Blind Super-Resolution for Single Remote Sensing Image via Sparse Representation and Transformed Self-Similarity

Yuhang Liu and Weidong Sun
Beijing National Research Center for Information Science and Technology, Dept. of Electronic Engineering, Tsinghua University, Beijing 100084, China
Email: liyuh17@mails.tsinghua.edu.cn, wdsun@tsinghua.edu.cn

Abstract. Super-resolution (SR) reconstruction is one of the effective ways to improve the spatial resolution of remote sensing images, but the blur kernel estimation without any additional prior information is the key to the quality of SR reconstruction. Facing the above problem, a blind super-resolution method via sparse representation and transformed self-similarity is proposed in this paper. In this method, the blur kernel as well as the reconstructed high-resolution image are estimated at the same time using the structural self-similarity from a single image, a down-sampled version of the observed image is used as the training samples for the dictionary learning, and the internal patch search space is expanded using the transformed self-similarity to improve the quality of estimation. Experiment results show that, our method performs well both in quality of kernel estimation and SR reconstruction.

1. Introduction
Remote sensing (RS) image plays an important role in target detection, object recognition, land-use classification, etc, in which spatial resolution is one of the most important critical indicators to those applications. Given many complex factors, such as the limitations of chips and sensors, as well as production costs, signal processing is still regarded as one of the most effective approaches to obtaining high-resolution (HR) images [1].

Super-resolution (SR) reconstruction aims to increase the spatial resolution of images from one or several low-resolution (LR) images of the same scene. It’s often assumed that the LR image \( l \) is obtained by blurring and then down-sampling from its HR version \( h \) as following:

\[
l = (h * k) \downarrow_s
\]  

(1)

where, \( k \) is the blur kernel and \( s \) is the down-sampling scale. In which, blur kernel \( k \) is often assumed as Gaussian or Bicubic [2-6]. However, the blur kernels relating with HR and LR images in real imaging systems are more complicated, we cannot take them as a simple Point Spread Function (PSF) of the camera [7,8]. So, SR methods may suffer undesired performance loss, like over-smoothing and ringing artifacts [11], with an inaccurate blur kernel. It is essential to give an estimation of the blur kernel before reconstructing the high-resolution one. This task refers to blind super-resolution, which assume the blur kernel of the degradation procedure is unavaiable, and the blind means that there is no any additional prior information can be used. Blind SR is a typical ill-posed problem, most solutions to this problem is by making model assumptions to the blur kernel [9-12], then solve the model by optimization or seeking help from internal priors. In [9], the authors use different parametered models as kernel candidates, like motion blur or Gaussian blur, they make parameter estimation for each candidate and select the best one. [10] uses an MAP estimation for both the blur kernel \( k \) and the HR image \( h \). In [11], the authors base on the observation that kernel mismatch could
bring either over-sharpening or over-smoothing, they use this prior to correct the kernel, then apply it to the next SR procedure.

Considering structural self-similarity between different scales can bring us some kinds of prior to the blur kernel estimation, our solution is trying to exploit the self-similarity over the scales, to help our blind SR just using one single LR image. In our method, the blur kernel as well as the reconstructed high-resolution image are estimated at the same time using the structural self-similarity from a single image, a down-sampled version of the observed image is used as the training samples for the dictionary learning, and the internal patch search space is expanded using the transformed self-similarity to improve the quality of estimation. Experiment results show that, our method performs well both in quality of kernel estimation and SR reconstruction.

2. Transformed Self-Similarity

Structural self-similarity in an image refers to that the structure in small patches will recur both in and across the scales. In [13], the authors did an exhaustive analysis on a very wide diversity of images, the result of their statistical analysis shows that patches tend to recur much more times inside the same image than in any other random external collection, this internal data redundancy is an often-used priori for patch based algorithms. RS image is also rich in self-similarity, such as, buildings, roads and rivers. Except the multi-scale similarity extracted directly from the image pyramid, there is also “rotation similarity”, especially within RS images. Consider the buildings and roads that have different orientations, they are similar after some rotation transformations, here we call this as “transformed self-similarity”.

To illustrate the transformed self-similarity, consider a patch \( p \in \mathbb{R}^{m \times m} \), where there’s another patch \( p_\theta \in \mathbb{R}^{m \times m} \), that is “similar” to \( p \) but has a rotation angle \( \theta \) around its centre. If we rotate \( p_\theta \) anticlockwise by \( \theta \), we will have the similar patch pairs. We express the rotation transformation as \( T_\theta(\ast) \), which rotates a patch clockwise by \( \theta \) around its centre, then \( T^{-1}_\theta(\ast) \) rotate the patch anticlockwise, thus we have:

\[
T^{-1}_\theta(p_\theta) = p
\]

we call \( p_\theta \) the transformed self-similar patch of \( p \) under angle \( \theta \). As shows in Figure 1, patch D is self-similar patch to patch A, while patch B and patch C are transformed self-similar patches of A with different rotation angles.

![Figure 1](image)

**Figure 1.** An example of transformed self-similarity

The rotation transformation can be applied during the sampling procedure, when we just sample the image by a rotated template. Consider an \( m \times m \) template, we rotate this template around its centre, then anchor the centre of the template to location \( [x, y] \), we can get all the corresponding coordinates and thus sample in a bilinear way, producing patches located at \( [x, y] \). To avoid the borders that cannot sample those rotated patches, we only sample the central part of the image with a margin width \( \sqrt{1/2m} \).
Let $\Omega_{xy} = \{ p_{[x,y]} | [x,y] \in I \}$ be the patch set obtained from the image, where $I$ is the set of pixel indices. As mentioned above, $[x,y]$ is the centre of each patch in image. For an input patch $q_{[i,j]}$, existing methods [2, 5, 6] go through the entire set $\Omega_{xy}$ or its subset around $q_{[i,j]}$ to find its similar patches:

$$\min_{x,y \in I} \text{dist}(q_{[i,j]}, p_{[x,y]})$$

(3)

where, dist() is the measurement of distance, which is often Euclidean. The searching space is a two-dimensional matrix along rows and columns. In this paper, we bring rotation to this, for each centre $[x,y]$, we extend another dimension, the rotation angle $\theta$. Thus, we get a set of patches at each centre $P_{[x,y]} = \{ p_{[x,y,\theta]} | \theta \in \Theta \}$, where $\Theta$ is the set of angles. For simplicity, we evenly discretize $2\pi$ by $n$, namely $\Theta = \{ \frac{2\pi i}{n} \mid i = 1, 2, ..., n - 1 \}$. The whole patch set from the image will be $\Omega_{xy\theta} = \{ P_{[x,y]} | [x,y] \in I \} = \{ p_{[x,y,\theta]} | [x,y] \in I, \theta \in \Theta \}$. In the searching stage, we make sure that there’s only one candidate chosen from every pixel index. For each target patch $q_{[i,j]}$, we first traverse the $\theta$ dimension at every pixel index $[x,y]$ to find the best candidate, then go through set I to find its nearest neighbours (NNs):

$$\min_{x,y \in I} \min_{\theta \in \Theta} \text{dist}(q_{[i,j]}, p_{[x,y,\theta]})$$

(4)

To evaluate the performance of our transformed self-similarity, searching experiment is conducted on the UC Merced Land Use Dataset [14]. We select 5 categories from the dataset, in which, the first 20 images in each category are used, and patch size is set to $5 \times 5$. This experiment is performed as follows: Each image is transformed into the YCbCr space, and we only use the Y channel as the input image. Then, we get all patches to forming the 3-dimensional patch set $\Omega_{xy\theta}$, and each origin patch $p_{[i,j]}$ is compared to $\Omega_{xy\theta}$ use method as Equation (4). We find their NNs and calculate the mean distance of its k-NNs. Here, we have an adapted threshold $\Delta d_{[i,j]}$, patch $q_{[x,y]}$ is similar to $p_{[i,j]}$ when dist($p, q$) < $\Delta d$. This threshold is calculated by the following way: we interpolate the original image and shift by 1 pixel, then down-sample it, thus we get another image $l$ which has a subpixel shift to the original one. We can get the corresponding patch $\tilde{p}_{[i,j]}$, then we have:

$$\Delta d_{[i,j]} = \beta \cdot \text{dist}(p_{[i,j]}, \tilde{p}_{[i,j]})$$

(5)

The results are averaged across all images within each group and are shown in Figure 2. We divide $2\pi$ by $n = \{ 1, 2, 4, 8, 16, 32 \}$, namely the discretization step for $\theta$ is $\{ 2\pi \cdot \frac{n}{2}, \frac{n}{4}, \frac{n}{8}, \frac{n}{16} \}$. We can see that airplane, harbour and buildings tend to have much more similar patches, and this is intuitive as they are often textured. As we search more accurately in the $\theta$ dimension, similar patches increase and the matching errors reduce. This indicates that transformed self-similarity can really help us to find better NNs and thus improve the performance of our following kernel estimation and dictionary learning based SR method.
3. Blind SR using Transformed Self-Similarity

Most SR methods are to train a super resolver from the low-resolution image $l$ itself or some external training set [1-6], and most of their super resolver is under a given kernel $k$ and upscale $s$, denoted as $h = \mathcal{SR}_{k,s}(l)$. But as a blind SR method, here, the kernel denoted as $\hat{k}$ will be estimated from the input image $l$, then the super-resolver will take $\hat{k}$ as the kernel parameter. This kernel estimator and blind super resolver can be described as:

\[
\hat{k} = \mathcal{F}_s(l) \quad (6) \\
\mathbf{h} = \mathcal{SR}_{\hat{k},s}(l) \quad (7)
\]

3.1. Kernel Estimator

Our kernel estimator bases on the assumption that every patch from LR image $l$ satisfies Equation (1). We will show that, we can exploit the self-similarities to find such HR-LR pairs inside the input image, and thus to form an estimation with them. As shown in Figure 3(a), let $f(x)$ be a pattern in the continuous scene, $f(x)$ recurs elsewhere but $s$ times larger denoted as $f\left(\frac{x}{s}\right)$. They are captured with the same low-resolution imaging device and are presented as $q[n]$, $r[n]$ in the discrete LR image $l[n]$:

\[
q[n] = \int f(x)b_l(n-x)dx \quad (8) \\
r[n] = \int f\left(\frac{x}{s}\right)b_l(n-x)dx \quad (9)
\]

where $b_l(x)$ is the PSF of the imaging device. Suppose the unknown HR image $h$ is $s$ times larger than $l$, the high-resolution versions of these two patterns are $Q[n]$ and $R[n]$ respectively. So, we have:

\[
Q[n] = \int f(x)b_h\left(\frac{n}{s}-x\right)dx \quad (10) \\
R[n] = \int f\left(\frac{x}{s}\right)b_h\left(\frac{n}{s}-x\right)dx \quad (11)
\]

where $b_h(x)$ is the PSF of the high-resolution imaging device. $b_h(x)$ is an optical zoom of $b_l(x)$ [13], their relations can be represented as:

\[
b_h(x) = sb_l(sx) \quad (12)
\]

Substituting Equation (10), (11) to (8) and making a variable change, we have:
\[ r[n] = \int f(x) b_L(n - sx) \, dx = \int f(x) b_L \left( \frac{n}{s} - x \right) \, dx = Q[n] \quad (13) \]

Equation (13) indicates that patch \( r \) can be seen as the HR version of \( q \), thus we can get HR-LR pairs to form Equation (1) for our kernel estimation.

Now let’s make it a more common case, we bring the rotation transformation to this. Consider another pattern that is \( s \) times larger to \( f(x) \) but has a rotation angle \( \theta \), denoted as \( f_\theta \left( \frac{x}{s} \right) \), like in Figure 3(b). They are \( q \) and \( r_\theta \) in image \( l \). Here we can say \( r_\theta \) is similar to \( q \)’s HR version under our rotated definition. Suppose there’s a rotated kernel \( k_\theta \), if \( r_\theta \) convolutes with \( k_\theta \) and then downsampled by scale \( s \), we will get a patch \( r_\theta^s \) similar to \( q \). We put \( r_\theta^s \) into the transformation \( T_\theta^{-1}(\ast) \), then have:

\[ T_\theta^{-1}(r_\theta^s) \approx q \quad (14) \]

As shown in Figure 3(c), it is equivalent that we first rotate patch \( r_\theta \) and then degrade, which is exactly what we do during the patch set sampling step. We have,

\[ (T_\theta^{-1}(r_\theta) \ast k) \downarrow s = (r \ast k) \downarrow s \approx q \quad (15) \]

We rewrite Equation (14) as:

\[ q_{M \times 1} = R_{M \times N} k_{N \times 1} \quad (16) \]

Figure 3. Illustration of the kernel estimation using transformed self-similarity.

where, \( M = m \times m \) is the pixels of patch \( q \), \( N = n \times n \) is the variable numbers of the blur kernel, and \( q_{M \times 1} \) and \( k_{N \times 1} \) are the patch and kernel in vector form, while \( R_{M \times N} \) refers to the procedure of degradation with patch \( r \). Then the kernel estimation can be presented by an optimization:

\[ \min_k \| q - Rk \|_2^2 + \lambda_x \| L_x k \|_2^2 + \lambda_y \| L_y k \|_2^2 \quad (17) \]

where, \( L_x, L_y \) are the regular constraints at \( x \) and \( y \) directions for kernel \( k \), parameter \( \lambda_x \) and \( \lambda_y \) is used to balance the smooth constraint of the kernel estimation. This problem has a closed form solution:
\[ \hat{k} = (R^TR + \lambda_x L_x^T L_x + \lambda_y L_y^T L_y)^{-1} R^T q \] (18)

Now comes our estimator’s algorithm. We start with some initial kernel \( \hat{k} \), use this to degrade the input image \( I \) producing a degraded image \( c \), and randomly select \( n \) patches that are rich in texture as the target patches. In our following experiments, we set \( n \) to 100, and use the Sobel filter to get the patches that are in the edge areas, as edges are often rich in texture information. Denote the target patches as \{ \( q_1, q_2, ..., q_n \) \}. For each patch \( q_i \), we search its t-NNs in the degraded image \( c \) using the method of Equation (3) as \{ \( r_{1,i}, r_{2,i}, ..., r_{l,i} \) \}. Their corresponding patch in \( I \) are \{ \( r_{1,i}, r_{2,i}, ..., r_{l,i} \) \}, and as shown in Figure 3(b), this can be seen as the HR version of \( q_i \). We then use the non-local mean of \{ \( r_{1,i}, r_{2,i}, ..., r_{l,i} \) \} to give an estimation of \( q_i \)’s HR version \( \hat{r}_i \):

\[ \hat{r}_i = \sum_{j=1}^{l} \omega_{ij} r_{ij} \] (19)

where \( \omega_{ij} \) is the weights derived from the NNs:

\[ \omega_{ij} = \frac{\exp(-\|r_i - r_{ij}\|_2^2/\gamma)}{\sum_j \exp(-\|r_i - r_{ij}\|_2^2/\gamma)} \] (20)

Here \( \gamma \) is a constraint proportional to \( \sigma^2 \) of the patch. After this, we get a set of HR patches \{ \( r_1, r_2, ..., r_l \) \}. Put these LR-HR pairs to Equation (18), we get an estimation \( \hat{k} \). Notice that Equation (18) updates the kernel which we use to down-sample, resulting in a recursive solution. In each iteration, \( \hat{k} \) is closer to the ground-truth and then we will get more accurate similar examples to form the equations. We will do this until convergence or max iteration. We say it is converged when the RMSE of estimated \( k \) between two iterations less than a given threshold. Here, our kernel estimation algorithm can be summarized as follows:

**Algorithm 1 The Kernel Estimation Algorithm**

**Inputs**: Low-resolution image \( I \), upscale \( s \).

**Outputs**: Blur kernel \( \hat{k} \).

**Step1**: Initial \( \hat{k} \) with delta function. Randomly select textured low-resolution patches \{ \( q_1, q_2, ..., q_n \) \} to form set Q.

**Step2**: Sample \( I \) with rotation, form patch set \( L_{xy\theta} \). Degrade \( L_{xy\theta} \) with \( \hat{k} \) and get the corresponding low-resolution patch set \( C_{xy\theta} \).

**Step3**: For each patch in set Q, find their similar patches in \( C_{xy\theta} \) and corresponding HR versions in \( L_{xy\theta} \). Reconstruct the HR version patch set \( R \) by non-local mean.

**Step4**: Updates \( \hat{k} \) by constraint from Q and R.

**Step5**: Repeat Step2, Step3 and Step4 until convergence or max iteration.

### 3.2. Super Resolver

Our super resolver is also a dictionary learning based method. We will learn a sparse dictionary from the internal HR-LR pairs, and then the HR image \( h \) is reconstructed by the sparse coding of each HR patch. Given the sampled training patch pairs in matrix \( Q = [q_1, q_2, ..., q_n] \) and \( R = [r_1, r_2, ..., r_m] \), we jointly train the sparse dictionaries \( D_I \) and \( D_H \) for LR patch and HR patch, so that forcing the LR and HR representations to share the same coefficients. The objective functions are as follows:

\[
\min_{D_I, D_H, A} \frac{1}{M} \|Q - D_I A\|_F^2 + \frac{1}{N} \|R - D_H A\|_F^2 + \lambda (\frac{1}{M} + \frac{1}{N})\|A\|_1
\] (21)

Here, \( M \) and \( N \) are the pixels of LR and HR image patches in the column form, \( A \) is the sparsity coefficients. The \( l^1 \)-norm here, enforce the sparsity. Equation (21) can be rewritten as:

\[
\min_{D_I, D_H} \|X - DA\|_F^2 + \hat{\lambda}\|A\|_1
\] (22)
where, $X$ and $D$ are combinations of $Q, R$ and $D_l, D_h$ respectively as follows:

$$X = \begin{bmatrix} \frac{1}{\sqrt{M}}Q \\ \frac{1}{\sqrt{N}}R \end{bmatrix}, \quad D = \begin{bmatrix} \frac{1}{\sqrt{M}}D_l \\ \frac{1}{\sqrt{N}}D_h \end{bmatrix}$$

(23)

Different from the approach ScSR in [13], which use an external database and the bicubic kernel to train the sparse dictionary, our dictionary is learned using the internal database and the estimated kernel. Furthermore, we use the origin image patch for both LR-HR dictionary training instead of the derivative features in [13]. We use our estimated kernel $\tilde{k}$ to degrade the input image $I$, producing image $E$, then the HR-LR patch pairs are extracted using the same method for our transformed self-similarity as discussed in Section 2.

In the SR reconstruction stage, for each input low-resolution patch $q$, we find its sparse representation from $D_l$, let the sparse coefficients be $\alpha$, then the problem can be formulated as:

$$\min_{\alpha} \| D_l \alpha - q \|_2^2 + \lambda \| \alpha \|_0$$

(24)

Equation (24) is NP-hard, which is often solved by minimizing the $l^1$-norm as:

$$\min_{\alpha} \| D_l \alpha - q \|_2^2 + \lambda \| \alpha \|_1$$

(25)

Finally, the HR version $r$ of patch $q$ is given as:

$$r = D_h \alpha$$

(26)

Here, the dictionary training problem in Equation (22) and the sparse representation problem in Equation (25) are both solved using the K-SVD [15] and Orthogonal Matching Pursuit (OMP) [16].

4. Experimental Results and Discussions

In this section, the performance of our proposed blind super-resolution method is verified using simulated RS images. Firstly, the performance of our kernel estimator is evaluated. Here, we use an HR image and a given kernel as the ground-truth to degrade producing the input LR image, and the kernel type is set to Gaussian blur and its size is set to $9 \times 9$, LR patch size for kernel estimation is set to $5 \times 5$, and the scale for the SR is set to $s = 2$. To avoid the border effects, the HR patch is set to $17 \times 17$, exactly the right size for degradation. Figure 4(a) is the input low-resolution image with its ground-truth kernel on the bottom-left corner, the estimated kernel and SR result are in Figure 4(b). We can see that our kernel estimator does well with the common blur kernel, the SR result is also quite well.

![Figure 4](image_url)

**Figure 4.** Kernel estimation and blind super-resolution results
Our proposed blind SR method is evaluated on 10 remote sensing images in the USC-SIPI dataset. We compared our work with ScSR [3], which uses the default Bicubic kernel. We also use the ground-truth kernel to SR as the benchmark. The PSNR and averaged PSNR metrics are used to compare the performance. From the comparison results between different methods shown in Table 1, we can see that, our proposed blind SR method improves the PSNR by 1.45dB than ScSR on average, and is much closer to the SR result using ground-truth kernel.

Table 1. Comparison results between different methods in PSNR.

| Image No. | SR using truth kernel | ScSR (Bicubic) | Our method |
|-----------|-----------------------|----------------|------------|
| 2.1.01    | 72.71                 | 71.28          | 72.52      |
| 2.1.02    | 69.97                 | 68.72          | 69.84      |
| 2.1.03    | 76.85                 | 75.46          | 76.75      |
| 2.1.04    | 73.19                 | 71.66          | 73.06      |
| 2.1.05    | 73.14                 | 71.92          | 72.97      |
| 2.1.06    | 72.89                 | 71.47          | 72.68      |
| 2.1.07    | 76.06                 | 74.27          | 76.11      |
| 2.1.08    | 76.91                 | 74.86          | 76.76      |
| 2.1.09    | 77.87                 | 76.24          | 78.22      |
| 2.1.10    | 74.33                 | 72.80          | 74.30      |
| Averaged  | 74.40                 | 72.87          | 74.32      |

5. Conclusion
In this paper, we introduced the transformed self-similarity into the self-similarity based blind SR method. The blur kernel as well as the reconstructed high-resolution image can be estimated at the same time using the structural self-similarity from a single image, and the internal patch search space is expanded using the transformed self-similarity to improve both the quality of kernel estimation and SR reconstruction.

6. Acknowledgment
This work was supported in part by the National Natural Science Foundation of China (41971294).

7. References
[1] Pan Z, Yu J, Huang H, Hu S, Zhang A, Ma H and Sun W 2013 Super-resolution based on compressive sensing and structural self-similarity for remote sensing images, IEEE Trans. on Geoscience and Remote Sensing, vol 51, no 9, pp 4864-4876
[2] Chang H, Yeung D and Xiong Y 2004 Super-resolution through neighbor embedding, IEEE Conference on Computer Vision and Pattern Recognition, pp 275-282
[3] Yang J, Wright J, Huang T and Ma Y 2010 Image super-resolution via sparse representation, IEEE Trans. on Image Processing, vol 19, no 11, pp 2861-2873
[4] Huang J, Singh A and Ahuja N 2015 Single image super-resolution from transformed self-exemplars, IEEE Conference on Computer Vision and Pattern Recognition, pp 5197-5206
[5] Freeman W, Jones T and Pasztor E 2002 Example-based super-resolution IEEE Computer Graphics and Applications, vol 22, no 2, pp 56-65
[6] Glasner D, Bagon S and Irani M 2009 Super-resolution from a single image IEEE International Conference on Computer Vision, pp 349-356
[7] Michaeli T and Irani M 2013 Nonparametric blind super-resolution IEEE International Conference on Computer Vision, pp 945-952
[8] Efrat N, Glasner D, Apartsin A, Nadler B and Levin A 2013 Accurate blur models vs. image priors in single image super-resolution IEEE International Conference on Computer Vision pp 2832-2839
[9] He Y, Yap K, Chen L and Chau L 2006 Blind super-resolution image reconstruction using a maximum a posteriori estimation *International Conference on Image Processing*, pp 1729-1732

[10] He Y, Yap K, Chen L and Chau L 2009 A soft MAP framework for blind super-resolution image reconstruction *Image and Vision Computing*, vol 27, no 4, pp 364-373

[11] Gu J, Lu H, Zuo W and Dong C 2019 Blind super-resolution with iterative kernel correction *Computer Vision and Pattern Recognition*, pp 1604-1613

[12] Begin I and Ferrie F 2007 PSF recovery from examples for blind super-resolution *IEEE International Conference on Image Processing*, pp 421-424

[13] Zontak M and Irani M 2011 Internal statistics of a single natural image *IEEE Conference on Computer Vision and Pattern Recognition*, pp 977-984

[14] Yang Y and Newsam S 2010 Bag-of-visual-words and spatial extensions for land-use classification *International Conference on Advances in Geographic Information Systems* pp 270-379

[15] Aharon M, Elad M and Bruckstein A 2006 The k-svd: an algorithm for designing of overcomplete dictionaries for sparse representation *IEEE Trans. on Signal Processing*, vol 54, no 11, pp 4311-4322

[16] Elad M, Rubinstein R and Zibulevsky M 2008 Efficient implementation of the k-svd algorithm using batch orthogonal match pursuit *Cs Technion*, 40