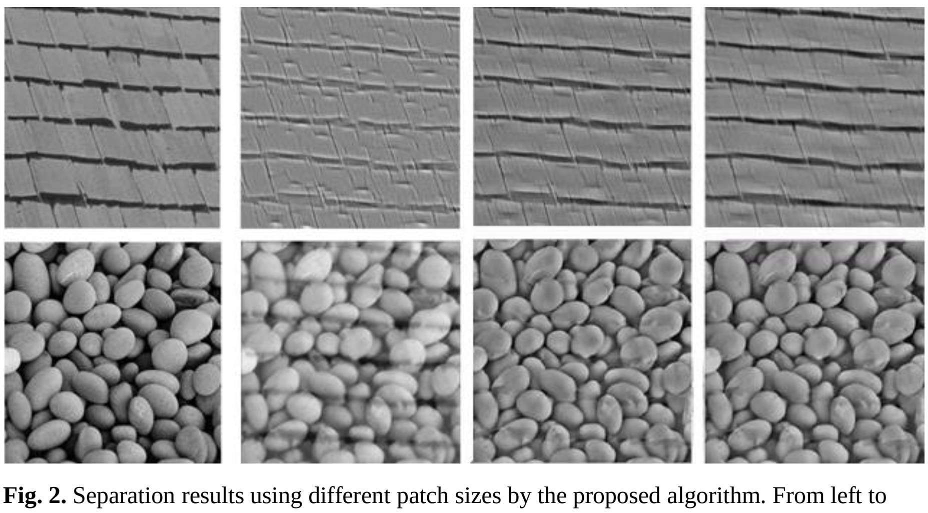

Fig. 2. Separation results using different patch sizes by the proposed algorithm. From left to right: the original components and the separated components using 10×10, 20×20, 30×30 patches, respectively.

4.2 Experiment 2

In this experiment, it is shown that using our learned dictionaries for algorithm initialization can lead to better separation performance of adaptive MCA. Experimental results are shown in Figure 3.

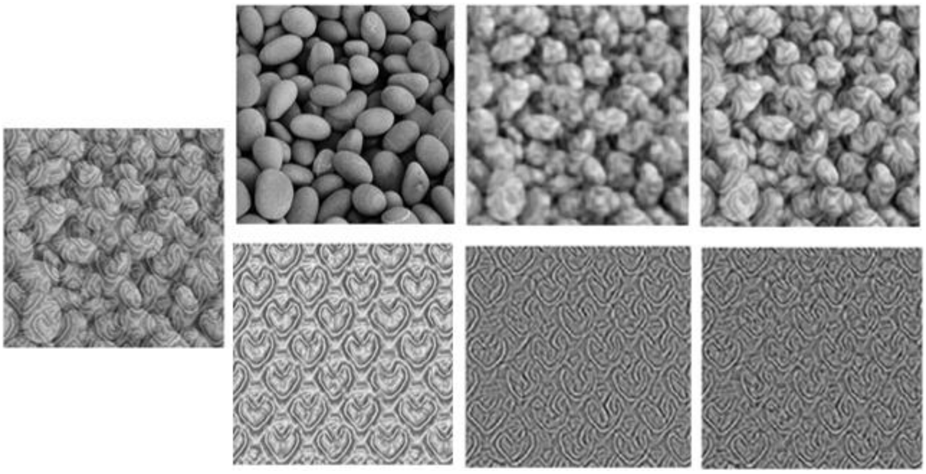

Fig. 3. Separation results using different initial dictionaries. From left to right: the mixture, the original components, and adaptive MCA separation results using learned dictionaries by our algorithm and by the K-SVD algorithm.

In the above results, the separation PSNR values were 15.56 (our algorithm) and 14.78 (K-SVD). It is clear that the initial dictionaries learned by our algorithm can lead to better individual components that are less contaminated by other components.

4.3 Experiment 3

Normally, higher overlapping rate can result in higher separation PSNR and better visual quality. However, high overlapping rate requires more computation, which can be very inconvenient in practice. For instance, the separation using full overlapping patches of a 256×256 picture with 8×8 patches will take about 60 times more time than using distinct patches. In our experiments, we show that our algorithm can still produce competitive results even when using distinct patches.

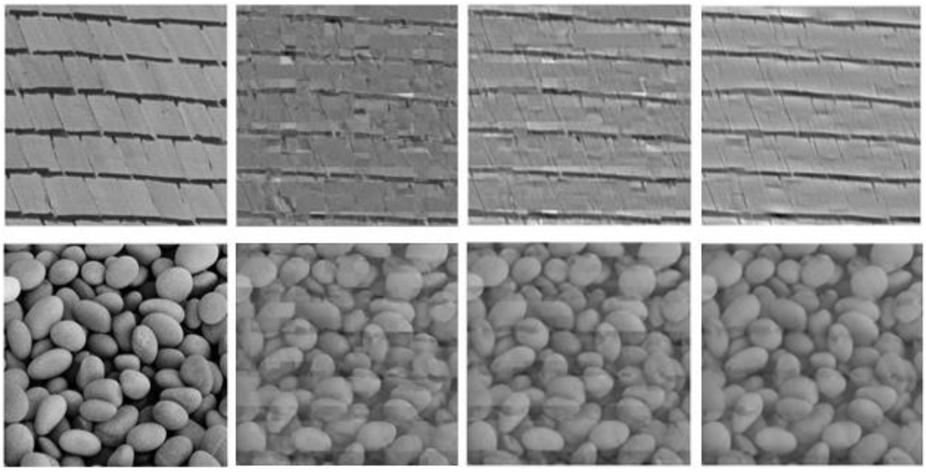

Fig. 4. Comparison of different extraction methods. From left to right: the original components, separation using K-SVD, our algorithm with distinct patches and separation using our algorithm with overlapping patches.

Table 5. Separation time using different patch extraction strategies

Time(s)	Test-1	Test-2	Test-3	Test-4	Test-5	Test-6	Test-7	Test-8	Test-9	Test-10	Avg.
Distinct	0.72	0.62	0.65	0.60	0.53	0.60	0.64	0.75	0.68	0.66	0.64
Sliding	38.35	38.18	37.66	41.44	40.70	41.71	40.55	39.89	38.60	40.98	39.81

Although in Figure 4 the quality with distinct patches was noticeably inferior to the quality with fully overlapping patches, the computational time shown in Table 5 was tremendously reduced, which can be very appealing in real world applications.

5 Conclusions

We proposed a new method to learn quality dictionaries for single image separation where training datasets are used to perform supervised online dictionary learning. Better trained dictionaries allow the use of simple sparse coding algorithms in the separation phase, which can greatly accelerate the separation process without compromising its performance. In addition, dictionaries learned by our method can also serve as the initial inputs for other methods using sparse representation to achieve better results. Furthermore, our algorithm can produce reasonable results using distinct patches, reducing the separation time greatly. In the future, we will conduct in-depth theoretical analysis of our algorithm and further accelerate the training process.

References

1. Luo, Y., Xu, Y. and Ji, H.: Removing Rain from a Single Image via Discriminative Sparse Coding. In: IEEE International Conference on Computer Vision, pp. 3397–3405 (2015)

2. Kong, N., Tai, Y. and Shin, J.: A Physically-based Approach to Reflection Separation: From Physical Modeling to Constrained Optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(2), pp. 209–221 (2014)
3. King, B. and Atlas, L.: Single-Channel Source Separation Using Complex Matrix Factorization. IEEE Transactions on Audio, Speech and Language Processing, 19(8), pp. 2591–2597 (2011)
4. Hao, Y., Xu, J., Bai, J. and Han, Y.: Image Decomposition Combining a Total Variational Filter and a Tikhonov Quadratic Filter. Multidimensional Systems and Signal Processing, 26(3), pp. 739–751 (2015)
5. Wang, G., Pan, Z., Zhao, Z. and Sun, X.: The Split Bregman Method of Image Decomposition Model for Ultrasound Image Denoising. In: 3rd International Conference on Image and Signal Processing, pp. 2870–2875 (2010)
6. Bobin, J., Starck, J., Fadili, J. M., Moudden, Y. and Donoho, D. L.: Morphological Component Analysis: An Adaptive Thresholding Strategy. IEEE Transactions on Image Processing, 16(11), pp. 2675–2681 (2007)
7. Kutyniok, G.: Data Separation by Sparse Representations. Comressed Sensing: Theory and Applications, Cambridge University Press, pp. 485–517 (2012)
8. Elad, M.: Sparse and Redundant Representations. Springer, New York (2010)
9. Liu, S., Hu, S. and Xiao, Y.: Image Separation Using Wavelet-Complex Shearlet Dictionary. Journal of Systems Engineering and Electronics, 25(2), pp. 314-321 (2014)
10. Engan, K., Aase, S. O. and Husoy, J. H.: Method of Optimal Directions for Frame Design. In: 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, 5, pp. 2443–2446 (1999)
11. Aharon, M., Elad, M. and Bruckstein, A.: K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. In: 2006 IEEE Transactions on Signal Processing, 54(11), pp. 4311–4322 (2006)
12. Mairal, J., Bach, F., Ponce, J. and Sapio, G.: Online Dictionary Learning for Sparse Coding. In: 26th Annual International Conference on Machine Learning, pp. 689–696 (2009)
13. Shoham, N. and Elad, M.: Algorithms for Signal Separation Exploiting Sparse Representations,with Applications to Texture Image Separation. In: 2008 IEEE Convention of Electrical and Electronics Engineers, pp. 538 – 542 (2008)
14. Peyr, G., Fadili, J. M. and Starck, J.: Learning the Morphological Diversity. Society for Industrial and Applied Mathematics on Imaging Sciences, 3(3), pp. 646–669 (2010)
15. Li, Y. and Feng, X.: Image Decomposition via Learning the Morphological Diversity. Pattern Recognition Letters, 33(2), pp. 111–120 (2012)
16. Liu, Q., Liu, J. and Liang, D.: Adaptive Image Decomposition via Dictionary Learning with Stuctural Incoherence. In: 2013 IEEE International Conference on Image Processing, pp. 280–284 (2013)
17. Rezaiifar, R.: Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. In: 27th Asilomar Conference on Signals, Systems and Computers, pp. 40–44 (1993)
18. Studer, C. and Baraniuk, R. G.: Stable Restoration and Separation of Approximately Sparse Signals. Applied and Computational Harmonic Analysis, 37(1), pp. 12–35 (2014)
19. Studer, C., Kuppinger, P., Pope, G. and Bölcskei, H.: Recovery of Sparsely Corrupted Signals. IEEE Transactions on Information Theory, 58(5), pp. 3115–3130 (2012)
20. Zhang, L., Zhang, L., Mou, X. and Zhang, D.: FSIM: A Fast Feature Similarity Index for Image Quality Assessment. IEEE Transactions on Image Processing, 20(8), pp. 2378–2386 (2011)