Unaligned Hyperspectral Image Fusion via Registration and Interpolation Modeling

Jiacheng Ying, Hui-Liang Shen, and Si-Yuan Cao

Abstract—In satellite remote sensing, the hyperspectral sensor acquires high-spectral-resolution and low-spatial-resolution hyperspectral images (HSIs). Conversely, the multispectral sensor acquires low-spectral-resolution and high-spatial-resolution multispectral images (MSIs). Thus, HSI and MSI fusion is required to promote both spatial and spectral resolutions. Currently, most algorithms are based on the assumption that the HSI and MSI are perfectly aligned. However, this is hardly achievable in real scenarios when the two sensors acquire images from different viewpoints. In this article, we propose a fusion algorithm that consists of two stages, i.e., image registration and image fusion. For image registration, we introduce the normalized edge difference (NED) for image similarity measure considering the different resolutions of the original images. For image fusion, we incorporate the interpolation process in the spatial degradation model to compensate for the interpolation error. Experimental results show that our algorithm performs better than the state of the arts for unaligned image fusion.

Index Terms—Hyperspectral image (HSI), image fusion, image interpolation, image registration, image super-resolution, multispectral image (MSI).

I. INTRODUCTION

HYPERSPECTRAL imaging can record rich spectral information and has been applied in many fields, such as material classification [1], face recognition [2], and geological exploration [3], [4]. However, due to the limitation of physical devices, images captured by hyperspectral sensors are usually of low spatial resolution. Conversely, multispectral sensors can capture high spatial resolution but low-spectral-resolution images. Hence, many image fusion algorithms [5]–[7] have been introduced to improve both the spatial and spectral resolutions. The aim of this work is to reconstruct a high-resolution hyperspectral image (HR-HSI) by fusing the low-resolution HSI (LR-HSI) and high-resolution multispectral image (HR-MSI).

Most of the current algorithms are implicitly based on the hypothesis that the HSI and the MSI are perfectly aligned. However, this situation can be hardly met if without special optical design. On the remote sensing satellite, two types of sensors acquire image pairs from different viewpoints, which will eventually cause image misalignment.

In this work, we propose an HSI super-resolution algorithm based on image registration and interpolation modeling. We assume that the spatial relationship between LR-HSI and HR-MSI can be depicted by affine transform. This assumption is reasonable since the satellite is far away from the ground scene to be imaged. Our algorithm contains a registration stage and a fusion stage. In the registration stage, we first apply spectral mapping and spatial upsampling such that the LR-HSI is of the same size as HR-MSI. With this treatment, the original registration problem is transformed to an MSI clear–blur pair registration problem. Generally, the edge difference (ED) between two aligned images is smaller than that between two unaligned images [8]. Hence, we define the normalized ED (NDE) to measure the misregistration error of two images. We also build edge image pyramids to perform multiscale registration. In the fusion stage, we apply a special strategy to utilize the affine parameters to construct a spatial degradation model from HR-HSI to LR-HSI. We also construct the spectral degradation model using the spectral mapping function of the image sensors. We decompose the HR-HSI to the spectral basis matrix and corresponding coefficients’ matrix. Our method is based on the prior that each pixel can be represented linearly using a small number of spectral bases. We obtain the spectral basis matrix by nonnegative matrix factorization (NMF) of LR-MSI and compute the corresponding coefficients’ matrix by solving a Sylvester equation [9] with L2-norm regularization.

We notice that Nie et al. [10] and Zhou et al. [11] also deal with the HSI super-resolution problem in the case of image misalignment. These works first estimate the transform parameters and then directly fuse the registered images. We note that, due to the transform bias and interpolation error, direct fusion is optimal for HSI super-resolution. Instead, we incorporate the interpolation process of the affine transform into the spatial degradation model in the image fusion stage. As can be seen in the experiment, this treatment can considerably improve the accuracy of image fusion.

In summary, the main contributions of this work are given as follows.

1) We propose a two-stage HS image super-resolution algorithm. Compared with the state of the arts, our algorithm takes into consideration that the remote sensing HSI and
MSI are unaligned, which is a more general situation in real-world applications.

2) For image registration, we introduce a new registration objective function called NED that is effective for clear–blur image similarity measurement.

3) For image fusion, we incorporate the interpolation process into the spatial degradation model. Compared with the regular treatment in previous works, our treatment effectively reduces the influence of transform bias and interpolation error in HR-HSI reconstruction.

II. RELATED WORK

In the following, we briefly review the work on image registration and fusion that closely relates to HSI super-resolution.

A. Image Registration

The objective of image registration is to find the best spatial transform between two images. Approximately, the registration algorithms can be classified into two categories: feature-based ones and intensity-based ones.

Feature-based algorithms detect feature points in images and perform image alignment by feature points matching. Scale-invariant feature transform (SIFT) [12] is a very common feature descriptor and has been widely used for image registration. It has good stability and is not sensitive to scaling, rotation, and luminance change. The speeded-up robust feature (SURF) descriptor [13] is a 64-D feature vector that is half of SIFT and can accelerate SIFT. It is good at dealing with blurred and rotated images. The Harris corner detector [14] extracts the intersection points between contours as features and is robust to viewpoint change. Brown et al. [15] extract SIFT feature points in images and use K-D tree and random sample consensus (RANSAC) [16] for image matching. Fan et al. [17] employ the Harris corner filter for pixel classification, which simultaneously improves the robustness of multimodal image registration and reduces computation complexity.

Intensity-based algorithms consider every pixel and introduce some measures that represent the image consistency for image alignment. Maes et al. [18] register images by employing the mutual information, which reaches the maximum when two images are perfectly aligned. Irani and Anandan [19] introduce the local normalized-correlation similarity, which measures the consistency of two images under different luminance. Myronenko and Song [20] align images by minimizing the residual complexity of two images, which can be represented by the sparseness of discrete cosine transform (DCT) coefficients. Chen et al. [8] introduce normalized total gradient (NTG) to represent the sparseness of gradient of the difference image, by minimizing which multispectral image alignment is achieved. Reddy and Chatterji [21] present phase correlation (PC) technique to cover translation, rotation, and scaling. Bulent et al. [22] apply diffeomorphic registration by optimizing a similarity measure for fine-tuned alignment after using SIFT and SURF for initial image alignment.

B. Image Fusion

Most HSI fusion algorithms are based on the assumption that two images are aligned. Guided by a secondary HR-MSI, the spatial resolution of the LR-HSI is improved. The typical algorithms in image fusion include matrix decomposition, tensor factorization, spectrum mapping, and deep learning.

Matrix decomposition algorithms are based on the assumption that the spectrum vector of each pixel can be represented as a linear combination of several spectral bases. The SRIF algorithm [23] employs principal component analysis in extracting spectral dictionary and incorporates gradient regularization in estimating the corresponding coefficients. Simoes et al. [5] extract the subspace of HSI as the spectral basis and employ the alternating direction method of multipliers (ADMM) to solve an optimization problem, with vector total variation of coefficients gradient as the regularizer. Yokoya et al. [24] use NMF to unmix both the HSI and MSI into endmember and abundance matrices and iteratively optimize one matrix with the other one fixed.

Tensor factorization algorithms decompose the HSI into a core tensor and three corresponding dictionaries. Li et al. [6] employ a regularizer to model the correlation between space and spectrum domains that improve the sparseness of the core tensor. Dian et al. [25] group the similar cubes together for the learning of dictionaries in three modes and use sparse coding to estimate core tensor. Ding et al. [52] deploy a block-term factorization model with multirank terms, which allows prior information incorporated for performance improvement.

Spectrum mapping algorithms map spectrum vectors of each pixel from low dimension to high dimension. Arad and Ben-Shahar [26] generate a large number of RGB atoms and hyperspectral atoms to construct the RGB image and HSI, respectively, with the weights of atoms shared. Similarly, Aeschbacher et al. [27] train the nearest neighbor relations between image pixels for HSI reconstruction.

Deep learning algorithms achieve super-resolution by constructing neural networks to train the mapping function from LR images to HR images in a supervised manner. Wang et al. [28] train a multitunit fusion network and iteratively plus the residual output of former units together as the final output, to increase the accuracy of the constructed images. Zhang et al. [29] build a two-stage SR network. The first stage uses a supervised fusion module to obtain a rough result, and the second stage deploys an unsupervised adaption module to further improve the image quality. Dian et al. [30] build a CNN network to train the relation between roughly constructed HR-HSIs and the residual of HR-HSIs, which can improve the accuracy of image reconstruction.

III. PROBLEM FORMULATION

We assume that the HR-MSI, denoted as \( \mathbf{X} \in \mathbb{R}^{M \times N \times L} \) includes \( M \times N \) pixels in \( L \) bands. The LR-HSI \( \mathbf{Y} \in \mathbb{R}^{m \times n \times L} \) includes \( m \times n \) pixels in \( L \) bands, with \( L \gg l \). The scale factor is denoted as \( b \), relating the spatial dimensions by \( M = b \times m \) and \( N = b \times n \). Our objective is to estimate the HR-HSI \( \mathbf{Z} \in \mathbb{R}^{M \times N \times L} \) from \( \mathbf{X} \) and \( \mathbf{Y} \).
The matrix forms of $X$, $Y$, and $Z$ are $\hat{X} \in \mathbb{R}^{L \times MN}$, $\hat{Y} \in \mathbb{R}^{L \times mn}$, and $\hat{Z} \in \mathbb{R}^{L \times MN}$. Then, $\hat{X}$ and $\hat{Y}$ can be linearly modeled with $\hat{Z}$ as

$$\hat{X} = R\hat{Z} \quad (1)$$

$$\hat{Y} = ZBS \quad (2)$$

where $R \in \mathbb{R}^{L \times L}$ denotes the spectral degradation function mapping the hyperspectral pixels to the multispectral ones; $B \in \mathbb{R}^{MN \times MN}$ represents the deblurring matrix obtained from the point spread function (PSF) of the hyperspectral sensors; and $S \in \mathbb{R}^{MN \times mn}$ is the spatial mapping matrix that maps the HR-HSI to the LR-HSI in the spatial domain. We note that $R$ and PSF are the parameters of the hyperspectral sensor and are supposed to be known.

As the spatial relation from $Z$ to $Y$ is identical to that from $X$ to $Y$, we compute $S$ by aligning $X$ and $Y$ in the registration stage of our algorithm. We first apply spectral mapping and spatial upsampling to $Y$ to generate a blurred HR-MSI $\tilde{X}$, which is also of size $M \times N \times l$. Then, the different scales’ registration problem is transferred to a clear–blur image pair registration problem. Let $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ denote the transform from the clear MSI to the blurred MSI, and

$$T(x) = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix} \Lambda$$

(3)

where $x = (x, y)^T$ is the original pixel position, $T(x) = (u, v)^T$ is the transformed pixel position, and $\Lambda = (a_1, a_2, \ldots, a_b)^T \in \mathbb{R}^{6 \times 1}$ is the affine parameter.

To be more specific, Fig. 1 illustrates the flowchart of our algorithm, which contains two stages. In the registration stage, the input LR-HSI is first converted to a blurred HR-MSI. Next, the clear HR-MSI and the blurred HR-MSI, which are unaligned, are used to construct the edge pyramids. Then, the affine parameter is optimized and transferred from the bottom layer to the top layer. In the fusion stage, the spectral basis matrix is solved by applying NMF, and the coefficient matrix is solved using Algorithm 1. Finally, the HR-HSI is reconstructed by multiplying two matrices.

IV. STAGE I: IMAGE REGISTRATION

The aim of image registration in our algorithm is to find the best affine transformation from HR-MSI $\tilde{X}$ to LR-HSI $\tilde{Y}$. More details will be elaborated on as follows.

A. Image Preprocessing

As the size of HR-MSI $\tilde{X}$ is inconsistent with that of LR-HSI $\tilde{Y}$ in the spectral and spatial domains, we apply spectral mapping and spatial upsampling to the LR-HSI $\tilde{Y}$ to generate a blurred HR-MSI $\tilde{Y} \in \mathbb{R}^{M \times N \times l}$ as

$$\tilde{Y} = [R \circ \tilde{Y}]_{[b]}$$

(4)

where $\circ$ denotes the spectral mapping operation, $[\cdot]_b$ the upsampling operation (here, we use the bicubic manner), and $b$ the scale factor. Here, we do not choose to downsample the HR-MSI $\tilde{X}$ to the size of $m \times n \times l$ because the registration error in the $m \times n$ sized pictures is $b$ times as large as that in the $M \times N$ sized pictures. Hence, the multiscale registration problem is transferred to a clear–blur image pair ($X$ and $\tilde{Y}$) registration problem.

B. Edge Difference

The image edge can well reflect the structure information of an image. A clear–blur image pair, whose pixels’ intensities are different, have similar edge structures. Generally, when two images are perfectly aligned, the ED reduces to the minimum.
Therefore, we define the ED of clear–blur image pair $I_c$ and $I_b$ as

$$ED(I_c, I_b) = \sum_j \|E_c - E_b\|_1$$  \hspace{1cm} (5)

where $\|\cdot\|_1$ denotes the L1-norm, $E_c$ and $E_b$ are the edges of $I_c$ and $I_b$, respectively, and $j$ denotes the image band. Here, we use the L1-norm to compute ED instead of L2-norm because L1-norm can promote the sparseness of ED and is robust to noise. The edge $E$ of the image $I$ is computed using

$$E(x, y, j) = \sqrt{\partial_x^2 I(x, y, j) + \partial_y^2 I(x, y, j)}$$  \hspace{1cm} (6)

where $E(x, y, j)$ is the edge intensity at pixel position $(x, y)$ at the band $j$, and $\partial_x I(x, y, j)$ and $\partial_y I(x, y, j)$ are the image gradients computed using the spatial convolution kernels $[-1, 0, 1]$ and $[-1, 0, 1]^T$, respectively.

Fig. 2 illustrates the edge information in a clear–blur image pair. It is observed that the clear image edge is sharper than the blurred image edge. Fig. 2(g) presents the cross-sectional distributions of the two kinds of edges. It is obvious that the distribution of clear image edge is of a higher crest and narrower width.

For demonstration, we use two 1-D Gaussian models with different variances to simulate the registration of clear–blur edge pair in Fig. 3. We compute the ED when the narrow Gaussian model moves from displacement $-7$ to $7$. It is observed that ED reaches the minimum when there is no displacement.

**C. Normalized Edge Difference**

To limit the range of the measure ED to $[0, 1]$, we define NED of the clear–blur image pair $I_c$ and $I_b$ as

$$NED(I_c, I_b) = \frac{\sum_j \|E_c - E_b\|_1}{\sum_j (\|E_c\|_1 + \|E_b\|_1)}$$  \hspace{1cm} (7)

where the denominator is the absolute sum of two image edges, and the numerator is the absolute sum of ED. If we treat the L1-norm as an energy operator, NED can be regarded as the relative energy of ED.

We note that NTG [8] has a similar form to our NED measure. We compare the two measures by imposing displacements on the clear images of six clear–blur pairs in both horizontal and vertical directions. Fig. 4 presents the NTG and NED values with respect to the displacement. The NTG measure is more likely to fall into the local optimum when the displacements are larger than 10 pixels. In comparison, NED has a wider capture range that is beneficial to find the real solution.

Consequently, the affine parameter $A$ of the clear–blur image pair registration problem can be solved by minimizing NED as

$$A = \arg \min_A J(A)$$  \hspace{1cm} (8)

where

$$J(A) = NDE(I_c(T(x, A)), I_b(x)).$$  \hspace{1cm} (9)

To compute $A$, we initialize it as $A_0 = (1, 0, 0, 0, 1, 0)^T$ and update it by gradient descent

$$A^{t+1} = A^t - \gamma \nabla_A J$$  \hspace{1cm} (10)

where the step size $\gamma = 0.5/(\max[M, N])$. $\nabla_A J$ denotes the current gradient value. More details about the derivation of $\nabla_A J$ can be found in the Appendix.

**D. Edge-Image Pyramids**

Only using the gradient descent method to minimize NED may cause the affine parameter to fall into the local optimum.
To solve this problem, we perform multiscale registration by building edge-image pyramids, whose lower layers are downsampled from the upper ones. We fix the downsampling factor as \( \alpha = 1.5 \) and require the short side of the bottom layer larger than 16 pixels. Starting from the bottom layer, we first initialize the affine parameter, update it using (10), and transfer it to the upper layer.

Let \( \mathbf{A}^t = (a_{1}^t, a_{2}^t, \ldots, a_{B}^t)^T \) denote the affine parameter at the \( t \)-th layer; we have

\[
\begin{bmatrix}
  u \\
  v
\end{bmatrix} = \begin{bmatrix}
  a_{1}^t x + a_{2}^t y + a_{3}^t \\
  a_{4}^t x + a_{5}^t y + a_{6}^t
\end{bmatrix}.
\tag{11}
\]

By transferring the affine parameter from the \( t \)-th layer to the \( (t+1) \)-th layer with \( \alpha \) times size enlargement, we have

\[
\begin{bmatrix}
  u \\
  v
\end{bmatrix} \rightarrow \begin{bmatrix}
  \alpha u \\
  \alpha v
\end{bmatrix}.
\tag{12}
\]

Hence, we need to update \( a_{1}^t \) and \( a_{6}^t \) to \( a_{1}^{t+1} \) and \( a_{6}^{t+1} \), respectively, for parameter transfer. Then, we use (10) to further optimize \( \mathbf{A}^{t+1} \) in the \( (t+1) \)-th layer. In this way, the affine parameter is solved in a coarse-to-fine manner.

We note that the employment of edge-image pyramids is essential to the image registration process. The parameter transfer between adjacent layers makes the algorithm more accurate and robust and also accelerates its convergence.

V. STAGE II: IMAGE FUSION

Based on the spectrum unmixing principle, each pixel spectrum \( \tilde{z}_i \in \mathbb{R}^L \) in the HR-HSI \( \tilde{X} \) can be represented as the linear combination of a few spectral bases as

\[
\tilde{z}_i = \mathbf{V} \mathbf{e}_i \tag{13}
\]

where \( \mathbf{V} = (v_1, v_2, \ldots, v_k) \) denotes the spectral basis matrix, in which each column is a spectral basis. \( \mathbf{e}_i \in \mathbb{R}^k \) is the coefficient vector. Combining all the pixels, we have

\[
\tilde{X} = \mathbf{V} \mathbf{E}
\tag{14}
\]

where \( \mathbf{E} = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{MN}) \) denotes the corresponding coefficient matrix. With (14), we rewrite (1) and (2) as

\[
\mathbf{X} = \mathbf{RVE}
\tag{15}
\]

and

\[
\hat{\mathbf{Y}} = \mathbf{VEB}
\tag{16}
\]

As mentioned in Section III, \( \mathbf{R} \) and \( \mathbf{B} \) are determined by the hyperspectral sensor and are, thus, known. \( \mathbf{S} \) is the spatial mapping matrix.

Most of the current algorithms treat the registration and fusion as two independent parts. These algorithms first warp LR-HSI (or warp HR-MSI) to register misaligned images and then fuse the two images. With this treatment, \( \mathbf{S} \) is constructed as a downsampling matrix, as shown in Fig. 5(a). Different from the previous work, we employ an interpolation modeling treatment to construct the matrix \( \mathbf{S} \) using the affine parameter obtained from Section IV. More specifically, a pixel \((x, y)\) in \( \hat{\mathbf{Y}} \) is mapped from subpixels in \( \mathbf{Z} \). We apply interpolation in a bilinear manner (the reason will be stated in Section V-C) by filling different interpolation weights \( \{w_{00}, w_{01}, w_{10}, w_{11}\} \) of nearest pixels \( \{(u_0, v_0), (u_1, v_0), (u_0, v_1), (u_1, v_1)\} \) into the corresponding positions when constructing \( \mathbf{S} \), as illustrated in Fig. 5(b). With our modeling treatment, LR-HSI does not need to be warped, and thus, the negative influence of interpolation error and registration bias to the reconstructing HSIs can be reduced.

\[\text{Equation (16) can be rewritten as}\]

\[
\hat{\mathbf{Y}} = \mathbf{VEB}.
\tag{18}\]

which demonstrates that LR-HSI \( \hat{\mathbf{Y}} \) can also be represented by spectral bases. Since \( \mathbf{E}, \mathbf{B}, \) and \( \mathbf{S} \) are all nonnegative and sparse, \( \mathbf{E}_B \) is also a nonnegative and sparse matrix. We employ NMF to estimate the spectral basis matrix, formulated as

\[
\mathbf{V} = \arg \min_{\mathbf{V}} \| \mathbf{E}_B \|_1
\]

s.t. \( \| \mathbf{Y} - \mathbf{V} \mathbf{E}_B \|_F^2 \leq \epsilon, \ \mathbf{V} \succeq 0, \ \mathbf{E}_B \succeq 0 \)

where \( \| \cdot \|_F \) denotes Frobenius-norm, and \( \epsilon \) is the tolerance to noise. Here, we treat matrix \( \mathbf{E}_B \) as a vector in the computation of L1-norm. We solve this problem using the SPArse Modeling Software (SPAMS) [32].

B. Solving the Coefficient Matrix

With the estimated spectral basis matrix \( \mathbf{V} \), we further solve the coefficient matrix \( \mathbf{E} \) for reconstructing \( \mathbf{Z} \). We formulate the problem as

\[
\mathbf{E} = \arg \min_{\mathbf{E}} \Psi(\mathbf{E})
\]

\[= \arg \min_{\mathbf{E}} \left\| \mathbf{Y} - \mathbf{V} \hat{\mathbf{S}} \right\|_F^2 + \eta \left\| \mathbf{X} - \mathbf{R} \mathbf{E} \right\|_F^2 + \gamma \left\| \mathbf{E} \right\|_F^2 \tag{20}\]

where \( \hat{\mathbf{S}} = \mathbf{BS} \) and \( \hat{\mathbf{R}} = \mathbf{RV} \). The first and second terms of the right-hand side are to minimize the reconstruction error, and

![Image](https://example.com/image.png)
the third term is a regularizer. The parameters $\eta$ and $\gamma$ control the balance of these terms; their influence will be analyzed in Section V-C.

The problem (20) can be solved by convex optimization. Let the derivation of $\Psi(E)$ be zero; we have

$$V^TVESS^T + (\eta\tilde{R}^T\tilde{R} + \gamma I)E = V^T\tilde{Y}S^T + \eta\tilde{R}^T\tilde{X}. \quad (21)$$

Left multiplying $(V^TV)^{-1}$ on both sides, (21) can be formulated as a Sylvester equation

$$PE + EQ = C \quad (22)$$

where

$$P = (V^TV)^{-1}(\eta\tilde{R}^T\tilde{R} + \gamma I) \quad (23)$$
$$Q = SS^T \quad (24)$$

and

$$C = (V^TV)^{-1}(V^T\tilde{Y}S^T + \eta\tilde{R}^T\tilde{X}). \quad (25)$$

To solve this problem, we conduct eigenvalue decomposition on $P$

$$P = H\Sigma H^{-1} \quad (26)$$

where $H \in \mathbb{R}^{k \times k}$ is the eigenvector matrix, each column of which is an eigenvector. $\Sigma = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_k)$, where $\lambda_i$ denotes the $i$th eigenvalue.

Inserting (26) into (22) and then left multiplying $H^{-1}$, we have

$$\Sigma H^{-1}E + H^{-1}EQ = H^{-1}C. \quad (27)$$

Let

$$F = (f_1, f_2, \ldots, f_k)^T = H^{-1}E \quad (28)$$
$$C' = (c_1, c_2, \ldots, c_k)^T = H^{-1}C \quad (29)$$

where $f_i$ and $c_i$ are the $i$th row of $F$ and $C'$, respectively. We solve (27) row by row

$$f_i = c_i(\lambda_i I + Q)^{-1}. \quad (30)$$

After recovering $F$, we compute $E$ as

$$E = HF. \quad (31)$$

Finally, the HR-HSI can be reconstructed by $\hat{Z} = VE$.

To summarize, Algorithm 1 presents the procedure of solving the coefficients’ matrix $E$.

### Algorithm 1 Solving the Coefficients’ Matrix $E$

**Input:** LR multispectral matrix $\hat{X}$, HR hyperspectral matrix $\hat{Y}$, spectral basis matrix $V$, spectral degradation function $R$, affine parameter $A$, PSF.

**Output:** HR hyperspectral matrix $\hat{Z}$.

1. Construct $B$ using PSF;
2. Construct $S$ using $A$;
3. Compute $P$ using (23);
4. Compute $Q$ using (24);
5. Compute $C$ using (25);
6. Compute $P$ using (23);
7. Compute $H$ and $\Sigma$ using (26);
8. Compute $C'$ using (29);
9. for $i = 1$ to $k$
   - Compute $f_i$ using (30);
10. end
11. Recover $F = (f_1, f_2, \ldots, f_k)^T$;

### C. Interpolation Manner and Parameter Choosing

Bilinear and bicubic manners are two common alternatives for image interpolation when constructing matrix $S$. We run our algorithm on six remote sensing hyperspectral datasets in unaligned cases using these two interpolation manners. Table I lists the average SAM and PSNR values of reconstructed HSIs and the running time. It is observed that the qualities of reconstructed HSIs are quite similar. Compared with the bicubic manner, the bilinear manner is computationally more efficient as it only needs to consider four nearest neighbor pixels, and accordingly, its spatial mapping matrix $S$ is sparser. Hence, we choose the bilinear manner in constructing $S$.

<table>
<thead>
<tr>
<th>$b = 4$</th>
<th>$b = 8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bilinear</td>
<td>Bicubic</td>
</tr>
<tr>
<td>SAM</td>
<td>2.30</td>
</tr>
<tr>
<td>RMSE</td>
<td>1.93</td>
</tr>
<tr>
<td>PSNR</td>
<td>43.03</td>
</tr>
<tr>
<td>Time (s)</td>
<td>21.89</td>
</tr>
<tr>
<td></td>
<td>25.31</td>
</tr>
</tbody>
</table>

We determine the appropriate values of the parameters $\gamma$ and $\eta$ in (20) from the range $\log_{10}\eta \in [-3, -2, -1, 0, 1, 2, 3]$ and $\log_{10}\gamma \in (-11, -10, -9, -8, -7, -6, -5, -4, -3, -2)$, respectively. Fig. 6 shows the average SAM and RMSE values of all the reconstructed HSIs with respect to the parameters $\eta$ and $\gamma$. We set the combination $\eta = 0.1$ and $\gamma = 10^{-5}$ that produce the lowest RMSE and SAM values.

### VI. Experiments

In this section, we first introduce the datasets we used in our experiments. Then, we introduce five quality metrics, one of which evaluates the registration degree and the others evaluate...
the reconstructed HSIs. Finally, we present the experimental results of our proposed algorithm.

We perform three kinds of experiments. First, we compare our NED measure in the registration stage with NTG [8], SAD [33], SSD [33], and NCC [19]. Then, we evaluate the influence of interpolation error in the fusion stage with or without using our interpolation modeling. Finally, we compare our HSI fusion algorithm with the state-of-the-art ones, including HySure [5], CSTF [6], SRIF [23], uSDN [34], IARF [11], HISRM [50], UAL [29], DBHIF [28], and DHSIS [30]. Similar to our algorithm, HySure, CSTF, and SRIF use degradation models, uSDN builds an unsupervised deep network. IARF and HISRM cope with the misaligned HSI-MSI fusion problem. UAL, DBHIF, and DHSIS are all supervised deep learning algorithms.

A. Datasets

We use six remote sensing datasets (see Fig. 7) to evaluate the algorithms. Datasets A and B were both taken by optics system imaging spectrometer (ROSIS) over Pavia, with the wavelength ranging from 0.43 to 0.86 μm [35]. The scenes are Pavia Center and Pavia University, respectively. We discard the noisy or water absorption channels and keep 93 bands for the experiments. We use a spatial region with 800 × 400 pixels in dataset A and 600 × 280 pixels in dataset B as HSI ground truth.

Dataset C was captured by the NASA Airborne Visible Infrared Imaging Spectrometer (AVIRIS) [36], covering the Cuprite in Las Vegas. It is the basic hyperspectral dataset for spectrum unmixing [37]–[40]. The dataset contains 224 bands ranging from 370 to 2480 nm. After eliminating noisy channels (1–3 and 221–224) and water absorption channels (107–113 and 153–168), 194 bands remain. We use an image region of 496 × 496 pixels as HSI ground truth.

Dataset D is the Kennedy Space Center (KSC) [36], [41], which was also taken by NASA AVIRIS over Florida at an altitude of about 20 km, with the wavelength ranging from 400 to 2500 nm. Getting rid of noisy channels and water absorption channels, 66 bands are preserved. A 496 × 496 spatial region is used in the experiments.

Dataset E is Urban [42], [43], one of the most widely used hyperspectral datasets in spectrum unmixing. It contains 210 bands ranging from 400 to 2500 nm with an increment of 10 nm. Removing the noisy bands and water absorption bands, we remained 162 bands. The 280 × 280 pixel-sized region is cut out from the image for the experiments.

Dataset F was captured by an airborne hyperspectral data flight line over the Washington DC Mall [44], with the wavelength ranging from 0.4 to 2.4 μm. We use 191 bands and cut out the image region with 376 × 272 pixels for the experiments.

To generate HR-MSIs, we use IKONOS-like spectral response filters [45] to map multiple bands to four bands. To generate the LR-HSIs, we blur the ground-truth HSIs spatially and downsampled them with scale factor b = 4, 8, and 16, respectively. The s × s Gaussian filters with the size s = (3b)/2 − 1 and variance σ² = (3b)/8 are used to blur the images.

Besides, the CAVE dataset [46] and the Harvard dataset [47] are also used to compare our algorithm with the supervised deep learning algorithms, i.e., UAL [29], DBHIF [28], and DHSIS [30]. The CAVE dataset contains 32 indoor scenes, and the Harvard dataset has 50 outdoor scenes.

B. Quantitative Metrics

Five metrics are used to evaluate our experimental results. The first one is registration error [48], formulated as

\[ e = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} \left\| \hat{p}(x, y, \Lambda_{\text{opt}}) - \hat{p}(x, y, \Lambda_{\text{gt}}) \right\|_F^2 \]  

(32)

where MN is the total number of pixels, x and y denote the coordinate of pixels, Λ_{opt} and Λ_{gt} denote the estimated and ground-truth affine parameters, respectively, and \( \hat{p} \) represents the pixel position after image warping.

The other four metrics evaluate the quality of the reconstructed HS images. The spectral angle mapper (SAM) [49]

1Available at http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes
2Available at http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes
3Available at https://sites.google.com/site/feiyunzhuhomepage/datasets-ground-truths
4Available at https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html
computes the average angle difference between two images as
\[
\text{SAM}(\mathbf{Z}, \hat{\mathbf{Z}}) = \frac{1}{MN} \sum_{j=1}^{M} \arccos \left( \frac{\mathbf{Z}_j^T \hat{\mathbf{Z}}_j}{\|\mathbf{Z}_j\|_2 \|\hat{\mathbf{Z}}_j\|_2} \right)
\]
where \( \mathbf{Z} \in \mathbb{R}^{M \times N \times L} \) and \( \hat{\mathbf{Z}} \in \mathbb{R}^{M \times N \times L} \) denote the estimated and ground-truth HR-HSIs, respectively. \( \mathbf{Z}_j \) and \( \hat{\mathbf{Z}}_j \) present the spectral vectors of the \( j \)th pixel of the images. The smaller the SAM, the better the spectrum reconstruction.

The third metric, the root mean square error (RMSE) [49], is defined as
\[
\text{RMSE}(\mathbf{Z}, \hat{\mathbf{Z}}) = \sqrt{\frac{1}{MN} \sum_{j=1}^{M} \sum_{i=1}^{L} (\mathbf{Z}_j(i) - \hat{\mathbf{Z}}_j(i))^2}
\]
where \( \mathbf{Z}_j(i) \) and \( \hat{\mathbf{Z}}_j(i) \) are the ground-truth and estimated intensities in the \( i \)th band of the images. The smaller the RMSE, the better the reconstruction quality.

The last metric is ERGAS [49], formulated as
\[
\text{ERGAS}(\mathbf{Z}, \hat{\mathbf{Z}}) = \frac{100}{b} \sqrt{\frac{1}{L} \sum_{i=1}^{L} \left( \frac{\text{RMSE}_i}{\mu \hat{\mathbf{Z}}_i} \right)^2}
\]
where \( b \) is the scale factor of the spatial resolution from the HR-MSI to LR-HSI, \( \text{RMSE}_i \) is the RMSE value in the \( i \)th band, and \( \mu \hat{\mathbf{Z}}_i \) is the mean of the ground-truth image in the \( i \)th band. The smaller the ERGAS, the better the reconstructed image quality.

### C. Evaluation of Image Registration

In the following, we evaluate the registration process of our algorithm. We first apply three affine transforms to HR-MSIs and downsample HSIs with 4× and 8× to create LR-HSIs. Then, the unaligned HR-MSI and LR-HSI pairs are put into our image registration algorithm. We impose small, middle, and large deformations on the images using the affine parameters
\[
\begin{align*}
A_1 &= (0.99, 0.05, -5, 0.04, 0.97, -5)^T \\
A_2 &= (1.02, 0.03, -10, -0.02, 0.98, -10)^T \\
A_3 &= (0.98, 0.03, -15, -0.03, 1.01, -15)^T.
\end{align*}
\]

Fig. 8 illustrates the blending images of six datasets before and after registration using our NED-based algorithm, shown as pseudo-RGBs. It is obvious that ghosts are significantly removed after registration. Fig. 9 shows the image registration results on the Pavia University dataset when the LR images are 8× downsampled. It is observed that our NED performs the best among the five measures with either visual or quantitative evaluation.

Table II lists the registration error on the six remote sensing datasets with scale factor \( b = 4 \) and 8. Our NED measure always produces the lowest registration error on the datasets B, E, and F in all cases. It performs the best or close to best on the rest datasets in most cases. Hence, we consider that our NED measure is suitable for the HSI-MSI fusion problem.

### D. Influence of Interpolation Error

We note that there exists an error between the intensity of real pixels and that computed by interpolation. Here, we analyze the influence of the interpolation error on the reconstructing HSIs without or with interpolation modeling. In the case of without interpolation modeling, we warp the LR-HSI using the ground-truth affine parameter and conduct image fusion. In the case of interpolation modeling, we incorporate the ground-truth affine parameter in the fusion process, as discussed in Section V. Table III compares the average...
Fig. 9. Image registration results on the Pavia University dataset using SSD [33], SAD [33], NCC [19], NTG [8], and our NED. The LR image and the HR image are unaligned. The spatial size relationship is $8 \times$. Blending images of original state and results are shown in pseudocolors. The second row shows the detail regions in red boxes.

TABLE III

<table>
<thead>
<tr>
<th></th>
<th>$b = 4$</th>
<th>$b = 8$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SAM</td>
<td>RMSE</td>
</tr>
<tr>
<td>Dataset A: Pavia Center</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o modeling</td>
<td>3.01</td>
<td>1.61</td>
</tr>
<tr>
<td>w/ modeling</td>
<td>2.58</td>
<td>1.37</td>
</tr>
<tr>
<td>Dataset B: Pavia University</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o modeling</td>
<td>2.44</td>
<td>2.20</td>
</tr>
<tr>
<td>w/ modeling</td>
<td>2.11</td>
<td>1.97</td>
</tr>
<tr>
<td>Dataset C: Cuprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o modeling</td>
<td>0.73</td>
<td>1.77</td>
</tr>
<tr>
<td>w/ modeling</td>
<td>0.68</td>
<td>1.61</td>
</tr>
<tr>
<td>Dataset D: KSC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o modeling</td>
<td>2.48</td>
<td>1.49</td>
</tr>
<tr>
<td>w/ modeling</td>
<td>1.87</td>
<td>1.16</td>
</tr>
<tr>
<td>Dataset E: Urban</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o modeling</td>
<td>3.41</td>
<td>3.93</td>
</tr>
<tr>
<td>w/ modeling</td>
<td>2.86</td>
<td>3.19</td>
</tr>
<tr>
<td>Dataset F: Washington DC Mall</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o modeling</td>
<td>3.91</td>
<td>2.34</td>
</tr>
<tr>
<td>w/ modeling</td>
<td>3.63</td>
<td>2.20</td>
</tr>
</tbody>
</table>

Fig. 10. PSNR values of the reconstructed HSIs on six remote sensing datasets and the CAVE dataset in the cases of with and without interpolation treatments. (a) PSNR values on six datasets. (b) PSNR values of the images in the CAVE dataset. The image index is sorted with respect to the PSNR values with interpolation modeling.

E. Evaluation of Image Fusion

In the following, we evaluate the image fusion process of our algorithm. We compare our algorithm with the state of the arts, including HySure [5], CSTF [6], SRIF [23], uSDN [34], IARF [11], HISRM [50], UAL [29], DBHIF [28], and DHSIS [30], among which the last three algorithms are supervised deep learning ones that need model training. IARF and HISRM deal with the unregistered HSI fusion problem, while the other competitors deal with aligned image pairs. Therefore, for HySure, CSTF, SRIF, uSDN, UAL, DBHIF, and reconstructed HSIs on six remote sensing datasets with LR images $4 \times$ downsampled. It is observed that the algorithm with interpolation modeling produces higher PSNR values. To further validate this, we test on a larger dataset, the CAVE dataset, by fusing unaligned RGB and MS image pairs. Fig. 10(b) shows that the algorithm with our interpolation modeling always performs better than that without modeling.

results on six datasets with scale factor $b = 4$ and 8. It is obvious that, on all metrics, the algorithm with interpolation modeling clearly improves the accuracy of HSI reconstruction compared with the traditional algorithms directly using image warping in image fusion.

In the HSI-MSI registration process, there is also the registration error in addition to the interpolation error. Hence, we then use our registration stage to estimate the affine parameter and compare the algorithms with and without interpolation modeling. Fig. 10(a) shows the PSNR values of
DHSIS, we need to align the HSI and MSI before performing HSI-MSI fusion.

We first evaluate the impacts of two warping manners (warping LR-HSI or warping HR-MSI) on image fusion since both manners cause interpolation errors. Table IV lists the SAM and RMSE values of the reconstructed HSIs when using different warping manners on the Pavia Center or CAVE datasets with 8 × resolution improvement. It is observed that, for HySure, CSTF, uSDN, and UAL, warping LR-HSI produces smaller SAM and RMSE values. SRIF performs better when warping HR-MSI. For DBHIF and DHSIS, warping LR-HSI produces

![Image](image-url)
lower RMSE values but higher SAM values compared with the HR-MSI warping manner. Our investigation indicates that the algorithms perform similarly on other datasets, and hence, for a fair comparison, we use different warping manners for individual algorithms in the following experiments. For clarity, we list the configurations of these algorithms in Table V, including warping manner, datasets, and the necessity of model training.

We first compare our algorithm with the competitors that do not need training, including HySure [5], CSTF [6], SRIF [23],
TABLE VII
AVERAGE SAM, RMSE, PSNR, AND ERGAS VALUES OF THE RECONSTRUCTED HSI FOR THE SAME-DATASET VALIDATION AND THE CROSS-DATASET VALIDATION WITH SCALE FACTOR \( b = 4, 8, \) AND 16

<table>
<thead>
<tr>
<th></th>
<th>( b = 4 )</th>
<th></th>
<th>( b = 8 )</th>
<th></th>
<th>( b = 16 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SAM</td>
<td>RMSE</td>
<td>PSNR</td>
<td>ERGAS</td>
<td>SAM</td>
</tr>
<tr>
<td>UAL [29]</td>
<td>5.93</td>
<td>2.26</td>
<td>42.41</td>
<td>2.40</td>
<td>6.35</td>
</tr>
<tr>
<td>DBHIF [28]</td>
<td>5.70</td>
<td>2.62</td>
<td>43.98</td>
<td>2.47</td>
<td>6.94</td>
</tr>
<tr>
<td>Ours</td>
<td>7.24</td>
<td>2.72</td>
<td>41.23</td>
<td>3.03</td>
<td>7.79</td>
</tr>
</tbody>
</table>

TABLE VIII
RUNNING TIMES OF HYSURE, CSTF, SRIF, USDN, IARF, HISRM, AND OUR ALGORITHM WHEN RECONSTRUCTING AN HR-HSI WITH 31 SPECTRAL BANDS, 496 × 496 SPATIAL RESOLUTION, AND 4× RESOLUTION IMPROVEMENT

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Time (s)</th>
<th>Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYSURE [5]</td>
<td>106.26</td>
<td>CPU</td>
</tr>
<tr>
<td>CSTF [6]</td>
<td>81.10</td>
<td>CPU</td>
</tr>
<tr>
<td>SRIF [23]</td>
<td>22.45</td>
<td>CPU</td>
</tr>
<tr>
<td>USDN [34]</td>
<td>725.66</td>
<td>CPU</td>
</tr>
<tr>
<td>IARF [11]</td>
<td>152.73</td>
<td>CPU</td>
</tr>
<tr>
<td>Ours</td>
<td>17.80</td>
<td>CPU</td>
</tr>
</tbody>
</table>

CPU (Intel Core i5-10600K) and 16-GB RAM, and USDN [34] is implemented using Python on the same computer. Deep learning methods, UAL [29], DBHIF [28], and DHSIS [30], are implemented using Python on GeForce RTX 2070 (GPU).

TABLE VII shows the running times of these algorithms when reconstructing an HR-HSI with 31 spectral bands, 496 × 496 spatial resolution, and 4× resolution improvement. We can see that our algorithm costs the least time compared to competitors running on CPU. Compared to the deep learning ones, our algorithm runs faster than UAL but slower than DBHIF and DHSIS. In general, our algorithm is computationally efficient.

VII. CONCLUSION
In this work, we have proposed an HSI fusion framework for unaligned LR-HSI and HR-MSI pairs. NED is introduced as a new measure to cope with the clear–blur image registration problem. Different from the previous work that directly warps the LR-HSIs, we employ the interpolation modeling treatment by incorporating the affine parameter into the spatial degradation model. Experimental results validate that the interpolation modeling treatment can effectively improve the quality of HSI reconstruction. The proposed image fusion algorithm is useful to real-world applications in which HSI and MSI cannot be assumed to be perfectly aligned.

APPENDIX
A. Affine Transformation
The affine transformation is formulated as

\[
T(x, y, A) = \begin{pmatrix} a_{1x} + a_{2y} + a_3 \\ a_{4x} + a_{5y} + a_6 \end{pmatrix} = XA
\]  

Authorized licensed use limited to: Zhejiang University. Downloaded on January 05,2022 at 02:11:43 UTC from IEEE Xplore. Restrictions apply.
where
\[
A = (a_1, a_2, \ldots, a_d)^\top \in \mathbb{R}^{d \times 1}
\]
\[
X = \begin{pmatrix}
  x & y & 1 & 0 & 0 & 0 \\
  0 & 0 & 0 & x & y & 1
\end{pmatrix} \in \mathbb{R}^{2 \times 6}.
\]

(39)
(40)

B. Approximation of L1-Norm

We use the Charbonnier penalty function \([51]\) to approximate the L1-norm
\[
\Psi(x) = \sum_{i=1}^{N} \sqrt{x_i^2 + \delta^2}, \quad x \in \mathbb{R}^N
\]
where \(\delta\) is fixed small constant. In this work, we set \(\delta = 0.01\).

The first derivative of the Charbonnier function is
\[
\nabla^T \Psi(x) = \begin{pmatrix}
  x_1/\sqrt{x_1^2 + \delta^2} \\
  \vdots \\
  x_N/\sqrt{x_N^2 + \delta^2}
\end{pmatrix} \top.
\]

(42)

C. Derivation of Gradient

By approximating the L1-norm with the Charbonnier function, NED can be rewritten as
\[
J(A) = \frac{\Psi(E_0 - E_c(T(x, y, A)))}{\Psi(E_0) + \Psi(E_c(T(x, y, A)))}
\]
Let
\[
P(A) = \Psi(E_0 - E_c(T(x, y, A)))
\]
and
\[
Q(A) = \Psi(E_0) + \Psi(E_c(T(x, y, A))).
\]

We have \(J(A) = (P(A))/Q(A)\) and
\[
\nabla_A J = \frac{(\nabla_A P) - J(\nabla_A Q)}{Q}.
\]

(46)

Then, we compute the first derivations of \(P\) and \(Q\) with respect to \(A\)
\[
\nabla_A P = (\nabla_A E_c)\nabla^T \Psi(E_0 - E_c)
\]
\[
\nabla_A Q = (\nabla_A E_c)\nabla^T \Psi(E_c).
\]

(47)
(48)

Combining (46)–(48), we have
\[
\nabla_A J(A) = \frac{(\nabla_A E_c)[\nabla^T \Psi(E_0 - E_c) - J\nabla^T \Psi(E_c)]}{Q}.
\]

(49)

Let
\[
w = \nabla^T \Psi(E_0 - E_c) - J\nabla^T \Psi(E_c) \in \mathbb{R}^N.
\]

(50)

We have
\[
\nabla_A J(A) = \frac{(\nabla_A E_c)w}{Q}.
\]

(51)

Since we only concern the direction of \(\nabla_A J(A)\), (51) can be simplified to
\[
\nabla_A J(A) \propto (\nabla_A E_c)w.
\]

(52)

Here, we denote \(\nabla A e_i = (\nabla A e_{1i}, \nabla A e_{2i}, \ldots, \nabla A e_{Ni})\), each column of which is computed as
\[
\nabla A e_i = \nabla A (T(x, y, A)) = (\nabla A T(x, y, A))(\nabla A e_i).
\]

(53)

According to (38), we have
\[
\nabla_A T(x, y, A) = \nabla_A X A = X^\top.
\]

(54)

In (53), \(\nabla e_c\) denotes the gradients at pixel \(e_i\) in the \(x\)- and \(y\)-directions, formulated as
\[
\nabla e_c = \begin{pmatrix}
  \nabla_x e_c \\
  \nabla_y e_c
\end{pmatrix}.
\]

(55)

Combining (52)–(55), we have
\[
\nabla_A J(A) \propto \sum_{i=1}^{N} w_i X^\top \nabla e_{ci}
\]
which completes the deduction of \(\nabla_A J\).

REFERENCES


