Adaptive Subspace-based Inverse Projections via Division into Multiple Sub-problems for Missing Image Data Restoration

Takahiro Ogawa, *Member, IEEE* and Miki Haseyama, *Senior Member, IEEE,*

***Abstract*—This paper presents adaptive subspace-based inverse projections via division into multiple sub-problems (ASIP-DIMS) for missing image data restoration. In the proposed method, a target problem for estimating missing image data is divided into multiple sub-problems, and each sub-problem is iteratively solved with constraints of other known image data. By projection into a subspace model of image patches, the solution of each sub-problem is calculated, where we call this procedure "subspace-based inverse projection" for simplicity. The proposed method can use higher-dimensional subspaces for finding unique solutions in each sub-problem, and successful restoration becomes feasible since a high level of image representation performance can be preserved. This is the biggest contribution of this paper. Furthermore, the proposed method generates several subspaces from known training examples and enables derivation of a new criterion in the above framework to adaptively select the optimal subspace for each target patch. In this way, the proposed method realizes missing image data restoration using ASIP-DIMS. Since our method can estimate any kind of missing image data, its potential in two image restoration tasks, image inpainting and super-resolution, based on several methods for multivariate analysis is also shown in this paper.**

***Index Terms*—Adaptive restoration, inverse problem, image restoration, inpainting, super-resolution.**

I. Introduction

MISSING image data restoration has numerous applications in image processing such as image inpainting and super-resolution (SR). In image inpainting, the missing image data correspond to missing intensities, and they are estimated from other known intensities within target images. In SR, the missing image data correspond to missing high-frequency components, and they are estimated from target low-resolution (LR) images with other training high-resolution (HR) images also being used as priors in some cases.

In order to realize these applications, many restoration methods based on various approaches that focus on characteristics of general images have been proposed. In this paper, we broadly categorize them into three approaches, structure-based approach [1], [2], example-based approach [3]–[5] and multivariate analysis-based approach [6]–[9]. The structure-based approach enables successful reconstruction of edges, and its specific algorithms are generally different for different applications. On the other hand, the example-based approach and the multivariate analysis-based approach enable successful reconstruction of textures, and their specific algorithms tend to be similar for different applications. Image representation by the multivariate analysis-based approach has been shown to be more accurate than that by the example-based approach [10], [11].

In this study, we focused on the multivariate analysis-based approach with the aim of improving its restoration performance for several applications such as image inpainting and SR. In the rest of this section, we present brief reviews of image inpainting and SR and then clarify the motivation for and the new proposition of our study.

1) *Image Inpainting*:

In the field of image inpainting, many variational methods using the structure-based approach have been proposed [12]–[17]. One of the pioneering works was carried out by Masnou et al. [12], and Bertalmio et al. proposed a representative method based on partial differential equations (PDEs) [1], [2]. Furthermore, the PDE-based method has been improved by many researchers [13], [15], [16]. Next, a pioneering work on the example-based approach was carried out by Efros et al. [18]. According to their idea, Drori et al. [3] and Criminisi et al. [4] developed more accurate image completion techniques. In recent years, the method proposed by Criminisi et al. has been improved in terms of computation speed and image representation performance [19]–[21]. Buyssens et al. proposed an example-based inpainting method using a new priority estimation scheme and an anisotropic spatial blending algorithm for improving inpainting quality as one of the state-of-the-art example-based approaches [20]. Ruzic et al. proposed a context-aware inpainting method using Markov random field (MRF) modeling and adopted a global approach that defines inpainting as a global optimization problem [21].

Most of the example-based approaches focus on a unique characteristic of strong self-similarities within target images, and restoration is achieved by finding the best-matched known examples. Thus, if this characteristic does not exist, accurate restoration is not guaranteed by only using the best-matched examples. Therefore, a number of methods based on the multivariate analysis-based approach, which approximate patches including missing areas by using low-dimensional subspaces generated from known patches within target images, have been proposed.

Since image inpainting is an inverse problem, many methods

T. Ogawa and M. Haseyama are with the Graduate School of Information Science and Technology, Hokkaido University, Sapporo, 060-0814, JAPAN.

This work was partly supported by JSPS KAKENHI Grant Numbers JP25280036 and JP15K12023.

try to solve this problem by approximating image patches in lower-dimensional subspaces. For example, Amano et al. proposed an effective restoration method using eigenspaces obtained by principal component analysis (PCA) [6]. This method is called back projection for lost pixels (BPLP) method. Several methods using kernel PCA (KPCA) [22] for successfully representing nonlinear visual features have also been proposed [7], [23], [24]. In recent years, many restoration methods using subspaces obtained by sparse representation have been proposed [8], [11], [25]–[31]. As similar ideas, several inpainting and error concealment methods based on neighbor embedding approaches have also been proposed [32], [33]. These methods are also derived from the aspect of manifold learning and provide good results. Image completion methods based on the concept of rank minimization have also been proposed [34], [35], [36]. Jin et al. proposed a low-rank patch-based block Hankel structured matrix approach for image inpainting [36].

2) *Super-resolution*:

The study of resolution enhancement has a long history [37], [38], and many researchers have studied SR in recent years. SR refers to the task of generating an HR image from one or more LR images by estimating missing high-frequency components. Generally, SR methods are divided into two categories [39]–[42], reconstruction-based and learning-based schemes, and the latter is matched to the main topic of this paper. Generally, in methods based on the learning-based scheme, HR images are obtained by using several other images as training data, and a number of learning-based SR methods have been proposed [43]–[47]. For example, Freeman et al. proposed representative example-based SR methods that estimate missing high-frequency components from mid-frequency components of a target image based on Markov networks for estimating HR images [5]. Glasner et al. proposed a representative method of SR [48] based on a combined approach of multi-image SR and example-based SR. This method realized successful SR from a single image based on observations that patches in a natural image redundantly recur many times inside the image both within the same scale and across different scales.

As a multivariate analysis-based approach, learning-based SR methods using PCA have been proposed for face hallucination [49]. Furthermore, Chakrabarti et al. improved the performance of face hallucination [50] based on KPCA [22]. Many methods perform SR for each local image patch within target images [44], [51], [52]. Kim et al. developed a global-based face hallucination method and a local-based SR method for general images by using KPCA [7]. Furthermore, more accurate SR methods have been realized by adopting multiple nonlinear eigenspaces [53], [54], and they enable selection of the optimal subspaces. In recent years, sparse representation-based methods [9], [55], [56] and neighboring embedding-based methods [57] have achieved successful generation of optimal subspaces for estimating missing high-frequency components. The above-mentioned methods are small parts of recent studies, and a number of new methods based on different ideas have also been proposed [58]–[60].

Furthermore, as state-of-the-art methods, many multivariate analysis-based methods using PCA, KPCA, sparse representation and rank minimization have been proposed. Zhang et al. proposed a single image SR method via learning multiple linear mappings that include a clustering-based scheme [61]. Zhou et al. also proposed a single image SR method based on compact KPCA coding and kernel regression, and their method also includes a clustering-based scheme [62]. Sun et al. [63] and Jiang et al. [64] respectively proposed sparse representation-based SR methods, and the latter one can simultaneously consider the manifold geometrical structure of patch manifold space and the support of the corresponding sparse coefficients. Cao et al. proposed an SR method based on image interpolation via low-rank matrix completion and recovery [65].

3) *Motivation and New Proposition*:

As shown in the above brief reviews related to image inpainting and SR, the multivariate analysis-based approach tries to approximate target patches including missing image data in lower-dimensional subspaces to find unique solutions of the target problem. In some methods [6], [30], [54], the maximum dimension for obtaining unique solutions can be analytically derived. If the number of missing pixels within target patches becomes larger in image inpainting, the dimension of subspaces must be set to a smaller value for obtaining unique solutions. If the magnification factor becomes larger in SR, the same problem generally occurs. Then since accurate approximation of target patches in lower-dimensional subspaces becomes difficult, the restoration performance also becomes worse.

In this paper, we present adaptive subspace-based inverse projections via division into multiple sub-problems (ASIP-DIMS) for missing image data restoration. We propose ASIP-DIMS for improving image representation performance of arbitrary multivariate analysis-based methods that approximate patches in low-dimensional subspaces. Specifically, the proposed method divides the target problem for estimating missing image data within a target patch into multiple sub-problems. Then each of these sub-problems is solved by projection into a subspace model of image patches, and this procedure is called "subspace-based inverse projection" for simplicity. Since the size of the target missing components becomes smaller in each sub-problem, we can use higher-dimensional subspaces for finding unique solutions, where missing components correspond to missing intensities or missing high frequency components in this paper. This approach provides a solution to the conventional problem of degradation of image representation performance. The above idea has not been proposed so far, i.e., no previous methods (not only our previously reported methods [23], [24], [30], [54] but also other existing methods) did not perform the division of missing components into multiple sub-groups. Then, by iteratively solving each sub-problem with the constraints of other known components within the target patch, the whole missing components can be successfully reconstructed since higher-dimensional subspaces can be used for approximating the target patch. This is the biggest contribution of our paper.

Next, since target images generally consist of several kinds of textures, they should be adaptively restored by the subspaces of the same kinds of textures, and we have proposed adaptive

restoration methods in [23], [24], [54][1]. Furthermore, in different fields of image restoration such as image denoising, the idea of adaptive restoration has also been introduced by using optimal subspaces for target patches [66], [67]. Therefore, the idea of these methods is introduced into our framework of the proposed method. Specifically, we define a new criterion from errors caused by solving the sub-problems for realizing adaptive selection of the optimal subspace. Then, for each patch including missing components, adaptive restoration becomes feasible. Although the novelty of this adaptive selection is less than that of the division into multiple sub-problems, it also helps successful image representation. Consequently, missing image data restoration using ASIP-DIMS becomes feasible.

The proposed method can be used for any applications if they involve restoration of missing image data. For example, by regarding both missing intensities and missing high-frequency components as missing image data, image inpainting and SR can be realized by the proposed method. In this paper, the proposed method is applied to these two applications to verify its high level of applicability as well as its high level of restoration performance. As described above, our method can be regarded as a boosting method for the multivariate analysis-based approach, and our method can also be regarded as an extended version of our previous works [30], [54]. Therefore, in this paper, we focus only on how ASIP-DIMS improve the performance of the multivariate analysis-based approach in the two applications. This means that we limit the main focus of this paper to the above point, not to inpainting of quite large regions and SR with high magnification factors. This is the placement of our paper.

This paper is organized as follows. First, in Section II, the basic concept of ASIP-DIMS for missing image data restoration is presented. In Section III, specific derivations of ASIP-DIMS using several multivariate analysis methods including PCA, KPCA and sparse representation are shown. In Section IV, results of experiments in which our method was applied to image inpainting and SR are presented to verify its performance. Finally, concluding remarks are presented in Section V. Note that since many mathematical notations are included in our paper, the main mathematical symbols and brief explanations of them are given in Appendix.

II. Adaptive Subspace-based Inverse Projections via Division into Multiple Sub-problems

ASIP-DIMS for missing image data restoration are presented in this section. An overview of ASIP-DIMS is shown in Fig. 1. From a target image, we clip a patch f ($w\times h$ pixels) for which information is partly missing and restore it on the basis of ASIP-DIMS that are derived by using subspaces obtained from known training examples, the procedures for acquiring training examples being different in each application, i.e., image inpainting and SR (see Section IV). For the following explanation, we denote known and unknown components within f as $\mathcal{K}$ and $\mathcal{U}$, respectively.

[1]Note that the methods in [23], [24] used a projection onto convex sets (POCS) algorithm, and their restoration procedures are completely different from ours.

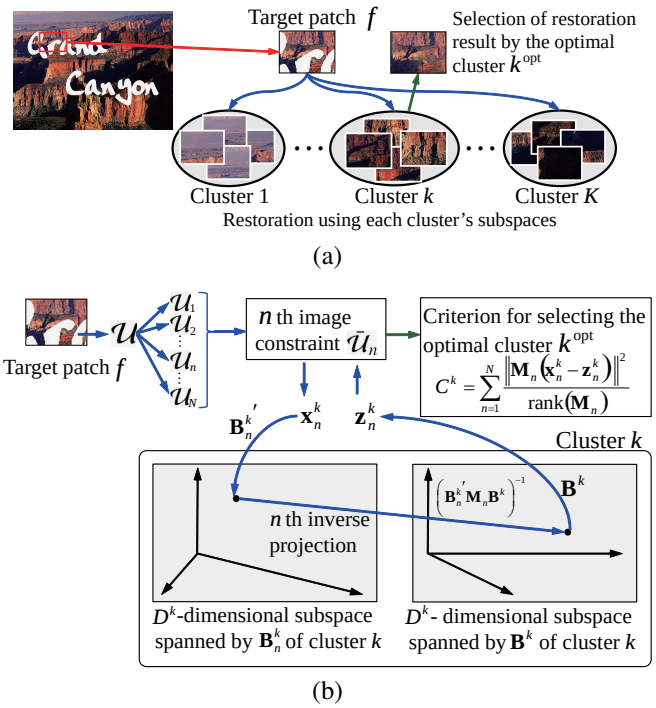


Fig. 1. Overview of ASIP-DIMS. As shown in (a), our method performs clustering of known training patches and achieves adaptive restoration of the target patch f by selecting the optimal cluster's restoration result. Furthermore, as shown in (b), the target inverse problem of estimating missing components $\mathcal{U}$ is divided into multiple sub-problems of estimating sub-components $\mathcal{U}_n$. Then each sub-component $\mathcal{U}_n$ is estimated by approximating the target patch f in the low-dimensional subspaces. The missing components $\mathcal{U}$ can be recovered by iteratively solving the sub-problems, where errors C^k caused in this iteration are used for selecting the optimal cluster k^{opt} in (a). The details are shown in II-A and II-B.

As shown in Fig. 1(a), the proposed method prepares K clusters by clustering known training patches to obtain K kinds of subspaces (see II-A). Then the restoration of the target patch f is performed by using all of the K subspaces, and then K restoration results are obtained. Among these K results, the proposed method enables selection of the optimal cluster's result. Note that in the restoration using each cluster's subspaces, we divide the missing components $\mathcal{U}$ into N sub-components $\mathcal{U}_n$ ($n = 1, 2, \cdots, N$) and iteratively reconstruct each sub-component $\mathcal{U}_n$ by using the other components $\bar{\mathcal{U}}_n$ within f as shown in Fig. 1(b), where we define $\bar{\mathcal{U}}_n = f - \mathcal{U}_n$. The estimation of each sub-component $\mathcal{U}_n$ is performed by projecting the target patch f into the subspaces of image patches. Furthermore, errors caused by this estimation are used for selecting the optimal cluster's result in Fig. 1(a). By combining the procedures of Figs. 1(a) and (b), the restoration using ASIP-DIMS can be realized (see II-B).

A. Subspace-based Clustering

In this subsection, we explain subspace-based clustering of training examples. Clustering using each cluster's subspace is based on the idea of our previously reported methods [23], [24], [54], and this section shows their generalized explanation for which subspaces are arbitrary, i.e., not limited to KPCA-based subspaces used in [23], [24], [54].

Given training examples, i.e., training patches, their clustering is performed to obtain $f^{k,j}$ ($j \in \mathcal{J}^k$) that represents the jth patch belonging to cluster k ($= 1, 2, \cdots, K$), where $\mathcal{J}^k$ is a set of indices, and K is the number of clusters. Note that we define $\mathbf{x}^{k,j} = [\boldsymbol{\kappa}^{k,j\top}, \boldsymbol{\mu}^{k,j\top}]^\top$, and $\boldsymbol{\kappa}^{k,j}$ ($\in \mathbf{R}^{D^\kappa}$) and $\boldsymbol{\mu}^{k,j}$ ($\in \mathbf{R}^{D^\mu}$) are vectors whose elements are respectively the components corresponding to $\mathcal{K}$ and $\mathcal{U}$ of the target patch f. Note that vector/matrix transpose is denoted by superscript $^\top$ in this paper. The dimensions D^κ, D^μ and D ($= D^\kappa + D^\mu$) depend on the method used for multivariate analysis methods (see Section III).

We perform clustering of known training patches by minimizing the following criterion:

$$C = \sum_{k=1}^{K} \sum_{j \in \mathcal{J}^k} \left\| \left(\mathbf{x}^{k,j} - \boldsymbol{\nu}^k\right) - \mathbf{B}^k \left(\mathbf{B}^{k\top}\mathbf{B}^k\right)^{-1} \mathbf{B}^{k\top} \left(\mathbf{x}^{k,j} - \boldsymbol{\nu}^k\right) \right\|^2, \quad (1)$$

where $\boldsymbol{\nu}^k = \frac{1}{|\mathcal{J}^k|} \sum_{j \in \mathcal{J}^k} \mathbf{x}^{k,j}$ is the mean vector of $\mathbf{x}^{k,j}$ ($j \in \mathcal{J}^k$), $|\mathcal{J}^k|$ representing the number of elements belonging to $\mathcal{J}^k$. The matrix $\mathbf{B}^k$ ($\in \mathbf{R}^{D \times D^k}$) is a basis matrix of cluster k obtained from its elements $\mathbf{x}^{k,j}$ ($j \in \mathcal{J}^k$), where $D^k < D$, and the bases in $\mathbf{B}^k$ depend on the method used for multivariate analysis such as PCA, KPCA and sparse representation. The matrix $\mathbf{B}^k \left(\mathbf{B}^{k\top}\mathbf{B}^k\right)^{-1}$ corresponds to the pseudo-inverse matrix of $\mathbf{B}^{k\top}$.

By iteratively performing two procedures, assignment of the known patches to the optimal clusters minimizing Eq. (1) and renewal of the basis matrices, which are similar to those of the k-means clustering algorithm, the proposed method realizes subspace-based clustering. Finally, the obtained basis matrix $\mathbf{B}^k$ of each cluster k provides the subspace optimally representing its elements $\mathbf{x}^{k,j}$ ($j \in \mathcal{J}^k$).

B. Restoration based on ASIP-DIMS

The details of restoration based on ASIP-DIMS are represented in this subsection. As shown in the previous subsection, we can obtain the subspace for each cluster k. The proposed method tries to estimate the missing components $\mathcal{U}$ of the target patch f by using this subspace. Given two vectors $\mathbf{x} = [\boldsymbol{\kappa}^\top, \boldsymbol{\mu}^\top]^\top$ ($\in \mathbf{R}^D$) and $\mathbf{y} = [\boldsymbol{\kappa}^\top, \mathbf{0}_{D^\mu}^\top]^\top$ ($\in \mathbf{R}^D$) respectively corresponding to an original vector of f and its corrupted vector for which components of $\mathcal{U}$ are missing, the following relationship holds: $\mathbf{y} = \mathbf{M}\mathbf{x}$, where $\mathbf{M}$ ($\in \mathbf{R}^{D \times D}$) is a diagonal matrix, and the first D^κ diagonal elements are one and the remaining D^μ diagonal elements are zero. From the definition of $\mathbf{M}$, its inverse matrix cannot be directly obtained since $\mathbf{M}$ is singular, i.e., $\mathrm{rank}(\mathbf{M}) < D$. Therefore, we try to derive corresponding pseudo-inverse by respectively approximating $\mathbf{x}$ and $\mathbf{y}$ in the D^k-dimensional subspaces, the dimensions of which must be smaller than $\mathrm{rank}(\mathbf{M})$. Selection of the optimal cluster among K clusters ($k = 1, 2, \cdots, K$) for the restoration of f is explained later.

If $\mathrm{rank}(\mathbf{M})$ becomes smaller, i.e., the size of the missing components (dimension D^μ) becomes larger, we have to reduce the dimension D^k since $D^k \leq \mathrm{rank}(\mathbf{M})$ must be satisfied for deriving the corresponding pseudo-inverse, where we assume that the basis matrices of the subspaces for respectively approximating $\mathbf{x}$ and $\mathbf{y}$ are full-rank matrices. Then the image representation performance also becomes worse by using the lower-dimensional subspaces.

In order to solve this problem, we divide the missing components $\mathcal{U}$ into N sub-components $\mathcal{U}_n$ ($n = 1, 2, \cdots, N$), performing this division in such a way that the number of elements in each sub-component $\mathcal{U}_n$ becomes the same. Then we iteratively estimate the elements within sub-components $\mathcal{U}_n$ ($n = 1, 2, \cdots, N$) by using the other parts $\bar{\mathcal{U}}_n$ of f. Here, we explain how the missing components $\mathcal{U}$ are divided into N sub-components $\mathcal{U}_n (n = 1, 2, \cdots, N)$. This can be clearly explained by showing how to derive $\mathbf{M}_n$ ($n = 1, 2, \cdots, N$) from $\mathbf{M}$, where $\mathbf{M}_n$ ($n = 1, 2, \cdots, N$) is a diagonal matrix obtained from sub-component $\mathcal{U}_n$. Since the matrix $\mathbf{M}$ is a diagonal matrix whose diagonal elements are zero or one, we equally divide its D^μ zero-diagonal elements into N groups from the top-left to the bottom-right of the matrix $\mathbf{M}$. Furthermore, the grouped $\frac{D^\mu}{N}$ zero-diagonal elements are assigned to $\mathbf{M}_n$ ($n = 1, 2, \cdots, N$). For example, given a diagonal matrix $\mathbf{M} = \mathrm{diag}\{1, 1, 1, 1, 0, 0, 0, 0, 0, 0\}$, i.e., $D = 10$ and $D^\mu = 6$, and when $N = 3$, $\mathbf{M}_1$, $\mathbf{M}_2$ and $\mathbf{M}_3$ are obtained as follows: $\mathbf{M}_1 = \mathrm{diag}\{1, 1, 1, 1, 0, 0, 1, 1, 1, 1\}$, $\mathbf{M}_2 = \mathrm{diag}\{1, 1, 1, 1, 1, 1, 0, 0, 1, 1\}$, and $\mathbf{M}_3 = \mathrm{diag}\{1, 1, 1, 1, 1, 1, 1, 1, 0, 0\}$, and $\mathbf{M} = \mathbf{M}_1\mathbf{M}_2\mathbf{M}_3$ is satisfied. Each diagonal matrix $\mathbf{M}_n$ ($n = 1, 2, 3$) contains $\frac{D^\mu}{N}$ ($= 2$) zero-diagonal elements.

From the definition of $\mathbf{M}_n$ ($n = 1, 2, \cdots, N$), $\mathrm{rank}(\mathbf{M}_n) > \mathrm{rank}(\mathbf{M})$ is satisfied, and we can use higher-dimensional subspaces for the restoration. From the above definition, $\mathrm{rank}(\mathbf{M}_n)$ ($n = 1, 2, \cdots, N$) and $\mathrm{rank}(\mathbf{M})$ become the numbers of non-zero diagonal elements included in the diagonal matrices $\mathbf{M}_n$ and $\mathbf{M}$, respectively. Therefore, since the number of the non-zero diagonal elements in $\mathbf{M}_n$ is larger than that in $\mathbf{M}$, $\mathrm{rank}(\mathbf{M}_n) > \mathrm{rank}(\mathbf{M})$ is always satisfied.

First, for each sub-component $\mathcal{U}_n$, the following equation is satisfied:

$$\mathbf{y}_n = \mathbf{M}_n\mathbf{x}, \quad (2)$$

where $\mathbf{y}_n$ is a corrupted vector including sub-component $\mathcal{U}_n$. By using the D^k-dimensional subspaces spanned by the bases in $\mathbf{B}^k$ and $\mathbf{B}_n^k$ of cluster k, $\mathbf{x}$ and $\mathbf{y}_n$ are approximated as

$$\mathbf{x} \cong \mathbf{B}^k\mathbf{p}^k + \boldsymbol{\nu}^k, \quad (3)$$

$$\mathbf{y}_n \cong \mathbf{B}_n^k\mathbf{q}_n^k + \mathbf{M}_n\boldsymbol{\nu}^k, \quad (4)$$

where $\mathbf{B}^k$ is obtained in the previous subsection, and $\mathbf{B}_n^k$ ($\in \mathbf{R}^{D \times D^k}$) is obtained from $\mathbf{M}_n\mathbf{x}^{k,j}$ ($j \in \mathcal{J}^k$) in the same way as $\mathbf{B}^k$. Furthermore, $\mathbf{p}^k$ and $\mathbf{q}_n^k$ in Eqs. (3) and (4) are respectively obtained as follows:

$$\mathbf{p}^k = \left(\mathbf{B}^{k\top}\mathbf{B}^k\right)^{-1} \mathbf{B}^{k\top} \left(\mathbf{x} - \boldsymbol{\nu}^k\right), \quad (5)$$

$$\mathbf{q}_n^k = \left(\mathbf{B}_n^{k\top}\mathbf{B}_n^k\right)^{-1} \mathbf{B}_n^{k\top} \left(\mathbf{y}_n - \mathbf{M}_n\boldsymbol{\nu}^k\right). \quad (6)$$

We introduce approximations of $\mathbf{x}$ and $\mathbf{y}_n$ respectively using $\mathbf{B}^k$ and $\mathbf{B}_n^k$ as shown in Eqs. (3) and (4) since the bases of $\mathbf{B}^k$ and $\mathbf{B}_n^k$ optimally represent $\mathbf{x}^{k,j}$ and $\mathbf{M}_n\mathbf{x}^{k,j}$, respectively. Thus, the basis matrices $\mathbf{B}^k$ and $\mathbf{B}_n^k$, which are used for optimally representing $\mathbf{x}$ and $\mathbf{y}_n$, respectively, are different. Furthermore, in the explanation of this section, $\mathbf{B}_n^k = \mathbf{M}_n\mathbf{B}^k$ does not necessary hold. As described before, it is assumed that $\mathbf{B}^k$

and $\mathbf{B}_n^k$ are full-rank matrices. By substituting Eqs. (3) and (4) into Eq. (2),

$$\mathbf{B}_n^k\mathbf{q}_n^k + \mathbf{M}_n\boldsymbol{\nu}^k \cong \mathbf{M}_n\left(\mathbf{B}^k\mathbf{p}^k + \boldsymbol{\nu}^k\right), \tag{7}$$

and by multiplying $\mathbf{B}_n^{k\top}$ into both sides of Eq. (7),

$$\mathbf{B}_n^{k\top}\mathbf{B}_n^k\mathbf{q}_n^k \cong \mathbf{B}_n^{k\top}\mathbf{M}_n\mathbf{B}^k\mathbf{p}^k. \tag{8}$$

Then, if $\mathbf{B}_n^k$ and $\mathbf{B}^k$ are full-rank matrices and $D^k \leq \mathrm{rank}(\mathbf{M}_n)$ is satisfied, $\mathbf{B}_n^{k\top}\mathbf{M}_n\mathbf{B}^k$ becomes non-singular, and its inverse matrix can be obtained as

$$\mathbf{p}^k \cong \left(\mathbf{B}_n^{k\top}\mathbf{M}_n\mathbf{B}^k\right)^{-1}\mathbf{B}_n^{k\top}\mathbf{B}_n^k\mathbf{q}_n^k. \tag{9}$$

Furthermore, by substituting Eq. (9) into Eq. (3),

$$\mathbf{x} \cong \mathbf{B}^k\left(\mathbf{B}_n^{k\top}\mathbf{M}_n\mathbf{B}^k\right)^{-1}\mathbf{B}_n^{k\top}\mathbf{B}_n^k\mathbf{q}_n^k + \boldsymbol{\nu}^k. \tag{10}$$

As shown in the following equation, the estimation result of $\mathbf{x}$ is obtained by substituting Eq. (6) into Eq. (10):

$$\mathbf{x} \cong \mathbf{B}^k\left(\mathbf{B}_n^{k\top}\mathbf{M}_n\mathbf{B}^k\right)^{-1}\mathbf{B}_n^{k\top}\left(\mathbf{y}_n - \mathbf{M}_n\boldsymbol{\nu}^k\right) + \boldsymbol{\nu}^k. \tag{11}$$

As shown in the above equation, we can estimate $\mathbf{x}$ without directly calculating the inverse matrix of $\mathbf{M}_n$, and it corresponds to the nth inverse projection in Fig. 1(b). By iteratively calculating Eq. (11) for all $\mathbf{M}_n$ ($n = 1, 2, \cdots, N$), we can estimate all of the missing components in $\mathcal{U}$. It should be noted that after calculating Eq. (11) for each $\mathbf{M}_n$, the other elements in $\bar{\mathcal{U}}_n$ also change, and they should be modified. This corresponds to the nth image constraint in Fig. 1(b). Therefore, given the nth result as $\mathbf{x}_n^k$, its update with the known component constraints can be written as

$$\begin{aligned}\mathbf{x}_{n+1}^k &= (\mathbf{I}_{D\times D} - \mathbf{M}_n)\left\{\mathbf{B}^k\left(\mathbf{B}_n^{k\top}\mathbf{M}_n\mathbf{B}^k\right)^{-1}\mathbf{B}_n^{k\top}\left(\mathbf{x}_n^k - \mathbf{M}_n\boldsymbol{\nu}^k\right) + \boldsymbol{\nu}^k\right\} \\ &\quad + \mathbf{M}_n\mathbf{x}_n^k \\ &= (\mathbf{I}_{D\times D} - \mathbf{M}_n)\,\mathbf{z}_n^k + \mathbf{M}_n\mathbf{x}_n^k \quad (n = 1, 2, \cdots, N),\end{aligned} \tag{12}$$

where $\mathbf{I}_{D\times D}$ is the $D \times D$ identity matrix, and we define $\mathbf{z}_n^k$ as

$$\mathbf{z}_n^k = \mathbf{B}^k\left(\mathbf{B}_n^{k\top}\mathbf{M}_n\mathbf{B}^k\right)^{-1}\mathbf{B}_n^{k\top}\left(\mathbf{x}_n^k - \mathbf{M}_n\boldsymbol{\nu}^k\right) + \boldsymbol{\nu}^k. \tag{13}$$

By iterating Eq. (12) for all n ($n = 1, 2, \cdots, N$), we finally obtain the estimation result of $\mathbf{x}$ with the constraint of $\mathcal{K}$. Specifically, our method iterates the update shown in Eq. (12) as $\mathbf{x}_1^k \to \mathbf{x}_2^k \to \cdots \to \mathbf{x}_N^k \to \mathbf{x}_1^k \to \mathbf{x}_2^k \to \cdots$. Note that the final result obtained by Eq. (12) does not necessarily converge to a fixed point. The proposed method tries to find a solution in the subspace with the constraint of the known components within the target patch. However, if the intersection of this subspace and the constraint does not exist, the final result does not converge to a fixed point. Thus, the proposed method simply outputs $\mathbf{x}_N^k$ after predetermined iterations.

As described above, since the initial inverse problem ($\mathbf{y} = \mathbf{M}\mathbf{x}$) is an ill-posed problem, direct calculation of its solution is impossible. In order to find its unique solution, we have to provide a regularization term for $\mathbf{x}$ or an assumption as its prior information. Our method adopts the assumption that $\mathbf{x}$ and $\mathbf{y}_n$ are in the D^k-dimensional subspaces as shown in Eqs. (3) and (4), respectively. In this approach, restoration performance directly depends on the dimension D^k used for deriving in Eq. (11) since D^k determines image representation performance in Eqs. (3) and (4). If we do not divide the target problem, the maximum dimension of D^k for obtaining a unique solution of $\mathbf{x}$ is $\mathrm{rank}(\mathbf{M})$. Thus, approximation in the D^k-dimensional subspace causes errors since $D - \mathrm{rank}(\mathbf{M})$ bases must be discarded. On the other hand, when adopting the division into multiple sub-problems, the target inverse problem is replaced with N sub-problems. Then the number of bases that must be discarded becomes $\frac{D-\mathrm{rank}(\mathbf{M})}{N}$ for each sub-problem. Consequently, the maximum dimension for obtaining a unique solution in each sub-problem is $D - \frac{D-\mathrm{rank}(\mathbf{M})}{N} = \frac{(N-1)D+\mathrm{rank}(\mathbf{M})}{N}$, and we can use more $\frac{(N-1)D+\mathrm{rank}(\mathbf{M})}{N} - \mathrm{rank}(\mathbf{M}) = \frac{N-1}{N}\{D - \mathrm{rank}(\mathbf{M})\}$ bases compared to the case directly solving the target inverse problem. Note that the higher-dimensional subspace means that the number of available bases is larger since we assume the basis matrices $\mathbf{B}_n^k$ and $\mathbf{B}^k$ are full-rank. The larger the number of bases available for representing image patches is, the smaller the residues caused by the image approximations in Eqs. (3) and (4) are. Since only these two approximations affect the derivation of Eq. (11), more accurate estimation of $\mathbf{x}$ becomes feasible by introducing our novel approach. On the other hand, if the dimension of the subspace becomes too large, i.e., N becomes a too large value, the overfitting occurs. Then, if the overfitting occurs, the estimation results by the proposed method tend not to change from initial values.

In this way, the missing components $\mathcal{U}$ of the target patch f can be estimated by the subspaces of cluster k. Note that if the target patch f belongs to cluster k, we can obtain optimal estimation results. However, since the target patch f contains missing components $\mathcal{U}$, we cannot perform its assignment to the optimal cluster k^{opt} based on the criterion shown in Eq. (1). In order to solve this problem, we monitor the following new criterion for assigning the target patch f to the optimal cluster k^{opt}:

$$C^k = \sum_{n=1}^{N}\frac{\|\mathbf{M}_n(\mathbf{x}_n^k - \mathbf{z}_n^k)\|^2}{\mathrm{rank}(\mathbf{M}_n)}. \tag{14}$$

In this equation, $\frac{\|\mathbf{M}_n(\mathbf{x}_n^k - \mathbf{z}_n^k)\|^2}{\mathrm{rank}(\mathbf{M}_n)}$ corresponds to the mean square error caused in $\bar{\mathcal{U}}_n$ by each nth estimation. Then C^k is the total sum of these mean square errors for all sub-known components $\bar{\mathcal{U}}_n$ ($n = 1, 2, \cdots, N$). It has been reported that the use of errors caused in the estimation is effective for selecting the optimal cluster or the optimal bases [23], [24], [30]. Specifically, in [30], the use of errors caused in the inverse projection provides better bases, i.e., better subspaces for representing the target patch compared to the use of approximation errors calculated from only the known components. Therefore, by selecting the cluster minimizing the converged error of C^k in Eq. (14), the proposed method can find the optimal cluster k^{opt} for the target patch f even if it contains missing components. Although the basic idea of selection of the optimal cluster for the restoration is based on our previously reported methods, we newly introduce the criterion in Eq. (14) derived through our new framework. Consequently, adaptive restoration becomes feasible by using the optimal cluster k^{opt}, and ASIP-DIMS can be realized.

III. Specific Derivations of ASIP-DIMS

In this section, we show the specific derivations of ASIP-DIMS for the following methods of multivariate analysis: PCA, KPCA and sparse representation. The derivation based on PCA and KPCA is shown in III-A, and the derivation based on sparse representation is shown in III-B. In the explanations of the previous section, the most important points of ASIP-DIMS are Eqs. (1), (12), (13) and (14). Therefore, in each subsection, we show how these equations are derived for each method of multivariate analysis.

A. Derivation Based on PCA and KPCA

In this subsection, we show the specific derivations of ASIP-DIMS based on PCA and KPCA. Since KPCA becomes PCA by setting its kernel function to the linear kernel, the derivation using KPCA is shown. Furthermore, we limit the explanation to the derivation using the linear kernel or the Gaussian kernel for simplicity.

From known training examples whose components in $\mathcal{K}$ and $\mathcal{U}$ are both known, we first perform their clustering to obtain $f^{k,j}$ ($j \in \mathcal{J}^k$). Given their vectors $\boldsymbol{\kappa}^{k,j}$ ($\in \mathbf{R}^{D^{\kappa}}$) and $\boldsymbol{\mu}^{k,j}$ ($\in \mathbf{R}^{D^{\mu}}$), we map $\boldsymbol{\kappa}^{k,j}$ into the feature space via a nonlinear map [22], and $\phi(\boldsymbol{\kappa}^{k,j})$ ($\in \mathbf{R}^{D^{\phi}}$) is obtained. Note that since the mapped results $\phi(\boldsymbol{\kappa}^{k,j})$ are high-dimensional or infinite-dimensional, it may not be possible to calculate them directly. Fortunately, it is well known that the following computational procedures depend only on inner products in the feature space, which can be efficiently obtained from a suitable kernel function [22]. If we use the linear kernel for this kernel function, the following derivation becomes that of ASIP-DIMS using PCA. It should be noted that an exact pre-image, which is the inverse mapping from the feature space back to the input space, typically does not exist [68], and we do not utilize the nonlinear map for $\boldsymbol{\mu}^{k,j}$ to avoid the pre-image estimation error.

Since Eq. (1) corresponds to the sum of errors caused in the input space by projecting each training example onto its cluster's subspace, the criterion C using KPCA can be rewritten as

$$C = \sum_{k=1}^{K} \sum_{j \in \mathcal{J}^k} \left(\left\| \boldsymbol{\kappa}^{k,j} - \tilde{\boldsymbol{\kappa}}^{k,j} \right\|^2 + \left\| \boldsymbol{\mu}^{k,j} - \tilde{\boldsymbol{\mu}}^{k,j} \right\|^2 \right). \tag{15}$$

Then, given $\mathbf{x}^{k,j} = [\phi(\boldsymbol{\kappa}^{k,j})^\top, \boldsymbol{\mu}^{k,j\top}]^\top$ and $\tilde{\mathbf{x}}^{k,j} = [\tilde{\boldsymbol{\phi}}^{k,j\top}, \tilde{\boldsymbol{\mu}}^{k,j\top}]^\top$, they satisfy

$$\tilde{\mathbf{x}}^{k,j} = \mathbf{B}^k \mathbf{B}^{k\top} \left(\mathbf{x}^{k,j} - \boldsymbol{\nu}^k \right) + \boldsymbol{\nu}^k, \tag{16}$$

where $\tilde{\boldsymbol{\phi}}^{k,j} \cong \phi(\tilde{\boldsymbol{\kappa}}^{k,j})$, and $\tilde{\boldsymbol{\kappa}}^{k,j}$ in Eq. (15) corresponds to the pre-image of $\tilde{\boldsymbol{\phi}}^{k,j}$. Furthermore, $\mathbf{B}^k$ is an eigenvector matrix of $\mathbf{X}^k \mathbf{H}^k \mathbf{H}^k \mathbf{X}^{k\top}$, where $\mathbf{X}^k = [\mathbf{x}^{k,1}, \mathbf{x}^{k,2}, \cdots, \mathbf{x}^{k,|\mathcal{J}^k|}]$ and $\mathbf{H}^k = \mathbf{I}_{|\mathcal{J}^k| \times |\mathcal{J}^k|} - \frac{1}{|\mathcal{J}^k|} \mathbf{1}_{|\mathcal{J}^k|} \mathbf{1}_{|\mathcal{J}^k|}{}^\top$ ($\in \mathbf{R}^{|\mathcal{J}^k| \times |\mathcal{J}^k|}$) is the centering matrix, and $\mathbf{1}_{|\mathcal{J}^k|}$ is a $|\mathcal{J}^k|$-dimensional vector whose elements are all one.

In Eq. (1), $\left\| \left(\mathbf{x}^{k,j} - \boldsymbol{\nu}^k \right) - \mathbf{B}^k \left(\mathbf{B}^{k\top} \mathbf{B}^k \right)^{-1} \mathbf{B}^{k\top} \left(\mathbf{x}^{k,j} - \boldsymbol{\nu}^k \right) \right\|^2$ represents the error caused in the input space by projecting a training patch $\mathbf{x}^{k,j}$ onto the subspace of cluster k to which it belongs. Similarly, $\left\| \boldsymbol{\kappa}^{k,j} - \tilde{\boldsymbol{\kappa}}^{k,j} \right\|^2 + \left\| \boldsymbol{\mu}^{k,j} - \tilde{\boldsymbol{\mu}}^{k,j} \right\|^2$ in Eq. (15) represents the error, which is caused in the input space by projecting a training patch $\mathbf{x}^{k,j}$ onto the linear/nonlinear eigenspace of cluster k to which it belongs. When we use the linear kernel, i.e., PCA, $\mathbf{x}^{k,j} = [\boldsymbol{\kappa}^{k,j\top}, \boldsymbol{\mu}^{k,j\top}]^\top$ and $\tilde{\mathbf{x}}^{k,j} = [\tilde{\boldsymbol{\kappa}}^{k,j\top}, \tilde{\boldsymbol{\mu}}^{k,j\top}]^\top$. Then Eq. (15) becomes the same as Eq. (1), that is, we do not have to rewrite Eq. (1) as Eq. (15). On the other hand, when we use the nonlinear kernel, i.e., KPCA, $\mathbf{x}^{k,j} = [\phi(\boldsymbol{\kappa}^{k,j})^\top, \boldsymbol{\mu}^{k,j\top}]^\top$ and $\tilde{\mathbf{x}}^{k,j} = [\tilde{\boldsymbol{\phi}}^{k,j\top}, \tilde{\boldsymbol{\mu}}^{k,j\top}]^\top$. Then, in order to obtain the error of $\mathbf{x}^{k,j}$ caused in the input space, we have to separately calculate the errors of $\boldsymbol{\kappa}^{k,j}$ and $\boldsymbol{\mu}^{k,j}$. Therefore, we have to rewrite Eq. (1) as Eq. (15). In this way, we define Eq. (15) in such a way that it can be commonly used in PCA and KPCA.

If we adopt nonlinear kernels, each column vector in $\mathbf{B}^k$ becomes high-dimensional, but we can perform the calculation using $\mathbf{B}^k$ based on the following scheme. Since the eigenvector matrix $\mathbf{B}^k$ satisfies the following singular value decomposition: $\mathbf{X}^k \mathbf{H}^k \cong \mathbf{B}^k \boldsymbol{\Lambda}^k \mathbf{V}^{k\top}$, it can be rewritten as

$$\mathbf{B}^k \cong \mathbf{X}^k \mathbf{H}^k \mathbf{V}^k \boldsymbol{\Lambda}^{k-1}, \tag{17}$$

where $\mathbf{V}^k$ is the eigenvector matrix of $\mathbf{H}^k \mathbf{X}^{k\top} \mathbf{X}^k \mathbf{H}^k$, and $\boldsymbol{\Lambda}^k$ is a singular value matrix. Then, by substituting Eq. (17) into Eq. (16), it can be rewritten as

$$\tilde{\mathbf{x}}^{k,j} \cong \mathbf{X}^k \mathbf{T}^k \mathbf{X}^{k\top} \mathbf{x}^{k,j} - \frac{1}{|\mathcal{J}^k|} \mathbf{X}^k \left(\mathbf{T}^k \mathbf{X}^{k\top} \mathbf{X}^k - \mathbf{I}_{|\mathcal{J}^k| \times |\mathcal{J}^k|} \right) \mathbf{1}_{|\mathcal{J}^k|}, \tag{18}$$

where

$$\mathbf{T}^k = \mathbf{H}^k \mathbf{V}^k \boldsymbol{\Lambda}^{k-2} \mathbf{V}^{k\top} \mathbf{H}^k, \tag{19}$$

and we use $\boldsymbol{\nu}^k = \frac{1}{|\mathcal{J}^k|} \mathbf{X}^k \mathbf{1}_{|\mathcal{J}^k|}$. Furthermore, since $\mathbf{X}^k = [\boldsymbol{X}_{\boldsymbol{\phi}}^{k\top}, \boldsymbol{X}_{\boldsymbol{\mu}}^{k\top}]^\top$, where $\boldsymbol{X}_{\boldsymbol{\phi}}^k = [\phi(\boldsymbol{\kappa}^{k,1}), \phi(\boldsymbol{\kappa}^{k,2}), \cdots, \phi(\boldsymbol{\kappa}^{k,|\mathcal{J}^k|})]$ and $\boldsymbol{X}_{\boldsymbol{\mu}}^k = [\boldsymbol{\mu}^{k,1}, \boldsymbol{\mu}^{k,2}, \cdots, \boldsymbol{\mu}^{k,|\mathcal{J}^k|}]$, Eq. (18) can be rewritten as follows:

$$\begin{aligned} \begin{bmatrix} \tilde{\boldsymbol{\phi}}^{k,j} \\ \tilde{\boldsymbol{\mu}}^{k,j} \end{bmatrix} &\cong \begin{bmatrix} \mathbf{X}_{\boldsymbol{\phi}}^k \\ \mathbf{X}_{\boldsymbol{\mu}}^k \end{bmatrix} \mathbf{T}^k \left[\mathbf{X}_{\boldsymbol{\phi}}^{k\top}, \mathbf{X}_{\boldsymbol{\mu}}^{k\top} \right] \begin{bmatrix} \phi(\boldsymbol{\kappa}^{k,j}) \\ \boldsymbol{\mu}^{k,j} \end{bmatrix} \\ &\quad - \frac{1}{|\mathcal{J}^k|} \begin{bmatrix} \mathbf{X}_{\boldsymbol{\phi}}^k \\ \mathbf{X}_{\boldsymbol{\mu}}^k \end{bmatrix} \left(\mathbf{T}^k \left[\mathbf{X}_{\boldsymbol{\phi}}^{k\top}, \mathbf{X}_{\boldsymbol{\mu}}^{k\top} \right] \begin{bmatrix} \mathbf{X}_{\boldsymbol{\phi}}^k \\ \mathbf{X}_{\boldsymbol{\mu}}^k \end{bmatrix} - \mathbf{I}_{|\mathcal{J}^k| \times |\mathcal{J}^k|} \right) \mathbf{1}_{|\mathcal{J}^k|} \\ &= \begin{bmatrix} \boldsymbol{X}_{\boldsymbol{\phi}}^k \\ \boldsymbol{X}_{\boldsymbol{\mu}}^k \end{bmatrix} \boldsymbol{\psi}^{k,j}, \end{aligned} \tag{20}$$

where

$$\begin{aligned} \boldsymbol{\psi}^{k,j} &= \mathbf{T}^k \left(\boldsymbol{X}_{\boldsymbol{\phi}}^{k\top} \phi(\boldsymbol{\kappa}^{k,j}) + \boldsymbol{X}_{\boldsymbol{\mu}}^{k\top} \boldsymbol{\mu}^{k,j} \right) \\ &\quad - \frac{1}{|\mathcal{J}^k|} \left\{ \mathbf{T}^k \left(\boldsymbol{X}_{\boldsymbol{\phi}}^{k\top} \boldsymbol{X}_{\boldsymbol{\phi}}^k + \boldsymbol{X}_{\boldsymbol{\mu}}^{k\top} \boldsymbol{X}_{\boldsymbol{\mu}}^k \right) - \mathbf{I}_{|\mathcal{J}^k| \times |\mathcal{J}^k|} \right\} \mathbf{1}_{|\mathcal{J}^k|}. \end{aligned} \tag{21}$$

Therefore, $\|\boldsymbol{\mu}^{k,j} - \tilde{\boldsymbol{\mu}}^{k,j}\|^2$ in Eq. (15) becomes $\|\boldsymbol{\mu}^{k,j} - \boldsymbol{X}_{\boldsymbol{\mu}}^k \boldsymbol{\psi}^{k,j}\|^2$ from Eq. (20). If we use the linear kernel, $\tilde{\boldsymbol{\kappa}}^{k,j} = \tilde{\boldsymbol{\phi}}^{k,j}$, and $\|\boldsymbol{\kappa}^{k,j} - \tilde{\boldsymbol{\kappa}}^{k,j}\|$ becomes $\|\boldsymbol{\kappa}^{k,j} - \boldsymbol{X}_{\boldsymbol{\phi}}^k \boldsymbol{\psi}^{k,j}\|^2$. On the other hand, if we adopt the Gaussian kernel, $\|\boldsymbol{\kappa}^{k,j} - \tilde{\boldsymbol{\kappa}}^{k,j}\|^2$ in Eq. (15) becomes

$$\begin{aligned} \left\| \boldsymbol{\kappa}^{k,j} - \tilde{\boldsymbol{\kappa}}^{k,j} \right\|^2 &= -\sigma^2 \log \left\{ \phi(\boldsymbol{\kappa}^{k,j})^\top \phi(\tilde{\boldsymbol{\kappa}}^{k,j}) \right\} \\ &= -\sigma^2 \log \left\{ \frac{\phi(\boldsymbol{\kappa}^{k,j})^\top \phi(\boldsymbol{\kappa}^{k,j}) + \phi(\tilde{\boldsymbol{\kappa}}^{k,j})^\top \phi(\tilde{\boldsymbol{\kappa}}^{k,j}) - \|\phi(\boldsymbol{\kappa}^{k,j}) - \phi(\tilde{\boldsymbol{\kappa}}^{k,j})\|^2}{2} \right\} \\ &\cong -\sigma^2 \log \left\{ \frac{2 - \|\phi(\boldsymbol{\kappa}^{k,j}) - \tilde{\boldsymbol{\phi}}^{k,j}\|^2}{2} \right\}, \end{aligned} \tag{22}$$

where σ^2 is a parameter of the Gaussian kernel, and $\|\phi(\boldsymbol{\kappa}^{k,j}) - \tilde{\phi}^{k,j}\|^2$ in the above equation is obtained as

$$\begin{aligned}\|\phi(\boldsymbol{\kappa}^{k,j}) - \tilde{\phi}^{k,j}\|^2 &= \phi(\boldsymbol{\kappa}^{k,j})^\top \phi(\boldsymbol{\kappa}^{k,j}) + \tilde{\phi}^{k,j\top}\tilde{\phi}^{k,j} - 2\phi(\boldsymbol{\kappa}^{k,j})^\top \tilde{\phi}^{k,j} \\ &\cong 1 + \boldsymbol{\psi}^{k,j\top} \boldsymbol{X}_{\phi}^{k\ \top} \boldsymbol{X}_{\phi}^{k} \boldsymbol{\psi}^{k,j} - 2\phi(\boldsymbol{\kappa}^{k,j})^\top \boldsymbol{X}_{\phi}^{k} \boldsymbol{\psi}^{k,j}. \quad (23)\end{aligned}$$

Thus, Eq. (22) can be calculated as follows:

$$\left\|\boldsymbol{\kappa}^{k,j} - \tilde{\boldsymbol{\kappa}}^{k,j}\right\|^2 \cong -\sigma^2 \log\left\{\frac{1 - \boldsymbol{\psi}^{k,j\top} \boldsymbol{X}_{\phi}^{k\ \top} \boldsymbol{X}_{\phi}^{k} \boldsymbol{\psi}^{k,j} + 2\phi(\boldsymbol{\kappa}^{k,j})^\top \boldsymbol{X}_{\phi}^{k} \boldsymbol{\psi}^{k,j}}{2}\right\}. \quad (24)$$

Consequently, we can calculate Eq. (15) to perform the clustering.

Furthermore, Eq. (12) of ASIP-DIMS is calculated, and $\mathbf{z}_n^k$ in Eq. (13) is obtained as $\mathbf{z}_n^k = \mathbf{X}^k \mathbf{w}_n^k$, where

$$\mathbf{w}_n^k = \boldsymbol{\Xi}_n^k \mathbf{X}_n^{k\top} \left(\mathbf{x}_n^k - \frac{1}{|\mathcal{J}^k|}\mathbf{X}_n^k \mathbf{1}_{|\mathcal{J}^k|}\right) + \frac{1}{|\mathcal{J}^k|}\mathbf{1}_{|\mathcal{J}^k|}, \quad (25)$$

$$\boldsymbol{\Xi}_n^k = \mathbf{H}^k \mathbf{V}^k \left(\mathbf{V}_n^{k\top} \mathbf{H}^k \mathbf{X}_n^{k\top} \mathbf{M}_n \mathbf{X}^k \mathbf{H}^k \mathbf{V}^k\right)^{-1} \mathbf{V}_n^{k\top} \mathbf{H}^k. \quad (26)$$

In the above equations, $\mathbf{X}_n^k = \mathbf{M}_n \mathbf{X}^k$, and the singular value decomposition $\mathbf{X}_n^k \mathbf{H}^k \cong \mathbf{B}_n^k \boldsymbol{\Lambda}_n^k \mathbf{V}_n^{k\top}$ is satisfied, i.e., $\mathbf{B}_n^k \cong \mathbf{X}_n^k \mathbf{H}^k \mathbf{V}_n^k \boldsymbol{\Lambda}_n^{k-1}$ is satisfied.

Finally, Eq. (14) is rewritten as

$$C^k = \sum_{n=1}^{N} \frac{\|\boldsymbol{\kappa} - \tilde{\boldsymbol{\kappa}}_n^k\|^2 + \|(\mathbf{I}_{D\times D} - \mathbf{M})\mathbf{M}_n(\mathbf{x}_n^k - \mathbf{z}_n^k)\|^2}{\mathrm{rank}\,(\mathbf{M}_n)}, \quad (27)$$

where $\boldsymbol{\kappa}$ corresponds to the known component of f. Then, if we use the linear kernel, we can directly calculate $\|\boldsymbol{\kappa} - \tilde{\boldsymbol{\kappa}}_n^k\|^2$ as $\|\boldsymbol{\kappa} - \mathbf{X}_{\phi}^k \mathbf{w}_n^k\|^2$ since $\tilde{\boldsymbol{\kappa}}_n^k = \mathbf{X}_{\phi}^k \mathbf{w}_n^k$. Furthermore, if we use the Gaussian kernel, $\phi(\tilde{\boldsymbol{\kappa}}_n^k) \cong \mathbf{X}_{\phi}^k \mathbf{w}_n^k$, and

$$\begin{aligned}\left\|\boldsymbol{\kappa} - \tilde{\boldsymbol{\kappa}}_n^k\right\|^2 &= -\sigma^2 \log\left\{\phi(\boldsymbol{\kappa})^\top \phi(\tilde{\boldsymbol{\kappa}}_n^k)\right\} \\ &= -\sigma^2 \log\left\{\frac{2 - \|\phi(\boldsymbol{\kappa}) - \phi(\tilde{\boldsymbol{\kappa}}_n^k)\|^2}{2}\right\} \\ &\cong -\sigma^2 \log\left\{\frac{1 - \mathbf{w}_n^{k\top} \mathbf{X}_{\phi}^{k\ \top} \mathbf{X}_{\phi}^k \mathbf{w}_n^k + 2\phi(\boldsymbol{\kappa})^\top \mathbf{X}_{\phi}^k \mathbf{w}_n^k}{2}\right\}. \quad (28)\end{aligned}$$

Then we can calculate Eq. (14).

In this way, the proposed method enables the derivation of ASIP-DIMS using PCA and KPCA. For the following explanation, we call ASIP-DIMS using PCA and KPCA "PCA-ASIP-DIMS" and "KPCA-ASIP-DIMS", respectively. PCA-ASIP-DIMS under the condition of $K = 1$, $N = 1$ and $\mathbf{B}_n^k = \mathbf{B}^k$ become equivalent to the algorithm of the BPLP method [6]. In addition, KPCA-ASIP-DIMS under the condition of $N = 1$ become equivalent to our previously reported KPCA-based method [54].

B. Derivation Based on Sparse Representation

In this subsection, we show the specific derivation of ASIP-DIMS based on sparse representation. As shown in [30], since we can adaptively select the optimal signal atoms for representing the target patch f in sparse representation, we do not have to adopt the clustering scheme shown in II-A. From the training examples, we calculate their vectors and estimate the optimal dictionary matrix $\mathbf{B}$ ($\in \mathbf{R}^{D\times L}$) by using the KSVD algorithm [25], where L is the number of signal atoms. Furthermore, from this dictionary matrix, we calculate N kinds of dictionary matrices $\mathbf{B}_n = \mathbf{M}_n \mathbf{B}$ ($\in \mathbf{R}^{D\times L}; n = 1, 2, \cdots, N$).

By using the dictionary matrices $\mathbf{B}$ and $\mathbf{B}_n$ ($n = 1, 2, \cdots, N$), the vectors $\mathbf{x}$ and $\mathbf{y}_n$ in the previous section are approximated as follows:

$$\mathbf{x} \cong \mathbf{B}\mathbf{r}_n \quad \text{subject to} \quad \|\mathbf{r}_n\|_0 \le T, \quad (29)$$

$$\mathbf{y}_n \cong \mathbf{B}_n \mathbf{s}_n \quad \text{subject to} \quad \|\mathbf{s}_n\|_0 \le T, \quad (30)$$

where $\|\bullet\|_0$ represents the l_0-norm, and T corresponds to D^k in II-A. Note that the selected signal atoms and their corresponding coefficients for representing $\mathbf{x}$ will change in each nth inverse projection, and we use a subscript n for $\mathbf{r}$ in Eq. (29). The above two equations are respectively rewritten as

$$\mathbf{x} \cong \mathbf{B}\mathbf{E}_n \mathbf{p}_n, \quad (31)$$

$$\mathbf{y}_n \cong \mathbf{B}_n \mathbf{F}_n \mathbf{q}_n. \quad (32)$$

The vectors $\mathbf{p}_n$ and $\mathbf{q}_n$ ($\in \mathbf{R}^T$) respectively contain nonzero elements of $\mathbf{r}_n$ and $\mathbf{s}_n$ in Eqs. (29) and (30). Furthermore, $\mathbf{E}_n$ and $\mathbf{F}_n$ ($\in \mathbf{R}^{L\times T}$) are extraction matrices for obtaining basis vectors, i.e., signal atoms, used for approximating $\mathbf{x}$ and $\mathbf{y}_n$, respectively. Specifically, the elements of the extraction matrices $\mathbf{E}_n$ and $\mathbf{F}_n$ are one or zero. If mth ($m = 1, 2, \cdots, T$) nonzero element exists in lth ($l = 1, 2, \cdots, L$) element of $\mathbf{x}$, the (l, m)th element of $\mathbf{E}_n$ becomes one, and the other elements become zero. The extraction matrix $\mathbf{F}_n$ is also defined in the same way as $\mathbf{E}_n$. The calculation schemes of these extraction matrices are shown later. Then $\mathbf{r}_n = \mathbf{E}_n \mathbf{p}_n$ and $\mathbf{s}_n = \mathbf{F}_n \mathbf{q}_n$ are satisfied. Note that $\mathbf{B}\mathbf{E}_n$ and $\mathbf{B}_n \mathbf{F}_n$ correspond to $\mathbf{B}^k$ and $\mathbf{B}_n^k$, respectively, in the previous section. Since we do not use the clustering scheme in ASIP-DIMS based on sparse representation, k is removed in this subsection. In addition, since centering is not adopted in this sparse representation approach, we do not use $\boldsymbol{\nu}^k$ shown in the previous section.

From the above definitions, Eq. (12) based on sparse representation is rewritten as

$$\begin{aligned}\mathbf{x}_{n+1} &= (\mathbf{I}_{D\times D} - \mathbf{M}_n)\,\mathbf{B}\mathbf{E}_n \left(\mathbf{F}_n{}^\top \mathbf{B}_n{}^\top \mathbf{M}_n \mathbf{B}\mathbf{E}_n\right)^{-1} \mathbf{F}_n{}^\top \mathbf{B}_n{}^\top \mathbf{x}_n + \ \mathbf{M}_n \mathbf{x}_n \\ &= (\mathbf{I}_{D\times D} - \mathbf{M}_n)\,\mathbf{z}_n + \mathbf{M}_n \mathbf{x}_n \ \ (n = 1, 2, \cdots, N), \quad (33)\end{aligned}$$

where $\mathbf{z}_n$ corresponding to Eq. (13) is defined as

$$\mathbf{z}_n = \mathbf{B}\mathbf{E}_n \left(\mathbf{F}_n{}^\top \mathbf{B}_n{}^\top \mathbf{M}_n \mathbf{B}\mathbf{E}_n\right)^{-1} \mathbf{F}_n{}^\top \mathbf{B}_n{}^\top \mathbf{x}_n. \quad (34)$$

It should be noted that in order to obtain Eq. (33), we have to calculate the two extraction matrices $\mathbf{E}_n$ and $\mathbf{F}_n$ ($n = 1, 2, \cdots, N$). Since $\mathbf{s}_n = \mathbf{F}_n \mathbf{q}_n$ from Eqs. (30) and (32), $\mathbf{F}_n$ can be obtained by solving

$$\min_{\mathbf{s}_n} \left\|\mathbf{M}_n (\mathbf{x}_n - \mathbf{B}_n \mathbf{s}_n)\right\|^2 \quad \text{subject to} \quad \|\mathbf{s}_n\|_0 \le T \quad (35)$$

based on the OMP algorithm [69], [70]. On the other hand, we cannot calculate $\mathbf{E}_n$ in the same way as Eq. (35) since the elements in sub-component $\mathcal{U}_n$ are unknown. Therefore, the proposed method adopts the solution of

$$\min_{\mathbf{E}_n} \left\|\mathbf{M}_n \left\{\mathbf{I}_{D\times D} - \mathbf{B}\mathbf{E}_n \left(\mathbf{F}_n{}^\top \mathbf{B}_n{}^\top \mathbf{M}_n \mathbf{B}\mathbf{E}_n\right)^{-1} \mathbf{F}_n{}^\top \mathbf{B}_n{}^\top\right\} \mathbf{x}_n\right\|^2 \quad \text{subject to} \quad \|\mathbf{r}_n\|_0 \le T \quad (36)$$

from $\mathbf{r}_n = \mathbf{E}_n \mathbf{p}_n$. In Eq. (36), this criterion represents the error caused in the other component $\bar{\mathcal{U}}_n$ by solving nth subproblem. Therefore, the extraction matrix $\mathbf{E}_n$ is determined in

such a way that this error becomes minimum. In the proposed method, the solution of Eq. (36) can be obtained by the basis selection algorithm in [30].

In this way, the proposed method can estimate the missing components by ASIP-DIMS based on sparse representation. As described above, ASIP-DIMS based on sparse representation do not adopt the clustering scheme, and we do not have to use Eq. (14) for this. However, as shown in the following section, if each restored pixel has multiple estimation results, we have to calculate Eq. (14) for finding the final output from these results. Thus, Eq. (14) is calculated as

$$\tilde{C} = \sum_{n=1}^{N} \frac{\|\mathbf{M}_n(\mathbf{x}_n - \mathbf{z}_n)\|^2}{\mathrm{rank}(\mathbf{M}_n)}. \quad (37)$$

As described above, the derivation of ASIP-DIMS based on sparse representation becomes feasible. For the following explanation, we call ASIP-DIMS using sparse representation "SP-SIP-DIMS". Since we do not have to perform clustering and adaptive selection of the optimal subspaces in this subsection, we use the notation SP-SIP-DIMS instead of SP-ASIP-DIMS. When $N = 1$, SP-SIP-DIMS become equivalent to our previously reported method (SP-SIP shown in the following section), inverse projection via sparse representation (IPVSR) [30]. Thus, the biggest difference from this method is the use of division into multiple sub-problems.

Algorithm 1 : Complete image inpainting algorithm based on ASIP-DIMS

Input: Image including missing areas Ω.
Output: Restored image.

1: Clip training known patches at the same interval from the input image to obtain their vectors.
2: **while** $|\Omega| \neq \emptyset$ **do**
3: $\forall \mathbf{p} \in \delta\Omega$, compute patch priority $P_{\mathbf{p}}$ based on [4].
4: Select the target patch f centered at $\mathbf{p}$ whose patch priority has the maximum value.
5: Calculate $\boldsymbol{\kappa}$ containing only known intensities within the target patch f, and define $\boldsymbol{\mu}$, whose elements correspond to missing intensities of f, $\mathbf{M}$ and $\mathbf{M}_n$.
{Clustering Part}
6: **while** clustering results $\mathbf{x}^{k,j}$ change **do**
7: Assign training patches to the optimal clusters based on Eq. (1).
8: $\forall k \in \{1, 2, \cdots, K\}$, update basis matrices $\mathbf{B}^k$ and $\mathbf{B}^k_n$ from $\mathbf{x}^{k,j}$ ($j \in \mathcal{J}^k$) belonging to cluster k.
9: **end while**
{Missing Intensity Estimation Part}
10: $\forall k \in \{1, 2, \cdots, K\}$,
11: **while** iterations are less than a predetermined number of times **do**
12: Update the estimation result $\mathbf{x}^k_n$ using cluster k based on Eq. (12).
13: **end while**
14: Select the final estimation result $\mathbf{x}^{k^{\mathrm{opt}}}_N$ of cluster k^{opt} minimizing C^k in Eq. (14).
15: Interpolate missing pixels within the target patch f from the estimation result $\mathbf{x}^{k^{\mathrm{opt}}}_N$.
16: Update the confidence factor of the target patch f for the following patch priority calculation according to [4].
17: **end while**

IV. Experimental Results

In this section, we present results of application of the proposed method to two kinds of image processing tasks, i.e., image inpainting and SR, for verifying its effectiveness. Experiments on image inpainting and SR are described in IV-A and IV-B, respectively.

A. Results of Image Inpainting

Experimental results for image inpainting are presented in this subsection. Image inpainting is achieved by simply regarding known and unknown components as known original intensities and unknown missing intensities, respectively. Specifically, known and unknown intensities within the target patch f correspond to $\mathcal{K}$ and $\mathcal{U}$, respectively, in the previous sections. Furthermore, training patches are clipped in the same interval from known parts within the target image. Then image inpainting based on ASIP-DIMS becomes feasible. We show the complete algorithm of image inpainting based on ASIP-DIMS in Algorithm 1. In Algorithm 1, Ω represents the whole missing areas within the target image, and $\partial\Omega$ represents their fill-fronts, these notations being based on [4]. Furthermore, $\mathbf{p}$ is the axis of a pixel. Note that when we adopt ASIP-DIMS using sparse representation, which does not need clustering, we can perform image inpainting by assuming $K = 1$, and this is the same for SR shown in the next subsection. In our method, the patch size was set to 15×15 pixels. In addition, we clipped patches f including missing areas according to the patch priority determined by [4] and performed their reconstruction to recover all of the missing areas.

In this experiment, we first prepared eight test images shown in Fig. 2. For Images 1–3 in this figure, we added text regions, which corresponded to missing areas, to these test images as shown in Fig. 3 with the assumption that the positions of the missing areas had been previously known. For the corrupted test images, we applied the three proposed types of ASIP-DIMS, i.e., PCA-ASIP-DIMS, KPCA-ASIP-DIMS and SP-SIP-DIMS. Furthermore, for comparison, we utilized the methods in [31], [33] and [36]. The method in [33] adopts neighboring embedding, and it can be regarded as a recently proposed method. The methods in [31] and [36] are state-of-the-art methods for image inpainting. Results of restoration obtained by the proposed methods, PCA-ASIP-DIMS, KPCA-ASIP-DIMS and SP-SIP-DIMS, and the above previously reported methods are shown in Figs. 4–6. Due to the limitation of space, we show a comparison only with the state-of-the-art methods[2]. In PCA-ASIP-DIMS and KPCA-ASIP-DIMS, we simply set the number of clusters to four ($K = 4$). In KPCA-ASIP-DIMS, the parameter of the Gaussian kernel, σ^2, was simply determined as the variance of $\|\phi(\boldsymbol{\kappa}^{k,i}) - \phi(\boldsymbol{\kappa}^{k,j})\|$ ($i, j \in \mathcal{I}; k = 1, 2, \cdots, K$). In SP-SIP-DIMS, the clipping interval of known patches was set to values in such a way that L became the closest to 3000. Furthermore, D^k and T were determined in such a way that the condition for calculating the pseudo-inverse was satisfied. The number of sub-problems, N, was set to four. Note that for color images, our methods and

[2]Other inpainting examples obtained by the proposed method and comparative methods in [4], [6], [24], [29], [31], [33] and [36] for the eight test images in Fig. 2 are shown in the supplemental materials.

(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4

(e) Image 5 (f) Image 6 (g) Image 7 (h) Image 8

Fig. 2. Eight test images used for verifying the inpainting performance of our method. The sizes of Images 1–8 are 640×480 pixels, 480×360 pixels, 640×480 pixels, 640×480 pixels, 480×360 pixels, 640×480 pixels, 480×360 pixels and 480×360 pixels, respectively.

(a) Corrupted image of Image 1 (b) Corrupted image of Image 2 (c) Corrupted image of Image 3

Fig. 3. Corrupted images obtained by adding missing areas to Images 1-3 in Fig. 2, respectively.

(a) Original image (b) Corrupted image (c) Reference [33]

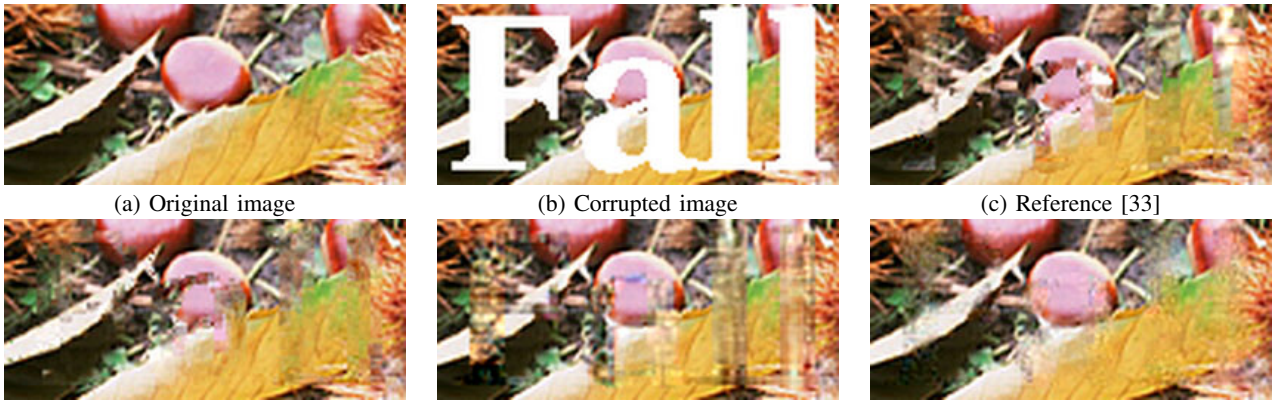
(d) Reference [31] (e) Reference [36] (f) Ours (PCA-ASIP-DIMS)

Fig. 4. Results of inpainting obtained by the previously reported methods [33], [31], [36] and our method for Image 1.

the previously reported methods calculated vectors that contain RGB values within patches and performed their restoration. Note that in our method, the initial RGB values within the missing areas ($\mathcal{U}$) of the target patch f were respectively set to the average RGB values within the known areas ($\mathcal{K}$). In the experiment, we simply set the number of iterations for updating $\mathbf{x}_n^k$ ($n = 1, 2, \cdots, N$) in Eq. (12) to ten, and this was the same for SR.

The results presented in Figs. 4–6 show that the proposed methods enable restoration with a high level of image representation performance. The biggest contribution of our method is the introduction of division into multiple sub-problems, and this enables the use of higher-dimensional subspaces to improve image representation performance. Since we also perform adaptive selection of the optimal subspaces in this framework, each missing texture tends not to be affected by other different textures.

Next, we show results of quantitative evaluation. We used the eight test images shown in Fig. 2 and randomly added missing blocks of 8×8 pixels in size with changes in the ratio of missing pixels. Figures 7 and 8 show the relationship between the ratio of missing pixels and the SSIM index calculated from the restored image, the details of which are shown below. It is well known that the MSE (PSNR) and its variants cannot successfully reflect visual image quality [71]. Since the SSIM index [72] is one of the representative criteria for



(a) Original image (b) Corrupted image (c) Reference [33]

(d) Reference [31] (e) Reference [36] (f) Ours (KPCA-ASIP-DIMS)

Fig. 5. Results of inpainting obtained by the previously reported methods [33], [31], [36] and our method for Image 2.

(a) (b) (c) (d) (e) (f)

Fig. 6. Results of inpainting obtained by the previously reported methods [33], [31], [36] and our method for Image 3. Images shown in (a)–(f) are respectively original image, corrupted image, results of [33], [31], [36] and our method (SP-SIP-DIMS), respectively.

measuring visual image quality, we adopted the SSIM index as the criterion for quantitative evaluation in this experiment. In addition to the previously reported methods in [31], [33], [36], we used several comparative methods that do not use "division into multiple sub-problems" or "adaptive selection of the optimal subspaces" (clustering scheme). As shown in Table I, the comparative methods we used were SIP (subspace-based inverse projection), SIP-DIMS (subspace-based inverse projections via division into multiple sub-problems) and ASIP (adaptive subspace-based inverse projections). The difference between ASIP-DIMS and SIP-DIMS is whether multiple clusters are used or not, the difference between ASIP and SIP being the same as the difference between ASIP-DIMS and SIP-DIMS. This means that when using SIP and SIP-DIMS, $K = 1$. Thus, we do not perform clustering of training patches, and only one subspace is calculated from all of the training patches. On the other hand, when using ASIP and ASIP-DIMS, we perform assignment of the training patches into K clusters and prepare K subspaces. Then the target patch including missing components is restored with adaptive selection of the optimal subspace among these K subspaces. Next, we explain the difference between ASIP-DIMS and ASIP, the difference between SIP-DIMS and SIP being the same as the difference between ASIP-DIMS and ASIP. The difference between ASIP-DIMS and ASIP is whether missing components $\mathcal{U}$ are divided into sub-components $\mathcal{U}_n$ ($n = 1, 2, \cdots, N$) or not. This means that when using SIP and ASIP, $N = 1$. Thus, we do not perform any division of missing components $\mathcal{U}$. On the other hand, when using SIP-DIMS and ASIP-DIMS, the missing components $\mathcal{U}$ are divided into N

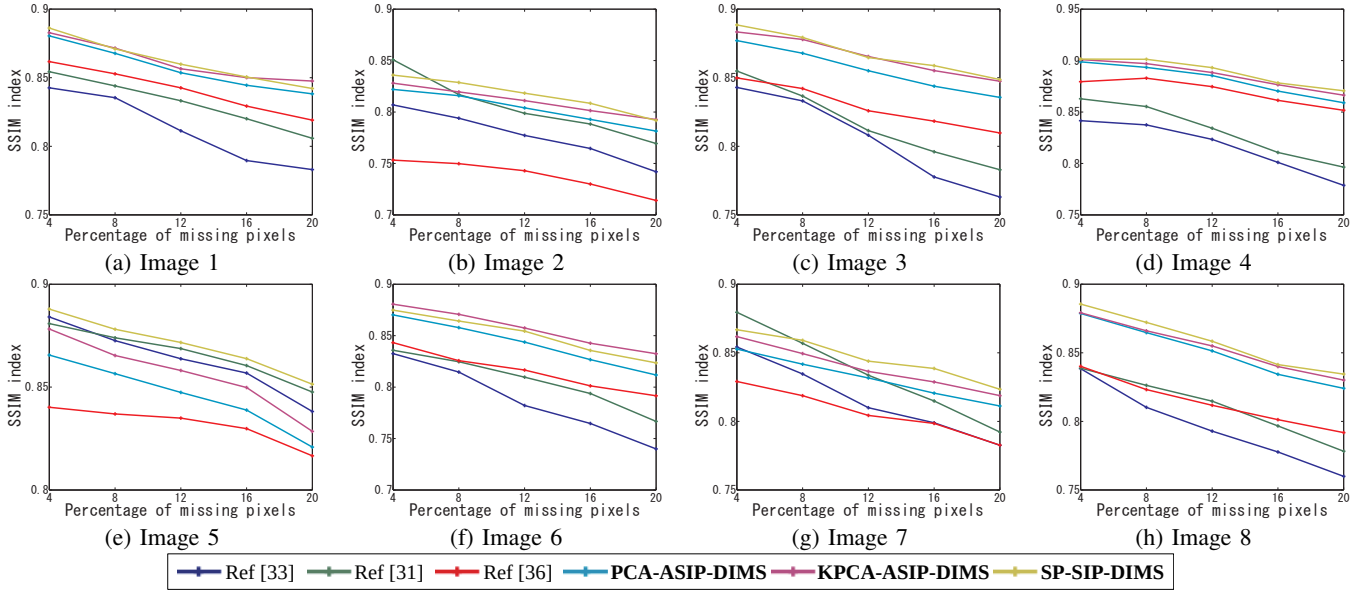


Fig. 7. Relationship between ratio of missing pixels and SSIM index of the restoration results obtained by each method: (a)–(h) respectively show the results obtained from Images 1–8 shown in Fig. 2.

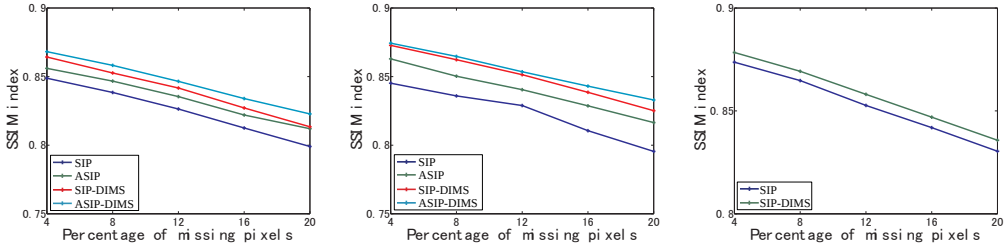


Fig. 8. Performance comparison between SIP, ASIP, SIP-DIMS and ASIP-DIMS. These graphs show the relationship between ratio of missing pixels and SSIM index of the restoration results obtained by each method. These results are average values obtained from Images 1–8.

sub-components $\mathcal{U}_n$, and restoration of these sub-components is performed. Note that when using sparse representation, we do not have to use the clustering scheme, and ASIP and ASIP-DIMS become equivalent to SIP and SIP-DIMS, respectively. Figure 7 shows a comparison of results obtained by using the proposed methods (PCA-ASIP-DIMS, KPCA-ASIP-DIMS and SP-SIP-DIMS) and the previously reported methods[3]. From the obtained results, we can see that the performance of the proposed methods is better than that of the previously reported methods. Among the proposed methods, SP-SIP-DIMS tended to give better results since the optimal bases can be selected one-by-one more flexibly than they can with PCA-ASIP-DIMS and KPCA-ASIP-DIMS. In Fig. 8, we also show a comparison of results obtained by using the proposed ASIP-DIMS and the other three comparative methods, SIP, ASIP and SIP-DIMS, shown in Table I. From these results, we can see that division into multiple sub-problems and adaptive selection of the optimal subspaces enable successful image inpainting. Furthermore, the introduction of division into multiple sub-problems tends to result in greater improvement than that with adaptive selection of the optimal subspaces. In this way, our method realizes successful image inpainting subjectively and quantitatively.

[3]Results of a comparison with not only state-of-the-art methods in [31], [33] and [36] but also other traditional methods in [4], [6], [24] and [29] are also shown in the supplemental materials.

TABLE I
RELATIONSHIP BETWEEN ASIP-DIMS AND THREE OTHER COMPARATIVE METHODS, SIP, ASIP AND SIP-DIMS, THAT DO NOT USE DIVISION INTO MULTIPLE SUB-PROBLEMS OR ADAPTIVE SELECTION OF THE OPTIMAL SUBSPACES (CLUSTERING SCHEME).

	Non-clustering ($K = 1$)	Clustering
Single problem ($N = 1$)	SIP	ASIP
Sub-problem division	SIP-DIMS	**ASIP-DIMS**

B. Results of Super-resolution

Super-resolution is achieved by regarding LR patches and their corresponding HR patches as $\mathcal{K}$ and $\mathcal{U}$, respectively, in the previous sections. Specifically, given a target image (Fig. 9(a)), we first regard patches clipped from this image as training HR patches. Next, patches clipped from the blurred target image (Fig. 9(b)) obtained by applying a low-pass filter to the target image (Fig. 9(a)) are regarded as training LR patches. We use the well-known Lanczos filter as a low-pass filter. Furthermore, by applying up-sampling to the target image (Fig. 9(a)), we obtain a blurred enlarged image (Fig.

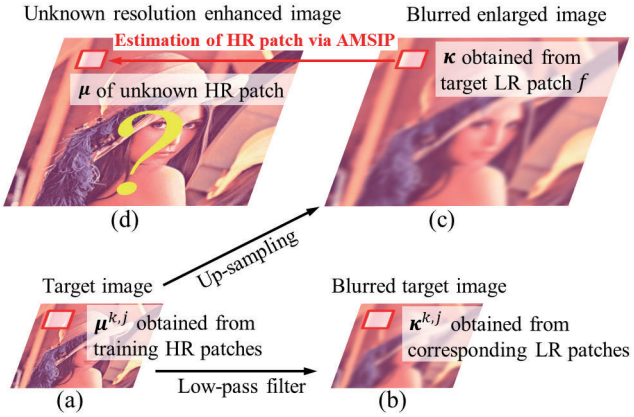


Fig. 9. Overview of the SR based on the proposed method.

9(c)). Then, from target patches clipped from the blurred enlarged image (Fig. 9(c)), we perform estimation of their corresponding HR patches to obtain an image with enhanced resolution (Fig. 9(d)). The complete algorithm of SR based on ASIP-DIMS is shown in Algorithm 2. In our method, the patch size was set to 8×8 pixels, and SR was performed in a raster scanning order for which the sliding interval was set to half the size of the patches. Since multiple estimation results are obtained for each pixel, we output the optimal results that minimize the criterion shown in Eq. (14). Furthermore, in our method, $K = 8$. Other parameter settings of the proposed method were performed in the same way as that shown in the previous subsection.

In this experiment, we first prepared nine test images shown in Fig. 10. For these test images, we performed SR by using the proposed methods and the previously reported methods in [61], [62] and [64]. These methods were selected since they are state-of-the-art methods[4]. Note that in all of the methods, SR was performed for only the luminance components since this scheme has been adopted in many studies in which the target is SR of color images. In our method, the initial intensity values of the HR patch ($\boldsymbol{\mu}$) are the same as those of the target LR patch ($\boldsymbol{\kappa}$). The results obtained by the proposed methods and the previously reported methods are shown in Figs. 11 and 12. In these figures, the magnification factor was set to four, i.e., we restored original size images from their quarter size images. Thus, the target images to be enlarged were 160×120 pixels. Furthermore, these images tended to include small objects, and we adopted such difficult SR problems to make the difference between the performance of our method and that of the other methods clearer. From the results, we can see that the proposed method enables successful SR with maintenance of sharpness and prevention of other artifacts. Compared to the problem of image inpainting, that of the SR is more difficult since the missing components tend to be larger. Therefore, the use of division into multiple sub-problems becomes important for solving this difficult problem.

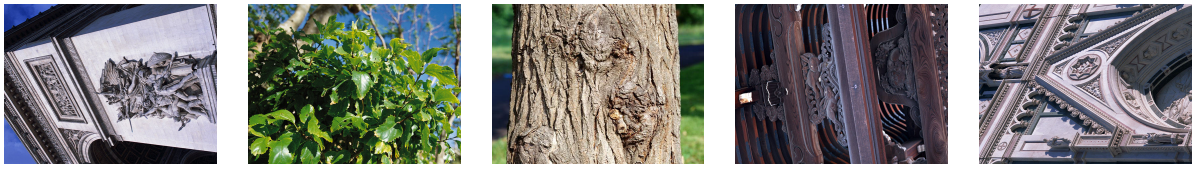
[4]Due to the limitation of space, we only performed a comparison between our method and state-of-the-art methods in [61], [62] and [64]. Examples of SR obtained by our method and the methods in [54], [57], [58], [60], [61], [62] and [64] are shown in the supplemental materials.

Algorithm 2 : Complete SR algorithm based on ASIP-DIMS

Input: Target image in Fig. 9(a).
Output: Result of SR in Fig. 9(d).

Calculate the blurred target image (Fig. 9(b)) from the target image (Fig. 9(a)) by using a low-pass filter.
Clip training HR patches at the same interval from the target image (Fig. 9(a)) to obtain their intensity vectors corresponding to $\boldsymbol{\mu}^{k,j}$.
Clip corresponding training LR patches from the blurred target image (Fig. 9(b)) to obtain their intensity vectors corresponding to $\boldsymbol{\kappa}^{k,j}$.
Define $\mathbf{x}^{k,j}$ from pairs of $\boldsymbol{\mu}^{k,j}$ and $\boldsymbol{\kappa}^{k,j}$ and $\mathbf{M}$ and $\mathbf{M}_n$
{Clustering Part}
while clustering results $\mathbf{x}^{k,j}$ change **do**
    Assign training patches to the optimal clusters based on Eq. (1).
    $\forall k \in \{1, 2, \cdots, K\}$, update basis matrices $\mathbf{B}^k$ and $\mathbf{B}_n^k$ from $\mathbf{x}^{k,j}$ ($j \in \mathcal{J}^k$) belonging to cluster k.
end while
{HR Patch Estimation Part}
Calculate the blurred enlarged image (Fig. 9(c)) by up-sampling the target image (Fig. 9(a)).
for each image patch f in the blurred enlarged image (Fig. 9(c)) **do**
    $\forall k \in \{1, 2, \cdots, K\}$,
    while iterations are less than a predetermined number of times **do**
        Update the estimation result $\mathbf{x}_n^k$ using cluster k based on Eq. (12).
    end while
    Select the final estimation result $\mathbf{x}_N^{k^{\text{opt}}}$ of cluster k^{opt} minimizing C^k in Eq. (14).
    if pixels within f have already included previously estimated intensities **then**
        if $C^{k^{\text{opt}}}$ calculated for the target patch f in Eq. (14) is smaller than those of the previously estimated intensities **then**
            Replace the previously estimated intensities with the intensities estimated for the current target patch f.
        end if
    end if
end for

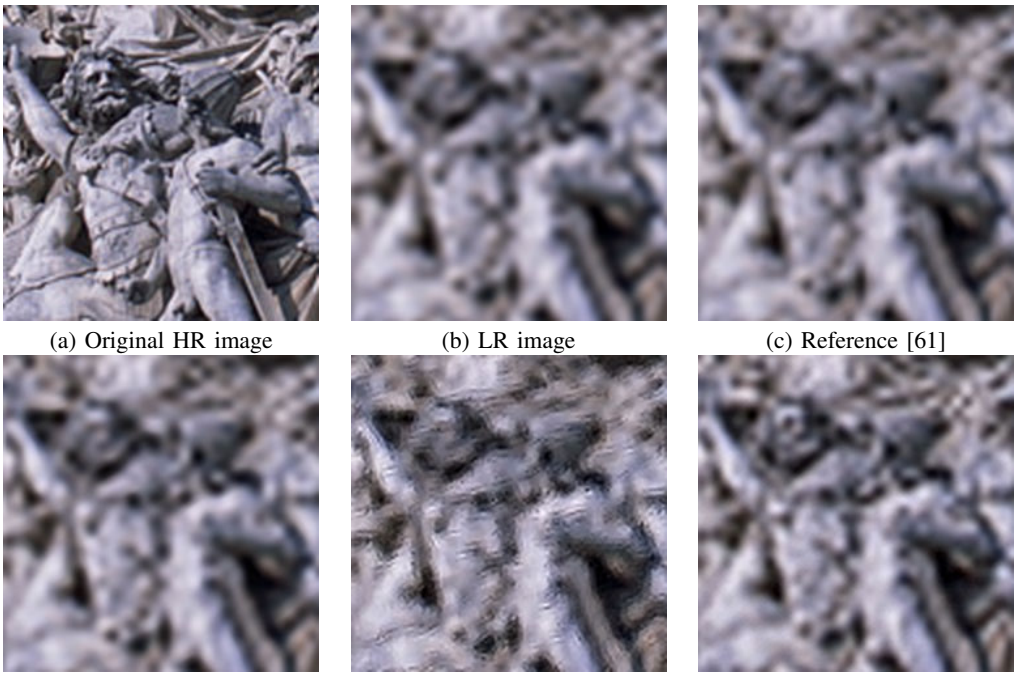
Finally, we show quantitative evaluation results. In this experiment, we reduced the size of the nine images in Fig. 10 to a quarter size and restored their original HR images by the proposed methods, the previously reported methods in [54], [57], [58], [60], [61], [62] and [64] and the comparative methods shown in Table I. Table II shows results for the SSIM index obtained from the estimated HR images. From the obtained results, we can see that the proposed method enables successful SR, and a tendency similar to that shown in the previous subsection can be found. Specifically, the improvement obtained by division into multiple sub-problems is greater than that obtained by selection of the optimal cluster. On the other hand, in some results, the SSIM index does not perfectly reflect perceptual image quality, e.g., results for Image 2. This indicates the necessity of image quality measures that are suitable for verifying SR performance, and this is the same for image inpainting.



(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4 (e) Image 5

(f) Image 6 (g) Image 7 (h) Image 8 (i) Image 9

Fig. 10. Nine test images used for verifying the SR performance of our method. The sizes of Images 1–9 are all 640×480 pixels.



(a) Original HR image (b) LR image (c) Reference [61]

(d) Reference [62] (e) Reference[64] (f) Ours (KPCA-ASIP-DIMS)

Fig. 11. Results of SR obtained by the previously reported methods [61], [62], [64] and our method for Image 1.

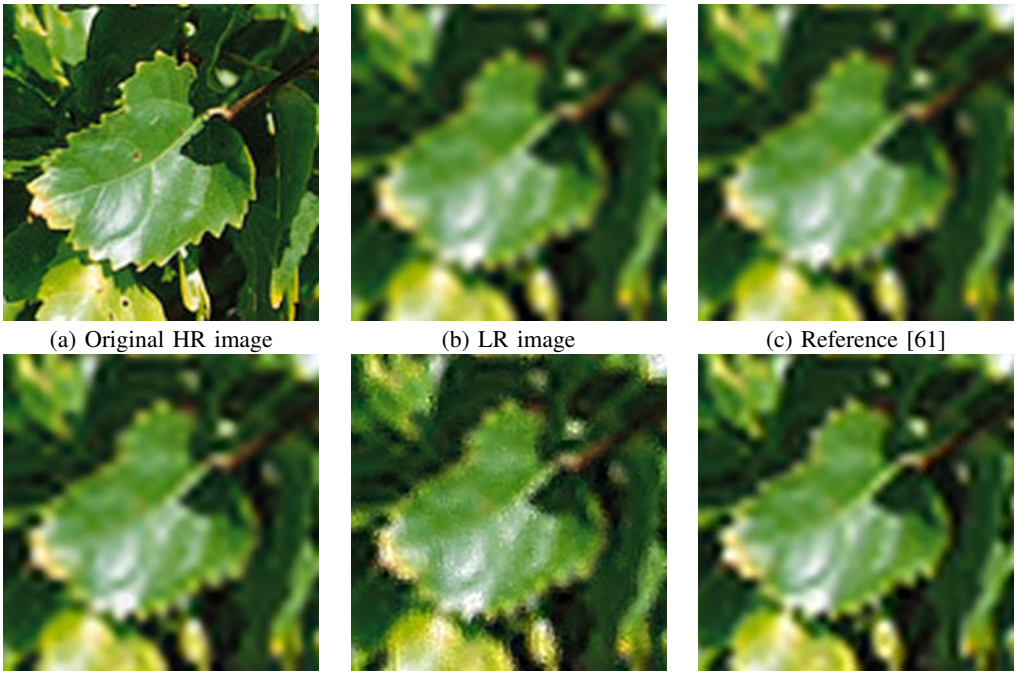
V. Conclusions

In this paper, we have presented ASIP-DIMS for missing image data restoration. The proposed method improves image representation performance by dividing the target problem into multiple sub-problems and solves these sub-problems by the projection into the subspaces. Furthermore, the proposed method enables selection of optimal subspaces for adaptively restoring missing image data. In this approach, we monitor errors caused by solving these sub-problems to realize the determination of optimal subspaces. Consequently, these approaches lead to successful restoration of missing image data.

In this paper, we present three types of ASIP-DIMS, i.e., PCA-ASIP-DIMS, KPCA-ASIP-DIMS and SP-SIP-DIMS, and verify the performance of each of these proposed methods by applying them to two image processing tasks, image inpainting and SR. From the obtained results, it can be seen that the proposed approaches achieve successful missing image data restoration.

Finally, we show the future work of our study. In this paper, we cannot mathematically prove the convergence of the proposed method. However, this is important for guaranteeing the reliability of the proposed method. Although degradation due to this problem did not occur in the results of our method in the experiments, it is necessary to mathematically show the condition of the convergence. Since the above point is the theoretical limitation of this paper, it should be addressed in our future work.

In this study, we manually set some parameters such as the number of sub-problems and the number of clusters. It is desirable for these values to be automatically determined from target images. Furthermore, since the proposed method needs to calculate several subspaces and perform the restoration for each patch, its computation cost is high. For example, when using PCA-ASIP-DIMS, KPCA-ASIP-DIMS and SP-SIP-DIMS, their average computation times for image inpaint-



(a) Original HR image (b) LR image (c) Reference [61]

(d) Reference [62] (e) Reference[64] (f) Ours (SP-SIP-DIMS)

Fig. 12. Results of SR obtained by the previously reported methods [61], [62], [64] and our method for Image 2.

ing of images that are 640×480 pixels and include 20% missing blocks (8×8 pixels) are 2.26×10^2 sec, 2.65×10^2 sec and 3.20×10^2 sec, respectively. In addition, the average computation times for obtaining the results of Table I in SR are 32.5 sec, 2.72×10^2 sec and 1.55×10^2 sec, respectively. The experiments were performed on a personal computer using Intel(R) Core(TM) i7 950 CPU 3.06 GHz with 8.0 Ggytes RAM. The implementation was performed by using Matlab. It is necessary to develop an algorithm with lower computation cost by replacing each procedure with a high-speed procedure. For example, it has been reported that the computation cost of the KSVD algorithm used in SP-SIP-DIMS is high, and this problem can be solved by much faster versions such as those in [73] and [74]. These issues will be the subject of subsequent reports.

Appendix

Since many mathematical notations are included in this paper, we show their brief explanations in this Appendix. The main mathematical symbols and their corresponding explanations are shown in Table III.

References

[1] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, "Image inpainting," *Proc. ACM SIGGRAPH Conference on Computer Graphics*, pp. 417–424, 2000.

[2] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. on Image Processing*, vol.10, no.8, pp.1200–1211, 2001.

[3] I. Drori, D. Cohen-Or, and H. Teshurun, "Fragment-based image completion," *Proc. ACM SIGGRAPH*, pp. 303–312, 2003.

[4] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[5] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[6] T. Amano and Y. Sato, "Image interpolation using BPLP method on the eigenspace," *Systems and Computers in Japan*, vol.38, no.1, pp.87–96, 2007.

[7] K. I. Kim, M. O. Franz , B. Schölkoph, "Iterative kernel principal component analysis for image modeling," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.27, no.9, pp. 1351–1366, 2005.

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

[9] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation ," *IEEE Trans. on Image Processing*, vol. 19, no. 11, 2010.

[10] C. Guillemot and O. Le Meur, "Image inpainting: Overview and recent advances," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 127–144, 2014.

[11] B. Shen, W. Hu, Y. Zhang and Y.-J. Zhang, "Image inpainting via sparse representation," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 697–700, 2009.

[12] S. Masnou and J. Morel, "Level lines based disocclusion," *Proc. IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 259–263, 1998.

[13] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation,* vol.12, no.4, pp.436–449, 2001.

[14] A. Rares, M. J. T. Reinders, and J. Biemond, "Edge-based image restoration," *IEEE Trans. on Image Processing,* vol.14, no.10, pp. 1454–1468, 2005.

[15] D. Z. M.-F. Auclair-Fortier, "A global approach for solving evolutive heat transfer for image denoising and inpainting," *IEEE Trans. on Image Processing*, vol. 15, no.9, pp. 2558–2574, 2006

[16] M. Bertalmio, "Strong-continuation, contrast-invariant inpainting with a third-order optimal PDE," *IEEE Trans. on Image Processing*, vol. 15, no. 7, pp. 1934–1938, 2006.

[17] M. Wang, B. Yan and K. N. Ngan, "An efficient framework for image/video inpainting," *Signal Processing: Image Communication*, vol. 28, no. 7, pp. 753–762, 2013.

[18] A. A. Efros and T. K. Leung, "Texture synthesis by nonparametric sampling," *Proc. IEEE International Conference on Computer Vision,* Corfu, Greece, pp.1033–1038, 1999.

TABLE II
PERFORMANCE COMPARISON (SSIM) OF THE PROPOSED METHOD, PREVIOUSLY REPORTED METHODS AND COMPARATIVE METHODS.

	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9	Average
Reference [58]	0.7923	0.8131	0.6285	0.8275	0.7559	0.7551	0.8772	0.7006	0.7940	0.7716
Reference [57]	0.8182	0.7768	0.6575	0.8713	0.7874	0.7620	0.8587	0.7591	0.7325	0.7804
Reference [54]	0.7819	0.8238	0.6641	0.8629	0.7629	0.7828	0.8957	0.7570	0.8502	0.7979
Reference [60]	0.7763	0.8029	0.6787	0.8574	0.7824	0.7650	0.8936	0.7762	0.8697	0.8002
Reference [61]	0.8231	0.8420	0.6628	0.8714	0.7850	0.7874	0.9064	0.7531	0.8465	0.8086
Reference [62]	0.8354	**0.8519**	0.6800	0.8743	0.8011	0.8037	0.9094	0.7621	0.8459	0.8182
Reference [64]	0.8225	0.8410	0.6747	0.8726	0.7841	0.7770	0.9090	0.7573	0.8566	0.8105
PCA-SIP	0.7741	0.7655	0.6348	0.8374	0.7234	0.7025	0.8697	0.6894	0.8110	0.7564
KPCA-SIP	0.7823	0.7932	0.6266	0.8110	0.7645	0.7457	0.8621	0.6985	0.8097	0.7660
PCA-ASIP	0.7773	0.7723	0.6457	0.8421	0.7331	0.7140	0.8733	0.6931	0.8152	0.7629
KPCA-ASIP	0.8040	0.8141	0.6583	0.8367	0.7879	0.7758	0.8839	0.7322	0.8368	0.7922
SP-SIP	0.8232	0.8341	0.6754	0.8663	0.7909	0.7846	0.8998	0.7428	0.8291	0.8051
PCA-SIP-DIMS	0.8023	0.8241	0.6523	0.8263	0.7733	0.7774	0.8768	0.7068	0.7782	0.7797
KPCA-SIP-DIMS	0.8251	0.8353	0.6795	0.8672	0.7935	0.7874	0.9005	0.7452	0.8301	0.8071
PCA-ASIP-DIMS	0.8291	0.8481	0.6817	0.8580	0.8014	0.8034	0.8970	0.7452	0.8127	0.8085
KPCA-ASIP-DIMS	0.8233	0.8245	**0.7060**	0.8748	0.8148	0.7933	0.9107	**0.7824**	**0.8849**	0.8239
SP-SIP-DIMS	**0.8386**	0.8486	0.7009	**0.8754**	**0.8254**	**0.8145**	**0.9127**	0.7810	0.8729	**0.8300**

TABLE III
MAIN MATH SYMBOLS USED IN OUR PAPER AND THEIR BRIEF EXPLANATIONS.

$\mathbf{B}^k$	Basis matrix optimal for representing $\mathbf{x}^{k,j}$ of cluster k (corresponding to $\mathbf{B}$ in III-B)
$\mathbf{B}_n^k$	Basis matrix optimal for representing $\mathbf{M}_n\mathbf{x}^{k,j}$ of cluster k (corresponding to $\mathbf{B}_n$ in III-B)
C	Criterion for clustering training patches
C^k	Criterion for classifying a target patch f into the optimal cluster (corresponding to $\tilde{C}$ in III-B)
$D^\bullet$	Dimension of $\bullet$
f	Target patch including missing components
$f^{k,j}$	jth training patches belonging to cluster k
$\mathbf{I}_{\bullet\times\bullet}$	Identity matrix of size $\bullet\times\bullet$
$\mathcal{J}^k$	Set of indices for training patches $f^{k,j}$ belonging to cluster k
$\mathbf{M}$	Diagonal matrix removing elements corresponding to $\mathcal{U}$
$\mathbf{M}_n$	Diagonal matrix removing elements corresponding to sub-component $\mathcal{U}_n$
$\mathbf{p}^k$	Coefficient vector for representing $\mathbf{x}$ using $\mathbf{B}^k$ (corresponding to $\mathbf{p}_n$ in III-B)
$\mathbf{q}_n^k$	Coefficient vector for representing $\mathbf{y}_n$ using $\mathbf{B}_n^k$ (corresponding to $\mathbf{q}_n$ in III-B)
$\mathbf{x}$	Original vector of f
$\mathbf{x}^{k,j}$	Vector of $f^{k,j}$
$\mathbf{z}_n^k$	Vector including final estimation results of nth sub-component by cluster k
$\mathbf{y}$	Corrupted vector ($=\mathbf{M}\mathbf{x}$) of f
$\mathbf{y}_n$	Corrupted vector ($=\mathbf{M}_n\mathbf{x}$) of f
$\mathcal{K}$	Known components in f
$\boldsymbol{\kappa}$	Vector of $\mathcal{K}$ in f
$\boldsymbol{\kappa}^{k,j}$	Vector corresponding to $\mathcal{K}$ in $f^{k,j}$
$\mathcal{U}$	Unknown components in f
$\mathcal{U}_n$	nth sub-components of $\mathcal{U}$
$\boldsymbol{\mu}$	Vector of $\mathcal{U}$ in f
$\boldsymbol{\mu}^{k,j}$	Vector corresponding to $\mathcal{U}$ in $f^{k,j}$
$\boldsymbol{\nu}^k$	Mean vector of $\mathbf{x}^{k,j}$ ($j\in\mathcal{J}^k$)
$\mathbf{1}_\bullet$	$\bullet$-dimensional vector whose elements are one
$\mathbf{0}_\bullet$	$\bullet$-dimensional zero vector

[19] T. H. Kwok, H. Sheung, and C. C. L. Wang, "Fast query for exemplar-based image completion," *IEEE Trans. on Image Processing*, vol. 19, no. 12, pp. 3106–3115, 2010.

[20] P. Buyssens, M. Daisy, D. Tschumperle and O. Lezoray, "Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions," *IEEE Trans. on Image Processing*, vol. 24, no. 6, pp. 1809–1824, 2015.

[21] T. Ruzic and A. Pizurica, "Context-aware patch-based image inpainting using Markov random field modeling," *IEEE Trans. on Image Processing*, vol. 24, no. 1, pp. 444–456, 2015.

[22] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola, "Input space versus feature space in kernel-based methods," *IEEE Trans. on Neural Networks*, vol.10, no.5, pp.1000–1017, 1999.

[23] T. Ogawa, M. Haseyama, "POCS-based texture reconstruction method using clustering scheme by kernel PCA," *IEICE Trans. Fundamentals*, vol. E90-A, no. 8, pp. 1519–1527, Aug. 2007.

[24] T. Ogawa, M. Haseyama, "Missing intensity interpolation using a kernel PCA-Based POCS algorithm and its applications," *IEEE Trans. on Image Processing*, vol. 20, no. 2, pp. 417–432, 2011.

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

[26] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[27] B. Wohlberg, "Inpainting with sparse linear combinations of exemplars," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 689–692, 2009.

[28] M. J. Fadili, J.-L. Starck and F. Murtagh, "Inpainting and zooming using sparse representations," *The Computer Journal*, vol. 52, no. 1, pp. 64–79, 2009.

[29] Z. Xu and J. Sun, "Image Inpainting by Patch Propagation Using Patch Sparsity," *IEEE Trans. on Image Processing*, vol. 19, no. 5, pp. 1153–1165, 2010.

[30] T. Ogawa and M. Haseyama, "Missing image data reconstruction based on adaptive inverse projection via sparse representation," *IEEE Trans. on Multimedia*, vol. 13, no. 5, pp. 974-992, 2011.

[31] Z. Li, H. He. H.-M. Tai, Z. Yin and F. Chen, "Color-direction patch-sparsity-based image inpainting using multidirection features," *IEEE Trans. on Image Processing*, vol. 24, no. 3, pp. 1138–1152, 2015.

[32] M. Turkan and C. Guillemot, "Locally linear embedding based texture synthesis for image prediction and error concealment," *Proc. IEEE International Conference on Image Processing*, pp. 3009–3012, 2012.

[33] C. Guillemot, M. Turkan, O. L. Meur, and M. Ebdelli, "Object removal and loss concealment using neighbor embedding methods," *Signal Processing: Image Communication*, vol. 28, no. 10, pp. 1405–1419, 2013.

[34] T. Takahashi, K. Konishi, and T. Furukawa, "Structured matrix rank minimization approach to image inpainting," *Proc. on IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 860–863, 2012.

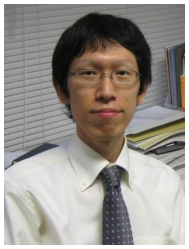
[35] J. Liu, P. Musialski, P. Wonka and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013.

[36] K. H. Jin and J. C. Ye, "Annihilating filter-based low-rank Hankel matrix approach for image inpainting," *IEEE Trans. on Image Processing*, vol. 24, no. 11, pp. 3498–3511, 2015.

[37] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. on Acoustics, Speech, Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[38] A. B. Oppenheim and R. W. Schafer, "Discrete-time signal processing, 2nd Edition," *Prentice Hall, New Jersey*, 1999.

[39] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE Signal Processing Magazine, Special issue of Super-Resolution Image Reconstruction*, 20(3), pp.21–36, 2003.
[40] S. Baker, S. and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.24, no.9, pp.1167–1183, 2002.
[41] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *International Journal of Imaging Systems and Technology*, vol. 14, no. 2, pp. 47–57, 2004.
[42] J.D. van Ouwerkerk, "Image super-resolution survey," *Image and Vision Computing*, vol24, no.10, pp.1039–1052, 2006.
[43] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," *Computer Graphics (Proc. Siggraph 2001)*, pp. 327–340, 2001.
[44] T. A. Stephenson and T. Chen, "Adaptive Markov random fields for example-based super-resolution of faces," *EURASIP Journal on Applied Signal Processing*, vol.2006, no.1, pp. 225–225, 2006.
[45] Q. Wang, X. Tang, and H. Shum, "Patch based blind image super resolution," *Proc. IEEE International Conference on Computer Vision*, vol. 1, pp. 709–716, 2005.
[46] X. Li, K.M. Lam, G. Qiu, L. Shen, and S. Wang, "An efficient example-based approach for image super-resolution," *Proc. International Conference on Neural Networks and Signal Processing*, pp.575–580, 2008.
[47] J. Sun, N. Zheng, H. Tao, and H. Shum, "Image hallucination with primal sketch priors," *Proc. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 729–736, 2003.
[48] D. Glasner, S. Bagon and M. Irani, "Super-resolution from a single image," *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 349–356, 2009.
[49] X. Wang and X. Trang, "Hallucinating face by eigentransformation," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 35, no. 3, pp. 425–434, 2005.
[50] A. Chakrabarti, A.N. Rajagopalan, and R. Chellappa, "Super-resolution of face images using kernel PCA-based prior," *IEEE Trans. on Multimedia*, vol. 9, no. 4, pp.888–892, 2007.
[51] C. Bishop, A. Blake, and B. Marthi, "Super-resolution enhancement of video," *Proc. Artificial Intelligence and Statistics (AISTATS)*, 2003.
[52] K.I. Kim, Y. Kwon, "Example-based learning for single-image super-resolution," *Lecture Notes in Computer Science, Pattern Recognition - 30th DAGM Symposium, Proceedings*, pp.456–465, 2008.
[53] Y. Hu, T. Shen, K. M. Lam, and S. Zhao, "A novel example-based super-resolution approach based on patch classification and the KPCA prior model," *Computational Intelligence and Security 2008*, vol. 1, pp.6–11, 2008.
[54] T. Ogawa and M. Haseyama, "Adaptive example-based super-resolution using kernel PCA with a novel classification approach," *EURASIP Journal on Advances in Signal Processing*, 2011:138.
[55] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Trans. on Image Processing*, vol. 20, no. 7, pp.1838–1857, 2011.
[56] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE Trans. on Image Processing*, vol. 21, no. 8, 2012.
[57] K. Zhang, X. Gao, X. Li, and D. Tao, "Partially supervised neighbor embedding for example-based image super-resolution," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 2, 2011.
[58] A. Kanemura, S. Maeda, S. Ishii, "Sparse Bayesian learning of filters for efficient image expansion," *IEEE Trans. on Image Processing*, vol. 19, no. 6, pp. 1480–1490, 2010.
[59] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. on Graphics*, vol. 30, no. 2, pp. 12:1–12:11, 2011.
[60] W. Dong, L. Zhang, R. Lukac and G. Shi, "Sparse representation based image interpolation with nonlocal autoregressive modeling," *IEEE Trans. on Image Processing*, vol. 22, no. 4, pp. 1382–1394, 2013.
[61] K. Zhang, D. Tao, X. Gao, X. Li and Z. Xiong, "Learning multiple linear mappings for efficient single image super-resolution," *IEEE Trans. on Image Processing*, vol. 24, no. 3, pp. 846–861, 2015.
[62] F. Zhou, T. Yuan, W. Yang and Q. Liao, "Single-image super-resolution based on compact KPCA coding and kernel regression," *IEEE Signal Processing Letters*, vol. 22, no. 3, pp. 336–340, 2015.
[63] Y. Sun, G. Gu, X. Sui and Y. Liu and C. Yang, "Single image super-resolution using compressive sensing with a redundant dictionary," *IEEE Photonics Journal*, vol. 7, no. 2, pp. 1–11, 2015.
[64] J. Jiang, X. Ma, Z. Cai and R. Hu, "Sparse support regression for image super-resolution," *IEEE Photonics Journal*, vol. 7, no. 5, pp. 1–11, 2015.
[65] F. Cao, M. Cai and Y. Tan, "Image Interpolation via Low-Rank Matrix Completion and Recovery," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 25, no. 8, pp. 1261–1270, 2015.
[66] L. Zhang, W. Dong, D. Zhang and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognition*, vol. 43, no. 4, pp. 1531–1549, 2010.
[67] C.-A. Deledalle, J. Salmon and A. Dalalyan, "Image denoising with patch based PCA: local versus global," *Proc. the British Machine Vision Conference*, pp. 25.1–25.10, 2011.
[68] J. T.-Y. Kwok, I. W.-H. Tsang, "The pre-image problem in kernel methods," *IEEE Trans. on Neural Networks*, vol.15, no.6, pp.1517–1525, 2004.
[69] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their applications to non-linear system identification," *Int. J. Contr.*, vol. 50, no. 5, pp. 1873–1896, 1989.
[70] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. on Information Theory*, vol. 50, no. 10,pp. 2231–2242, 2004.
[71] Z. Wang and A. C. Bovik, "Modern image quality assessment," *Morgan & Claypool Publishers*, Mar. 2006.
[72] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
[73] S. K. Sahoo and A. Makur, "Dictionary training for sparse representation as generalization of k-means clustering," *IEEE Signal Processing Letters*, vol.20, no.6, pp.587-590, 2013.
[74] S. K. Sahoo and A. Makur, "Sparse sequential generalization of k-means for dictionary training on noisy signals," *Signal Processing*, vol. 129, pp. 62–66, 2016.



Takahiro Ogawa (S'03–M'08) received his B.S., M.S. and Ph.D. degrees in Electronics and Information Engineering from Hokkaido University, Japan in 2003, 2005 and 2007, respectively. He joined Graduate School of Information Science and Technology, Hokkaido University in 2008. He is currently an associate professor in the Graduate School of Information Science and Technology, Hokkaido University. His research interests are multimedia signal processing and its applications. He has been an Associate Editor of ITE Transactions on Media Technology and Applications. He is a member of the IEEE, ACM, EURASIP, IEICE, and ITE.

Miki Haseyama (S'88–M'91–SM'06) received her B.S., M.S. and Ph.D. degrees in Electronics from Hokkaido University, Japan in 1986, 1988 and 1993, respectively. She joined the Graduate School of Information Science and Technology, Hokkaido University as an associate professor in 1994. She was a visiting associate professor of Washington University, USA from 1995 to 1996. She is currently a professor in the Graduate School of Information Science and Technology, Hokkaido University. Her research interests include image and video processing and its development into semantic analysis. She has been a Vice-President of the Institute of Image Information and Television Engineers, Japan (ITE), an Editor-in-Chief of ITE Transactions on Media Technology and Applications, a Director, International Coordination and Publicity of The Institute of Electronics, Information and Communication Engineers (IEICE). She is a member of the IEEE, IEICE, ITE and ASJ.