Image Super-Resolution via Analysis Sparse Prior

Qiang Ning, Kan Chen, Li Yi, Chuchu Fan, Yao Lu and Jiangtao Wen, *Fellow, IEEE*

**Abstract**

In this paper, we present a new algorithm for a single image super-resolution using the analysis sparse prior in the \( la\beta \) color space. Experimental results show that our algorithm outperforms other existing state-of-the-art methods. In addition, due to the high scalability of our algorithm, key modules of the proposed algorithm can be integrated with other super resolution algorithms.
Image Super-Resolution via Analysis Sparse Prior

I. INTRODUCTION

T he goal of image super-resolution (SR) is to reconstruct a high-resolution (HR) image from one or more low-resolution (LR) images. Algorithms for image super resolution can generally be categorized into following families: Interpolation based [1], [2], Reconstruction based [3], [4], Self-based [5], [6] and Learning based methods [7]–[9]. Interpolation based methods are the simplest and fastest. Reconstruction based methods apply constraints to the HR images based on some prior such as image categorization. Self based methods utilizes redundancy across scales. Learning based methods make use of information from a library of images.

Based on the assumption that an image can be restored as a linear combination of atoms from an over-complete dictionary, sparse representation has been proposed for image SR [8], [9]. The sparsity model used in these papers is a generative model known as synthesis sparsity. Recently, a different sparsity model known as analysis sparsity has been proposed by Elad et al. [10].

According to [10], the synthesis model is useful for finding where a signal can lie, whereas the analysis model can be used to find where a signal cannot lie. When an over-complete dictionary with high coherence is used, it becomes difficult to determine where the signal may lie under the synthesis sparsity prior. On the contrary, finding where the signal can not be becomes relatively easier. The SR algorithm proposed in this paper was motivated by this intuition.

In the remainder of the paper, we will first introduce our analysis sparsity based super-resolution model in Section 2. Section 3 explains our algorithm in detail, while experiments and discussions are given in Section 4.

II. PROBLEM FORMULATION

A. Analysis Reconstruction with Synthesis-based Solvers

Under the traditional synthesis sparsity model, a signal $x \in \mathbb{R}^d$ is said to be $k$-sparse if it can be represented as a linear combination of atom signals from $D \in \mathbb{R}^{d \times K}$.

$$x = D\gamma, \quad s.t. \|\gamma\|_0 = k.$$  

(1)

On the other hand, a signal $x$ is said to be $l$-cosparse if it follows the analysis sparsity model and produces a sparse output when analyzed with an operator $\Omega \in \mathbb{R}^{K \times d}$.

$$\gamma_A = \Omega x, \quad s.t. \|\gamma_A\|_0 = K - l.$$  

(2)

Most recently, inherent connections between the two models has been revealed [10]. The equivalence of analysis exact reconstruction with an augmented synthesis problem was studied in [11]. As [11] pointed out, if $y = Mx$ has been observed, where $M \in \mathbb{R}^{m \times d}$ is the acquisition matrix, the solution of (3) is identical to the solution of an augmented synthesis recovery problem (4):

$$\hat{x} = \arg\min_x \|\Omega x\|_0 \quad \text{with} \quad y = Mx,$$  

(3)

$$\hat{x} = D \times \arg\min_{\gamma} \|\gamma\|_0 \quad \text{with} \quad \tilde{y} = \hat{A}\gamma,$$  

(4)

where $D = \Omega^\dagger$ is the pseudo-inverse of $\Omega$, $P_D$ is any projector onto the nullspace of $D$, $\tilde{y} = [y, 0]'$, $\hat{A} = [MD, P_D]'$.

Equations (3) and (4) show that the analysis sparsity model is a least-squares constrained synthesis sparsity model. Due to the robustness of synthesis solvers, when we refer to the analysis sparsity model in the rest of this paper, we always have the corresponding enhanced synthesis model in mind. We still use the traditional synthesis dictionary training model proposed by Yang et al. in [9] for our experiments.

B. Analysis-Based Super-Resolution

We use $X$ and $Y$ to denote the HR and LR images, and $x$ and $y$ the corresponding image patches. $\Omega_D$ is the trained analysis operator for image $I$ and $D_I = \Omega_I'$ is the pseudo-inverse of $\Omega_I$. $S$ and $U$ denote a downsampling operator and a base upscaling method (for example, bicubic), respectively. $H$ is the blurring filter. Another important operator, $\Delta$, is defined as the difference between an identity matrix $I$ and $USH$.

Using the operators defined above, we have:

$$Y = SHX + n_0,$$  

(5)

$$X = UY + \Delta X,$$  

(6)

where $n_0$ represents gaussian noise. Note that we ignore the noise term in (6), and use back projection (BP) [12] at the end of our algorithm to denoise the result images.

The inverse problem then becomes a prediction of $\Delta X$, instead of $X$ by making use of the base method followed by deficiency compensation. This inverse problem becomes less ill-posed after we do this replacement, too. To regularize this inverse process, we solve the following sparse representation problem,

$$\min_{\gamma} \|\gamma\|_0, \quad s.t. \quad \|\tilde{y} - \tilde{D}\gamma\|_2^2 < \varepsilon,$$  

(7)

where $\tilde{y} = [y, 0]'$, $\tilde{D} = [FDy, P_D]'$ and the $F$ operator is a feature extraction operator used to lay constraints on the closeness of the approximation of $\gamma$ for $y$ that will be discussed in more detail later.

Similar to synthesis based SR proposed by [9], the HR and LR images are “connected” via a vector $\gamma$:

$$\tilde{\Delta}x = D\Delta X\cdot\gamma,$$  

(8)
and therefore we can recover $X$ by putting all the reconstructed $\Delta x$ into $\Delta X$ and calculate (6).

In practice, we use the $l_0$ norm as opposed to the $l_1$ norm to convert (9) to a convex optimization problem, which is easier to solve and can still guarantee some sparsity.

$$\min_{\gamma} \| \gamma \|_1, \quad \text{s.t.} \quad \| \hat{y} - D\gamma \|_2^2 < \epsilon.$$  

(9)

C. Iterative Refinements

From (6), we can see that our model can be generalized as an improvement to other existing methods. In our model, $\Delta$ can be regarded as a deficiency learning operator, as it represents information that cannot be recovered from $U$. Although there have been many studies on high frequency detail enhancement since, for example, Freeman [7], the deficiency learning perspective has not received sufficient attention until recently [13].

Since an upscaling method $U$ can be refined using (6), if we denote this scheme by $M_U$, we can repeat the refinement using (6) for $M_I$ iteratively until and if the iterations converge. According to (6), computational complexity of $M$ is:

$$C(M_U) = C(U) + C(\Delta),$$  

(10)

where $C(\Delta)$ represents the complexity of (7), and is constant for a given $\epsilon$. Then,

$$C(M^n_U) = C(U) + nC(\Delta)$$  

(11)

for the n-th iteration ($M^n_U = U$). It is easy to see that the complexity of this iteration only grows linearly with respect to $n$.

III. ALGORITHM DESCRIPTION

A. $l\alpha\beta$ Color space

To up-scale color images, people usually conduct the SR only in one of the color channels, usually the luminance. The other two channels then undergo simple interpolation-based SR in order to reduce the computational complexity. When correlations exist between different color components, using different algorithms for different components may lead to noticeable degradation to color consistency after up-scaling.

In [14], Ruderman et al. developed the $l\alpha\beta$ color space, where the 3 principal components were produced using an orthogonal decorrelation. This allows for applying different operations in the 3 decorrelated channels without undesirable cross-channel incompatibilities. Experiments show that using the $l\alpha\beta$ space can lead to a significant and robust improvement in PSNR.

Usually, there are three steps when converting an image in the RGB space to the $l\alpha\beta$ space, namely RGB to XYZ tristimulus conversion, XYZ to LMS conversion [15], and LMS to $l\alpha\beta$ conversion [14]. In our algorithm, in order not to lose high frequency information, we use the following simplified conversion:

$$\begin{bmatrix}
1 \\
\alpha \\
\beta
\end{bmatrix} =
\begin{bmatrix}
0.3475 & 0.6823 & 0.5559 \\
0.2162 & 0.4336 & -0.6411 \\
0.1304 & -0.1033 & -0.0268
\end{bmatrix}
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}.$$  

(12)

As the conversion matrix above is invertible, we can easily convert the enhanced image in the $l\alpha\beta$ space back to the RGB space.

B. Variance Prediction for $\Delta x$

In practice, (8) should turn into:

$$\Delta x_N = D\Delta x \cdot \gamma.$$  

(13)

The subscript $N$ appears because the scale of $\Delta x$ has been altered several times during the normalization both in the dictionary training step and reconstruction step. As a result, it is necessary to predict the scale of $\Delta x$ before obtaining the final result.

From (5) and (6) and ignoring noise, $SH\Delta X = Y - SHUY$.

Based on both intuition and experimental results (Fig. 1), we found that for a signal $\omega$, $Var(SH\omega)$ has a strong linear correlation with $Var(\omega)$, so that $\Delta x$ can be predicted as the following:

$$\Delta x = \frac{\beta\Delta x_N \sigma(Y - SHUY)}{\sigma(\Delta x_N)},$$  

(15)

where $\sigma$ represents the standard deviation and $\sigma(\omega) = \sqrt{Var(\omega)}$.

![Fig. 1: The Variance of $\Delta x$ is linearly depended on $SH\Delta x$.](image)

C. Neighbor Compatibility

In [9], Yang et al. enhanced neighbor compatibility by incorporating information from previously reconstructed high-resolution patches to the patch that is currently being processed. This method suffers from several drawbacks. Firstly, the continuous stitching means that patches have to be processed one by one, making parallel computing impossible. Secondly, a previously reconstructed patch may introduce a strong enough component that “dwarfs” information obtained from other sources. In our algorithm, similar to [9], [16], we also used four 1-D filters to extract first- and second-order derivatives:

$$f_1 = [-1, 0, 1], f_2 = f_1, f_3 = [1, 0, -2, 0, 1], f_4 = f_3.$$  

(16)
The difference is that we also apply the above filters to lower scales, thereby incorporating multi-scale feature extraction and mitigating deficiencies introduced by single-scale features. In practice, however, too many scales may degrade too much of the importance of original scale. So we choose two scales in experiments and also apply coefficient weighing the second scale features, so the final form of our feature would be following.

\[ F_y = \left[ \frac{J_y}{\mu f_{sp}} \right] \]  

where \( f \) and \( F \) is the single- and multi-scale features extraction operator respectively, and \( \mu \) is the weight of second-scale features.

Our algorithm can be explained using the pseudo code as following:

**Require:** Input color image \( Y \), dictionary \( D_Y \) and \( D_{\Delta X} \) using base method \( U \)

**Ensure:** High resolution reconstruction \( \hat{X} \)

1. Convert \( Y \) from \( RGB \) to \( la_{\beta} \) using (12)
2. Find the correct scale of \( \Delta X \) using (14). Denote the result as \( P_s \).
3. For each \( 4x4 \) patch \( y \), taken with overlap in each direction do
4. Solve the minimization problem in (9) and find the sparse representation \( \gamma \)
5. Extract patch \( p_s \) from \( P_s \) corresponding to the location of \( y \) and calculate its variance
6. Referring to (13)(15), recover the rescaled \( \Delta x \)
7. Add \( \Delta x \) to \( \Delta \hat{X} \), averaging overlapped area
8. End for
9. Find \( \hat{X} \) using (6)
10. Denoising using BP

**IV. Experiments**

In our study, we first investigated the benefit introduced by each of the steps in our algorithm. In the experiments, the value of \( \beta \) in (15) was set to 1.

In Table I, the average gain provided by every step of our modifications is presented. Step 1 is the technique of [9] (but without BP denoising), step 2 changes target from \( X \) to \( \Delta X \), step 3 converts the color space to \( la_{\beta} \) and finally step 4 replaces the synthesis model by the analysis model. Finally BP is applied to denoise the output from the previous steps.

In Table II we compare our results with Bicubic, SAI proposed by Zhang and Wu [2], self-based method Glasner by [5], and Yang et al. [9].

Furthermore, in Table III we show the results when the iterative refinement is applied to the Bicubic (B), SAI and \( M_B \) algorithms, without the BP denoising (therefore the results in this table were slightly different from Table II for some algorithms because BP was skipped).

**V. Discussion**

Preliminary experimental results reported in the previous section show that the proposed algorithm significantly out-performs other algorithms.

<table>
<thead>
<tr>
<th>Image</th>
<th>Bicubic</th>
<th>SAI</th>
<th>Glasner</th>
<th>Yang</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brickhouse</td>
<td>36.04</td>
<td>33.84</td>
<td>38.73</td>
<td>37.86</td>
<td>39.16</td>
</tr>
<tr>
<td>Child</td>
<td>44.08</td>
<td>39.27</td>
<td>45.26</td>
<td>42.99</td>
<td>48.14</td>
</tr>
<tr>
<td>Lena</td>
<td>42.67</td>
<td>38.67</td>
<td>43.06</td>
<td>42.35</td>
<td>46.41</td>
</tr>
<tr>
<td>Mandrill</td>
<td>35.08</td>
<td>33.56</td>
<td>37.31</td>
<td>37.05</td>
<td>37.45</td>
</tr>
<tr>
<td>Statue</td>
<td>38.77</td>
<td>35.17</td>
<td>40.07</td>
<td>39.68</td>
<td>43.43</td>
</tr>
<tr>
<td>Sparrow</td>
<td>41.41</td>
<td>38.84</td>
<td>43.23</td>
<td>42.47</td>
<td>44.95</td>
</tr>
<tr>
<td>Zebra</td>
<td>35.23</td>
<td>30.40</td>
<td>35.85</td>
<td>36.66</td>
<td>40.10</td>
</tr>
<tr>
<td>Monarch</td>
<td>34.48</td>
<td>31.99</td>
<td>36.78</td>
<td>36.97</td>
<td>37.56</td>
</tr>
<tr>
<td><strong>Gain</strong></td>
<td>-3.25</td>
<td>1.57</td>
<td>1.65</td>
<td>3.68</td>
<td></td>
</tr>
</tbody>
</table>

Due to the complexity involved in solving the sparse representation, our current implementation of the proposed algorithm is not yet current time. However, many optimizations of the algorithm exist, and, as discussed earlier, the proposed scheme is more parallel-processing friendly than many other state-of-the-art algorithms.

Some limitations exist for the proposed algorithm. First-ly, the performance scale prediction used in the proposed algorithm degrades when back projection is used within base method \( U \). According to (14) and (15), our prediction does not work if \( Y - SH U Y = 0 \), which is exactly what back projection achieves. As a result, we remove the BP from the up-scaling method \( U \) before iterative refinement. This limitation may be eliminated if a scale prediction approach without calculating (15) is used.

In addition, theoretically, the proposed iterative refinement will work if the deficiency “image” of a up-scale method \( U \) could be sparsely represented. This is the case

<table>
<thead>
<tr>
<th>Image</th>
<th>Bicubic</th>
<th>SAI</th>
<th>Glasner</th>
<th>Yang</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brickhouse</td>
<td>36.04/37.51</td>
<td>31.20/35.38</td>
<td>37.51/35.78</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Child</td>
<td>44.08/46.41</td>
<td>36.26/43.20</td>
<td>46.41/46.70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lena</td>
<td>42.67/45.02</td>
<td>35.12/42.36</td>
<td>45.02/45.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mandrill</td>
<td>35.08/36.29</td>
<td>31.08/34.50</td>
<td>36.29/36.52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Statue</td>
<td>38.77/41.18</td>
<td>31.65/38.41</td>
<td>41.18/41.88</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sparrow</td>
<td>41.41/43.50</td>
<td>35.90/41.31</td>
<td>43.50/43.92</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zebra</td>
<td>35.23/37.63</td>
<td>26.52/34.68</td>
<td>37.63/38.46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monarch</td>
<td>34.48/36.12</td>
<td>30.22/34.42</td>
<td>36.12/36.53</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Gain</strong></td>
<td>1.98</td>
<td>5.90</td>
<td>0.45</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
for linear up-scale operators of $U$, but may not be true universally.

Finally, it is theoretically unclear if the proposed iterative refinement converges and under what conditions.

VI. Conclusion

In this paper, we introduce a novel algorithm for color image super-resolution using iterative refinement and the analysis sparsity model (or enhanced synthesis sparsity model). Preliminary experiments with a relatively large number of images show significant improvement in SR performance as compared with other techniques.

REFERENCES


