www.ietdl.org

Published in IET Image Processing
Received on 10th April 2014
Revised on 22nd September 2014
Accepted on 15th October 2014
doi: 10.1049/iet-ipr.2014.0313

ISSN 1751-9659

Single-image super-resolution in RGB space via group sparse representation

Ming Cheng[1], *Cheng Wang*[1], *Jonathan Li*[2]

[1]*Department of Computer Science, Xiamen University, Xiamen 361005, People's Republic of China*
[2]*Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON, Canada N2 L 3G1*
E-mail: chm99@xmu.edu.cn

Abstract: Super-resolution (SR) is the problem of generating a high-resolution (HR) image from one or more low-resolution (LR) images. This study presents a new approach to single-image super-resolution based on group sparse representation. Two dictionaries are constructed corresponding to the LR and HR image patches, respectively. The sparse coefficients of an input LR image patch in terms of the LR dictionary are used to recover the HR patch from the HR dictionary. When constructing the dictionaries, the three colour channels in a training image patch are considered a group composed of three atoms. The whole group is selected simultaneously when representing an image patch so that the correlations between the colour channels can be retained. A dictionary training method is also designed in which the two dictionaries are trained jointly to ensure that the corresponding LR and HR patches have the same sparse coefficients. Experimental results demonstrate the effectiveness of the proposed method and its robustness to noise.

1 Introduction

High-resolution (HR) images are desired in many practical cases such as remote sensing, medical imaging, video surveillance and entertainment. However, HR images are not always available because of technological and economic constraints. Super-resolution (SR) is the problem of generating an HR image from one or more low-resolution (LR) images [1]. The SR task is cast as the inverse problem of recovering the original HR image by fusing the LR images based on reasonable assumptions or prior knowledge about the observation model that maps the HR image to the LR ones [2]. Conventional approaches for generating an SR image require multiple LR images of the same scene, typically aligned with sub-pixel accuracy. Recently, single-image super-resolution (SISR), which recovers the HR image from a single LR image, has attracted increasing interest.

Interpolation is a simple and fast SISR approach that has been widely used in image processing [3–5]. Commonly used interpolation methods include bilinear interpolation and bicubic interpolation. Interpolation methods are based on generic smoothness priors and tend to generate overly smooth images, especially when dealing with natural images that have complex textures.

More recently, some learning-based SISR methods have been proposed by different researchers. This group of methods assumes that the high-frequency details lost in an LR image can be learned from a training set of LR and HR image pairs [6]. Freeman *et al.* [7, 8] proposed an example-based SR algorithm for creating plausible high frequency details in zoomed images, where a Markov network is used to model the spatial relationships between image patches. Chang *et al.* [9], assuming small image patches in LR and HR images form manifolds with similar local geometry, adopted the philosophy of local linear embedding from manifold learning. Their algorithm maps the local geometry of the LR patch space to the HR patch space, generating the HR patch as a linear combination of neighbours. However, using a fixed number K neighbours for reconstruction limits the performance of the algorithm. This neighbour embedding approach is further developed in [10–12]. Yang *et al.* [2, 13, 14] proposed a sparse representation based method for adaptively choosing the relevant reconstruction neighbours to represent the relationship between the training data and the input patches. Wu *et al.* [15] proposed a learning-based SR algorithm using kernel partial least squares consisting of two steps: estimation of the primitive super-resolved image and compensation with the residual image.

In this paper, we focus on the problem of sparse representation based SISR, and here we give a brief review of this approach. Sparse representation has been widely used in image processing. The assumption that natural images admit a sparse decomposition over a redundant dictionary leads to many efficient algorithms [16]. We consider image patches of size $\sqrt{n} \times \sqrt{n}$ pixels, ordered lexicographically as column vectors $\{\boldsymbol{x}_i, \ i = 1, \ 2, \ \ldots\}$, $\boldsymbol{x}_i \in \mathbb{R}^n$. Let $\boldsymbol{D} \in \mathbb{R}^{n \times K}$ be an over-complete dictionary of K atoms, $K \gg n$. An image patch is represented as $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$. Since there are many possible $\boldsymbol{\alpha}$ that satisfy $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$, our aim is to find the $\boldsymbol{\alpha}$ with the fewest non-zero elements. Thus, $\boldsymbol{\alpha}$ is called the sparse coefficients of $\boldsymbol{x}$ with dictionary $\boldsymbol{D}$. In practice, we observe only a small set of measurements

$\boldsymbol{y}$ of $\boldsymbol{x}$:

$$\boldsymbol{y} = \boldsymbol{L}\boldsymbol{x} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{\alpha} \tag{1}$$

where $\boldsymbol{L} \in \mathbb{R}^{m\times n}$ with $m < n$ is a projection matrix. In SR context, $\boldsymbol{x}$ is an HR image patch and $\boldsymbol{y}$ is its LR counterpart (or features extracted from the LR counterpart). Under mild conditions, the sparsest solution to (1) is unique. Furthermore, if $\boldsymbol{D}$ satisfies an appropriate near-isometry condition, then, for a wide variety of matrices $\boldsymbol{L}$, the HR patch $\boldsymbol{x}$ can be recovered (almost) perfectly by $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is the sparsest solution to (1) [17, 18].

In practice, we use two coupled dictionaries: $\boldsymbol{D}_{\mathrm{h}}$ for HR patches and $\boldsymbol{D}_{\mathrm{l}}$ for LR ones. The sparse representation of an LR patch in terms of $\boldsymbol{D}_{\mathrm{l}}$ is used directly to recover the corresponding HR patch from $\boldsymbol{D}_{\mathrm{h}}$. Formally, the sparse coefficients of an LR patch are found by solving the problem

$$\min\|\boldsymbol{\alpha}\|_0 \ \text{s.t.} \ \left\|\boldsymbol{D}_{\mathrm{l}}\boldsymbol{\alpha} - \boldsymbol{y}\right\|_2^2 \le \varepsilon \tag{2}$$

Searching the minimum l_0-norm is combinatorial, and the above optimisation is an NP-hard problem. Commonly used methods for approximately solving the problem are greedy algorithms, such as matching pursuit (MP) and orthogonal matching pursuit (OMP) [19]. OMP is an iterative, greedy algorithm that selects, at each step, the atom in the dictionary most correlated with the current residual. That atom is then added to the set of selected atoms, and the residual is updated by projecting the signal onto the linear subspace spanned by the atoms that have already been selected. These processes are repeated until the signal is satisfactorily decomposed or the number of selected atoms reaches a pre-defined threshold. Another relaxation strategy is to replace the l_0-norm with l_1-norm

$$\min\|\boldsymbol{\alpha}\|_1 \ \text{s.t.} \ \left\|\boldsymbol{D}_{\mathrm{l}}\boldsymbol{\alpha} - \boldsymbol{y}\right\|_2^2 \le \varepsilon \tag{3}$$

It has been proved [20] that when the sparse coefficients are sufficiently sparse, the minimal l_1-norm solution is also the sparsest one. The l_1-norm minimisation problem is convex, and many effective solving techniques have been developed. Given the optimal solution $\boldsymbol{\alpha}^*$ to (2) or (3), the HR patch can be reconstructed as $\boldsymbol{x} = \boldsymbol{D}_{\mathrm{h}}\boldsymbol{\alpha}^*$.

The aforementioned method can be applied directly to greyscale images. When dealing with colour images in RGB space, in most of the literature, the images are converted to luminance/chrominance colour models (e.g. the YCrCb model in [2] and the YIQ model in [9]) and the SR operations are applied only to the luminance channel. The chrominance channels are simply interpolated. Since chrominance information does not play a role in the SR process, we achieve better results if we exploit all the information in the images. In this work, we choose to stay with the original RGB space. There are two simple ways to fulfil directly the SR operations in RGB space. One way is to concatenate the three colour channels into a single vector. (The two dictionaries $\boldsymbol{D}_{\mathrm{h}}$ and $\boldsymbol{D}_{\mathrm{l}}$ are constructed in the same way.) However, this approach forces the three channels to use the same sparse coefficients, which limit the approach's performance. Another way is to apply separately the SR algorithm to each colour channel. This approach is even worse because it does not consider the strong correlations between the channels.

Inspired by recent progress in group sparse representation (GSR) [21, 22], we propose a GSR based SISR method for colour images. In GSR, the atoms in a dictionary are organised as groups, where the sparse coefficients in the same group tend to be zeros or non-zeros simultaneously. This characteristic is very useful to maintain the correlations between the colour channels.

The rest of the paper is organised as follows: Section 2 introduces the GSR concept and the corresponding group sparse coding algorithm. In Section 3 the GSR technique is applied to solve the SISR problem of colour images, and a dictionary training algorithm is designed as well. Section 4 gives the experimental results. Section 5 concludes the paper.

2 Group sparse representation

Let $\boldsymbol{D} \in \mathbb{R}^{n\times K}$ be an overcomplete dictionary of K atoms, and suppose $K = Nk$. We divide the atoms into N groups, each containing k atoms

$$\boldsymbol{D} = \left[\boldsymbol{d}_{1,1}, \boldsymbol{d}_{2,1}, \ldots, \boldsymbol{d}_{N,1}, \boldsymbol{d}_{1,2}, \ldots, \boldsymbol{d}_{N-1,k}, \boldsymbol{d}_{N,k}\right]$$

For each atom, the first subscript is the group index, and the second subscript is the index in the group. The linear representation of a vector $\boldsymbol{x} \in \mathbb{R}^n$ in terms of $\boldsymbol{D}$ is also grouped:

$$\boldsymbol{\alpha} = \left[\alpha_{1,1}, \alpha_{2,1}, \ldots, \alpha_{N,1}, \alpha_{1,2}, \ldots, \alpha_{N-1,k}, \alpha_{N,k}\right]^{\mathrm{T}}$$

In this representation, the goal is that the atoms in the same group are selected (or not selected) simultaneously, and the number of selected groups is as small as possible. This optimisation problem is formulated as

$$\min\|\boldsymbol{\alpha}\|_{2,0} \ \text{s.t.} \ \boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha} \tag{4}$$

The mixed norm $\|\boldsymbol{\alpha}\|_{2,0}$ is defined as $\|\boldsymbol{\alpha}\|_{2,0} = \sum_{j=1}^{N} I\left(\left\|\boldsymbol{\alpha}_j\right\|_2 > 0\right)$, where $\boldsymbol{\alpha}_j = [\alpha_{j,1}, \alpha_{j,2}, \ldots, \alpha_{j,k}]^{\mathrm{T}}$ corresponds to the jth group of the atoms and $I\left(\left\|\boldsymbol{\alpha}_j\right\|_2 > 0\right) = 1$ if $\left\|\boldsymbol{\alpha}_j\right\|_2 > 0$.

The problem (4) is called the GSR problem [23, 24]. Now we consider its solution. Group orthogonal matching pursuit (GOMP) [22], which follows the idea of OMP, is a robust, fast algorithm used to solve (4). GOMP is an iterative process. At each step, one atom group that has non-zero coefficients is identified. Then, the group is added to the set of selected groups, and the residual is updated by projecting $\boldsymbol{x}$ onto the linear subspace spanned by all the selected atoms. These processes are repeated until a pre-determined stopping criterion is met.

The details of GOMP are summarised in Algorithm 1 as follows

Algorithm 1 Group Orthogonal Matching Pursuit

1. Given dictionary $\boldsymbol{D}$ whose atoms are grouped and vector $\boldsymbol{x}$ to be represented.
2. Initialise the residual $\boldsymbol{r} = \boldsymbol{x}$ and the selected group index set $\Lambda = \phi$.
3. Compute the correlations $\boldsymbol{c} = \boldsymbol{D}^{\mathrm{T}}\boldsymbol{r} = [\boldsymbol{r}^{\mathrm{T}}\boldsymbol{d}_{1,1}, \boldsymbol{r}^{\mathrm{T}}\boldsymbol{d}_{2,1}, \ldots, \boldsymbol{r}^{\mathrm{T}}\boldsymbol{d}_{N,k}]^{\mathrm{T}}$, divide $\boldsymbol{c}$ into groups: $\boldsymbol{c}_j = [\boldsymbol{r}^{\mathrm{T}}\boldsymbol{d}_{j,1}, \boldsymbol{r}^{\mathrm{T}}\boldsymbol{d}_{j,2}, \ldots, \boldsymbol{r}^{\mathrm{T}}\boldsymbol{d}_{j,k}]^{\mathrm{T}}$, $j = 1, \ldots, N$ and select the group with the highest

doi: 10.1049/iet-ipr.2014.0313

average correlation:

$$j = \arg\max \operatorname{mean}(|\boldsymbol{c}_j|)$$
$$\Lambda = \Lambda \cup \{j\}$$

4. Update residual $\boldsymbol{r}$ by projecting $\boldsymbol{x}$ onto the linear subspace spanned by all selected groups: $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{D}_\Lambda (\boldsymbol{D}_\Lambda)^+ \boldsymbol{x}$ where $\boldsymbol{D}_\Lambda$ is composed of all selected groups and $(\boldsymbol{D}_\Lambda)^+$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{D}_\Lambda$.
5. Repeat steps 3 and 4 until the minimum bound on the residual or the maximum number of selected groups is reached.
6. Final answer $\boldsymbol{\alpha}^*$ is $(\boldsymbol{D}_\Lambda)^+ \boldsymbol{x}$ combined with zeros for those unselected groups.

3 SISR via GSR and dictionary training

Our algorithm directly operates on the original RGB images. When constructing a dictionary ($\boldsymbol{D}_\text{h}$ or $\boldsymbol{D}_\text{l}$), the three colour channels in an image patch are considered a group composed of three atoms. When the dictionary is used to represent an image patch, our solution to capture the correlations between the colour channels is to select the three atoms in a group simultaneously (yet with different coefficients).

3.1 SISR via GSR

To construct the two dictionaries, colour images are collected and the smooth parts are discarded. Then image patches are selected, and each patch's colour channel is ordered lexicographically as a column vector. The same colour channels of the selected patches are concatenated horizontally to form a matrix $\boldsymbol{D}_{\text{h,C}}$, $C \in \{R, G, B\}$ and the dictionary $\boldsymbol{D}_\text{h}$ is constructed as follows

$$\boldsymbol{D}_\text{h} = \begin{bmatrix} \boldsymbol{D}_{\text{h,R}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_{\text{h,G}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{D}_{\text{h,B}} \end{bmatrix} = \operatorname{BlockDiag}\left(\boldsymbol{D}_{\text{h,R}}, \boldsymbol{D}_{\text{h,G}}, \boldsymbol{D}_{\text{h,B}}\right)$$

In dictionary $\boldsymbol{D}_\text{h}$, the atoms from the same image patch are considered one group. If the dictionary is composed of P image patches, then the ith, $(i+P)$th and $(i+2P)$th atoms constitute one group, $i = 1, 2, \ldots, P$.

The images are then blurred and downsampled to obtain their LR versions. When constructing $\boldsymbol{D}_\text{l}$, we do not directly use LR patches. In the literature, some authors have suggested extracting features from the LR images in order to boost the prediction accuracy. A feature extraction operator $\boldsymbol{F}$ is defined and applied to each of the LR image channels; then, the feature patches corresponding to the selected HR patches are determined. Each channel of the feature patches form a matrix $\boldsymbol{D}_{\text{l,C}}$, $C \in \{R, G, B\}$ and we have

$$\boldsymbol{D}_\text{l} = \operatorname{BlockDiag}\left(\boldsymbol{D}_{\text{l,R}}, \boldsymbol{D}_{\text{l,G}}, \boldsymbol{D}_{\text{l,B}}\right)$$

The atoms in $\boldsymbol{D}_\text{l}$ are organised as groups in the same way as in $\boldsymbol{D}_\text{h}$. We remove the mean value for each column of $\boldsymbol{D}_{\text{h,C}}$ and $\boldsymbol{D}_{\text{l,C}}$, so that the dictionaries represent image textures rather than absolute intensities. In the recovery process, the mean value of each channel's HR patch is predicted by the patch's LR version.

For an input LR patch $\boldsymbol{y}$, we transform the patch to a feature patch $\hat{\boldsymbol{y}}$. The values in each channel of $\hat{\boldsymbol{y}}$ are ordered as a column vector $\hat{\boldsymbol{y}}_\text{C}$, $C \in \{R, G, B\}$ and we represent $\hat{\boldsymbol{y}}$ as $\hat{\boldsymbol{y}} = \left[\hat{\boldsymbol{y}}_\text{R}^\text{T}, \hat{\boldsymbol{y}}_\text{G}^\text{T}, \hat{\boldsymbol{y}}_\text{B}^\text{T}\right]^\text{T}$. The problem of finding the sparse representation of $\hat{\boldsymbol{y}}$ is formulated as

$$\min \|\boldsymbol{\alpha}\|_{2,0} \text{ s.t. } \hat{\boldsymbol{y}} = \boldsymbol{D}_\text{l}\boldsymbol{\alpha} \tag{5}$$

when calculating the $l_{2,0}$-norm of $\boldsymbol{\alpha}$, the elements in $\boldsymbol{\alpha}$ that correspond to the atoms in the same group are considered one group. The optimal solution $\boldsymbol{\alpha}^*$ to (5) is obtained by the GOMP algorithm, and the HR patch is reconstructed as $\boldsymbol{x} = \boldsymbol{D}_\text{h}\boldsymbol{\alpha}^*$.

Solving (5) individually for each local patch does not guarantee compatibility between adjacent patches. We segment the input LR image into overlapping patches, and the reconstructed HR patches are averaged in the overlapping regions.

The entire SR process is summarised as Algorithm 2 as follows:

Algorithm 2 SISR via GSR

1. Input: dictionaries $\boldsymbol{D}_\text{h}$ and $\boldsymbol{D}_\text{l}$, an LR image $\boldsymbol{Y}$.
2. Transform $\boldsymbol{Y}$ to a feature image $\hat{\boldsymbol{Y}}$ by $\boldsymbol{F}$, then segment $\boldsymbol{Y}$ and $\hat{\boldsymbol{Y}}$ into overlapping patches.
3. For each patch $\hat{\boldsymbol{y}}$ in $\hat{\boldsymbol{Y}}$, taken starting from the upper-left corner,

   (1) Remove the mean value for each channel.
   (2) Solve the following optimisation problem:

$$\boldsymbol{\alpha}^* = \arg\min \|\boldsymbol{\alpha}\|_{2,0} \text{ s.t. } \hat{\boldsymbol{y}} = \boldsymbol{D}_\text{l}\boldsymbol{\alpha}$$

   (3) Generate the HR patch $\boldsymbol{x} = \boldsymbol{D}_\text{h}\boldsymbol{\alpha}^*$.
   (4) Add mean value to each channel of $\boldsymbol{x}$, obtained from the corresponding LR patch $\boldsymbol{y}$.
   (5) Put $\boldsymbol{x}$ into an HR image $\boldsymbol{X}$, averaging the overlapping regions.

4. Output: HR image $\boldsymbol{X}$.

3.2 Dictionary training

In the previous subsection, we generated dictionaries by randomly sampling raw patches from training images. Such a strategy results in large dictionaries and high computational cost. To accelerate the computation, we train a more compact dictionary pair $\left\{\boldsymbol{D}_\text{h}^*, \boldsymbol{D}_\text{l}^*\right\}$ with the form

$$\boldsymbol{D}_\text{h}^* = \operatorname{BlockDiag}(\boldsymbol{D}_{\text{h,R}}^*, \boldsymbol{D}_{\text{h,G}}^*, \boldsymbol{D}_{\text{h,B}}^*)$$
$$\boldsymbol{D}_\text{l}^* = \operatorname{BlockDiag}(\boldsymbol{D}_{\text{l,R}}^*, \boldsymbol{D}_{\text{l,G}}^*, \boldsymbol{D}_{\text{l,B}}^*)$$

To ensure the corresponding HR and feature patches have the same group sparse coefficients with respect to $\boldsymbol{D}_\text{h}^*$ and $\boldsymbol{D}_\text{l}^*$, we jointly train the two dictionaries.

Let $\boldsymbol{S}_\text{h} = [\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^\text{P}]$ be a collection of sampled HR image patches and $\boldsymbol{S}_\text{l} = \left[\hat{\boldsymbol{y}}^1, \hat{\boldsymbol{y}}^2, \ldots, \hat{\boldsymbol{y}}^\text{P}\right]$ be the set of corresponding feature patches. Each patch is represented as a column vector, formed by concatenating the three colour channels

$$\boldsymbol{x}^p = \left[(\boldsymbol{x}_{\mathrm{R}}^p)^{\mathrm{T}}, (\boldsymbol{x}_{\mathrm{G}}^p)^{\mathrm{T}}, (\boldsymbol{x}_{\mathrm{B}}^p)^{\mathrm{T}}\right]^{\mathrm{T}}$$
$$\hat{\boldsymbol{y}}^p = \left[(\hat{\boldsymbol{y}}_{\mathrm{R}}^p)^{\mathrm{T}}, (\hat{\boldsymbol{y}}_{\mathrm{G}}^p)^{\mathrm{T}}, (\hat{\boldsymbol{y}}_{\mathrm{B}}^p)^{\mathrm{T}}\right]^{\mathrm{T}}, \quad p = 1, \ldots, P$$

P is the total number of training patch pairs. Let $\boldsymbol{S} = \begin{bmatrix} \boldsymbol{S}_{\mathrm{h}} \\ \boldsymbol{S}_{\mathrm{l}} \end{bmatrix}$, $\boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}_{\mathrm{h}}^* \\ \boldsymbol{D}_{\mathrm{l}}^* \end{bmatrix}$ and $\boldsymbol{A} = \left[\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \ldots, \boldsymbol{\alpha}^P\right]$, $\boldsymbol{\alpha}^p$ being the group sparse coefficients of the pth column of $\boldsymbol{S}$ in terms of $\boldsymbol{D}$. We describe the dictionary training problem as follows

$$\min_{\boldsymbol{D},\boldsymbol{A}} \sum_{p=1}^{P} \left\|\boldsymbol{\alpha}^p\right\|_{2,0} \text{ s.t. } \|\boldsymbol{S} - \boldsymbol{D}\boldsymbol{A}\|_{\mathrm{F}}^2 \le \varepsilon \qquad (6)$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm of the matrix.

The optimisation problem (6) is minimised iteratively. Each iteration includes two stages. The first stage is sparse coding, in which we assume that $\boldsymbol{D}$ is fixed, and find the best coefficient matrix $\boldsymbol{A}$. The problem is decoupled to P distinct problems of the form

$$\boldsymbol{\alpha}^p = \arg\min\|\boldsymbol{\alpha}\|_{2,0} \text{ s.t. } \boldsymbol{s}^p = \boldsymbol{D}\boldsymbol{\alpha}, \; p = 1, \ldots, P \qquad (7)$$

where $\boldsymbol{s}^p = \left[(\boldsymbol{x}^p)^{\mathrm{T}}, (\hat{\boldsymbol{y}}^p)^{\mathrm{T}}\right]^{\mathrm{T}}$ is the pth column of $\boldsymbol{S}$. This is a GSR problem and is solved by GOMP.

A second stage is performed to search for a better dictionary with fixed $\boldsymbol{A}$

$$\boldsymbol{D} = \arg\min\|\boldsymbol{S} - \boldsymbol{D}\boldsymbol{A}\|_{\mathrm{F}}^2 \qquad (8)$$

This problem is also decoupled to three distinct problems of the form

$$\boldsymbol{D}_{\mathrm{C}} = \arg\min\left\|\boldsymbol{S}_{\mathrm{C}} - \boldsymbol{D}_{\mathrm{C}}\boldsymbol{A}_{\mathrm{C}}\right\|_{\mathrm{F}}^2, \; C \in \{R, G, B\} \qquad (9)$$

where

$$\boldsymbol{D}_{\mathrm{C}} = \begin{bmatrix} \boldsymbol{D}_{\mathrm{h,C}}^* \\ \boldsymbol{D}_{\mathrm{l,C}}^* \end{bmatrix}, \boldsymbol{S}_{\mathrm{C}} = \left[\begin{bmatrix} \boldsymbol{x}_{\mathrm{C}}^1 \\ \hat{\boldsymbol{y}}_{\mathrm{C}}^1 \end{bmatrix}, \ldots, \begin{bmatrix} \boldsymbol{x}_{\mathrm{C}}^P \\ \hat{\boldsymbol{y}}_{\mathrm{C}}^P \end{bmatrix}\right]$$

and $\boldsymbol{A}_{\mathrm{R}}$, $\boldsymbol{A}_{\mathrm{G}}$, $\boldsymbol{A}_{\mathrm{B}}$ are the first, second and last one-third rows of $\boldsymbol{A}$, respectively. The optimisation problem (9) is a quadratically constrained quadratic programming that is readily solved in many optimisation packages. In our implementation, we used the Matlab package developed in [25].

The complete framework of our dictionary training algorithm is summarised as Algorithm 3.

Algorithm 3 Dictionary training

1. Input

$\boldsymbol{S}_{\mathrm{h}}$ and $\boldsymbol{S}_{\mathrm{l}}$: HR and feature training sets, each containing P patches.
N: Number of groups in each trained dictionary, $N \ll P$.

2. Initialise $\boldsymbol{D}_{\mathrm{h}}^*$ and $\boldsymbol{D}_{\mathrm{l}}^*$ with the first N patches in $\boldsymbol{S}_{\mathrm{h}}$ and $\boldsymbol{S}_{\mathrm{l}}$, respectively.
3. Repeat the following steps until some predefined criteria are met

(a) Group sparse coding step: Fix $\boldsymbol{D}$, for $p = 1, \ldots, P$
(i) Solve the GSR problem

$$\boldsymbol{\alpha}^p = \arg\min\|\boldsymbol{\alpha}\|_{2,0} \text{ s.t. } \boldsymbol{s}^p = \boldsymbol{D}\boldsymbol{\alpha}$$

(ii) For each group $\boldsymbol{\alpha}_j^p = [\alpha_{j,\mathrm{R}}^p, \alpha_{j,\mathrm{G}}^p, \alpha_{j,\mathrm{B}}^p]^{\mathrm{T}}$ in $\boldsymbol{\alpha}^p$, $j = 1, \ldots, N$, if $\max_{\mathrm{C}}\left|\alpha_{j,\mathrm{C}}^p\right|$ is smaller than a pre-defined threshold, then set $\boldsymbol{\alpha}_j^p = \boldsymbol{0}$.

(b) Dictionary updating step: Fix $\boldsymbol{A}$, update the three colour components of $\boldsymbol{D}$, respectively.

$$\boldsymbol{D}_{\mathrm{C}} = \arg\min\left\|\boldsymbol{S}_{\mathrm{C}} - \boldsymbol{D}_{\mathrm{C}}\boldsymbol{A}_{\mathrm{C}}\right\|_{\mathrm{F}}^2, \; C \in \{R, G, B\}$$

4. Output: Trained dictionaries $\boldsymbol{D}_{\mathrm{h}}^*$ and $\boldsymbol{D}_{\mathrm{l}}^*$.

3.3 Feature extraction for LR images

Typically, the feature extraction operator $\boldsymbol{F}$ is chosen as a kind of high-pass filter. In [2, 9, 22], the first- and second-order derivatives of the images are used as the features. Since of their simplicity and effectiveness, we use the same options in our realisation. The four one-dimensional filters to extract the derivatives are

$$\boldsymbol{f}_1 = [-1, 0, 1] \quad \boldsymbol{f}_2 = \boldsymbol{f}_1^{\mathrm{T}}$$
$$\boldsymbol{f}_3 = [1, 0, -2, 0, 1] \quad \boldsymbol{f}_4 = \boldsymbol{f}_3^{\mathrm{T}}$$

Furthermore, we adopt the trick proposed in [2] that we up-sample the LR image by a factor of 2 using Bicubic interpolation before feature extraction. Application of the four filters yields four gradient maps for each LR image. When constructing a feature patch, the values in the four gradient maps are concatenated.

4 Experimental results

In this section, we report the proposed method's SR results and compare them with the results for the state-of-the-art methods. Our results are evaluated both visually and quantitatively in PSNR. Then, the robustness to noise of the proposed method is tested. The PSNR is defined as

$$\text{PSNR} = 10 \times \log_{10}\frac{255^2}{\text{MSE}} \qquad (10)$$

where MSE is the mean square error calculated among all three colour channels.

In our experiments, a magnification factor of 3 is considered. In the SR process, the input LR images are segmented to 3×3 patches with an overlap of one pixel between adjacent patches, which, in HR images, corresponds to 9×9 patches with an overlap of three pixels. The HR and feature patches are represented as column vectors whose dimensions are 243 and 432 (i.e. $9 \times 9 \times 3$ and $6 \times 6 \times 4 \times 3$), respectively.

4.1 SR results

We applied our method to natural images such as animals, buildings and people. The original HR images were blurred and down-sampled by a factor of 3 to generate the LR

IET Image Process., 2015, Vol. 9, Iss. 6, pp. 461–467
doi: 10.1049/iet-ipr.2014.0313

Fig. 1 *Test images*

From left- to right-hand side: butterfly, Lena, tiger, Parthenon and flower

images. The SR results were compared with the original images to obtain the PSNR. Both raw-patch and trained dictionaries, denoted as GSR-SR1 and GSR-SR2, respectively, were tested. Four methods were selected for comparison: (1) Bicubic interpolation, (2) Yang's sparse representation based method (SR-SR) [2], (3) He's Gaussian process regression (GPR) method [26] and (4) Wang's semi-coupled dictionary learning (SCDL) method [27]. The GPR method uses Gaussian process regression to predict each pixel's neighbours in the SR process. The SCDL method uses two semi-coupled dictionaries to represent HR and LR image patches, respectively, indicating that the representations of corresponding patches have a stable mapping. For the latter three methods, the SR operations were applied only to the luminance channel (The chrominance channels were simply interpolated), and the results were converted back to RGB space before visual and quantitative evaluation (The quantitative results reported in [2, 26, 27] were carried out on only the luminance channel).

We used 50 000 HR and feature patch pairs to construct the raw-patch dictionaries $\boldsymbol{D}_\text{h}$ and $\boldsymbol{D}_\text{l}$. The same patch pairs are also used to train the compact dictionaries $\boldsymbol{D}_\text{h}^*$ and $\boldsymbol{D}_\text{l}^*$, each containing 5 000 groups of atoms. We briefly describe our HR patch selection strategy. From the Internet, we collect about 1 000 colour images including natural scenes, animals, plants, buildings, people etc. The images are segmented to 9×9 patches, and patches with constant colour are discarded. Thus, we obtain about 10 million HR patches. The patches are classified according to the standard deviations of the three colour channels, σ_C, $C \in \{R, G, B\}$. The value of each σ_C is divided into twelve unequally spaced intervals. (The interval is short when σ_C is small and becomes longer when σ_C increases.) All patches are classified into $12^3 = 1{,}728$ categories. The number of patches in each category is counted. If the number is < 50, all patches in the category are selected; otherwise, only part of the category is selected.

As shown in Fig. 1, we use five colour images to test the methods. For visual evaluation, the SR results of butterfly, Lena and tiger are shown in Figs. 2–4. Some parts of the images are enlarged in order to facilitate comparison. The results of Bicubic and SCDL are overly smooth and lose many details; the results of SR–SR are a little noisy. GPR tends to generate plaques, making the recovered images look unnatural. GSR-SR1 generates sharp edges with rare artefacts and is closest to the ground truth. In all figures, there is no significant difference between the results of GSR-SR1 and GSR-SR2, which suggests that the trained dictionary pair has representation ability similar to the raw-patch dictionary pair.

The quantitative evaluation results are listed in Table 1. The results are consistent among the five test images. For all images, GSR-SR1 provides the highest PSNR (0.76 dB higher than SR-SR, 0.93 dB higher than GPR and 0.63 dB higher than SCDL, on the average). The results of GSR-SR2 are slightly worse than the results of GSR-SR1, but still better than the results of the other four methods.

The most time-consuming operation in GSR-SR2 is to calculate the pseudo-inverse of $\boldsymbol{D}_\Lambda$ (Steps 4 and 6 in Algorithm 1). In our experiments we found that five atom groups are enough to accurately represent the feature patch $\hat{\boldsymbol{y}}$. The correlation coefficient between $\hat{\boldsymbol{y}}$ and $\boldsymbol{D}_\text{l}\boldsymbol{\alpha}^*$ is > 0.99 and $\|\boldsymbol{r}\|_2 / \|\hat{\boldsymbol{y}}\|_2$ is < 0.05, where $\boldsymbol{r} = \hat{\boldsymbol{y}} - \boldsymbol{D}_\text{l}\boldsymbol{\alpha}^*$ is the residual. The computation of $(\boldsymbol{D}_\Lambda)^+$ is quite rapid because the number of columns in $\boldsymbol{D}_\Lambda$ is small. In the experiments, we used a computer with Intel Core i7 @ 3.06 GHz CPU

Fig. 2 *Comparison of the SR results on butterfly*

From left- to right-hand side and top to bottom: HR image, LR image, Bicubic, SR-SR, GPR, SCDL, GSR-SR1 and GSR-SR2

Fig. 3 *Comparison of the SR results on Lena*

From left- to right-hand side and top to bottom: HR image, LR image, Bicubic, SR-SR, GPR, SCDL, GSR-SR1 and GSR-SR2

Fig. 4 *Comparison of the SR results on tiger*

From left- to right-hand side and top to bottom: HR image, LR image, Bicubic, SR-SR, GPR, SCDL, GSR-SR1 and GSR-SR2

Table 1 PSNR (dB) of the recovered images

Image	Bicubic	SR-SR	GPR	SCDL	GSR-SR1	GSR-SR2
Lena	27.63	28.38	28.09	28.64	29.07	28.91
Tiger	25.77	26.01	25.86	25.89	27.14	26.87
Butterfly	19.18	19.63	19.36	20.19	20.69	20.42
Parthenon	22.43	22.74	22.63	22.64	23.00	22.89
Flower	23.14	23.21	23.15	23.24	23.84	23.69
Average	23.63	23.99	23.82	24.12	24.75	24.56

and 8 GB RAM. Using the trained dictionary, the computer processes about 22 image patches per second.

4.2 Robustness to noise

We tested the robustness of the proposed SISR method by adding different levels of Gaussian noise to the input LR image. Two noise levels, with standard deviations of 2 and 5, respectively, were used in the experiments. Table 2 shows the quantitative results from different levels of noisy inputs. In terms of PSNR, GSR-SR1 and GSR-SR2 provided the first and second best results in all cases.

Table 2 PSNR (dB) of the recovered images from different levels of noisy inputs

Methods	Lena		Tiger		Butterfly	
	$\sigma = 2$	$\sigma = 5$	$\sigma = 2$	$\sigma = 5$	$\sigma = 2$	$\sigma = 5$
Bicubic	27.38	26.60	25.47	24.87	18.98	18.57
SR-SR	27.78	26.72	25.74	24.71	19.14	18.61
GPR	27.61	26.80	25.56	24.81	19.01	18.68
SCDL	27.91	26.70	25.52	24.64	19.37	18.59
GSR-SR1	28.46	26.98	26.70	25.10	19.92	18.94
GSR-SR2	28.23	26.77	26.45	24.88	19.51	18.80

IET Image Process., 2015, Vol. 9, Iss. 6, pp. 461–467
doi: 10.1049/iet-ipr.2014.0313

5 Conclusions

In this paper, we proposed a novel SISR method based on GSR. The method works in RGB space and, by grouping the atoms in the dictionaries, retains the correlations between the colour channels. A dictionary training method was also designed to reduce the size of the dictionaries. Experimental results demonstrated the proposed method's effectiveness for natural image SR tasks and robustness to noise. The overall performance of our method is very competitive with the performance of state-of-the-art methods. Since only one trained dictionary pair was tested in the experiments, the optimal dictionary size is an open question that will be investigated in our future work. Since the SR operation is performed independently on each patch, another research direction will be parallelisation.

6 References

1 Park, S.C., Park, M.K., Kang, M.G.: 'Super-resolution image reconstruction: a technical overview', *IEEE Signal Process. Mag.*, 2003, **20**, (3), pp. 21–36

2 Yang, J., Wright, J., Huang, T.S., Ma, Y.: 'Image super-resolution via sparse representation', *IEEE Trans. Image Process.*, 2010, **19**, (11), pp. 2861–2873

3 Li, X., Orchard, M.T.: 'New edge-directed interpolation', *IEEE Trans. Image Process.*, 2001, **10**, (10), pp. 1521–1527

4 Zhang, L., Wu, X.: 'An edge-guided image interpolation algorithm via directional filtering and data fusion', *IEEE Trans. Image Process.*, 2006, **15**, (8), pp. 2226–2238

5 Liu, X., Zhao, D., Xiong, R., Ma, S., Gao, W., Sun, H.: 'Image interpolation via regularized local linear regression', *IEEE Trans. Image Process.*, 2011, **20**, (12), pp. 3455–3469

6 Gao, X., Zhang, K., Tao, D., Li, X.: 'Joint learning for single-image super-resolution via a coupled constraint', *IEEE Trans. Image Process.*, 2012, **21**, (2), pp. 469–480

7 Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: 'Learning low-level vision', *Int. J. Comput. Vis.*, 2000, **40**, (1), pp. 25–47

8 Freeman, W.T., Jones, T.R., Pasztor, E.C.: 'Example-based super-resolution', *IEEE Comput. Graph. Appl.*, 2002, **22**, (2), pp. 56–65

9 Chang, H., Yeung, D.-Y., Xiong, Y.: 'Super-resolution through neighbor embedding'. Proc. IEEE Conf. CVPR, Washington, DC, USA, 2004, pp. 275–282

10 Fan, W., Yeung, D.-Y.: 'Image hallucination using neighbor embedding over visual primitive manifolds'. Proc. IEEE Conf. CVPR, Minneapolis, MN, USA, 2007, pp. 1–7

11 Chan, T.-M., Zhang, J., Pu, J., Huang, H.: 'Neighbor embedding based super-resolution algorithm through edge detection and feature selection', *Pattern Recognit. Lett.*, 2009, **30**, (5), pp. 494–502

12 Zhang, K., Gao, X., Li, X., Tao, D.: 'Partially supervised neighbor embedding for example-based image super-resolution', *IEEE J. Sel. Topics Signal Process.*, 2011, **5**, (2), pp. 230–239

13 Yang, J., Wright, J., Huang, T., Ma, Y.: 'Image super-resolution as sparse representation of raw image patches'. Proc. IEEE Conf. CVPR, Anchorage, AK, USA, 2008, pp. 1–8

14 Yang, J., Wang, Z., Lin, Z., Cohen, S., Huang, T.: 'Coupled dictionary training for image super-resolution', *IEEE Trans. Image Process.*, 2012, **21**, (8), pp. 3467–3478

15 Wu, W., Liu, Z., He, X.: 'Learning-based super resolution using kernel partial least squares', *Image Vis. Comput.*, 2011, **9**, (6), pp. 394–406

16 Elad, M., Figueiredo, M.A.T., Ma, Y.: 'On the role of sparse and redundant representations in image processing', *Proc. IEEE*, 2010, **98**, (6), pp. 972–982

17 Donoho, D.L.: 'Compressed sensing', *IEEE Trans. Inf. Theory*, 2006, **52**, (4), pp. 1289–1306

18 Rauhut, H., Schnass, K., Vandergheynst, P.: 'Compressed sensing and redundant dictionaries', *IEEE Trans. Inf. Theory*, 2008, **54**, (5), pp. 2210–2219

19 Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.: 'Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition'. Proc. 27th Annual Asilomar Conf. Signals, Syst. Computation, 1993, pp. 40–44

20 Donoho, D.: 'For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution', *Commun. Pure Appl. Math.*, 2006, **59**, (6), pp. 797–829

21 Majumdar, A., Ward, R.K.: 'Compressed sensing of color images', *Signal Process.*, 2010, **90**, pp. 3122–3127

22 Liu, H., Li, S., Yin, H.: 'Infrared surveillance image super resolution via group sparse representation', *Opt. Commun.*, 2013, **289**, pp. 45–52

23 Eldar, Y.C., Mishali, M.: 'Robust recovery of signals from a structured union of subspaces', *IEEE Trans. Inf. Theory*, 2009, **55**, (11), pp. 5302–5316

24 Stojnic, M., Parvaresh, F., Hassibi, B.: 'On the reconstruction of block-sparse signals with an optimal number of measurements', *IEEE Trans. Signal Process.*, 2009, **57**, (8), pp. 3075–3085

25 Lee, H., Battle, A., Raina, R., Ng, A.Y.: 'Efficient sparse coding algorithms'. Advances in Neural Information Processing Systems, 2007, pp. 801–808

26 He, H., Siu, W.-C.: 'Single image super-resolution using Gaussian process regression'. Proc. IEEE Conf. CVPR, Colorado Springs, CO, USA, 2011, pp. 449–456

27 Wang, S., Zhang, L., Liang, Y., Pan, Q.: 'Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis'. Proc. IEEE Conf. CVPR, Providence, RI, USA, 2012, pp. 2216–2223