Joint 2D SSA with GSVD and binary linear programming based image super-resolution

Ziyin Huang and Yui-Lam Chan
Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, China.
shmillehzy@gmail.com and enylchan@polyu.edu.hk
Bingo Wing-Kuen Ling
School of Information Engineering, Guangdong University of Technology, Guangzhou, 51006, China.
yongquanling@gdut.edu.cn
Huan Ye
Guangzhou Academy of Fine Arts, Guangzhou, Guangdong Province, China.
18620114040@qq.com

Abstract—This paper proposes a joint two dimensional (2D) singular spectrum analysis (SSA) with the generalized singular value decomposition (GSVD) and the binary linear programming based method for performing the super-resolution. For a given low resolution image, first both the upsampling operation and a lowpass filtering are applied on each column of the image to obtain an enlarged image. Second, apply the 2D Hankelization to both the low resolution image and the enlarged image to obtain their corresponding trajectory matrices. Third, both the GSVD and the 2D de-Hankelization are applied to these two trajectory matrices to obtain their corresponding sets of the de-Hankelized 2D SSA components. Here, it is proved that the exact perfect reconstruction is achieved. In order to enhance the high frequency contents of the enlarged image, the selection of the de-Hankelized 2D SSA components is formulated as a binary linear programming problem. Computer numerical simulation results show that the proposed method outperforms the state-of-the-art methods.

Keywords—Super-resolution, 2D SSA, GSVD, binary quadratic programming, binary linear programming.

1. Introduction
Super-resolution images are the high resolution images constructed using the low resolution images [1]. The low resolution images are usually obtained due to the limitations of the acquisition hardware [1]. In this case, the acquisition process can be modeled by the downsampling operation [2]. Since the downsampling operation results to the lost of the information and the occurrence of the aliasing, in general the downsampling operation is irreversible [3]. Hence, it is very challenges to reconstruct the high resolution images using these low resolution images. However, if the reconstruction error is suppressed, then the details of the images can be seen more clearly. This result is very useful to the image forensics and medical diagnosis [4].

To address the above difficulty, the low resolution images are first upsampled. Then, the upsampled images are processed via the lowpass interpolation filters [5]. Since the upsampling operation and the lowpass filtering are the linear operators, the obtained performances are limited by these linear operators. On the other hand, the deep learning approach [6] is proposed. Although the deep learning approach is a kind of nonlinear methods, the required computational power is very high.

The outline of this paper is as follows. Section 2 presents our proposed method based on the joint 2D SSA with the GSVD and binary linear programming. Section 3 presents the computer numerical simulation results. Finally, a conclusion is drawn in Section 4.

2. Method
This paper aims to perform a nonlinear and adaptive post processing to the upsampled and filtered images to reduce the reconstruction error. The research work is designed as follows. First, the trajectory matrix is constructed via the 2D Hankelization [7]. The details are discussed in Section 2.1. Second, the GSVD [8] is employed to decompose the trajectory matrices to obtain the 2D SSA components of both the low resolution images as well as the corresponding upsampled and filtered images. The details are discussed in Section 2.2. Third, the 2D SSA components are selected via a binary linear programming approach [9]. The details are discussed in Section 2.4. Finally, the post processing images are obtained via the 2D de-Hankelization. The details are discussed in Section 2.3.

2.1. 2D Hankelization
Let a low resolution image be \( Y \) and the size of \( Y \) be \( \tilde{L}_1 \times \tilde{L}_2 \). That is, \( Y \in \mathbb{R}^{\tilde{L}_1 \times \tilde{L}_2} \). First, \( Y \) is divided into some blocks. Let the sizes of these blocks be \( L_1 \times L_2 \). Here, it is assumed that \( \tilde{L}_1 \) is an integer multiple of \( L_1 \) and \( \tilde{L}_2 \) is an integer multiple of \( L_2 \). That is, \( \frac{\tilde{L}_1}{L_1} \) and \( \frac{\tilde{L}_2}{L_2} \) are positive integers. These blocks are taken from \( Y \) by shifting one pixel horizontally and \( L_i \) pixels vertically. Let the \( (m,n) \)th block of \( Y \) be \( Y_{m,n} \) for \( m = 1, \ldots, \frac{\tilde{L}_1}{L_1} \) and for \( n = 1, \ldots, \frac{\tilde{L}_2}{L_2} + 1 \). Obviously, \( Y_{m,n} \in \mathbb{R}^{L_1 \times L_2} \) for
for \( n = 1, \ldots, \tilde{L} - L + 1 \). Let the \( j \)th column of \( \mathbf{Y}_{m,n} \) be \( \mathbf{y}_{m,n+l-1} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \), for \( n = 1, \ldots, \tilde{L} - L + 1 \) and for \( l = 1, \ldots, L \). Obviously, \( \mathbf{y}_{m,n+l} \in \mathbb{R}^{L} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( l = 1, \ldots, \tilde{L} \) as well as \( \mathbf{Y}_{m,n} = [\mathbf{y}_{m,n} \ \cdots \ \mathbf{y}_{m,n+L-1}] \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( n = 1, \ldots, \tilde{L} - L + 1 \). Then, we have

\[
\mathbf{Y} = \begin{bmatrix}
\mathbf{y}_{1,1} & \cdots & \mathbf{y}_{1,L-1} \\
\vdots & \ddots & \vdots \\
\mathbf{y}_{L-1,1} & \cdots & \mathbf{y}_{L-1,L-1}
\end{bmatrix}
\]

To construct the trajectory matrix, first it is required to perform the vectorization on \( \mathbf{Y}_{m,n} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( n = 1, \ldots, \tilde{L} - L + 1 \). Let the obtained vectors be \( \tilde{\mathbf{y}}_{m,n} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( n = 1, \ldots, \tilde{L} - L + 1 \). That is, \( \tilde{\mathbf{y}}_{m,n} = [\mathbf{y}_{m,n}^{T} \ \cdots \ \mathbf{y}_{m,n+L-2}^{T}] \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( n = 1, \ldots, \tilde{L} - L + 1 \). Here, the superscript “*T” denotes the transposition operator. Obviously, \( \tilde{\mathbf{y}}_{m,n} \in \mathbb{R}^{L \times L} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( n = 1, \ldots, \tilde{L} - L + 1 \). Define

\[
\overline{\mathbf{y}}_{m,n} = [\tilde{\mathbf{y}}_{m,1} \ \cdots \ \tilde{\mathbf{y}}_{m,L-1}]
\]

for \( m = 1, \ldots, \frac{\tilde{L}}{L} \). Obviously, \( \overline{\mathbf{y}}_{m,n} \in \mathbb{R}^{L \times (L - 1)} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \). Let the trajectory matrix be \( \tilde{\mathbf{Y}} \). Now, the trajectory matrix is constructed by putting \( \overline{\mathbf{y}}_{m} \) horizontally for \( m = 1, \ldots, \frac{\tilde{L}}{L} \). That is,

\[
\tilde{\mathbf{Y}} = \begin{bmatrix}
\overline{\mathbf{y}}_{1} & \cdots & \overline{\mathbf{y}}_{\frac{\tilde{L}}{L}}
\end{bmatrix}
\]

Obviously, \( \tilde{\mathbf{Y}} \in \mathbb{R}^{L \times (L - 1) \times \frac{\tilde{L}}{L}} \).

Let the upsampled and filtered image be \( \mathbf{Z} \) and the upsampling ratio be \( S \). It is assumed that \( S > 1 \) as well as both \( SL_{x} \) and \( SL_{y} \) are integers, but \( S \) is not necessary to be an integer. Obviously, the size of \( \mathbf{Z} \) is \( SL_{x} \times L_{z} \). That is, \( \mathbf{Z} \in \mathbb{R}^{SL_{x} \times L_{z}} \). Similarly, \( \mathbf{Z} \) is also divided into some blocks. Denote the sizes of these blocks as \( SL_{x} \times L_{z} \). Similarly to \( \mathbf{Y} \), the blocks are taken from \( \mathbf{Z} \) by shifting one pixel horizontally but \( SL_{y} \) pixels vertically. Let the \((m,n)\)th block of \( \mathbf{Z} \) be \( \mathbf{Z}_{m,n} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( n = 1, \ldots, \tilde{L} - L + 1 \). Obviously, \( \mathbf{Z}_{m,n} \in \mathbb{R}^{SL_{x} \times L_{z}} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( n = 1, \ldots, \tilde{L} - L + 1 \). Let the \( i \)th column of \( \mathbf{Z}_{m,n} \) be \( \mathbf{z}_{m,n+i-1} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \), for \( n = 1, \ldots, \tilde{L} - L + 1 \) and for \( i = 1, \ldots, L \). Obviously, \( \mathbf{z}_{m,n+i} \in \mathbb{R}^{L} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( i = 1, \ldots, \tilde{L} \). Also, \( \mathbf{Z}_{m,n} = [\mathbf{z}_{m,n} \ \cdots \ \mathbf{z}_{m,n+L-1}] \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( n = 1, \ldots, \tilde{L} - L + 1 \). Then, we have

\[
\mathbf{Z} = \begin{bmatrix}
\mathbf{z}_{1,1} & \cdots & \mathbf{z}_{1,L-1} \\
\vdots & \ddots & \vdots \\
\mathbf{z}_{L-1,1} & \cdots & \mathbf{z}_{L-1,L-1}
\end{bmatrix}
\]

Similarly, the vectorization on \( \mathbf{Z}_{m,n} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( n = 1, \ldots, \tilde{L} - L + 1 \) is performed. Let the obtained vectors be \( \tilde{\mathbf{z}}_{m,n} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( n = 1, \ldots, \tilde{L} - L + 1 \). That is, \( \tilde{\mathbf{z}}_{m,n} = [\mathbf{z}_{m,n}^{T} \ \cdots \ \mathbf{z}_{m,n+L-2}^{T}] \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( n = 1, \ldots, \tilde{L} - L + 1 \). Obviously, \( \tilde{\mathbf{z}}_{m,n} \in \mathbb{R}^{L \times L} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \) and for \( n = 1, \ldots, \tilde{L} - L + 1 \). Define

\[
\overline{\mathbf{z}}_{m,n} = [\tilde{\mathbf{z}}_{m,1} \ \cdots \ \tilde{\mathbf{z}}_{m,L-1}]
\]

for \( m = 1, \ldots, \frac{\tilde{L}}{L} \). Obviously, \( \overline{\mathbf{z}}_{m,n} \in \mathbb{R}^{L \times (L - 1)} \) for \( m = 1, \ldots, \frac{\tilde{L}}{L} \). Let the trajectory matrix be \( \tilde{\mathbf{Z}} \). Now, the trajectory matrix is constructed by putting \( \overline{\mathbf{z}}_{m} \) horizontally for \( m = 1, \ldots, \frac{\tilde{L}}{L} \). That is,

\[
\tilde{\mathbf{Z}} = \begin{bmatrix}
\overline{\mathbf{z}}_{1} & \cdots & \overline{\mathbf{z}}_{\frac{\tilde{L}}{L}}
\end{bmatrix}
\]

Obviously, \( \tilde{\mathbf{Z}} \in \mathbb{R}^{L \times (L - 1) \times \frac{\tilde{L}}{L}} \).

2.2. GSVD

Instead of performing the singular value decomposition (SVDs) of \( \tilde{\mathbf{Y}} \) and \( \tilde{\mathbf{Z}} \) individually, this paper proposes to perform the GSVD on both \( \tilde{\mathbf{Y}} \) and \( \tilde{\mathbf{Z}} \) simultaneously. In particular, denote the superscript “*H” as the conjugate transposition operator. Let \( \mathbf{I}_{a} \) be the \( a \times a \) identity matrix as well as \( \mathbf{U} \) and \( \mathbf{V} \) be the \( L_{x} \times L_{y} \) unitary matrix and the \( SL_{x} \times SL_{y} \) unitary matrix, respectively.
Obviously, \( U \in C^{l_1 \times d_{i_1} d_{i_2}} \) and \( V \in C^{s_{i_1} \times d_{i_1} d_{i_2}} \). That is, 
\[ U^{HH} = U^H U = I_{d_{i_1} d_{i_2}} \quad \text{and} \quad V^{HH} = V^H V = I_{s_{i_1} d_{i_2}} \].

Let 
\[ \text{rank}(\bar{Y}) = r_1 \leq \min \left( L_1 L_2, \frac{L_1}{L_1} (L_2 - L_1 + 1) \right) \]
and 
\[ \text{rank}(\bar{Z}) = r_2 \leq \min \left( S_1 L_2, \frac{L_1}{L_1} (L_2 - L_1 + 1) \right) \]
as well as \( r_1 = r_1 + r_2 \). Here, \( \text{rank}(A) \) refers to the total number of the independent rows or the total number of the independent columns of \( A \). Also, \( \min(a, b) \) refers to the minimum value of \( a \) and \( b \). Moreover, \( \text{diag}(a) \) denotes the square diagonal matrix with its diagonal elements being the elements of \( a \). Define two non-negative vectors \( c \in \mathbb{R}^s \) and \( s \in \mathbb{R}^r \). That is, \( c \geq 0 \_s \) and \( s \geq 0 \_r \). Let 
\[ \bar{C} = \begin{bmatrix} 0_{(L_2 - r_2) \times (r_2 - r_2)} & 0_{(L_2 - r_2) \times r_2} \\ c_1 & \cdots & c_r \end{bmatrix} \in \mathbb{R}^{L_2 \times r_2} \]
and 
\[ \bar{S} = \begin{bmatrix} \text{diag}(s) & 0_{s \times (r_2 - r_2)} \\ 0_{s \times (L_2 - r_2)} & 0_{s \times (L_2 - r_2)} \end{bmatrix} \in \mathbb{R}^{s \times L_2} \]
as well as \( X \in C^{L_2 \times r_2} \). Then, we have \( \bar{Y} = U \bar{C} X^H \), \( \bar{Z} = V \bar{S} X^H \) and \( \bar{C}^H \bar{C} + \bar{S}^H \bar{S} = I_{r_2} \). Let the columns of \( U \) be \( U_l \) for \( l = 1, \ldots, L_1 \) and the columns of \( V \) be \( V_l \) for \( l = 1, \ldots, S_1 \). Also, let \( c = [c_1 \ldots c_r]^T \) and 
\[ s = [s_1 \ldots s_r]^T \] as well as the column of \( X \) be \( X_l \) for \( l = 1, \ldots, r_2 \). Define the Hankelized 2D SSA components as 
\[ \tilde{Y}_l = [\tilde{Y}_{l,1} \tilde{Y}_{l,2} \cdots \tilde{Y}_{l,r_2}] \]
and 
\[ \tilde{Z}_l = [\tilde{Z}_{l,1} \tilde{Z}_{l,2} \cdots \tilde{Z}_{l,r_2}] \] for \( l = 1, \ldots, r_2 \). Obviously, \( \tilde{Y}_l \in \mathbb{R}^{L_{l,1} \times (L_{l,1} - r_2 + 1)} \) for \( l = 1, \ldots, r_1 \) and \( \tilde{Z}_l \in \mathbb{R}^{S_{l,1} \times (S_{l,1} - r_2 + 1)} \) for \( l = 1, \ldots, r_2 \).

It is worth noting that the right decomposition matrices obtained via applying the GSVD to both \( \bar{Y} \) and \( \bar{Z} \) are \( X \). Hence, the Hankelized 2D SSA components of both \( \bar{Y} \) and \( \bar{Z} \) would be similar to each other. As a result, the super-resolution image would be close to the original image.

2.3. 2D de-Hankelization

Now, \( \tilde{Y}_l \) for \( l = 1, \ldots, r_2 \) is divided into \( \frac{L_1}{L_1} \) blocks horizontally. Let the \( m \)th block of \( \tilde{Y}_l \) be \( \tilde{Y}_{l,m} \) for \( l = 1, \ldots, r_2 \) and for \( m = 1, \ldots, \frac{L_1}{L_1} \). That is, 
\[ \tilde{Y}_l = \begin{bmatrix} \tilde{Y}_{l,1} & \cdots & \tilde{Y}_{l,r_2} \end{bmatrix} \] for \( l = 1, \ldots, r_2 \). Obviously, 
\[ \tilde{Y}_{l,m} \in \mathbb{R}^{L_{l,1} \times (L_{l,1} - r_2 + 1)} \] for \( l = 1, \ldots, r_2 \) and for \( m = 1, \ldots, \frac{L_1}{L_1} \).

Likewise, \( \tilde{Z}_l \) for \( l = 1, \ldots, r_2 \) is divided into \( \frac{L_1}{L_1} \) blocks horizontally. Let the \( m \)th block of \( \tilde{Z}_l \) be \( \tilde{Z}_{l,m} \) for \( l = 1, \ldots, r_2 \) and for \( m = 1, \ldots, \frac{L_1}{L_1} \). Obviously, 
\[ \tilde{Z}_{l,m} = \begin{bmatrix} \tilde{Z}_{l,m,1} & \cdots & \tilde{Z}_{l,m,r_2} \end{bmatrix} \] for \( l = 1, \ldots, r_2 \) and for \( m = 1, \ldots, \frac{L_1}{L_1} \). That is, 
\[ \tilde{Z}_{l,m} \in \mathbb{R}^{S_{l,1} \times (S_{l,1} - r_2 + 1)} \] for \( l = 1, \ldots, r_2 \) and for \( m = 1, \ldots, \frac{L_1}{L_1} \).
\[ Z_i = \begin{bmatrix} \hat{Z}_{i,1} & \cdots & \hat{Z}_{i,r} \end{bmatrix} \quad \text{for} \quad l = 1, \ldots, r \]

That is, \( \hat{Z}_{i,m} \in \mathbb{R}^{S_L, L_2} \) for \( l = 1, \ldots, r \) and for \( m = 1, \ldots, \frac{L_1}{L_i} \).

Let the \( p^{\text{th}} \) column of \( \hat{Z}_{i,m} \) be \( \hat{Z}_{i,m,p} \) for \( l = 1, \ldots, r \), for \( m = 1, \ldots, \frac{L_1}{L_i} \) and for \( p = 1, \ldots, \tilde{L}_2 - \tilde{L}_2 + 1 \).

That is, \( \hat{Z}_{i,m} = \begin{bmatrix} \hat{Z}_{i,m,1} & \cdots & \hat{Z}_{i,m,r} \end{bmatrix} \) for \( l = 1, \ldots, r \) and for \( m = 1, \ldots, \frac{L_1}{L_i} \). Obviously, \( \hat{Z}_{i,m, p} \in \mathbb{R}^{S_L, L_2} \) for \( l = 1, \ldots, r \), for \( m = 1, \ldots, \frac{L_1}{L_i} \) and for \( p = 1, \ldots, \tilde{L}_2 - \tilde{L}_2 + 1 \).

Further divided into \( L_2 \) blocks vertically. Let the \( q^{\text{th}} \) block of \( \hat{Z}_{i,m, p} \) be \( \hat{Z}_{i,m,p,q} \) for \( l = 1, \ldots, r \), for \( m = 1, \ldots, \frac{L_1}{L_i} \), for \( p = 1, \ldots, \tilde{L}_2 - \tilde{L}_2 + 1 \) and for \( q = 1, \ldots, L_2 \). That is, \( \hat{Z}_{i,m,p} = \begin{bmatrix} \hat{Z}_{i,m,p,1} & \cdots & \hat{Z}_{i,m,p,r} \end{bmatrix} \) for \( l = 1, \ldots, r \), for \( m = 1, \ldots, \frac{L_1}{L_i} \) and for \( p = 1, \ldots, \tilde{L}_2 - \tilde{L}_2 + 1 \).

For \( l = 1, \ldots, r \), for \( m = 1, \ldots, \frac{L_1}{L_i} \), and for \( p = 1, \ldots, \tilde{L}_2 - \tilde{L}_2 + 1 \), \( \hat{Z}_{i,m, p} \) for \( q = 1, \ldots, L_2 \).

Similarly, the vector based 2D de-Hankelization is performed on \( \hat{Z}_{i,m, p} \) for \( l = 1, \ldots, r \) and \( m = 1, \ldots, \frac{L_1}{L_i} \).

Define

\[
\hat{Z}_{i,m,q} = \begin{cases} \frac{1}{q} \sum_{k=1}^{q} \hat{Z}_{i,m,q-k+1, k} & \text{for } 1 \leq q < L_2 \\ \frac{1}{L_2} \sum_{k=1}^{L_2} \hat{Z}_{i,m,q-k+1, k} & \text{for } L_2 \leq q < \tilde{L}_2 - \tilde{L}_2 + 1 \\ \frac{1}{\tilde{L}_2 - \tilde{L}_2 + 1} \sum_{k=1}^{\tilde{L}_2 - \tilde{L}_2 + 1} \hat{Z}_{i,m,q-k+1, k} & \text{for } \tilde{L}_2 - \tilde{L}_2 + 1 \leq q \leq \tilde{L}_2 
\end{cases}
\]

for \( l = 1, \ldots, r \), for \( m = 1, \ldots, \frac{L_1}{L_i} \) and for \( q = 1, \ldots, L_2 \).

Observe, \( \hat{Z}_{i,m,q} \in \mathbb{R}^{S_L} \) for \( l = 1, \ldots, r \), for \( m = 1, \ldots, \frac{L_1}{L_i} \) and for \( q = 1, \ldots, L_2 \). Define the de-Hankelized 2D SSA components as

\[
\hat{Y}_l = \sum_{i=1}^{n} Y_i \quad \text{and} \quad \hat{Z}_l = \sum_{i=1}^{n} Z_i
\]

It is worth noting that the diagonal averaging method guarantees the exact perfect reconstruction of the image if the conventional SVD is employed in the SSA and the blocks taken from both \( Y \) and \( Z \) are shifted by one pixel both horizontally and vertically. However, the exact perfect reconstruction condition is unknown if the GSVD is employed in the SSA as well as the blocks taken from both \( Y \) and \( Z \) are not shifted by one pixel vertically. To address these issues, we have the following results:

**Theorem 1**

\[
\hat{Y} = \sum_{i=1}^{n} Y_i \quad \text{and} \quad \hat{Z} = \sum_{i=1}^{n} Z_i
\]

**Proof:**

Since \( \hat{Y} = u_{\mathbb{C}^{n}}X^{\theta} \)

\[
= \begin{bmatrix} U_{1} & \cdots & U_{n-1} \end{bmatrix} \theta_{1, \cdots, n-1} \begin{bmatrix} X_{1}^{\theta_{1, \cdots, n-1}} \\ \vdots \\ X_{n}^{\theta_{1, \cdots, n-1}} \end{bmatrix}
\]

\[
= \begin{bmatrix} U_{1} & \cdots & U_{n-1} \end{bmatrix} \begin{bmatrix} \theta_{1, \cdots, n-1}X_{1}^{\theta_{1, \cdots, n-1}} \\ \vdots \\ \theta_{1, \cdots, n-1}X_{n}^{\theta_{1, \cdots, n-1}} \end{bmatrix}
\]

\[
= \begin{bmatrix} u_{1} & \cdots & u_{n-1} \end{bmatrix} \begin{bmatrix} X_{1}^{\theta_{1, \cdots, n-1}} \\ \vdots \\ X_{n}^{\theta_{1, \cdots, n-1}} \end{bmatrix} = \hat{Y}
\]

\[
= \begin{bmatrix} v_{1} & \cdots & v_{n} \end{bmatrix} \begin{bmatrix} X_{1}^{\theta_{1, \cdots, n}} \\ \vdots \\ X_{n}^{\theta_{1, \cdots, n}} \end{bmatrix} = \hat{Z}
\]

this completes the proof.

Theorem 1 reveals that the sum of the Hankelized 2D SSA components is equal to the trajectory matrix.

**Theorem 2**

\[
\hat{Y} = \sum_{i=1}^{n} \hat{Y}_i \quad \text{and} \quad \hat{Z} = \sum_{i=1}^{n} \hat{Z}_i
\]

**Proof:**

Since \( \hat{Y}_l = \begin{bmatrix} \hat{Y}_{l,1} & \cdots & \hat{Y}_{l,r} \end{bmatrix} \) for \( l = 1, \ldots, r \)

and for \( m = 1, \ldots, \frac{L_1}{L_i} \), as well as \( \hat{Y}_{i,m,p} = \begin{bmatrix} \hat{Y}_{i,m,p,1} & \cdots & \hat{Y}_{i,m,p,r} \end{bmatrix} \) for
for $m = 1, \ldots, \frac{\tilde{L}_1}{L_1}$ and for $p = 1, \ldots, \tilde{L}_2 - L_2 + 1$.

we have $\hat{Y}_{l,m} = \begin{bmatrix} \hat{Y}_{l,m,l_1} & \ldots & \hat{Y}_{l,m,l_2-1} \\ \vdots & \ddots & \vdots \\ \hat{Y}_{l,m,l_2-1} & \ldots & \hat{Y}_{l,m,l_2-1} \end{bmatrix}$ for $l = 1, \ldots, r_f$,

and for $m = 1, \ldots, \frac{\tilde{L}_1}{L_1}$. On the other hand, since $\tilde{Y} = \begin{bmatrix} \tilde{Y}_1 & \ldots & \tilde{Y}_l \\ \vdots & \ddots & \vdots \\ \tilde{Y}_l \end{bmatrix}$, we have $\bar{Y}_{m} = \frac{\tilde{L}_1}{L_1} \sum_{l=1}^{n} \tilde{Y}_{l,m}$ for $m = 1, \ldots, \frac{\tilde{L}_1}{L_1}$.

Besides, since $\bar{Y}_{m} = \begin{bmatrix} Y_{m,1} & \ldots & Y_{m,l_2-1} \\ \vdots & \ddots & \vdots \\ Y_{m,l_2-1} & \ldots & Y_{m,l_2} \end{bmatrix}$ for $m = 1, \ldots, \frac{\tilde{L}_1}{L_1}$, we have

$$\begin{bmatrix} Y_{m,1} & \ldots & Y_{m,l_2-1} \\ \vdots & \ddots & \vdots \\ Y_{m,l_2-1} & \ldots & Y_{m,l_2} \end{bmatrix} = \frac{\tilde{L}_1}{L_1} \sum_{l=1}^{n} \tilde{Y}_{l,m,l_2-1}$$

for $m = 1, \ldots, \frac{\tilde{L}_1}{L_1}$. Moreover, as

$$\hat{Y}_{l,m,q} = \begin{cases} \frac{1}{L_2} \sum_{k=1}^{L_2} \tilde{Y}_{l,m,q} & \text{for } 1 \leq q < L_2 \\ \frac{1}{L_2 - q + 1} \sum_{k=1}^{L_2 - q + 1} \tilde{Y}_{l,m,q} & \text{for } L_2 - q + 1 \leq q \leq L_2 \end{cases}$$

for $l = 1, \ldots, r_f$, for $m = 1, \ldots, \frac{\tilde{L}_1}{L_1}$ and for $q = 1, \ldots, \frac{\tilde{L}_2}{L_2}$.

Furthermore, as

$$\hat{Y} = \begin{bmatrix} \hat{Y}_{1,1} & \ldots & \hat{Y}_{1,l_2} \\ \vdots & \ddots & \vdots \\ \hat{Y}_{l_2-1,1} & \ldots & \hat{Y}_{l_2-1,l_2} \end{bmatrix} \quad \text{and} \quad \hat{Y}_{l} = \begin{bmatrix} \hat{Y}_{l,1} & \ldots & \hat{Y}_{l,l_2} \\ \vdots & \ddots & \vdots \\ \hat{Y}_{l,l_2-1} & \ldots & \hat{Y}_{l,l_2-1} \end{bmatrix}$$

for $l = 1, \ldots, r_f$, we have $\hat{Y} = \sum_{l=1}^{r_f} \hat{Y}_{l}$.

Likewise, since $\hat{Z}_{l,m} = \begin{bmatrix} \hat{Z}_{l,m,1} & \ldots & \hat{Z}_{l,m,l_2-1} \end{bmatrix}$ for $l = 1, \ldots, r_f$ and for $m = 1, \ldots, \frac{\tilde{L}_1}{L_1}$ as well as

$$\hat{Z}_{l,m,p} = \begin{bmatrix} \hat{Z}_{l,m,p,1} \\ \vdots \\ \hat{Z}_{l,m,p,l_2-1} \end{bmatrix}$$

for $l = 1, \ldots, r_f$, for $m = 1, \ldots, \frac{\tilde{L}_1}{L_1}$ and for $p = 1, \ldots, \tilde{L}_2 - L_2 + 1$.

we have

$$\hat{Z}_{l,m} = \begin{bmatrix} \hat{Z}_{l,m,1} & \ldots & \hat{Z}_{l,m,l_2-1} \\ \vdots & \ddots & \vdots \\ \hat{Z}_{l,m,l_2-1} & \ldots & \hat{Z}_{l,m,l_2-1} \end{bmatrix}$$

for $l = 1, \ldots, r_f$ and for $m = 1, \ldots, \frac{\tilde{L}_1}{L_1}$.

On the other hand, we have $\hat{Z} = \begin{bmatrix} \hat{Z}_1 & \ldots & \hat{Z}_l \\ \vdots & \ddots & \vdots \\ \hat{Z}_l \end{bmatrix}$.

we have

$$\hat{Z}_{l,m} = \begin{bmatrix} \hat{Z}_{l,m,1} & \ldots & \hat{Z}_{l,m,l_2-1} \\ \vdots & \ddots & \vdots \\ \hat{Z}_{l,m,l_2-1} & \ldots & \hat{Z}_{l,m,l_2-1} \end{bmatrix}$$

for $l = 1, \ldots, r_f$ and for $m = 1, \ldots, \frac{\tilde{L}_1}{L_1}$.

Besides, since $\hat{Z} = \begin{bmatrix} \hat{Z}_1 & \ldots & \hat{Z}_l \\ \vdots & \ddots & \vdots \\ \hat{Z}_l \end{bmatrix}$, we have

$$\hat{Z}_{l,m} = \begin{bmatrix} \hat{Z}_{l,m,1} & \ldots & \hat{Z}_{l,m,l_2-1} \\ \vdots & \ddots & \vdots \\ \hat{Z}_{l,m,l_2-1} & \ldots & \hat{Z}_{l,m,l_2-1} \end{bmatrix}$$

for $l = 1, \ldots, r_f$ and for $m = 1, \ldots, \frac{\tilde{L}_1}{L_1}$.

for $m = 1, \ldots, \frac{\tilde{L}_1}{L_1}$.

Moreover, as

$$\tilde{Z}_{l,m,q} = \begin{cases} \frac{1}{L_2} \sum_{k=1}^{L_2} \tilde{Z}_{l,m,q} & \text{for } 1 \leq q < L_2 \\ \frac{1}{L_2 - q + 1} \sum_{k=1}^{L_2 - q + 1} \tilde{Z}_{l,m,q} & \text{for } L_2 - q + 1 \leq q \leq L_2 \end{cases}$$

for $l = 1, \ldots, r_f$, for $m = 1, \ldots, \frac{\tilde{L}_1}{L_1}$ and for $q = 1, \ldots, \frac{\tilde{L}_2}{L_2}$.

Furthermore, as

$$Z = \begin{bmatrix} z_{1,1} & \ldots & z_{1,l_2} \\ \vdots & \ddots & \vdots \\ z_{l_2-1,1} & \ldots & z_{l_2-1,l_2} \end{bmatrix} \quad \text{and} \quad \tilde{Z} = \begin{bmatrix} \tilde{Z}_{1,1,1} & \ldots & \tilde{Z}_{1,1,l_2} \\ \vdots & \ddots & \vdots \\ \tilde{Z}_{l_2-1,1,1} & \ldots & \tilde{Z}_{l_2-1,1,l_2} \end{bmatrix}$$

for $l = 1, \ldots, r_f$, we have $Z = \sum_{l=1}^{r_f} \tilde{Z}_{l}$. This completes the proof.

Theorem 2 reveals that the sum of the de-Hankelized 2D SSA components is equal to the original image.

2.4. Selection of de-Hankelized 2D SSA components via binary linear programming
It is worth noting that the size of $Z$ is larger than that of Y, so it is difficult to define the objective function in the pixel domain to select $\hat{Z}_l$ for $l = 1, \ldots, r_2$. On the other hand, the selection of $\tilde{Z}_l$ for $l = 1, \ldots, r_2$ is performed in the frequency domain so that the high frequency contents of Z can be enhanced.

Since $\tilde{Z}_l$ for $l = 1, \ldots, r_2$ are processed in the block based manner and some spatial regions in Z contain more information than other spatial regions, the selection of $\tilde{Z}_l$ for $l = 1, \ldots, r_2$ is also performed in the block based manner. This can provide a higher flexibility for reconstructing Z.

For the conventional SSA, the de-Hankelized 2D SSA components are either selected or dropped. Hence, the selection problem is formulated as a binary programming problem. Let the $(m,n)^{th}$ block of $\tilde{Z}_l$ be $\hat{B}_{m,n,l}$ for $m = 1, \ldots, \frac{L_1}{L_l}$, for $n = 1, \ldots, \frac{L_2}{L_l}$ and for $l = 1, \ldots, r_2$.

Obviously, $\hat{B}_{m,n,l} \in \mathbb{R}^{L_1 \times L_2}$ for $m = 1, \ldots, \frac{L_1}{L_l}$ and for $l = 1, \ldots, r_2$. Let the $(u,v)^{th}$ element of $\hat{B}_{m,n,l}$ be $\hat{B}_{m,n,l,uv}$ for $u = 1, \ldots, L_1$, for $v = 1, \ldots, L_2$, for $m = 1, \ldots, \frac{L_1}{L_l}$, for $n = 1, \ldots, \frac{L_2}{L_l}$ and for $l = 1, \ldots, r_2$. Likewise, Y is also divided into the blocks. Let the $(m,n)^{th}$ block of Y be $B_{m,n}$ for $m = 1, \ldots, \frac{L_1}{L_l}$ and for $n = 1, \ldots, \frac{L_2}{L_l}$.

Obviously, $B_{m,n} \in \mathbb{R}^{L_1 \times L_2}$ for $m = 1, \ldots, \frac{L_1}{L_l}$ and for $n = 1, \ldots, \frac{L_2}{L_l}$. Let the $(u,v)^{th}$ element of $B_{m,n}$ be $B_{m,n,uv}$ for $u = 1, \ldots, L_1$, for $v = 1, \ldots, L_2$, for $m = 1, \ldots, \frac{L_1}{L_l}$ and for $n = 1, \ldots, \frac{L_2}{L_l}$. Let $b_{m,n,l}$ be the selection coefficient of $\hat{B}_{m,n,l}$ for $m = 1, \ldots, \frac{L_1}{L_l}$, for $n = 1, \ldots, \frac{L_2}{L_l}$ and for $l = 1, \ldots, r_2$. Let $b_{m,n} = [b_{m,n,1}, \ldots b_{m,n,r_2}]^T$ for $m = 1, \ldots, \frac{L_1}{L_l}$ and for $n = 1, \ldots, \frac{L_2}{L_l}$. Obviously, $b_{m,n} \in \{0,1\}^r$ for $m = 1, \ldots, \frac{L_1}{L_l}$ and for $n = 1, \ldots, \frac{L_2}{L_l}$. Let $W(\omega_1, \omega_2)$ for $(\omega_1, \omega_2) \in [-\pi,\pi] \times [-\pi,\pi]$ be a weighted function. Let $J(b_{m,n})$ be the objective function for selecting $\hat{B}_{m,n,l}$ for $m = 1, \ldots, \frac{L_1}{L_l}$, for $n = 1, \ldots, \frac{L_2}{L_l}$ and for $l = 1, \ldots, r_2$. Since most of $\hat{B}_{m,n,l}$ for $m = 1, \ldots, \frac{L_1}{L_l}$, for $n = 1, \ldots, \frac{L_2}{L_l}$ and for $l = 1, \ldots, r_2$ contain important information for reconstructing Z, most of $\hat{B}_{m,n,l}$ for $m = 1, \ldots, \frac{L_1}{L_l}$, for $n = 1, \ldots, \frac{L_2}{L_l}$ and for $l = 1, \ldots, r_2$ should be selected. In other words, let $v_{l_2} = [1 \cdots 1]^T$ be an $r_2 \times 1$ column vector. Then, the total number of the nonzero elements in $v_{l_2} - b_{m,n}$ should be very small for $m = 1, \ldots, \frac{L_1}{L_l}$ and for $n = 1, \ldots, \frac{L_2}{L_l}$. Therefore, we define $J(b_{m,n}) = \|v_{l_2} - b_{m,n}\|$ for $m = 1, \ldots, \frac{L_1}{L_l}$ and for $n = 1, \ldots, \frac{L_2}{L_l}$. Since $J(b_{m,n})$ involves the $L_0$ norm operator which is highly nonconvex, in general there is more than one global optimal solution. In this case, the global optimal solution is not unique. To avoid this difficulty, the convex relaxation is performed. We define $\tilde{J}(\hat{b}_{m,n}) = \|v_{l_2} - \hat{b}_{m,n}\|$ for $m = 1, \ldots, \frac{L_1}{L_l}$ and for $n = 1, \ldots, \frac{L_2}{L_l}$. In this case, the local optimal solution is unique and the obtained local optimal solution is the global optimal solution. On the other hand, in order to enforce Z to be close to Y, a constraint is imposed on the weighted difference between the frequency response of $B_{m,n}$ and that of $\hat{B}_{m,n,l}$ for $m = 1, \ldots, \frac{L_1}{L_l}$ and for $n = 1, \ldots, \frac{L_2}{L_l}$. It is worth noting the frequency response of $B_{m,n}$ and that of $\hat{B}_{m,n,l}$ are

$$E_{m,n}(\omega_1, \omega_2) = \frac{1}{\sqrt{L_1L_2}} \sum_{u=1}^{L_1} \sum_{v=1}^{L_2} \hat{B}_{m,n,uv} e^{-j[(u-1)\omega_1 + (v-1)\omega_2]} ,$$

and

$$f_{m,n,l}(\omega_1, \omega_2) = \frac{1}{\sqrt{SL_1L_2}} \sum_{u=1}^{SL_1} \sum_{v=1}^{SL_2} \hat{B}_{m,n,uv} e^{-j[(u-1)\omega_1 + (v-1)\omega_2]}$$

for $l = 1, \ldots, r_2$ and for $(\omega_1, \omega_2) \in [-\pi,\pi] \times [-\pi,\pi]$ , respectively. Define

$$F_{m,n}(\omega_1, \omega_2) = \left[ f_{m,n,1}(\omega_1, \omega_2) \cdots f_{m,n,r_2}(\omega_1, \omega_2) \right]^T$$

for $m = 1, \ldots, \frac{L_1}{L_l}$, for $n = 1, \ldots, \frac{L_2}{L_l}$ and for $(\omega_1, \omega_2) \in [-\pi,\pi] \times [-\pi,\pi]$. Then, the frequency response of

$$\sum_{l=1}^{r_2} b_{m,n,l} \hat{B}_{m,n,l}$$

is $F_{m,n}^T(\omega_1, \omega_2) b_{m,n}$ and the weighted difference between $E_{m,n}(\omega_1, \omega_2)$ and $F_{m,n}^T(\omega_1, \omega_2) b_{m,n}$ is

$$W(\omega_1, \omega_2) F_{m,n}^T(\omega_1, \omega_2) b_{m,n} - E_{m,n}(\omega_1, \omega_2)$$

for $m = 1, \ldots, \frac{L_1}{L_l}$, for $n = 1, \ldots, \frac{L_2}{L_l}$ and for $(\omega_1, \omega_2) \in [-\pi,\pi] \times [-\pi,\pi]$. However,
$F_m^r (\omega_1, \omega_2) b_{m,n} - E_m (\omega_1, \omega_2)$ is a complex valued function for $m = 1, \cdots, \frac{L_1}{L_2}$, and for $n = 1, \cdots, \frac{L_2}{L_2}$ and for $(\omega_1, \omega_2) \in [-\pi, \pi] \times [-\pi, \pi]$. Define the specifications on the upper bounds of the real parts and the imaginary parts of $W(\omega_1, \omega_2) F_m^r (\omega_1, \omega_2) b_{m,n} - E_m (\omega_1, \omega_2)$ be $\varepsilon$ for $m = 1, \cdots, \frac{L_1}{L_2}$, and for $n = 1, \cdots, \frac{L_2}{L_2}$ and for $(\omega_1, \omega_2) \in [-\pi, \pi] \times [-\pi, \pi]$. That is,

$$W(\omega_1, \omega_2) \left[ \text{real}\left( F_m^r (\omega_1, \omega_2) b_{m,n} - E_m (\omega_1, \omega_2) \right) \right] \leq \varepsilon$$

and

$$W(\omega_1, \omega_2) \left[ \text{imag}\left( F_m^r (\omega_1, \omega_2) b_{m,n} - E_m (\omega_1, \omega_2) \right) \right] \leq \varepsilon$$

for $m = 1, \cdots, \frac{L_1}{L_2}$, and for $n = 1, \cdots, \frac{L_2}{L_2}$ and for $(\omega_1, \omega_2) \in [-\pi, \pi] \times [-\pi, \pi]$. This is equivalent to

$$\begin{bmatrix}
W(\omega_1, \omega_2) \text{real}(F_m^r (\omega_1, \omega_2)) \\
-W(\omega_1, \omega_2) \text{real}(F_m^r (\omega_1, \omega_2)) \\
W(\omega_1, \omega_2) \text{imag}(F_m^r (\omega_1, \omega_2)) \\
-W(\omega_1, \omega_2) \text{imag}(F_m^r (\omega_1, \omega_2))
\end{bmatrix} b_{m,n} \leq \begin{bmatrix}
W(\omega_1, \omega_2) \text{real}(E_m (\omega_1, \omega_2)) \\
-W(\omega_1, \omega_2) \text{real}(E_m (\omega_1, \omega_2)) \\
W(\omega_1, \omega_2) \text{imag}(E_m (\omega_1, \omega_2)) \\
-W(\omega_1, \omega_2) \text{imag}(E_m (\omega_1, \omega_2))
\end{bmatrix} \varepsilon$$

for $m = 1, \cdots, \frac{L_1}{L_2}$, and for $n = 1, \cdots, \frac{L_2}{L_2}$ and for $(\omega_1, \omega_2) \in [-\pi, \pi] \times [-\pi, \pi]$. Let

$$A(\omega_1, \omega_2) = \begin{bmatrix}
W(\omega_1, \omega_2) \text{real}(F_m^r (\omega_1, \omega_2)) \\
-W(\omega_1, \omega_2) \text{real}(F_m^r (\omega_1, \omega_2)) \\
W(\omega_1, \omega_2) \text{imag}(F_m^r (\omega_1, \omega_2)) \\
-W(\omega_1, \omega_2) \text{imag}(F_m^r (\omega_1, \omega_2))
\end{bmatrix}$$

and

$$P(\omega_1, \omega_2) = \begin{bmatrix}
W(\omega_1, \omega_2) \text{real}(E_m (\omega_1, \omega_2)) \\
-W(\omega_1, \omega_2) \text{real}(E_m (\omega_1, \omega_2)) \\
W(\omega_1, \omega_2) \text{imag}(E_m (\omega_1, \omega_2)) \\
-W(\omega_1, \omega_2) \text{imag}(E_m (\omega_1, \omega_2))
\end{bmatrix} + \varepsilon$$

for $m = 1, \cdots, \frac{L_1}{L_2}$, and for $n = 1, \cdots, \frac{L_2}{L_2}$ and for $(\omega_1, \omega_2) \in [-\pi, \pi] \times [-\pi, \pi]$. Now, the selection of $\hat{b}_{m,n}$ for $m = 1, \cdots, \frac{L_1}{L_1}$, and for $n = 1, \cdots, \frac{L_2}{L_2}$ and for $l = 1, \cdots, r_2$ is formulated as the following binary programming problem:

Problem $(P_{m,n})$

$$\min_{b_{m,n}} \left\| \hat{b}_{m,n} - b_{m,n} \right\|,$$

subject to $A(\omega_1, \omega_2) b_{m,n} \leq P(\omega_1, \omega_2)$ for $m = 1, \cdots, \frac{L_1}{L_1}$, and for $n = 1, \cdots, \frac{L_2}{L_2}$ and for $(\omega_1, \omega_2) \in [-\pi, \pi] \times [-\pi, \pi]$.

It is worth noting that the constraint is a function of $\omega_1$ and $\omega_2$. Also, the frequency domain is a continuous set and a continuous set consists of an infinite element. Therefore, Problem $(P_{m,n})$ for $m = 1, \cdots, \frac{L_1}{L_1}$ and for $n = 1, \cdots, \frac{L_2}{L_2}$ is an infinite constrained optimization problem. In general, it is very challenging to guarantee that all these infinite number of constraints are satisfied. To address this difficulty, the frequency domain is sampled into a finite number of points. Let $K$ be the total number of sampling points in the frequency domain and these sampling points be $(\omega_{k,1}, \omega_{k,2})$ for $k = 0, \cdots, K - 1$. Then, Problem $(P_{m,n})$ is approximated by:

Problem $(\tilde{P}_{m,n})$

$$\min_{\tilde{b}_{m,n}} \left\| \tilde{b}_{m,n} - b_{m,n} \right\|,$$

subject to $A(\omega_{k,1}, \omega_{k,2}) b_{m,n} \leq P(\omega_{k,1}, \omega_{k,2})$ for $m = 1, \cdots, \frac{L_1}{L_1}$, and for $n = 1, \cdots, \frac{L_2}{L_2}$ and for $k = 0, \cdots, K - 1$.

Now, Problem $(\tilde{P}_{m,n})$ for $m = 1, \cdots, \frac{L_1}{L_1}$ and for $n = 1, \cdots, \frac{L_2}{L_2}$ can be casted as a linear programming problem and the solution can be found via the conventional bounce and bound method.

3. Experimental results

This work does not involve any human participant as well as any data from the human tissue and the animal. A set of color images with various types of contents is employed for the demonstration. In particular, the set of images include the pictures of a building, a toy, a natural landscape, an airplane, a tree and a baby. Here, these are the low resolution images. Therefore, their sizes are very small. In particular, the sizes of the image “building”, the image “toy”, the image “natural landscape”, the image “airplane”, the image “tree”, the image “baby” are chosen as $L_1 \times L_1 = 60 \times 48$, $L_1 \times L_2 = 64 \times 48$, $L_1 \times L_2 = 60 \times 40$, $L_1 \times L_2 = 60 \times 40$ and $L_1 \times L_2 = 60 \times 40$, respectively. Figure 1 shows some of these low resolution images.

(a)  (b)  (c)
Figure 1. The original low resolution images. (a) A building. (b) A toy. (c) A natural landscape. (d) An airplane. (e) A tree. (f) A baby.

To perform the super-resolution, the total numbers of the rows of the high resolution images are chosen as the double of those of the low resolution images. That is, \( S = 2 \). To generate the enlarged images, the triangular interpolation is employed. That is, the values of the elements in the inserted rows are the average values of the elements of the rows before and after the inserted rows.

In the algorithm, the block size is chosen as \( L_1 \times L_2 = 2 \times 2 \). Also, the frequency domain is sampled using 225 points. That is, \( K = 225 \). Moreover, the specifications on the upper bounds of the real parts and the imaginary parts of the constraint functions are set as \( \varepsilon = 70 \).

In order to compare our proposed method, a similar nonlinear adaptive approach is compared. In particular, the empirical mode decomposition (EMD) based method [10] is compared. Here, the problem of finding the high resolution image is formulated as an optimization problem in a similar manner. Figure 2 and Figure 3 show the super-resolution images obtained by our proposed approach and the EMD approach [10], respectively. It can be seen that our proposed approach can achieve the better super-resolution performances qualitatively.

Figure 2. The high resolution images obtained via our proposed method. (a) A building. (b) A toy. (c) A natural landscape. (d) An airplane. (e) A tree. (f) A baby.

Figure 3. The high resolution images obtained via the EMD approach [10]. (a) A building. (b) A toy. (c) A natural landscape. (d) An airplane. (e) A tree. (f) A baby.

4. Discussion

Since the proposed method yields an image with a higher resolution, the proposed method can be applied to the medical images so that the medical officers have the medical images with the higher resolutions to improve the accuracy of the diagnosis. However, as the proposed method requires the upsampling and the filtered images, it depends on the employed interpolation filter. To address this issue, the nonlinear and adaptive approaches such as the two dimensional singular spectrum analysis will be employed to generate the images with the larger sizes.

5. Conclusions

This paper proposes a joint 2D SSA with the GSVD and the binary linear programming based method for performing the super-resolution. In particular, the exact perfect reconstruction condition is derived when the GSVD is employed in the SSA as well as the blocks taken from both the original image and the enlarged image are not shifted by one pixel vertically. Moreover, as the Hankelized 2D SSA components of both the original image and the enlarged image would be similar to each others, the super-resolution image would be close to the original image.

List of abbreviations

| 2D | Two dimensional |
|----|-----------------|
| SSA | Singular spectrum analysis |
| GSVD | Generalized singular value decomposition |
| SVD | Singular value decomposition |
| EMD | Empirical mode decomposition |

Availability of data and material
All the images can be downloaded from the public domain.

Competing interests
There is no conflict of interest.

Fundings
This paper was supported partly by the National Nature Science Foundation of China (no. U1701266, no. 61671163 and no. 62071128), the Team Project of the Education Ministry of the Guangdong Province (no. 2017KCXTD011), the Guangdong Higher Education Engineering Technology Research Center for Big Data on Manufacturing Knowledge.
Patent (no. 501130144) and Hong Kong Innovation and Technology Commission, Enterprise Support Scheme (no. S/E/070/17).

Authors’ contributions
Ziyin Huang is responsible for the conceptualization, deriving the theory, implementation of the Matlab program and writing the draft paper. Bingo Wing-Kuen Ling is responsible for finding a funding to support this research, verification of the developed theory and revising the draft paper. Yui-Lam Chan and Huan Ye are responsible for checking the results.

Acknowledgements
The authors would like to thank for the reviewers to provide the valuable comments to improve the quality of this paper.

References
[1] Reuben A. Farrugia and Christine Guillemot, “A simple framework to leverage state-of-art single-image super-resolution methods to restore light fields,” online available in Signal Processing: Image Communication, vol. 80, 2020.
[2] Zhen Li, Qilei Li, Wei Wu, Jinglei Yang, Zuoyong Li and Xiaomin Yang, “Deep recursive up-down sampling networks for single image super-resolution,” online available in Neurocomputing, vol. 25, 2019.
[3] Charlotte Yuk-Fan Ho, Bingo Wing-Kuen Ling and Peter Kwong-Shun Tam, “Representations of linear dual-rate system via single SISO LTI filter, conventional sampler and block sampler,” IEEE Transactions on Circuits and Systems—II: Express Briefs, vol. 55, no. 2, pp. 168-172, 2008.
[4] Salvador Villena, Miguel Vega, Javier Mateos, Duska Rosenberg, Fionn Murtagh, Rafael Molina and Aggelos K. Katsaggelos, “Image super-resolution for outdoor digital forensics. Usability and legal aspects,” Computer in Industry, vol. 98, pp. 34-47, 2018.
[5] Dong Cheng and Kit-lan Kou, “FFT multichannel interpolation and application to image super-resolution,” Signal Processing, vol. 162, pp. 21-34, 2019.
[6] Defu Qiu, Shengxiang Zhang, Ying Liu, Jianqing Zhu and Lixin Zheng, “Super-resolution reconstruction of knee magnetic resonance imaging based on deep learning,” online available in Computer Methods and Programs in Biomedicine, 2019.
[7] Nina Golyandina, Anton Korobeynikov, Alex Shlemov and Konstantin Usevich, “Multivariate and 2D extensions of singular spectrum analysis with the rssa package,” Journal of Statistical Software, vol. 67, no. 2, pp. 1-78, 2015.
[8] Alessandro Buccini, Mirjeta Pasha and Lothar Reichel, “Generalized singular value decomposition with iterated Tikhonov regularization,” online available in Journal of Computational and Applied Mathematics, vol. 7, 2019.
[9] Yiqiao Cai, Jiahai Wang, Jian Yin and Yalan Zhou, “Memetic clonal selection algorithm wit EDA vaccination for unconstrained binary quadratic programming problems,” Expert Systems with Applications, vol. 38, no. 6, pp. 7817-7827, 2011.
[10] Andrey S. Krylov, Andrey V. Nasonov and Dmitry V. Sorokin, “Face image super-resolution from video data with non-uniform illumination,” Proceedings of 18th International Conference on Computer Graphics, GraphiCon, pp. 150-155, 2008.