Chapter 3

Development of Pansharpening-based Fast MS Image SR using Sparse Representations

3.1 Introduction

Traditionally, sparse representation finds an accurate reconstruction of an image from a few transformed domain coefficients, which has many applications, like image restoration, image enhancement, image segmentation, super-resolution, and so on [1, 31, 115]. Dictionary learning-based single-image SR technique is reported to be capable of removing limitations of traditional SR methods by employing a dictionary learned from both LR and HR image patches [17]. It is able to estimate information missed out in the traditional interpolation methods by examining the HR and LR image patch pairs' co-occurrences, while the dictionary is trained [30, 121]. Works reported by Yang *et al.* [115] and Ramos *et al.* [2] demonstrate the superiority of dictionary learning-based SR methods over traditional methods such as bicubic interpolation or new edge directed interpolation (NEDI) [60]).

As discussed in Chapter 2, sparse representation-based algorithms are required to solve several regularization problems, which are highly computationally intensive. Large data size of satellite images further increases the computational complexity. The remote sensing applications either require continuous monitoring of large number of such images or be able to refresh/reload the captured images frequently. Therefore, acceleration of a sparse representation-based algorithm is cruicial for practical systems.

In this chapter, we aim to develop a fast sparse representation-based SISR method for remote sensing using multicore parallel processing approach. We have proposed three SISR algorithms using the patch-based sparse representation and overcomplete dictionary learning techniques. These algorithms are designed based on the types of datasets used and corresponding dictionaries that could be learned from these datasets. First, a fast SR imaging technique is developed using multicore processing from RGB images, where SR carried out on the luminance channel, Y after transformation of the RGB to YCbCr format. Coupled LR-HR dictionaries are trained from external HR image datasets using the joint sparse coding technique. Next, we have developed a new dictionary learning scheme by creating an HR dataset from patches of multiple HR PAN images, and an LR dataset formed by patches extracted from the corresponding LR MS images. Then, each band of the test LR MS image is reconstructed individually, and subsequently, super-resolved bands are merged to get the desired HR MS image. The overall algorithm is implemented using multicore parallel processing for faster execution. Finally, we have developed a MS image SR algorithm using the concept of pansharpening and sparse representations. In pansharpening method, HR MS images are obtained via fusion of PAN and MS images. False RGB image obtained from the MS image is converted to the YCbCr format and then Y-channel is replaced by the HR PAN image to get the HR MS image. We train a pair of adaptive overcomplete LR-HR dictionaries from the HR PAN image and the test MS image; perform band-wise SR of the LR MS image though sparse representation. This method overcomes the limitation of spectral resolution of pansharpening methods and results in high spatial as well as spectral resolution MS images.

3.2 Fast SISR using sparse representations and multicore processing

3.2.1 Introduction

In this chapter, to reduce the computation time in practical remote sensing applications, like border monitoring, disaster management, etc., we propose a parallel sparse representation-based SR algorithm and implement the same using multicore processing. SR algorithms are generally developed for single image i.e. either for a grayscale (PAN) or the luminance component (Y-channel) of the YCbCr image obtained from the RGB MS image. Now, we present a fast SR image reconstruction method from RGB MS images via sparse representation.

3.2.2 Proposed method

3.2.2.1 Sparse representation-based SISR reconstruction

The general block diagram of the proposed SISR method is shown in Fig. 3.1. The input image $\mathbf{X} \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$ is divided into several overlapping patches $\mathbf{x}$ of dimension $\sqrt{n}\times\sqrt{n}$ by traversing the image in a lexicographical manner. Let $\mathbf{y} \in \mathbb{R}^n$ be a patch vector extracted from the observed LR image $\mathbf{Y}$ and arranged into a column vector of dimension $n\times 1$. In the basic sparse representation-based SISR approach reported in [115], first sparse coefficients $\boldsymbol{\alpha} \in \mathbb{R}^K$ are estimated for $\mathbf{y}$ using the LR dictionary $\mathbf{D}_\ell \in \mathbb{R}^{n\times K}$, trained on the LR image patches obtained after applying downsampling and blurring operators to the input $\mathbf{X}$. These coefficients are then multiplied with an HR dictionary $\mathbf{D}_h \in \mathbb{R}^{\bar{n}\times K}$, trained directly on HR patches extracted from $\mathbf{X}$, while it is constrained to have the same sparse representations for $\mathbf{x}$ and $\mathbf{y}$ to estimate the unknown target HR patch $\mathbf{x}$. Here, $\bar{n}$ is equal to n multiplied by zoom factor.

Given the dictionary $\mathbf{D}_\ell$, we obtain $\boldsymbol{\alpha} \in \mathbb{R}^K$ for each patch $\mathbf{y}$ by solving the following minimization problem with a local image fidelity constraint [2]:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad \text{such that } \|\mathbf{D}_\ell\boldsymbol{\alpha} - \mathbf{y}\|_2^2 \leq \varepsilon, \tag{3.1}$$

where $\mathbf{y} = E_p(\mathbf{Y})$ represents overlapping patches of $\mathbf{Y}$ extracted using the patch extraction operator E_p. Eq. 3.1 can be reformulated into an unconstrained optimization problem using the Lagrangian form as follows:

$$\hat{\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}} \|\mathbf{D}_l\boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 , \tag{3.2}$$

where λ is regularization parameter. Subsequently, HR patches $\mathbf{x} \in \mathbb{R}^{\bar{n}}$ can be obtained by multiplying $\hat{\boldsymbol{\alpha}}$ obtained above with $\mathbf{D}_h$, i.e.

$$\mathbf{x} = \mathbf{D}_h\hat{\boldsymbol{\alpha}}. \tag{3.3}$$

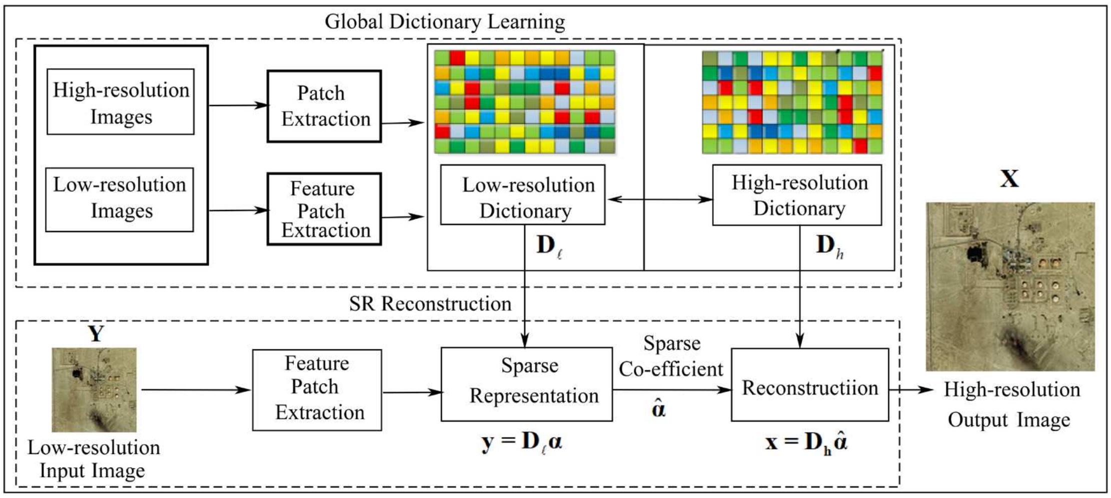

Global Dictionary Learning
High-resolution Images
Low-resolution Images
Patch Extraction
Feature Patch Extraction
Low-resolution Dictionary
High-resolution Dictionary
$\mathbf{D}_\ell$
$\mathbf{D}_h$
X
SR Reconstruction
Y
Feature Patch Extraction
Sparse Representation
Sparse Co-efficient
$\hat{\boldsymbol{\alpha}}$
Reconstructiion
High-resolution Output Image
Low-resolution Input Image
$\mathbf{y} = \mathbf{D}_\ell\boldsymbol{\alpha}$
$\mathbf{x} = \mathbf{D}_h\hat{\boldsymbol{\alpha}}$

Figure 3.1: Overview of SISR based on sparse representation and dictionary learning

Suppose an HR image $\mathbf{X}_0$ is obtained by tiling the estimated HR patches $\mathbf{x}$ in reference to their positions in the original image. Due to noise and other measurement errors $\mathbf{X}_0$ may not exactly fit to the assumed image acquisition model i.e. $\mathbf{Y} = SH\mathbf{X}$, where S and H are respectively the downsampling and blurring operators of the imaging model. In order to overcome this, a global reconstruction constraint on $\mathbf{X}_0$ may be imposed by solving the following least-squares minimization problem:

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \|SH\mathbf{X} - \mathbf{Y}\|_2^2 + c\,\|\mathbf{X} - \mathbf{X}_0\|_2^2 . \tag{3.4}$$

Eq. 3.4 is efficiently solved by applying the gradient descent method [115] by formulating a simpler iterative equation as follows:

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \nu \left[H^T S^T \left(\mathbf{Y} - SH\mathbf{X}_i \right) + c \left(\mathbf{X} - \mathbf{X}_0 \right) \right], \tag{3.5}$$

where c and ν are regularization parameter and steps size, respectively. The final solution $\mathbf{X}^*$ obtained after i^{th} iteration gives the desired SR output image.

3.2.2.2 Global dictionary learning

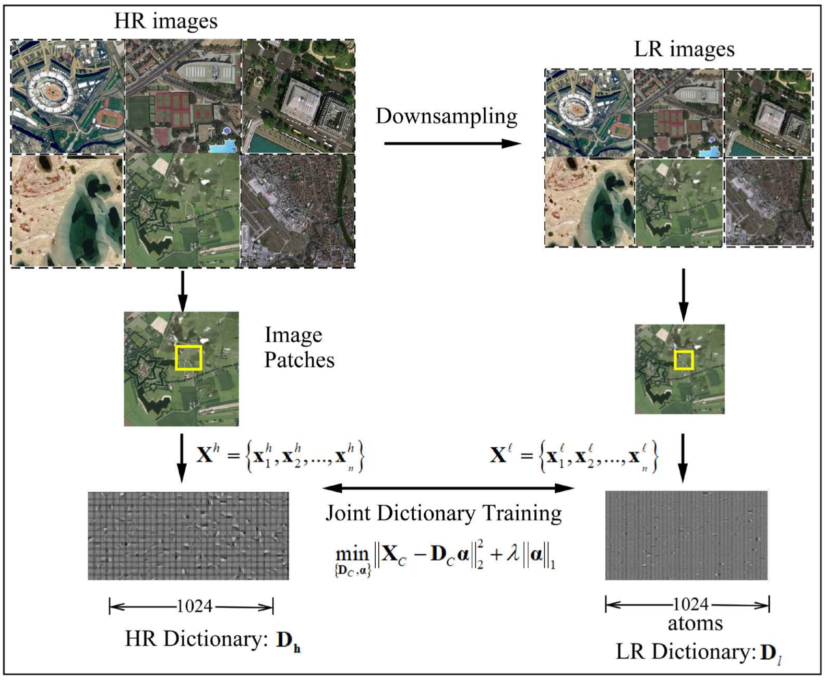

HR images
LR images
Downsampling
Image Patches
$\mathbf{X}^h = \{\mathbf{x}_1^h, \mathbf{x}_2^h, ..., \mathbf{x}_n^h\}$
$\mathbf{X}^\ell = \{\mathbf{x}_1^\ell, \mathbf{x}_2^\ell, ..., \mathbf{x}_n^\ell\}$
Joint Dictionary Training
$\min_{\{\mathbf{D}_C, \boldsymbol{\alpha}\}} \|\mathbf{X}_C - \mathbf{D}_C\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$
1024
HR Dictionary: $\mathbf{D_h}$
1024
atoms
LR Dictionary: $\mathbf{D}_l$

Figure 3.2: Sparse coding-based joint dictionary learning.

Fig. 3.2 shows the schematic of sparse coding-based coupled LR-HR dictionary training method. Suppose $\mathbf{X}^h = \left\{ \mathbf{x}_1^h, \mathbf{x}_2^h, ..., \mathbf{x}_n^h \right\}$ and $\mathbf{X}^\ell = \left\{ \mathbf{x}_1^\ell, \mathbf{x}_2^\ell, ..., \mathbf{x}_n^\ell \right\}$ are the sets of related HR and LR image patches in the training dataset. The combined sparse coding problem using both LR and HR patches is given by

$$\min_{\{\mathbf{D}_h, \mathbf{D}_\ell, \boldsymbol{\alpha}\}} \frac{1}{p} \left\| \mathbf{X}^h - \mathbf{D}_h \boldsymbol{\alpha} \right\|_2^2 + \frac{1}{q} \left\| \mathbf{X}^\ell - \mathbf{D}_\ell \boldsymbol{\alpha} \right\|_2^2 + \lambda \left(\frac{1}{p} + \frac{1}{q} \right) \|\boldsymbol{\alpha}\|_1, \tag{3.6}$$

where p and q are sizes of HR and LR image patch vectors, respectively. The ℓ_1 -norm term $\|\boldsymbol{\alpha}\|_1$ enforces the sparsity for both the LR and HR dictionaries, allowing to share the same sparse codes between them. Assuming $\mathbf{X}_C = \begin{bmatrix} \frac{1}{\sqrt{p}}\mathbf{X}^h \\ \frac{1}{\sqrt{q}}\mathbf{X}^\ell \end{bmatrix}$ and $\mathbf{D}_C = \begin{bmatrix} \frac{1}{\sqrt{p}}\mathbf{D}_h \\ \frac{1}{\sqrt{q}}\mathbf{D}_\ell \end{bmatrix}$ the HR-LR combined training patch set and HR-LR coupled dictionary, respectively, Eq. 3.6 can be re-written as

$$\{\mathbf{D}_C, \boldsymbol{\alpha}\} = \min_{\{\mathbf{D}_C, \boldsymbol{\alpha}\}} \|\mathbf{X}_C - \mathbf{D}_C\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 . \tag{3.7}$$

Learning $\mathbf{D}_C$ using Eq. 3.7 is similar to a single dictionary learning problem, which alternatively minimizes $\mathbf{D}_C$ and $\boldsymbol{\alpha}$ using the feature-sign-search algorithm [56] as done in [115].

3.2.3 Speed-up using multicore parallel processing

As shown in Fig. 3.1, the sparse representation-based SR algorithm has three major tasks- patch extraction, sparse representation, and SR reconstruction, which are implemented using a multicore-based parallel algorithm. Since, we train a global dictionary from a large dataset, the learned dictionary is applicable to any given test image from similar satellites. Therefore, the dictionary training is performed offline, parallelization is done only to the SR image reconstruction part.

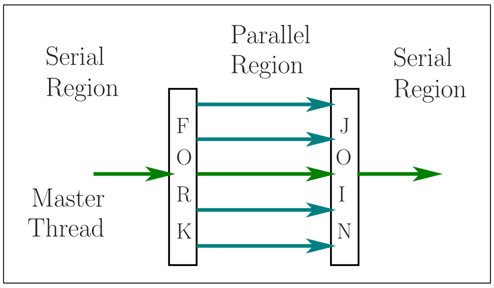


Figure 3.3: The fork-join model of parallel computing

The image processing tasks that involve execution of the same basic operations over a large number of pixels usually exhibits data level parallelism. We may divide such a huge set of basic operations into many smaller subsets such that the operations involved in a particular subset is not dependent on the result of any other subset, implying that they can be processed simultaneously. Based on the fork-join parallel model (shown in Fig. 3.3), a parallel sparse representation-based image super-resolution (Parallel ScSR) algorithm is implemented as given in Algorithm 1.

Algorithm 1: : Parallel Single Image SR Algorithm

```
Input: LR observed image $\mathbf{Y}$; dictionaries $\mathbf{D}_\ell$ and $\mathbf{D}_h$; c and maxItr

use #pragma omp parallel for
for each $5 \times 5$ patch $\mathbf{y}_i$ in $\mathbf{Y}$ do
    $m_i \leftarrow$ mean pixel value of $\mathbf{y}_i$;
    $\mathbf{y}_i \leftarrow \mathbf{y}_i - m_i$
end for
use #pragma omp parallel for
$xx \leftarrow$ patch indexes in $x$-dimension of SR image
$yy \leftarrow$ patch indexes in $y$-dimension of SR image
for (int $xx = 0$; $xx < rows$; $xx^{++}$) do
    for (int $yy = 0$; $yy < cols$; $yy^{++}$) do
        $\hat{\boldsymbol{\alpha}} = \min_{\hat{\boldsymbol{\alpha}}} \|\mathbf{D}_\ell \boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1$
    end for
end for
for each HR patch do
    $\mathbf{x}_i \leftarrow \mathbf{D}_h \hat{\boldsymbol{\alpha}}$; $\mathbf{x}_i \leftarrow \mathbf{x}_i + m_i$
end for
use #pragma omp parallel for
for (int $i = 0$; $i <$ (maxIter); $i^{++}$) do
    $\mathbf{X}^* = \arg \min_{\mathbf{X}} \|SH\mathbf{X} - \mathbf{Y}\|_2^2 + c \|\mathbf{X} - \mathbf{X}_0\|_2^2$
end for
Output: super-resolution image $\mathbf{X}^*$
```

The sparse representation-based SR algorithm is modified to incorporate loop level parallelism directives of OpenMP Multi-processing (OpenMP) into it which is written as follows: #pragma omp <directive> [clauses]. Among different directives, the 'parallel for' directive provides loop level parallelism in a signal or image processing algorithm in a easy manner. For example, let us consider a code that finds negative of an image by subtracting each pixel value from maximum gray level intensity i.e. 255. Using OpenMP we can parallelize this code simply as shown below:

Sequential loop	Parellelized loop using OpenMP

for(int x = 0; x<input.rows; x^{++}){
  for(int y = 0; y<input.cols; y^{++}){
    output.at(x,y) = 255 - input.at(x,y);
  }
}
	# pragma omp parallel for
	for(int x = 0; x<input.rows; x^{++}){
	  for(int y = 0; y<input.cols; y^{++}){
	    output.at(x,y) = 255 - input.at(x,y);
	  }
	}

The number of threads or cores among which the task is to be divided can be set during runtime by using *omp_set_num_threads* (*integer*). In a loop level parallelism, if we have total 200 iterations, we may assign five threads/cores to compute 40 iterations in each. The five threads will run simultaneously and the master thread resumes execution after all forked threads finish individual tasks.

3.2.4 Experiments and results

3.2.4.1 Simulation setup

A. Computing environment

We use C++ programming on PARAMTEZ, a high-performance computing (HPC) server from CDAC, India having the following specifications: 12 TF HPC system with 01 Master, 06 Nodes, 24 Cores, 64 GB RAM, and 50 TB Storage. Open source computer vision (OpenCV) library package- OpenCV2.4.9 and g++ (GCC) 4.8.5 compiler are used for the implementation of the proposed parallel SR algorithm in the server. For parallelization during compilation, OpenMP is integrated into the OpenCV environment. OpenMP includes a set of environment variables and runtime instructions that directs the compiler about how a program to be parallelized. The advantages of using OpenMP are due to its platform or operating system independence and easier in programming.

B. Database preparation

The database is divided into two parts: training and testing datasets. Training dataset consists of about 120 different RGB satellite images with spatial resolutions varying from 550×550 to 1200×1200 and collected from ISRO's public domain repository[1]. These images captured by the Cartosat-2 series satellite generally provide scene specific SPOT (Satellite for Observation of Earth) imagery useful for various cartographic applications. We obtain corresponding LR versions by applying the blurring and downsampling operations on them. Finally, patches are extracted from both the LR and HR images. For testing, few images which are not considered in the training are taken and then LR images of sizes 128×128, 256×256 and 512×512 are obtained from them.

C. Dictionary training

In this work, a pair of dictionaries $\mathbf{D}_h$ and $\mathbf{D}_\ell$ are learned by the joint dictionary training model defined in Eq. 3.7. The dictionaries consist of 1024 atoms in each and are learned using 1,00,000 LR-HR image patch pairs from the training dataset. The regularization parameter λ is taken as 0.15. Here, we have trained the dictionaries offline using the sequential programming, and on an average, it takes approximately 4 hours for convergence of the dictionary learning process in about 30 iterations.

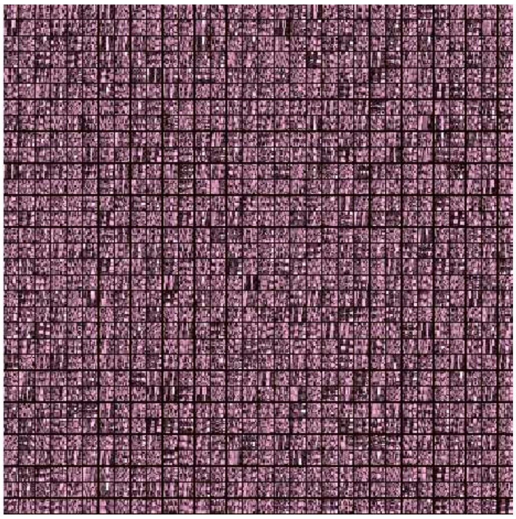


Figure 3.4: An example of the HR dictionary obtained by the joint dictionary training.

Fig. 3.4 presents an example of patch-based HR dictionary trained using 1,00,000 LR-HR patch pairs sampled from the luminance component (Y-channel) of the YCbCr image obtained from the RGB images for training. The size of each atom of the trained dictionary is taken as 25×1 and represented as a 5×5 patch in the figure.

[1] Images from Cartosat-2 Series Satellite, `http://www.isro.gov.in/pslv-c38-cartosat-2-series-satellite/images-cartosat-2-series-satellite`

3.2.4.2 Performance evaluations

A. Visual study

Super-resolution outputs for three different inputs Test1, Test2 and Test3 having resolutions of 128×128, 256×256 and 512×512 are shown in Fig. 3.5 for 2× zooming. A small portion of the zoomed image is also shown superimposed on the output image in the left hand side to highlight the visual quality of reconstruction.

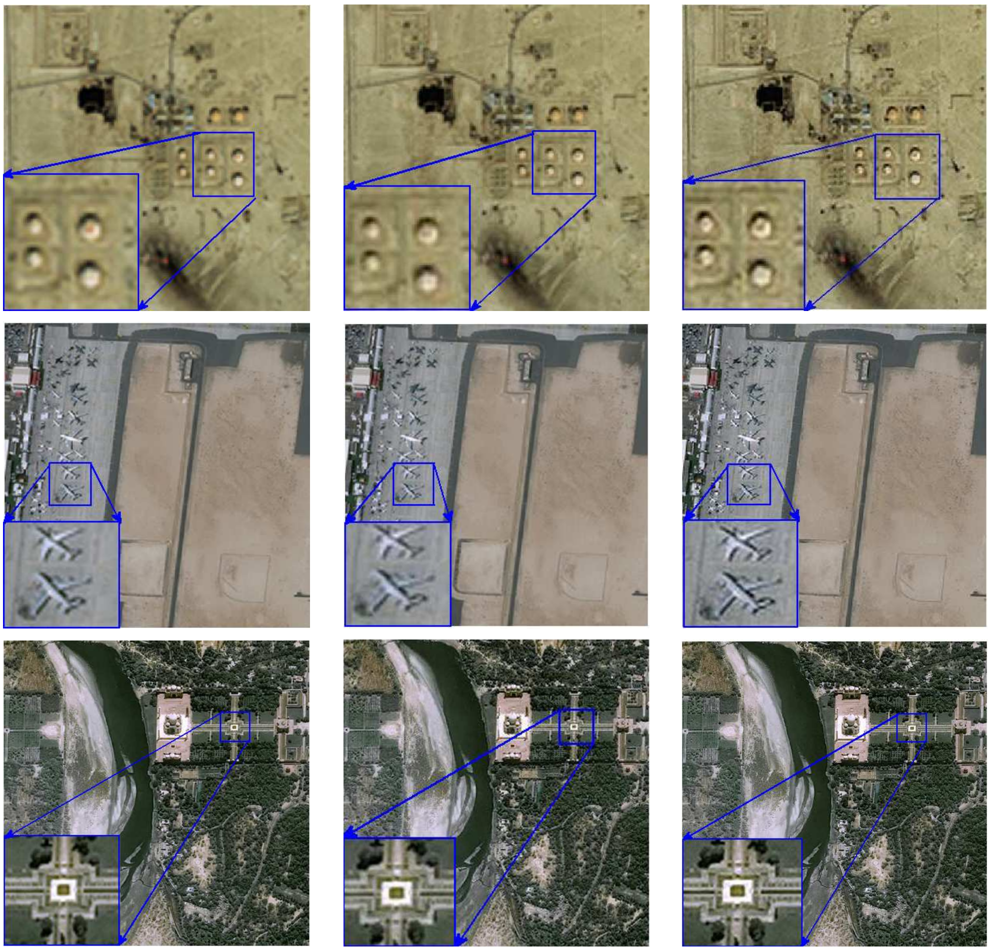


Figure 3.5: Results for a zooming factor of 2. First row left to right: Test1 and results of bicubic and the proposed method, respectively. Second row left to right: Test2 and results of bicubic and the proposed method. Third row left to right: Test3 and results of bicubic and the proposed method, respectively

B. Quantitative evaluation

Five quantitative metrics such as Peak signal-to-noise ratio (PSNR),mean structural similarity index measurements (MSSIM), spatial correlation coefficient (sCC), quality index (Q-index), and non-reference image quality evaluator (NIQE) are shown for the reconstructed images in Table 3.1 with zooming 2 and 4. It can be seen that the proposed method is able to give better results in terms of PSNR and MSSIM for all the test images compared to bicubic interpolation. Also, the other parameters such as Q-index, NIQE are significantly improved over the bicubic method. However, the Yang's method [115] when implemented using MATLAB, yields similar results with the proposed method, but at the cost of high computational time.

Table 3.1: Quantitative evaluation for the three test images with different zoom factors.

Parameters	**Test1**				**Test2**				**Test3**			
	Zoom = 2		Zoom = 4		Zoom = 2		Zoom = 4		Zoom = 2		Zoom = 4	
	Bicubic	Proposed	Bicubic	Proposed	Bicubic	Proposed	Bicubic	Proposed	Bicubic	Proposed	Bicubic	Proposed
PSNR (dB)	29.30	30.16	26.57	26.83	29.31	30.33	27.03	27.36	30.40	31.76	28.69	29.21
MSSIM	0.756	0.805	0.740	0.778	0.959	0.968	0.945	0.953	0.969	0.979	0.946	0.955
sCC	0.9296	0.9411	0.8820	0.8888	0.9492	0.9590	0.9206	0.9264	0.9828	0.9866	0.9453	0.9524
Q-index	0.6817	0.7440	0.3980	0.4608	0.5956	0.6448	0.3225	0.3656	0.8078	0.8447	0.5179	0.5904
NIQE	14.73	10.51	29.71	22.14	12.44	8.955	22.97	19.82	11.83	8.796	26.01	20.90

C. Time complexity and speed-up

The execution time is computed for SR reconstruction of the given test images in both sequential and parallel approaches. A measurement of speed-up i.e. the ratio of time taken in serial to the time taken in parallel executions, is obtained to find out the affect of varying the number of cores for parallel implementation. Fig. 3.6 shows that the Yang's method requires 98.41 secs in MATLAB for reconstructing a 256×256 image from a 128×128 image, whereas, in PARAMTEZ, using the sequential approach, we implement the same in 91 secs. On the contrary, with the application of parallel processing the time is reduced up to 8.4 secs., while using 24 cores simultaneously. Thus, the overall speed-up achieved using the proposed method is $= 91/8.4 = 10.83$.

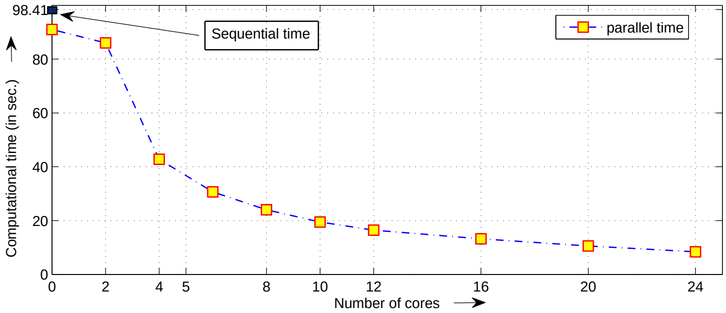


Figure 3.6: Plot of computation time versus number of cores used in PARAMTEZ

3.3 Parallel MS image SR based on sparse representations

3.3.1 Introduction

The data format and properties of multispectral (MS) satellite images are different from those of natural RGB images; extending a SISR method for MS images requires proper analysis of the MS image data and their processing. First, each band in the MS dataset have some geo-specific significance and they are not available in the RGB format. Secondly, some RGB images have varying amounts of luminance and chrominance distributions, which results in some chrominance channels that may contain some useful information for SR [69]. So, SR of the luminance channel alone may not give the best result. Lastly, if we generate an RGB image (false colour) by selecting only three bands it leads to spectral loss. Therefore, as shown in Section 3.2, we can not perform YCbCr colour transformation on MS images.

Here, we have proposed a fast MS image super-resolution (MSISR) work based on sparse representations and experiments are conducted using the QuickBird MS satellite image datasets available in the public domain at Global Land Cover Facility (GLCF)[2]. As shown in Fig. 3.7, the dataset consists of HR panchromatic (PAN) and four LR MS bands. Spatial resolution of MS bands are ≈2.8 m and that of PAN image is 0.72 m. The details of dictionary training and super-resolution image reconstruction process of the proposed method are discussed below. All implementations are carried out using OpenCV and C++ programming.

3.3.2 MS image sparse overcomplete dictionary learning

A schematic of the proposed dictionary learning procedure is detailed in Fig. 3.8. In this work, we learned a coupled sparse overcomplete dictionary by using the coupled

[2] GLCF: Global Land Cover Facility, available at `http://glcf.umd.edu/data/quickbird/`, accessed on November 20, 2018

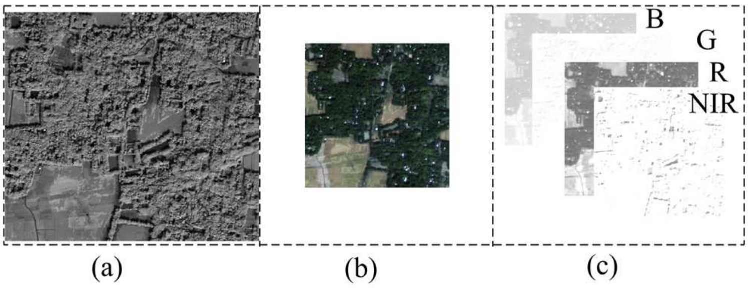


Figure 3.7: Example of a QuickBird MS image dataset: (a) PAN image of 0.7 m resolution (b) MS image of 2.8 m resolution (c) MS bands- blue, green, red and near infrared (NIR).

K-SVD dictionary training approach [1] on the MS training dataset.

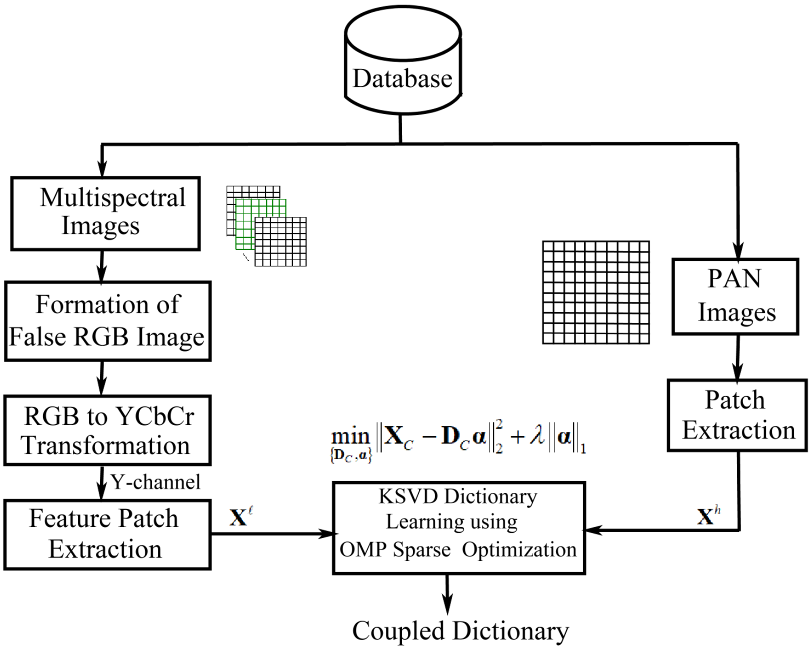

Database
Multispectral Images
Formation of False RGB Image
RGB to YCbCr Transformation
Y-channel
Feature Patch Extraction
$\mathbf{X}^{\ell}$
$\min_{\{\mathbf{D}_C,\boldsymbol{\alpha}\}} \|\mathbf{X}_C - \mathbf{D}_C\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$
KSVD Dictionary Learning using OMP Sparse Optimization
PAN Images
Patch Extraction
$\mathbf{X}^{h}$
Coupled Dictionary

Figure 3.8: Proposed dictionary training scheme for MSISR method.

An HR dataset $\mathbf{X}^h$ consisting of 5×5 patches extracted from all the PAN images in the database is created. For LR dataset $\mathbf{X}^\ell$, first, we generate a false RGB image from each MS image in the database and then trnasform them into YCbCr. Next, the luminance band image $\mathbf{Y}$ is applied to four 1-D high-pass filters as done in [115] for feature extraction. Finally, similar to HR dataset, patches of size 5×5 are extracted from the four feature images and concatenated to obtain a feature patch vector of length 100×1 against the 25×1 patch vector of the PAN image. In this way, by concatenating all the feature patch vectors into a matrix, we obtain the LR feature patch set $\mathbf{X}^\ell$. A combined dataset $\mathbf{X}_C$ is created by stacking the above generated HR and LR patch sets $\mathbf{X}^h$ and $\mathbf{X}^\ell$. We apply the K-SVD algorithm [1] to train a coupled dictionary $\mathbf{D}_C$ using the above dataset enforcing the single sparse coefficients matrix $\boldsymbol{\alpha}$ for both the HR-PAN and LR-MS patch sets. Before the training, the LR and HR dictionaries $\mathbf{D}_\ell$ and $\mathbf{D}_h$ are initialized using the sample patches directly. We avoid the random or DCT transform-based initializations here for simplicity. As illustrated in Algorithm 2, in K-SVD, the dictionary learning is done iteratively until convergence, where it solves a sparse regularization problem by applying the orthogonal matching pursuit (OMP) followed by update of the dictionary atoms one by one. To train $\mathbf{D}_C = [\mathbf{D}_\ell; \mathbf{D}_h]$ from 50,000 patches, on an average, five K-SVD iterations are required to achieve convergence. In our simulations, we learn coupled dictionaries of size 256, 512 and 1024, respectively. A view of the obtained dictionary by the proposed learning technique is shown in Fig. 3.9.

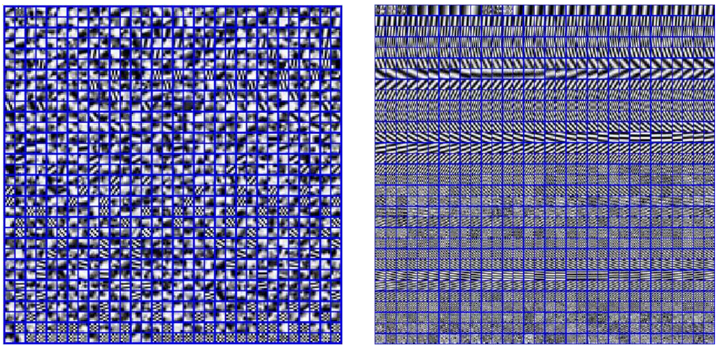
The coupled dictionary learning problem is given as follows:

$$\{\mathbf{D}_C, \boldsymbol{\alpha}\} = \min_{\{\mathbf{D}_C, \boldsymbol{\alpha}\}} \|\mathbf{X}_C - \mathbf{D}_C\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1, \tag{3.8}$$

where $\mathbf{X}_C = \begin{bmatrix} \frac{1}{\sqrt{p}}\mathbf{X}^h \\ \frac{1}{\sqrt{q}}\mathbf{X}^\ell \end{bmatrix}$ is the combined dataset of MS and PAN image patches, $\mathbf{D}_C = \begin{bmatrix} \frac{1}{\sqrt{p}}\mathbf{D}_h \\ \frac{1}{\sqrt{q}}\mathbf{D}_\ell \end{bmatrix}$ is the trained coupled dictionary, 'λ' is the regularization parameter and $\boldsymbol{\alpha}$ is the sparse vector common to both $\mathbf{D}_h$ and $\mathbf{D}_\ell$. Here, the size of HR patch- and LR feature-patch vectors are denoted by 'p' and 'q' respectively. If we select 1024 atoms, the size of LR and HR dictionaries $\mathbf{D}_\ell$ and $\mathbf{D}_h$ will be 100×1024 and 25×1024, respectively. For simulation, the value of regularization parameter λ is set to 0.1 and the error goal for the OMP is set to 0.001.

Algorithm 2: : K-SVD algorithm for Dictionary Learning

```
Input: Data $\mathbf{X}_C = \{\mathbf{x}_i\}_N \in^{(p+q)\times N}$ and Dictionary $\mathbf{D}_C \in \mathbb{R}^{(p+q)\times K}$
Initialize $\mathbf{D}_C^{(0)}$ using patches and $\ell_2$-normalized the columns;
Set $J = 1$.
while not converge do
    The sparse coding step
    for $i = 1$ to $N$ do
        Solve $\min_{\boldsymbol{\alpha}_i} \|\mathbf{x}_i - \mathbf{D}_C\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1$ using OMP;
    end for
    The dictionary update step
    for $k = 1$ to $K$ do
        $I \leftarrow \{j|\boldsymbol{\alpha}_{kj} \neq 0\}$;
        $\mathbf{E}_k^R \leftarrow \mathbf{X}_C - \sum_{j\neq k} \mathbf{d}_j \boldsymbol{\alpha}_I^j$;
        Apply SVD decomposition $\mathbf{E}_k^R = \mathbf{U}\Lambda\mathbf{V}^T$;
        $\mathbf{d}_k \leftarrow \mathbf{U}_{:,1}$;
        $\boldsymbol{\alpha}_R^k \leftarrow \mathbf{V}_{:,1}\Lambda(1,1)$;
    end for
    Set $J = J + 1$
end while
```



(a) (b)

Figure 3.9: Patches of trained overcomplete dictionaries by the proposed technique: (a) HR dictionary (b) LR dictionary

3.3.3 Super-resolution image reconstruction

Here, we propose MSISR based on patch-wise sparse representation of each band. As depicted in Fig. 3.8, we split the bands of a given QuickBird LR multispectral image into four bands- red, green, blue and near infrared (NIR). HR version of each band image is separately reconstructed using sparse representation and then applied to a global multispectral image reconstruction constraint-based regularization to obtain the final super-resolved MS image. An overview of the proposed MS image reconstruction using sparse representations over the trained coupled overcomplete dictionary is shown in Fig. 3.10.

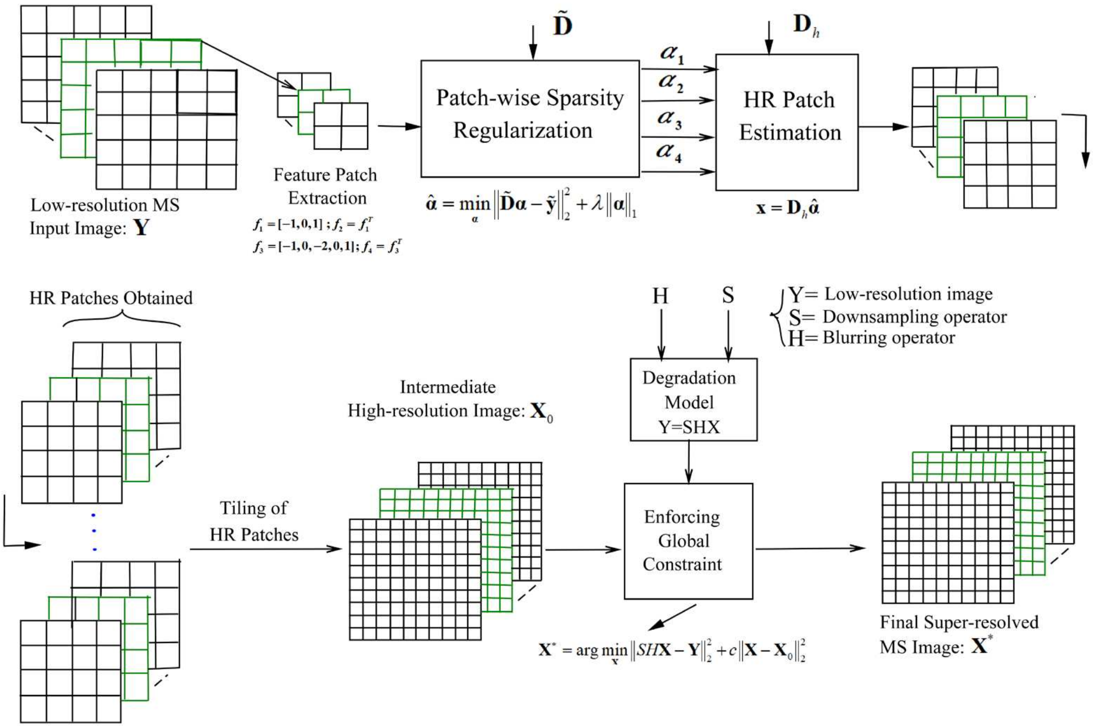

$\tilde{\mathbf{D}}$
$\mathbf{D}_h$
α_1
α_2
α_3
α_4
Patch-wise Sparsity Regularization
HR Patch Estimation
Feature Patch Extraction
Low-resolution MS Input Image: $\mathbf{Y}$
$f_1 = [-1,0,1]$; $f_2 = f_1^T$
$f_3 = [-1,0,-2,0,1]$; $f_4 = f_3^T$
$\hat{\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}} \|\tilde{\mathbf{D}}\boldsymbol{\alpha} - \tilde{\mathbf{y}}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$
$\mathbf{x} = \mathbf{D}_h\hat{\boldsymbol{\alpha}}$
HR Patches Obtained
H
S
Y= Low-resolution image
S= Downsampling operator
H= Blurring operator
Degradation Model Y=SHX
Intermediate High-resolution Image: $\mathbf{X}_0$
Tiling of HR Patches
Enforcing Global Constraint
$\mathbf{X}^* = \arg\min_{\mathbf{X}} \|SH\mathbf{X} - \mathbf{Y}\|_2^2 + c\|\mathbf{X} - \mathbf{X}_0\|_2^2$
Final Super-resolved MS Image: $\mathbf{X}^*$

Figure 3.10: Overview of the proposed MS image super-resolution reconstruction algorithm

As done in [72], for sparse representation problem, we consider dividing an LR band $\mathbf{Y} \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$ into overlapping patches of dimension $\sqrt{n} \times \sqrt{n}$. Next, we use the LR $\mathbf{D}_\ell \in \mathbb{R}^{n\times K}$ and HR $\mathbf{D}_h \in \mathbb{R}^{\bar{n}\times K}$ dictionaries, which are already trained using available training images. Here, $\bar{n}$ is equal to n multiplied by zoom factor. The reconstruction problem is implemented in two steps. In the first step, sparse representation co-efficient for each LR feature patch is obtained by sparse coding as follows:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \text{ such that } \ & \|\mathbf{D}_\ell\boldsymbol{\alpha} - \mathbf{y}\|_2^2 \leqslant \varepsilon_1 \ \text{ and} \\ & \|E_P\mathbf{D}_h\boldsymbol{\alpha} - \mathbf{w}\|_2^2 \leqslant \varepsilon_2, \end{aligned} \tag{3.9}$$

where E_P is a patch extraction operator that finds regions of overlap between the previously reconstructed HR patches and the patch under consideration, while ε represents error tolerance. Eq. 3.9 can be reformulated into an unconstrained problem using the Lagrangian form as follows:

$$\hat{\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}} \left\| \tilde{\mathbf{D}}\boldsymbol{\alpha} - \tilde{\mathbf{y}} \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha} \right\|_1 \tag{3.10}$$

where $\tilde{\mathbf{D}} = \begin{bmatrix} \mathbf{D}_\ell \\ E_P\mathbf{D}_h \end{bmatrix}$ and $\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{w} \end{bmatrix}$ and 'λ' is regularization parameter of the optimization problem. Next, the desired HR patches $\mathbf{x} \in R^{\bar{n}}$ can be obtained by multiplying $\hat{\boldsymbol{\alpha}}$ obtained above with $\mathbf{D_h}$, i.e.

$$\mathbf{x} = \mathbf{D_h}\hat{\boldsymbol{\alpha}}\mathbf{.} \tag{3.11}$$

All the HR patches '$\mathbf{x}_i$' obtained from Eq. 3.11 are positioned into an HR grid which results in an upscaled version '$\mathbf{X}_0$' for each input band '$\mathbf{Y}$'. However, the image generated in this manner may not have the same point spread function (PSF) as the original HR image. Besides, the generated image also has limitations in terms of inconsistency in patch positioning.

In the second step, we incorporate the image acquisition model equation i.e. $\mathbf{Y} = SH\mathbf{X}$ as an additional constraint into the solution space of $\mathbf{X_0}$. This helps in producing an image which is closer to $\mathbf{X}$ following the imaging model of actual HR image. The final SR image is obtained by solving the following minimization problem.

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \left\| SH\mathbf{X} - \mathbf{Y} \right\|_2^2 + \lambda \left\| \mathbf{X} - \mathbf{X}_0 \right\|_2^2 \tag{3.12}$$

Eq. 3.12 is efficiently solved by applying the gradient descent method and the final solution $\mathbf{X}^*$ is estimated iteratively.

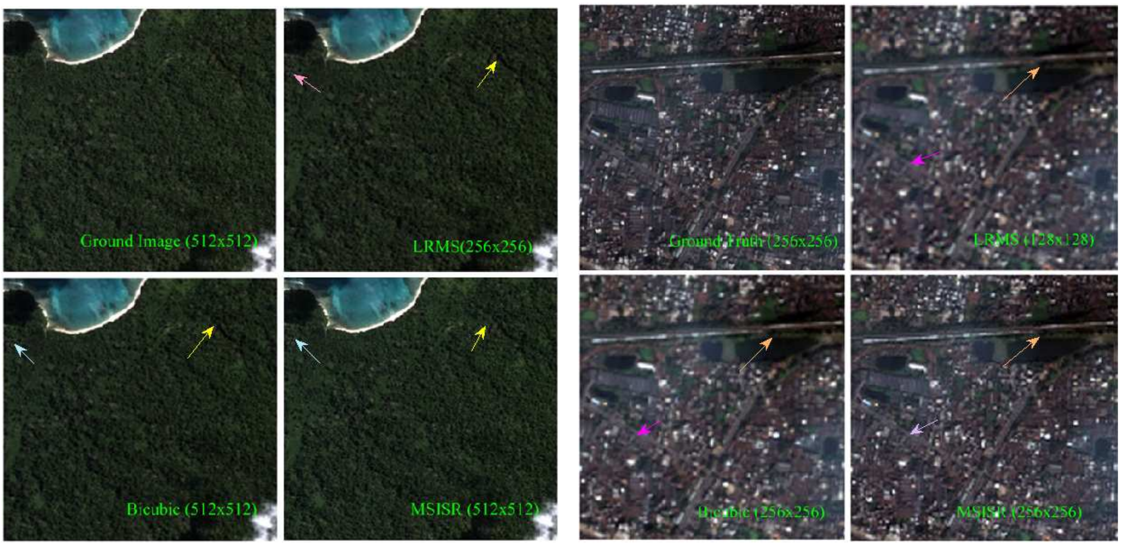
3.3.4 Multicore parallel implementation

Single instruction multiple data (SIMD) operations are divided among different processors to process rapidly. The proposed MSISR algorithm has three major computationally heavy sections- patch feature extraction, sparse optimization for LR feature patches and back-projection using the global imaging constraint during SR image reconstruction. We have designed a parallel version of the proposed algorithm to speed-up its execution on a multicore computing system.

3.3.5 Experiments and results

3.3.5.1 Visual study

The proposed MS image SR based on sparse representation is tested on two low-resolution MS images: a 256×256 land cover image of the areas of Indonesia and Ujong, and a 128×128 image of the areas of Sundarban, India. Output images for all the methods are displayed in Fig. 3.11 below. Few regions of the results are also marked with arrows, where noticeable visual enhancements are observed.



(a) Test1

(b) Test2

Figure 3.11: SR results of 'Test1' and 'Test2' images corresponding to upscale ratio 2. First row left to right: ground-truth and LR MS image downsampled by 2. Second row left to right: results of bicubic and the proposed methods.

3.3.5.2 Objective Evaluations

Besides peak signal-to-noise ratio (PSNR) and mean structural similarity (MSSIM) index, we evaluate three other objective parameters for quantitative evaluation of the output images. They are, namely, universal image quality index (Q-index), spectral angle mapper (SAM), and erreur relative globale adimensionnelle de synthese (ERGAS).

Table 3.2: Comparison of quantitative parameters for Test1 and Test2 MS Images

Parameters	Test1 Input: 256×256, Output: 512×512			Test2 Input: 128×128, Output: 256×256		
	Bicubic	SparseFI [133]	MSISR	Bicubic	SparseFI [133]	MSISR
PSNR	29.40	30.39	30.42	28.44	29.32	29.38
MSSIM	0.961	0.993	0.994	0.810	0.848	0.859
Q-index	0.823	0.845	0.885	0.960	0.988	0.992
SAM	1.330	1.250	1.254	0.779	0.767	0.767
ERGAS	1.543	1.400	1.390	1.230	1.124	1.110

Results for Test1 and Test2 MS images are shown in Table 3.2. We observe that for MS images, the proposed method outperforms bicubic interpolation and gives comparable results with SparsFI.

3.3.5.3 Time complexity and speed-up

In the proposed method, the major time consuming operations in the algorithm are as mentioned in Table 3.3. Here, most of the time is consumed in feature extraction, sparse coding using ℓ_1-minimization, and generation of high-resolution patches.

Table 3.3: Sequential execution time (in secs.)

Function	**Total time**
Feature extraction	46
ℓ_1-minimization	97
HR patch generation + others	5

Time required to execute the algorithm is computed both for sequential as well as multicore parallel implementations. It is observed that the sequential program takes

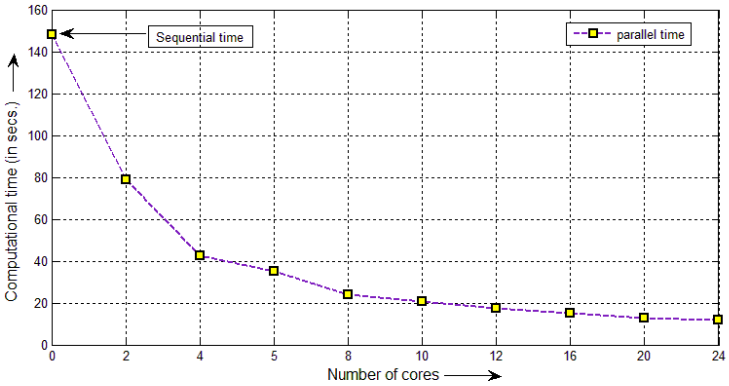


Figure 3.12: Plot of computation time vs. number of cores.

around 148 secs. of time for an 128×128 image to upscale by 2. However, as shown in Fig. 3.12, this time is reduced to approximately 12 secs using OpenMP. Thus, overall speed-up that is achieved using the proposed method is $s = 148/12 = 12.33$.

3.3.5.4 Effects of dictionary size

We have tested the proposed method using three dictionary sizes - 256, 512 and 1024. Using 50,000 training patches, and 40 iterations of the K-SVD dictionary training requires 2 hr 16 mins., 3 hr 05 mins., and 5 hr 50 mins., respectively. Fig. 3.13

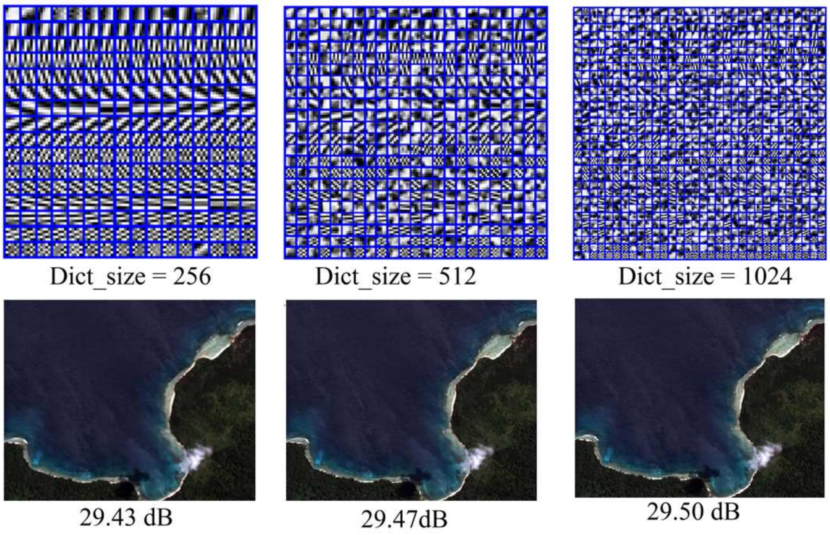


Figure 3.13: Samples of trained dictionaries of sizes 256, 512 and 1024 and corresponding reconstructed images.

shows the trained dictionaries of various size and corresponding SR results for an

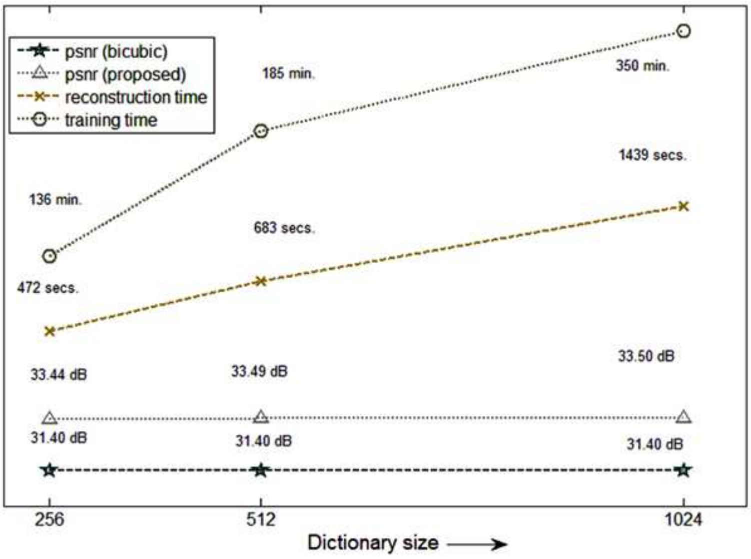


Figure 3.14: Effect of dictionary size on reconstruction time and quality (in terms of PSNR).

example test image. It is found that with increasing dictionary size, the computation time required for SR image reconstruction also increases linearly. However, there is a light increase in the quality of reconstructed images for higher dictionary sizes beyond 512. Fig. 3.14 demonstrates the impact of dictionary size on reconstruction time and output PSNR for upscaling of a 256×256 image by 2.

3.3.5.5 Selection of sparse coding technique

In this work, to obtain faster results based on sparse representation method, we have considered four state-of-the-art ℓ_1-minimization tools for the sparse representation of sample image patches using the same overcomplete dictionary. They are, namely, the iterative shrinkage and thresholding algorithm (ISTA) [29], the fast ISTA i.e. FISTA [8], the orthogonal matching pursuit (OMP) [13], and the feature-sign search algorithm [56]. We compute reconstruction time taken by each method for a fixed dictionary of size 256 and λ=0.15, and shown in Table 3.4 for an 128×128 size image upscaled by 2.

Table 3.4: Time complexity of different sparse coding techniques

Tools	ISTA	FISTA	OMP	Feature-sign
Time (in secs.)	659.15	620.05	81.02	49.71
PSNR (in dB)	29.92	30.00	29.92	30.07
MSSIM	0.797	0.800	0.797	0.802

It has been observed that for patch-wise sparse regularization problems, the ℓ_1-feature-sign-search-based method produces the quickest results, while yielding the highest PSNR value at the same time.

3.4 Pansharpening-based spatial/spectral SR of MS images

Among the remote sensing satellites, several satellites, such as Landsat, QuickBird, SPOT, and others have both the LR MS sensors as well as band images as well as a corresponding HR PAN sensor on board. For example, as given in Table 3.5, QuickBird MS image dataset consist of a PAN image (band) acquired by the HR PAN sensor (with 0.65 m resolution) and four LR MS bands acquired by the MS sensor (with 2.62 m resolution). Similarly, the MS image dataset of the Landsat-7 satellite is comprised of PAN and MS images with resolutions of 15 m and 30 m, respectively. The LR spectral bands are combined to provide an MS satellite image, whereas the PAN image is a single band grayscale image. Fig. 3.15 shows a comparison of MS and PAN images of the same location, demonstrating the visual differences between the two.

Table 3.5: Details of QuickBird multi-spectral image dataset

Satellite	Band Information/ Wavelengths	Swath	Resolution
QuickBird	PAN: 0.45-0.90 μm Blue: 0.45-0.52 μm Green: 0.52-0.60 μm Red: 0.63-0.69 μm NIR: 0.76-0.90 μm	16.8 Km	PAN: 0.65 m MS: 2.62 m

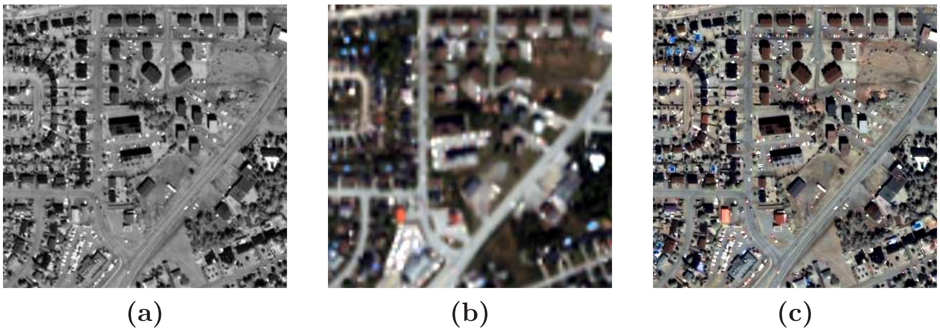

(a) (b) (c)

Figure 3.15: Example of QuickBird dataset of the a specific area: HR PAN image (left), LR MS image (centre), and Pan-sharpened image (right).

HR MS images can be produced by fusion of the HR PAN and the LR MS images. These methods are known as pansharpening as the resolution of MS images are made equal to that of the PAN images. A large number of works on the fusion of MS and PAN images are explained in [4]. Some of the existing pansharpening methods, which are very popular are based on - intensity-hue-saturation (IHS)[106], principal component analysis (PCA) [90] and Brovery transform [129] methods. A brief overview of these methods are given in Table 3.6.

Table 3.6: Comparison of different pan-sharpening and SISR works

Methods	Salient Features	Remarks
IHS [106]	A new look at IHS-like image fusion methods	– Its a PAN injection method – RGB to IHS transform
PCA [90]	An efficient pan-sharpening method via a combined adaptive PCA approach and contourlets	– Adaptive PCA-contourlet approach – Reduces spectral distortion
Brovery [129]	Problems in the fusion of commercial high-resolution satellite as well as Landsat-7 images and initial solutions	– Multiplies each MS band with PAN – Divides by sum of all MS bands
SparseFI [133]	A sparse image fusion algorithm with application to pansharpening	– Sparse Fusion based on compressive sensing – Less spectral distortions and better spatial resolution
ScSR [115]	Image super-resolution via sparse representation	– Global dictionary training – High accuracy natural image SISR

These are substitution-based methods, where the MS image is first transformed into a color image and then an resized or interpolated MS image is obtained to get a spatial resolution in the order of the PAN image. Next, the pixels of the luminance channel of the color transformed MS image is replaced with those of the PAN image and the final HR image is obtained through an inverse color transformation. One major limitation in this approach is due to the difference in statistical distribution of the pixels of the luminance channel and the PAN image; the output images produced by such methods suffer from significant spectral distortions [91].

Table 3.7: SISR over the Pansharpening methods

SISR	Pansharpening
Uses the LR input only	Uses the LR MS and HR PAN images
Reconstructs high spatial/spectral output	Output has spectral distortions
Applicable to any MS dataset	Not applicable to MS images without PAN band
Example: Resourcesat-2, LISS-III, and LISS-IV, QuickBird, etc.	Example: Landsat, ETM, ETM+, QuickBird, SPOT, etc.

The SISR method, on the other hand, generates an HR version of a single band, and for the MS images, each reconstructed band is concatenated to produce the final image. The prime objective in this work is to produce an HR MS image from the given LR MS image using the SR methods. Although the approaches of pansharpening and SISR are similar, they differ in that pansharpening only improves the spatial information of MS images, whereas SR attempts to estimate the spatial and spectral information of the target HR image from the LR image itself. Table 3.7 gives a comparison of the pansharpening- and SR-based reconstruction methods of MS images. Most of the sparse representation-based SR works in the literature focus-on learning a pair of dictionaries (LR and HR) from a given dataset of HR RGB images and reconstructing the luminance channel of the LR input using a patch-wise sparse representation technique. However, in the case of SR of MS images, a standard HR MS image dataset is usually not available to learn the dictionary pairs, and reconstruction from a transformed RGB image in pansharpening results in spectral distortions.

To address the dictionary learning issue, we first focus on using input HR PAN images to train the LR/HR dictionary pair because they contain high spatial details, which are desired in the target HR image. Zhu *et al.* [133] present a similar pansharpening work in which they utilize PAN image patches as dictionary atoms. We concentrate on training of coupled sparse overcomplete dictionaries based on the extraction of features, such as edges, from PAN image patches, so that these features can be used for improved representation of LR patches during reconstruction. Second, rather than reconstructing an RGB MS image, we reconstruct each MS band separately and combine them to obtain the resulting HR MS image, allowing each band to retain its original spectral properties.

3.4.1 Proposed method

The proposed SR algorithm considers the input PAN image $\mathbf{X}_P$ for learning a pair of overcomplete dictionaries $\mathbf{D}_\ell$ and $\mathbf{D}_h$. Then, patch-based sparse representations of the input LR MS image $\mathbf{Y}$ is carried out to generate the desired output HR MS image $\mathbf{X}$. The work flow of the proposed SR algorithm is depicted in Fig. 3.16. It comprises of two main steps: dictionary training and SR reconstruction.

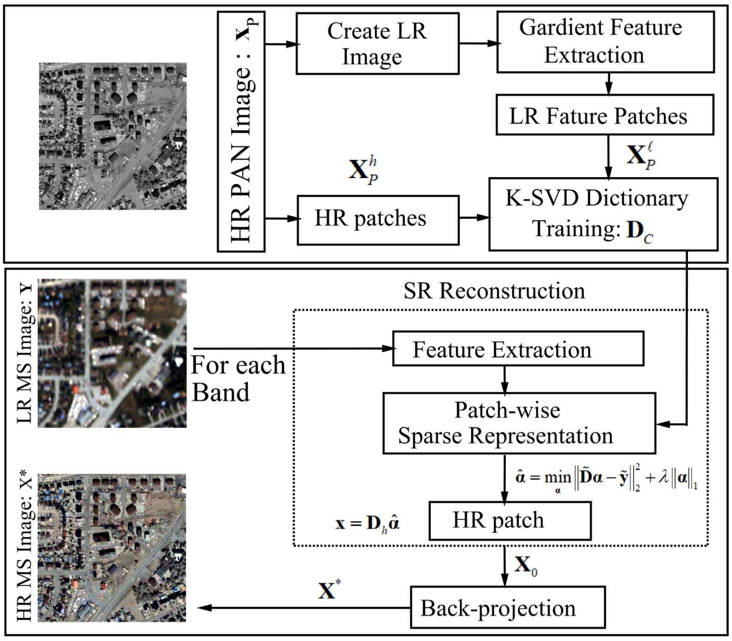

HR PAN Image : $\mathbf{X}_P$
Create LR Image
Gardient Feature Extraction
LR Fature Patches
$\mathbf{X}_P^\ell$
$\mathbf{X}_P^h$
HR patches
K-SVD Dictionary Training: $\mathbf{D}_C$
LR MS Image: Y
For each Band
SR Reconstruction
Feature Extraction
Patch-wise Sparse Representation
$\hat{\boldsymbol{\alpha}} = \min_{\alpha} \|\tilde{\mathbf{D}}\boldsymbol{\alpha} - \tilde{\mathbf{y}}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1$
$\mathbf{x} = \mathbf{D}_h \hat{\boldsymbol{\alpha}}$
HR patch
$\mathbf{X}_0$
HR MS Image: X*
$\mathbf{X}^*$
Back-projection

Figure 3.16: Proposed MS image SR algorithm.

3.4.1.1 Dictionary training

Before extraction of LR patches from the given PAN image $\mathbf{X}_P$, it is blurred and downsampled according to the image degradation model so that the actual PSF of the input MS image fits into it approximately. The blurred and downsampled PAN image is then passed through four 1D-filters of orders 1 and 2, to extract both horizontal and vertical gradient features from it.

$$f_1 = [-1, 0, 1], \quad f_2 = [1, 0, -2, 0, 1], \quad f_3 = f_1^T, \; f_4 = f_3^T, \tag{3.13}$$

The four feature vectors obtained from each patch after filtering are concatenated into a single vector that represents an LR feature patch. Again, HR patches are also extracted directly from the given PAN image. Two metrices $\mathbf{X}_P^h$ and $\mathbf{X}_P^\ell$ are created which contain all the HR and LR feature vectors stacked into the columns of the metrices. A sparse representation problem is then formulated to train the LR-HR dictionary pair as a coupled dictionary $\mathbf{D}_C$ from the combined inputs of $\mathbf{X}_P^h$ and $\mathbf{X}_P^\ell$.

$$\{\mathbf{D}_C, \boldsymbol{\alpha}\} = \min_{\{\mathbf{D}_C, \boldsymbol{\alpha}\}} \|\mathbf{X}_C - \mathbf{D}_C\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 , \tag{3.14}$$

where $\mathbf{D}_C = \begin{bmatrix} \frac{1}{\sqrt{p}}\mathbf{D}_h \\ \frac{1}{\sqrt{q}}\mathbf{D}_\ell \end{bmatrix}$ and $\mathbf{X}_C = \begin{bmatrix} \frac{1}{\sqrt{p}}\mathbf{X}_P^h \\ \frac{1}{\sqrt{q}}\mathbf{X}_P^\ell \end{bmatrix}$; here p and q represent the sizes of the HR and LR feature patches in vector form. The least-square minimization based problem in Eq. 3.14 is solved by utilizing the optimized K-SVD training algorithm [87] to obtain the coupled trained overcomplete dictionary $\mathbf{D}_C$.

3.4.1.2 Super-resolution reconstruction

The LR MS image is processed band-wise for SR reconstruction. A selected band image is first applied to the feature extraction stage to get the feature patch vectors as done in dictionary training. Next, for each feature patch vector $\mathbf{y}$ in LR MS image, a sparse representation problem is formulated using the trained dictionaries $\mathbf{D}_\ell$ and $\mathbf{D}_h$ as $\tilde{\mathbf{D}}$ and it is given as:

$$\hat{\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}} \left\|\tilde{\mathbf{D}}\alpha - \tilde{y}\right\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 , \tag{3.15}$$

where $\tilde{\mathbf{D}} = \begin{bmatrix} \mathbf{D}_\ell \\ E_p\mathbf{D}_h \end{bmatrix}$ and $\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{w} \end{bmatrix}$; E_p extracts the overlapping region between the current patch under reconstruction and the previously reconstructed HR patch; w represents the pixels of overlapped region, and 'λ' is a regularization parameter of the optimization problem. We estimate the sparse coefficient vector $\hat{\boldsymbol{\alpha}}$ by solving the Eq. 3.15 using the feature-sign search based convex optimization algorithm [56].

Since, the patches of both LR and HR images share a common sparse representation with their individual dictionaries, HR image patches can be reconstructed as follows:

$$\mathbf{x} = \mathbf{D}_h\hat{\boldsymbol{\alpha}} \tag{3.16}$$

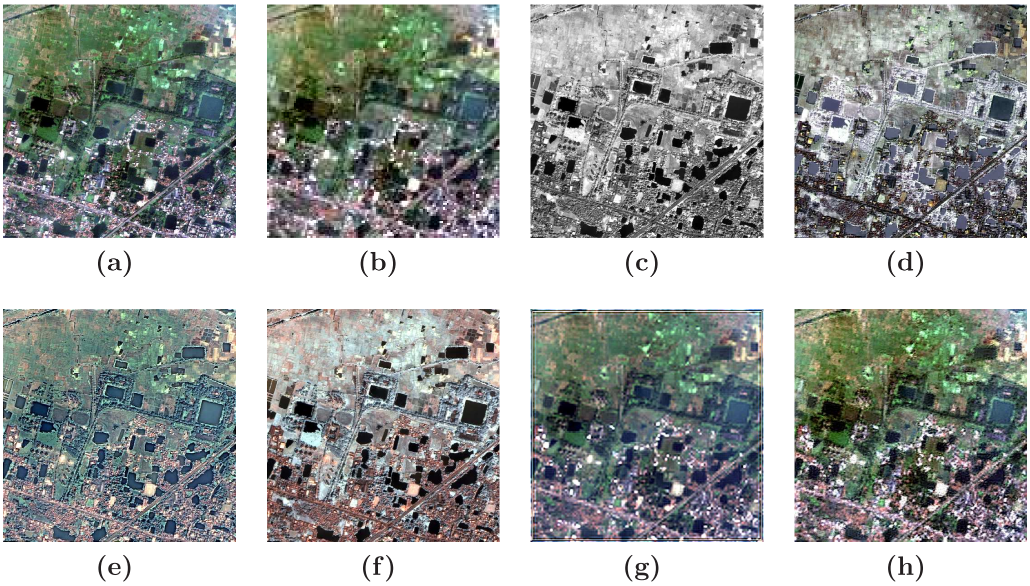
Tiling all the reconstructed patches in its corresponding band yields an intermediate HR image $\mathbf{X}_0$. Finally, back-projection is applied to satisfy the global imaging model constraint i.e $\mathbf{Y} = SH\mathbf{X}$ on $\mathbf{X}_0$ to obtain the final HR image $\mathbf{X}^*$. Mathematically,

$$\mathbf{X}'^* = \arg\min_{\mathbf{X}} \|SH\mathbf{X} - \mathbf{Y}\|_2^2 + c\,\|\mathbf{X} - \mathbf{X}_0\|_2^2 \tag{3.17}$$

Eq. 3.17 is efficiently solved using the gradient descent method.

3.4.2 Experiments and Results

Experiments are carried out using the proposed method on two test MS images and results are shown with four existing MS image SR methods based on pansharpening, namely, the IHS [106], the PCA [90], the Brovery [129], and the SparseFI [133]. Datasets containing PAN and MS images of size 2048×2048 and 512×512, respectively are acquired from the QuickBird sensor over the regions- the Sundarbans, India captured on 02 November, 2002 and collected from GLCF[3].


(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.17: Sample images from QuickBird and reconstructed outputs using different methods. First row (from left): Ground-truth MS, Downsampled MS, Downsampled PAN, and IHS. Second row (from left): PCA, Brovery, SparseFI and the Proposed.

In this experiment, the input PAN and MS images are downsampled by a factor 4. Thus, for both the datasets, the dimensions of the test LR MS image and trainable

[3] Global Land Cover Facility, (Accessed on 20 March, 2019). ftp://ftp.glcf.umd.edu/glcf/Quick Bird/

Table 3.8: Performance evaluation for QuickBird first dataset.

Parameter	IHS [11]	PCA [9]	Brovery [13]	SparseFI [14]	Proposed
RMSE	36.10	33.37	33.97	24.23	23.62
CC	0.589	0.738	0.715	0.769	0.813
SD	38.88	29.25	17.66	15.55	13.18
SAM	9.04	8.29	10.63	7.20	7.02
UIQI	0.742	0.842	0.878	0.805	0.860
ERGAS	22.15	12.82	18.24	8.24	8.11

PAN image are resized to 128×128 and 512×512, respectively. During the training phase, a coupled overcomplete dictionary consisting of 1024 atoms is learned using 10000 sample patches taken from the PAN image. Here, we consider extracting patches of size 7×7 as reported by Zhu *et al.* [133] and shown better results. In this work, each LR MS band image is upscaled individually.

For quality assessment of the resulted images of different methods, the quantitative metrics computed are as follows: root mean-square error (RMSE), spatial correlation coefficient (sCC), spectral distortion (SD), universal image quality index (UIQI), spectral angle mapper (SAM), and erreur relative globale adimensionnelle de synthese (ERGAS). The output images by different methods for the two test

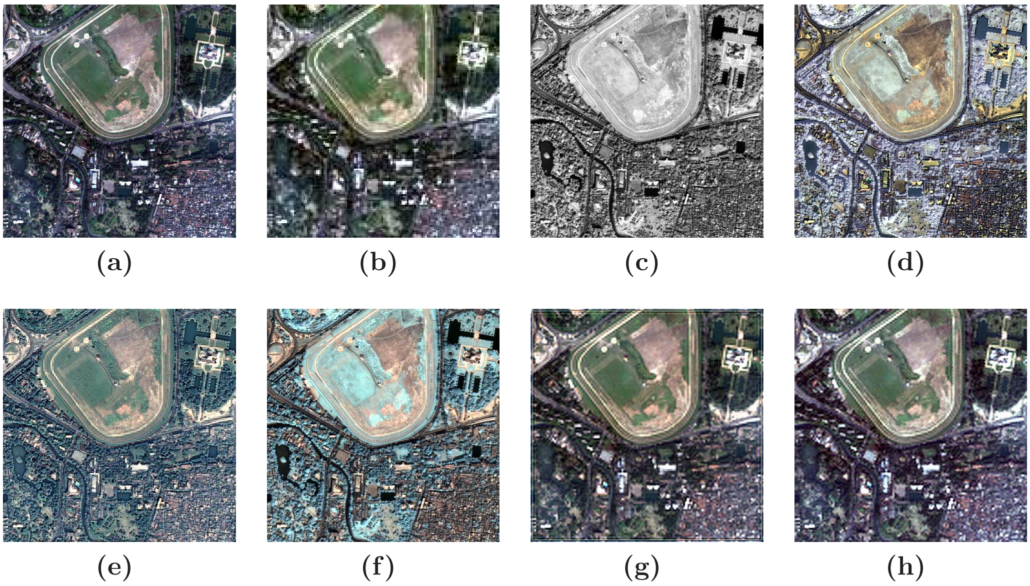


Figure 3.18: QuickBird images and results using different methods. First row (from left): Ground-truth MS, Downsampled MS, Downsampled PAN, and IHS. Second row (from left): PCA, Brovery, SparseFI and the Proposed.

datasets are visually presented in Figs. 3.17–3.18. The visual outputs for two different datasets are quantitatively validated using the above mentioned parameters in Tables 3.8-3.9. Results show that the proposed method provides better spatial information, while causing less spectral distortion than others.

Table 3.9: Performance evaluation for QuickBird second dataset.

Parameter	IHS [11]	PCA [9]	Brovery [13]	SparseFI [14]	Proposed
RMSE	24.51	18.90	20.86	14.10	13.25
CC	0.714	0.807	0.817	0.828	0.839
SD	31.70	27.93	14.52	12.31	10.92
SAM	8.76	7.97	9.39	7.45	0.703
UIQI	0.773	0.813	0.818	0.845	0.894
ERGAS	25.14	13.57	20.14	9.06	8.76

3.5 Conclusion

In this chapter, we have discussed three different algorithms of fast MS image SR using sparse representations. The first algorithm presents an accelerated multicore parallel processing-based SISR algorithm for RGB satellite images. Experiments are carried out using ISRO's Cartosat-2 series satellite images. Results show reasonable PSNR and MSSIM improvements besides better visual quality.

The second algorithm presents a sparse overcomplete dictionary-based parallel MS image super-resolution method. Here, we have utilized a database consisting of MS and PAN images from QuickBird sensor as LR and HR training datasets and performed band-wise SR of the given LR MS image. Remote sensing image specific objective evaluation parameters, like SAM, ERGAS and Q-index, etc., are measured from reconstructed images to evaluate and compare the visual quality. However, it is observed that the execution time for the 4-band QuickBird image reconstruction is very high compared to other color/gray image SR. Furthermore, dictionary training is a time consuming affair. We demonstrate that parallel processing-based implementation technique greatly reduces the reconstruction time and able to achieve speed-up of around 10 to 12 times.

In the third algorithm, a pansharpening-based MS image SR algorithm is presented. Here, we have trained a pair of LR/HR dictionaries from a single PAN image, then the LR MS image is processed using sparse representation with these dictionaries to get the target HR MS image. Results are compared with other sparse representation-based SISR and pansharpening methods. It is found that the proposed method is able to generate MS images having better spatial as well as spectral resolutions than the other comparing methods.

As a future work, we can consider more effective feature extraction strategies useful for MS remote sensing imagery. This can help to improve the sparse representation as well as the quality of trained dictionaries. Additionally, improved regularization problems incorporating new *a priori* information can be considered to reconstruct the spatial and spectral information. Moreover, in this work, we have shown speed-up of the SR image reconstruction only to obtain near real-time outputs. However, in practical case, we may need to train a dictionary for a given set of unknown images to perform SR of new images. Therefore, it is also important to develop a parallel version of the sparse representation-based coupled over complete dictionary, which may be used in real-time applications.