

## Article

# Progressive Sample Processing of Band Selection for Hyperspectral Image Transmission

Keng-Hao Liu <sup>1,\*</sup>, Shih-Yu Chen <sup>2</sup> , Hung-Chang Chien <sup>1</sup> and Meng-Han Lu <sup>1</sup>

<sup>1</sup> Department of Mechanical and Electro-Mechanical Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan; m043020080@student.nsysu.edu.tw (H.-C.C.); m053020108@student.nsysu.edu.tw (M.-H.L.)

<sup>2</sup> Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin 64002, Taiwan; sychen@yuntech.edu.tw

\* Correspondence: keng3@mail.nsysu.edu.tw; Tel.: +886-7-5252000-4220

Received: 18 December 2017; Accepted: 16 February 2018; Published: 26 February 2018

**Abstract:** Band selection (BS) is one of the important topics in hyperspectral image (HSI) processing. Many types of BS algorithms were proposed in the last decade. However, most of them were designed for off-line use. They can only be used with pre-collected data, and are sometimes ineffective for applications that require timeliness, such as disaster prevention or target detection. This paper proposes an online BS method that allows us obtain instant BS results in a progressive manner during HSI data transmission, which is carried out under band-interleaved-by-sample/pixel (BIS/BIP) format. Such a revolutionary method is called progressive sample processing of band selection (PSP-BS). In PSP-BS, BS can be done recursively pixel by pixel, so that the instantaneous BS can be achieved without waiting for all the pixels of an image. To develop a PSP-BS algorithm, we proposed PSP-OMPBS, which adopted the recursive version of a self-sparse regression BS method (OMPBS) as a native algorithm. The experiments conducted on two real hyperspectral images demonstrate that PSP-OMPBS can progressively output the BS with extremely low computing time. In addition, the convergence of BS results during transmission can be further accelerated by using a pre-defined pixel transmission sequence. Such a significant advantage not only allows BS to be done in a real-time manner for the future satellite data downlink, but also determines the BS results in advance, without waiting to receive every pixel of an image.

**Keywords:** band selection (BS); progressive sample processing (PSP); real-time processing

## 1. Introduction

Due to the use of hundreds of spectral bands, hyperspectral imaging (HSI) generally has enormous data volume and contains vast amount of information. This special characteristic results in several issues. First, the inter-band correlation of HSI is very high, and adjacent bands may contain redundant spectral information. This would lead to the well-known problem called the “curse of dimensionality” in data analysis. Second, the computational complexity of processing HSI data is very high. Third, storing HSI data usually requires large amount of disc space. Finally, transmitting HSI data requires higher bandwidths. Under such circumstances, removing partial data without significant loss of an image’s spectral information is necessary. One commonly used approach is band selection (BS) [1]. BS takes advantage of such high-band correlation to remove the redundant bands, in order to achieve a wide range of applications, such as dimensionality reduction, data storage, data transmission, target detection, and classification.

Many different types of BS algorithms [2–15] have been proposed in the past two decades. Most of them make an assumption that the BS problem is an optimization problem, which maximizes or minimizes a pre-defined objective function that can measure the amount of information or inter-band

redundancy contained by the currently selected bands. For instance, Keshava et al. [2] adopts common distance measures, such as Euclidean distance and a spectral angle mapper, to measure the similarity of bands. Du et al. [3] uses linear regression and orthogonal subspace projection to sequentially select a new band based on the complementary vector space. Chang et al. [4] regards a spectral band as a signal, and designed a BS method based on a constrained energy minimization algorithm. Martínez-Usó et al. [5] introduces a hierarchical clustering-based method to select bands from the band groups that were pre-clustered by a specific method, with some informational measures. In addition, there has been a lot of research that adopts the perspective of information theory. The works in [6–8] adopt mutual information as the similarity index to find the maximum information spanned by the bands. In recent years, sparse regression models were used to perform BS. For instance, Sun et al. [9–11] uses a self-sparse regression (SSR) model to select bands. In an SSR problem, finding a new basis is equivalent to selecting the most representation bands in an HSI image. Lai et al. [12] proposed a SSR-based BS method, which adopts an orthogonal matching pursuit (OMP) algorithm to sequentially find the next band in an efficient manner. For the purpose of HSI data transmission, Chang et al. [13,14] proposed a progressive BS (PBS) method for spectral unmixing. In PBS, the bands are first prioritized by a certain criterion, and are transmitted by the order of prioritization. Du et al. [15] proposed a BS-based dimensional reduction method for change detection in multi-temporal hyperspectral images. Except for the above-mentioned work, there were still various kinds of BS research published in the literature [16–25].

Since many existing HSI algorithms were designed to deal with off-line problems, they usually require more computing time for pursuing extremely high accuracy. Undoubtedly, those sophisticated methods may not be appropriate for dealing with timeliness applications. Hence, developing real-time and efficient approaches is urgently needed. Based on that point, some research started to develop on-board computing approaches [26–29], where the data compression could be carried out on satellites to reduce the demand of downlink bandwidth, focusing additionally on real-time image classification with the aid of GPU acceleration [30,31].

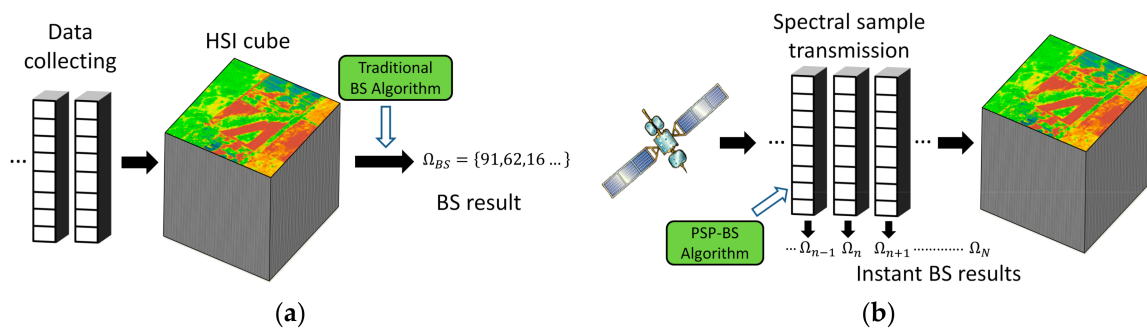
On the other hand, processing HSI data in a real-time progressive fashion under the data transmission held from satellite to ground station has become another attractive topic. The term “progressive processing” for HSI data originated from [32]. According to [32], the term refers to a type of signal processing method. It decomposes entire data processing into a finite number of stages, and processes data progressively stage by stage, in the sense that the results obtained by previous stages can be used to update or improve the results to be processed in subsequent stages. Recently, the concept of progressive processing was utilized to process HSI data. For example, in order to achieve real-time spectral unmixing during band transmission, a revolutionary concept called progressive band processing (PBP) [33], was developed. PBP assumes that the HSI data is transmitted by band sequential (BSQ) format, and processes the image immediately after receiving a new band. Under the PBP framework, many common topics, such as spectral unmixing, target detection, and anomaly detection could be realized [33–35]. Unlike the traditional algorithms, which can only be implemented on the collected data, PBP makes use of recursive processing, to instantly process the current data segment. More specifically, PBP methods preserve the past information obtained at previous stage, and use it to accelerate the computation at the current stage. The most attractive feature of PBP is that the computing time will not significantly increase as the growth of number of received bands. Therefore, PBP algorithms have significant potential for the uses of applications in data transmission and communication.

Based on the importance of BS, and the urgent demand of real-time algorithms, this paper proposes a real-time progressive BS method for applications related to data transmission. To our best knowledge, there were no BS methods proposed for this purpose before. In our case, we assume that HSI data is transmitted in band-interleaved-by-pixel (BIP) or band-interleaved-by-sample (BIS) format, and is processed pixel by pixel (sample by sample). According to [32], such a process operated on BIP/BIS transmission is called progressive sample processing (PSP). To sum up, we developed a novel

concept called progressive sample processing of band selection (PSP-BS), which combines BS with PSP. Figure 1 illustrates the difference of traditional BS and PSP-BS.

To realize PSP-BS, we had to choose an appropriate BS algorithm as the core of PSP-BS. In this paper, a self-sparse regression BS algorithm, called orthogonal matching pursuit-based BS (OMPBS) [12], has been adopted because (1) the OMPBS can sequentially find the most information-complementary bands from the image, without any prior knowledge or any complicated optimization scheme; and (2) The algorithm is easily re-formulated to fulfill recursive processing through mathematical decomposition of the optimization equations of OMPBS. In the later sections, we first introduce OMPBS, and then derive its recursive version, called recursive OMPBS (Re-OMPBS), as the key algorithm to realize PSP-BS. In addition, we make an assumption that the pixel transmission sequence (PTS) could be defined before transmission. We propose three PTSs to make BS results converge more quickly during the PSP-BS. We call Re-OMPBS combined with PTS PSP-OMPBS.

To evaluate the effectiveness of PSP-OMPBS, two real hyperspectral datasets were used in the experiments. We conducted a comparison of computing time and the corresponding BS accuracy at every time stage in a progressive manner. Land cover classification was also adopted to evaluate the quality of the selected bands. According to the experimental results, the computational efficiency of PSP-OMPBS is stably high so that the process could be carried out in real time. The BS accuracy can be improved early on by using particular PTSs.



**Figure 1.** An illustration of the difference of traditional band selection (BS) and progressive sample processing of band selection (PSP-BS). (a) The traditional BS algorithm is implemented on the collected hyperspectral image (HSI) data. (b) The PSP-BS is implemented in data sample transmission. In PSP-BS, the real-time BS monitoring can be achieved.

## 2. Orthogonal Matching Pursuit-Based BS (OMPBS)

The OMPBS [12] is a sequential BS method based on a self-sparse regression (SSR) model [36]. It aims to find a set of representative bands that can represent all bands based on the minimization of reconstructed errors in the SSR model, by using an orthogonal matching pursuit (OMP) algorithm [37]. Suppose  $L$  is the number of bands and  $N$  is number of pixels in a hyperspectral image. Let  $\mathbf{b}_i \in R^{N \times 1}$  presents the  $i$ th band vector. In an SSR model, both the observation matrix as well as the dictionary matrix are the set of band vectors denoted by  $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_L]$  and the coefficient matrix is denoted by  $\mathbf{C}$ . Obviously, the goal is to find the optimal coefficient matrix  $\mathbf{C}$  that minimizes the reconstructed error  $\|\mathbf{B} - \mathbf{BC}\|_F^2$ . Since only  $k$  bands need to be selected, we impose the sparsity constraint  $\|\mathbf{C}\|_{0,2} \leq p$  where  $l_0/l_2$  norm  $\|\mathbf{C}\|_{0,2}$  counts the number of the non-zero rows in  $\mathbf{C}$ . Thus, our sparse-BS optimization problem is

$$\hat{\mathbf{C}} = \arg \min \|\mathbf{B} - \mathbf{BC}\|_F^2 \dots \text{s.t. } \|\mathbf{C}\|_{0,2} \leq p, \quad (1)$$

where  $\|\mathbf{B} - \mathbf{BC}\|_F^2$  is the reconstructed error,  $\|\mathbf{C}\|_{0,2} \leq p$  is the sparsity constraint, and  $p$  is the sparsity level. The sub-optimization problem (find  $\mathbf{C}$ ) is solved by least square estimator  $\hat{\mathbf{Q}} = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T \mathbf{B}$ ,

where  $\mathbf{P}$  is temporary basis matrix composed by the candidate bands. The original OMPBS algorithm is composed of a doubly-nested loop. So, it is required to solve  $L!/(L-p)!$  times of (1), so the overall computation is time-consuming. To improve its efficiency, we revised the original algorithm by residual perspective. The revised algorithm (Algorithm 1) is shown as follows.

---

**Algorithm 1. OMPBS**


---

**Objective:** Find  $p$  representative bands for a HSI cube

**Input:**  $\mathbf{B}$  ( $N \times L$  matrix), sparsity =  $p$

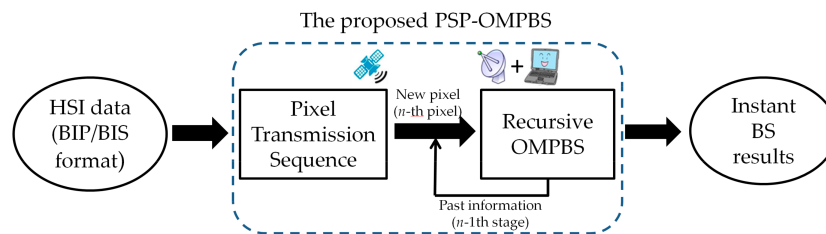
**Output:** Band indices set  $\Omega_p = \{b_1, b_2, \dots, b_p\}$

**Steps:**

1. Initialization: Set  $\Omega_0 = \emptyset$ ,  $\mathbf{R}_0 = \mathbf{B}$ ,  $\mathbf{P}_0 = \emptyset$ ,  $j = 1$ .
  2. At  $j$ -th iteration, calculate the row norm vector  $\mathbf{k} \in L \times 1$  of matrix  $\mathbf{R}_j^T \mathbf{R}_j$ .
  3. Find the row index  $b_j$  whose value is maximum at  $\mathbf{k}$ . Then set  $\mathbf{P}_j = [\mathbf{P}_{j-1} \ \mathbf{b}_{b_j}]$ ,  $\Omega_j = \Omega_{j-1} \cup \{b_j\}$ .
  4. Solve the sub optimization problem  $\hat{\mathbf{Q}}_j = \arg \min_{\mathbf{Q}_j} \|\mathbf{B} - \mathbf{P}_j \mathbf{Q}_j\|_F^2$  by  $\mathbf{Q}_j = (\mathbf{P}_j^T \mathbf{P}_j)^{-1} \mathbf{P}_j^T \mathbf{B}$ .
  5. Update the residual matrix  $\mathbf{R}_{j+1} = \mathbf{B} - \mathbf{P}_j \mathbf{Q}_j$ .
  6. If  $j = p$ , break; otherwise, set  $j \leftarrow j + 1$  and go to Step 2.
- 

### 3. Progressive Sample Processing-Based OMPBS (PSP-OMPBS)

The proposed PSP-OMPBS method is fully introduced in this section. It consists of two parts, as shown in Figure 2. The first part controls the data transmission order, in which the pixel transmission sequence will be formed. The second part is the real-time BS computing core, recursive OMPBS (Re-OMPBS). Section 3.1 introduces the derivation of Re-OMPBS, where we convert some steps in OMPBS to their recursive forms. Section 3.2 describes the condition for implementing recursive equations introduced in Section 3.1. Section 3.3 summaries the details of the Re-OMPBS algorithm. Finally, the formation of the pixel transmission sequence is introduced in Section 3.4.



**Figure 2.** Conceptual flowchart of progressive sample processing-based OMPBS (PSP-OMPBS). The center part is the proposed PSP-OMPBS method which includes two blocks: formation of the pixel transmission sequence (PTS), and implementation of the recursive OMPBS (Re-OMPBS) algorithm. The transmission is carried out from the PTS block to the Re-OMPBS block.

#### 3.1. Derivation of Recursive OMPBS (Re-OMPBS)

In this section, we aim to integrate OMPBS into PSP. According to the definition, PSP-BS can instantly acquire the current BS result during the data transmission by using two types of information: the processed information provided in previous stage (i.e., the  $(n-1)$ th stage), and the new information provided in current stage (i.e.,  $n$ th stage). More specifically, PSP-BS can immediately update the BS result when a new pixel (the  $n$ th pixel) is received. Assume that  $n$  presents the number of currently received pixels. We would like to utilize the past information obtained in the  $(n-1)$ th stage to reduce the computational burden of the  $n$ th stage. In doing so, the optimization process of OMPBS must be broken down.



### 3.1.1. Decomposition of the Least Square Estimator

In the following derivation, we assume that  $j$  is a fixed number and do not annotate it for each term. In the OMPBS algorithm, there are two calculations that can be decomposed. The first one is the least square estimator in Step 4:

$$\mathbf{Q}(n) = \left( \mathbf{P}(n)^T \times \mathbf{P}(n) \right)^{-1} \times \mathbf{P}(n)^T \times \mathbf{B}(n). \quad (2)$$

As we know, computing Equation (2) is required for each iteration  $j$  in the inner loop of OMPBS. This computation would be time-consuming when  $n$  is large. This would be unfavorable to progressive processing. To resolve this issue, we must convert Equation (2) to its recursive version. To do that, we define two matrices:

$$\mathbf{B}(n) = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1L} \\ b_{21} & b_{22} & \cdots & b_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nL} \end{bmatrix}, \quad \mathbf{P}(n) = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1p} \\ p_{21} & p_{22} & \cdots & p_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{np} \end{bmatrix}. \quad (3)$$

$\mathbf{B}(n)$  represents the total pixel data after the  $n$ th pixel is received, and  $\mathbf{P}(n)$  presents the data's temporary basis matrix, composed by selected band vectors. Let  $\mathbf{r}_n = [b_{n1}, b_{n2}, \dots, b_{nL}]$  be the new pixel and  $\tilde{\mathbf{p}}_n = [p_{n1}, p_{n2}, \dots, p_{np}]$ . Next, we define auxiliary matrix  $\mathbf{H}(n)$

$$\mathbf{H}(n) = \mathbf{P}(n)^T \times \mathbf{P}(n) = \begin{bmatrix} \mathbf{P}(n-1)^T & \tilde{\mathbf{p}}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{P}(n-1) \\ \tilde{\mathbf{p}}_n \end{bmatrix} = \mathbf{P}(n-1)^T \times \mathbf{P}(n-1) + \tilde{\mathbf{p}}_n^T \tilde{\mathbf{p}}_n. \quad (4)$$

Substituting (4) into (2), we get

$$\mathbf{Q}(n) = \left( \mathbf{P}(n-1)^T \times \mathbf{P}(n-1) + \tilde{\mathbf{p}}_n^T \tilde{\mathbf{p}}_n \right)^{-1} \times \mathbf{P}(n)^T \times \mathbf{B}(n). \quad (5)$$

Now use Woodberry's Identity [38]

$$\left( \mathbf{A} + \mathbf{c}\mathbf{c}^T \right)^{-1} = \mathbf{A}^{-1} - \frac{(\mathbf{A}^{-1}\mathbf{c})(\mathbf{c}^T\mathbf{A}^{-1})}{1 + \mathbf{c}^T\mathbf{A}^{-1}\mathbf{c}} \quad (6)$$

to decompose Equation (5). Let  $\mathbf{A} = \mathbf{P}(n-1)^T \mathbf{P}(n-1)$ ,  $\mathbf{c} = \tilde{\mathbf{p}}_n^T$ ,  $\mathbf{v} = \mathbf{A}^{-1}$  and  $\rho = \mathbf{c}^T \mathbf{A}^{-1} \mathbf{c}$ , and Equation (6) becomes

$$\mathbf{Q}(n) = \left( \mathbf{H}(n-1)^{-1} - \frac{\mathbf{v}_{n|(n-1)} \mathbf{v}_{n|(n-1)}^T}{1 + \rho_{n|(n-1)}} \right) \begin{bmatrix} \mathbf{P}(n-1) \\ \tilde{\mathbf{p}}_n \end{bmatrix}^T \begin{bmatrix} \mathbf{B}(n-1) \\ \mathbf{r}_n \end{bmatrix} \quad (7)$$

where  $\mathbf{v}_{n|(n-1)} = (\mathbf{P}_{n-1}^T \mathbf{P}_{n-1})^{-1} \tilde{\mathbf{p}}_n^T = \mathbf{H}(n-1)^{-1} \tilde{\mathbf{p}}_n^T$ ,  $\rho_{n|(n-1)} = \tilde{\mathbf{p}}_n^T (\mathbf{P}_{n-1}^T \mathbf{P}_{n-1})^{-1} \tilde{\mathbf{p}}_n = \tilde{\mathbf{p}}_n^T \mathbf{v}_{n|(n-1)}$ .

We expand Equation (7), and further simplify it by  $\mathbf{Q}(n-1) = \mathbf{H}(n-1)^{-1} \mathbf{P}(n-1)^T \mathbf{B}(n-1)$ , to get

$$\begin{aligned} \mathbf{Q}(n) &= \mathbf{H}(n-1)^{-1} \left( \mathbf{P}(n-1)^T \mathbf{B}(n-1) \right) + \tilde{\mathbf{p}}_n^T \mathbf{r}_n \\ &\quad - \frac{\mathbf{v}_{n|(n-1)} \mathbf{v}_{n|(n-1)}^T}{1 + \rho_{n|(n-1)}} \left( \mathbf{P}(n-1)^T \mathbf{B}(n-1) + \tilde{\mathbf{p}}_n^T \mathbf{r}_n \right) \\ &= \mathbf{Q}(n-1) + \mathbf{H}(n-1)^{-1} \tilde{\mathbf{p}}_n^T \mathbf{r}_n - \frac{\mathbf{v}_{n|(n-1)} \mathbf{v}_{n|(n-1)}^T}{1 + \rho_{n|(n-1)}} \left( \mathbf{P}(n-1)^T \mathbf{B}(n-1) + \tilde{\mathbf{p}}_n^T \mathbf{r}_n \right). \end{aligned} \quad (8)$$

Multiply the 3rd term of Equation (8) by  $\mathbf{H}(n-1)\mathbf{H}(n-1)^{-1}$ , and then re-organize it. Finally we obtain

$$\begin{aligned}\mathbf{Q}(n) &= \mathbf{Q}(n-1) - \frac{\mathbf{v}_{n|(n-1)}\mathbf{v}_{n|(n-1)}^T}{1+\rho_{n|(n-1)}}\mathbf{H}(n-1)\mathbf{Q}(n-1) \\ &\quad + \left(\mathbf{H}(n-1)^{-1} - \frac{\mathbf{v}_{n|(n-1)}\mathbf{v}_{n|(n-1)}^T}{1+\rho_{n|(n-1)}}\right)\tilde{\mathbf{p}}_n^T\mathbf{r}_n \\ &= \left(\mathbf{I}_{L\times L} - \frac{\mathbf{v}_{n|(n-1)}\mathbf{v}_{n|(n-1)}^T}{1+\rho_{n|(n-1)}}\mathbf{H}(n-1)\right)\mathbf{Q}(n-1) \\ &\quad + \left(\mathbf{H}(n-1)^{-1} - \frac{\mathbf{v}_{n|(n-1)}\mathbf{v}_{n|(n-1)}^T}{1+\rho_{n|(n-1)}}\right)\tilde{\mathbf{p}}_n^T\mathbf{r}_n\end{aligned}\quad (9)$$

Equation (9) is the recursive version of Equation (2). In Equations (7)–(9),  $\mathbf{Q}(n-1)$  and  $\mathbf{H}(n-1)$  are the past information obtained at the  $(n-1)$ th stage;  $\mathbf{v}_{n|(n-1)}$  and  $\rho_{n|(n-1)}$  are the so-called innovation information [32], whose calculation is involved with both previous information and current information; and  $\tilde{\mathbf{p}}_n$  and  $\mathbf{r}_n$  are provided by the new incoming pixel. Once  $\mathbf{Q}(n)$  is obtained, we calculate  $\mathbf{H}(n) = \mathbf{H}(n-1) + \tilde{\mathbf{p}}_n^T\tilde{\mathbf{p}}_n$  for the use in the next stage. It should be noted that Equation (9) can only be used when the updating condition is met. This detail will be presented in Section 3.2.

### 3.1.2. Decomposition of the Residual Multiplication Term

On the other hand, in OMPBS, the other calculation that can be simplified is the residual multiplication term  $\mathbf{R}(n)_{j+1}^T\mathbf{R}(n)_{j+1}$  in Step 5 of OMPBS. Calculating this term would be time-consuming when  $n$  becomes large. According to OMPBS, the residual term is formed by

$$\mathbf{R}_{j+1}(n) = \mathbf{B}(n) - \mathbf{P}_j(n)\mathbf{Q}_j(n). \quad (10)$$

We expand  $\mathbf{R}(n)_{j+1}^T\mathbf{R}(n)_{j+1}$  in terms of Equation (10):

$$\begin{aligned}\mathbf{R}_{j+1}(n)^T \times \mathbf{R}_{j+1}(n) &= (\mathbf{B}(n) - \mathbf{P}_j(n)\mathbf{Q}_j(n))^T (\mathbf{B}(n) - \mathbf{P}_j(n)\mathbf{Q}_j(n)) \\ &= \mathbf{B}(n)^T\mathbf{B}(n) - \mathbf{Q}_j(n)^T\mathbf{P}_j(n)^T\mathbf{B}(n) - \mathbf{B}(n)^T\mathbf{P}_j(n)\mathbf{Q}_j(n) + \mathbf{Q}_j(n)^T\mathbf{P}_j(n)^T\mathbf{P}_j(n)\mathbf{Q}_j(n).\end{aligned}\quad (11)$$

There are four terms required to be computed in Equation (11). The first term is  $\mathbf{B}(n)^T\mathbf{B}(n)$ , which can be obtained by the following recursive equation

$$\mathbf{B}(n)^T\mathbf{B}(n) = \begin{bmatrix} \mathbf{B}(n-1)^T & \mathbf{r}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{B}(n-1) \\ \mathbf{r}_n \end{bmatrix} = \mathbf{B}(n-1)^T\mathbf{B}(n-1) + \mathbf{r}_n^T\mathbf{r}_n. \quad (12)$$

Similarly, the second term can be represented by

$$\mathbf{Q}_j(n)^T\mathbf{P}_j(n)^T\mathbf{B}(n) = \mathbf{Q}_j(n)^T \begin{bmatrix} \mathbf{P}_j(n-1)^T & \tilde{\mathbf{p}}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{B}(n-1) \\ \mathbf{r}_n \end{bmatrix} = \mathbf{Q}_j(n)^T \left( \mathbf{P}_j(n-1)^T\mathbf{B}(n-1) + \tilde{\mathbf{p}}_n^T\mathbf{r}_n \right), \quad (13)$$

where  $\mathbf{Q}_j(n)^T$  is supposed to be known by Equation (9). The third term  $\mathbf{B}(n)^T\mathbf{P}_j(n)\mathbf{Q}_j(n)$  is the transposition of the second term, so it is unnecessary to re-compute it. Finally, the fourth term can be calculated by

$$\mathbf{Q}_j(n)^T\mathbf{P}_j(n)^T\mathbf{P}_j(n)\mathbf{Q}_j(n) = \mathbf{Q}_j(n)^T\mathbf{H}_j(n)\mathbf{Q}_j(n) = \mathbf{Q}_j(n)^T \left( \mathbf{H}_j(n-1) + \tilde{\mathbf{p}}_n^T\tilde{\mathbf{p}}_n \right) \mathbf{Q}_j(n) \quad (14)$$

In conclusion, term  $\mathbf{R}(n)_{j+1}^T\mathbf{R}(n)_{j+1}$  can be quickly calculated by using Equations (11)–(14).

### 3.2. Condition for Applying Recursive Equations

Unlike related work about PBP [33–35], where the recursive calculation can be applied to each iteration of PBP, Equation (9) of Re-OMPBS can only be implemented under particular conditions. To understand it, we assume that  $\Omega_j(n)$  presents the BS result at the  $j$ th iteration in the  $n$ th stage, and the result was just obtained. Now we move to calculate  $\mathbf{Q}_j(n)$  for finding the residual. If the condition  $\Omega_j(n) = \Omega_j(n-1)$  holds, we can use Equation (9) to calculate  $\mathbf{Q}_j(n)$ . More specifically, if the selected bands' indices of the  $j$ th iteration of the previous stage are the same as the band indices of the  $j$ th iteration in the current stage, we can use the recursive equation to save computing time. This is simply because the sparse coefficient matrix  $\mathbf{Q}_j(n)$  can only be “updated” when the currently-selected bases (i.e., band indices) are exactly the same as the previous ones. More specifically, the information in the horizontal dimension of temporary matrices  $\mathbf{P}_j(n-1)$  and  $\mathbf{P}_j(n)$  are aligned so that the information of previous stage can be shared with the current stage. In the following, we call it the “updating condition”.

Satisfying the updating condition not only reduces computing time for  $\mathbf{Q}_j(n)$ , but also slightly accelerates Equations (12)–(14). One key of implementing Re-OMPBS is that we have to store the computing results obtained at the  $(n-1)$ th stage, such as terms  $\{\mathbf{Q}_{j-1}(n)\}_{j=1}^p$ ,  $\{\mathbf{H}_{j-1}(n)\}_{j=1}^p$ ,  $\mathbf{B}(n-1)^T \mathbf{B}(n-1)$ , and  $\{\mathbf{P}(n-1)^T \mathbf{B}(n-1)\}_{j=1}^p$ , for applying in Equation (9) and Equations (11)–(14), when the updating condition is held at the  $n$ th stage. One should note that meeting this condition is not necessary for Re-OMPBS. For those cases that do not meet the conditions, we directly use the original least square formula  $\mathbf{Q}_j(n) = (\mathbf{P}_j(n)^T \mathbf{P}_j(n))^{-1} \mathbf{P}_j(n)^T \mathbf{B}(n)$  to calculate  $\mathbf{Q}_j(n)$ .

One might think that the probability of reaching these conditions would be low. However, this is not the case. In fact, when the amount of received pixels  $n$  reaches a certain amount, a single new incoming pixel will not obviously influence the BS result. In other words, the BS result is expected to be consistent. This phenomenon will be demonstrated in our experiment section. Moreover, Re-OMPBS is regarded as the “approximate” version of OMPBS. Theoretically, there would be some tiny numerical errors accumulated in  $\mathbf{Q}_j(n)$  if the updating condition has been reached continuously for a long period. In this case, an occasional lack of meeting the updating condition exactly gives the chance to reset the numerical error to zero.

### 3.3. Re-OMPBS Algorithm

Based on the above-mentioned content, the Re-OMPBS algorithm (Algorithm 2) is listed below.

**Algorithm 2. Re-OMPBS**Objective: Find the BS of the  $n$ th stage during BIP/BIS transmissionInput:

1. Sparsity  $p$
2. Processed information:
  - Previous BS result:  $\Omega_p(n-1) = \{b_1^{(n-1)}, b_2^{(n-1)}, \dots, b_p^{(n-1)}\}$
  - Previous sparse coefficient matrices:  $\{\mathbf{Q}_{j-1}(n)\}_{j=1}^p$
  - Previous auxiliary matrix:  $\{\mathbf{H}_{j-1}(n)\}_{j=1}^p$
  - Other terms  $\mathbf{B}(n-1)^T \mathbf{B}(n-1)$  and  $\{\mathbf{P}(n-1)^T \mathbf{B}(n-1)\}_{j=1}^p$
3. New information:  $n$ -th pixel:  $\mathbf{r}_n = [r_{n1}, r_{n2}, \dots, r_{nL}]$

Output: Band indices  $\Omega_p(n) = \{b_1^{(n)}, b_2^{(n)}, \dots, b_p^{(n)}\}$ Steps:

1. Initialization: Set  $\Omega_0(n) = \emptyset$ ,  $\mathbf{P}_0(n) = \emptyset$ ,  $j=1$ ,  $\mathbf{R}_1(n) = \mathbf{B}(n-1)^T \mathbf{B}(n-1) + \mathbf{r}_n^T \mathbf{r}_n$ .
2. At the  $j$ th iteration, calculate the vector-wise norms of all rows in matrix  $\mathbf{R}(n)_j^T \mathbf{R}(n)_j$ , and denote it by column vector  $\mathbf{k}$ ,  $\mathbf{k} \in L \times 1$ .
3. Find the row index whose value is maximum is  $\mathbf{k}$ , denoted by  $b_j$ . Then set  $\mathbf{P}_j(n) = [\mathbf{B}_{j-1}(n) \quad \mathbf{b}_{b_j}(n)]$ ,  $\Omega_j(n) = \Omega_{j-1}(n) \cup b_j$ .
4. If  $\Omega_j(n) = \Omega_j(n-1)$ , using Equation (9) to calculate  $\mathbf{Q}_j(n)$ ; otherwise,  $\mathbf{Q}_j(n) = (\mathbf{P}_j(n)^T \mathbf{P}_j(n))^{-1} \mathbf{P}_j(n)^T \mathbf{B}(n)$ .
5. Update auxiliary matrix  $\mathbf{H}_j(n)$  by equation  $\mathbf{H}_j(n) = \mathbf{H}_j(n-1) + \tilde{\mathbf{P}}_n^T \tilde{\mathbf{P}}_n$ .
6. Update the residual term  $\mathbf{R}(n)_{j+1}^T \mathbf{R}(n)_{j+1}$  by Equations (11)–(14).
7. If  $j = p$ , break; otherwise,  $j \leftarrow j + 1$  and go to Step 2.

**3.4. Design of Pixel Transmission Sequence (PTS)**

The original pixel transmission order in BIP/BIS format is based on a horizontal line scan [35]. If we regard a hyperspectral image as a 2D matrix, the pixel transmission starts from the upper left corner. Once the first row has been transmitted, then it transmits the second row, and so on. However, using this sequence for Re-OMPBS might mean that the BS cannot converge until the almost all pixels are transmitted. Ideally, in a progressive processing, we might expect to see the final results as soon as possible. In this case, adopting an appropriate sampling technique in the spatial domain to select the representative pixels for early transmission is necessary. There are three pixel transmission sequences we considered in this paper.

**3.4.1. Original Band-Interleaved-by-Sample/Pixel (BIS/BIP) Sequence**

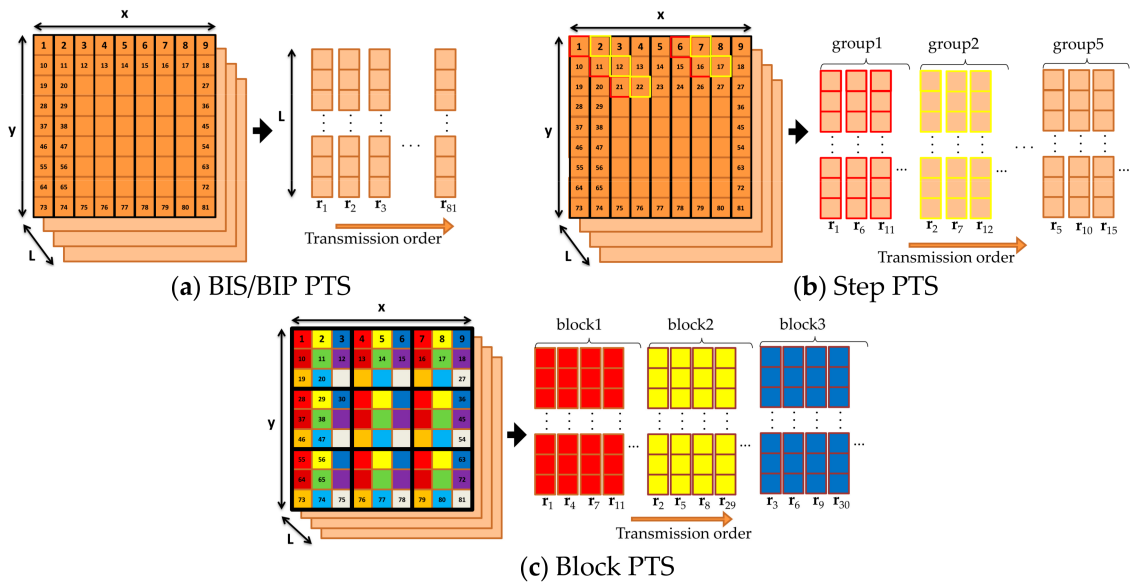
As mentioned above, the original pixel transmission sequence is the same as the storage sequence in the BIS/BIP format. Suppose the size of a hyperspectral image cube  $\mathbf{I}$  is  $x \times y \times L$ . The number of total pixels is  $N = x \times y$ . We reshape  $\mathbf{I}$  to an  $L \times N$  matrix form in row-major order,  $[\mathbf{r}_1 \mathbf{r}_2 \dots \mathbf{r}_N]_{L \times N}$ , where the column vectors are spectral pixels of  $\mathbf{I}$ . The BIS/BIP transmission sequence is defined by  $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\}$ . A simple illustration is shown in Figure 3a.

**3.4.2. Step Sequence**

The step method refers to uniformly selecting pixels in an original sequence with interval  $k$ . To do this, we uniformly segment pixel set  $[\mathbf{r}_1 \mathbf{r}_2 \dots \mathbf{r}_N]$  into  $k$  groups. For each transmission, one pixel is selected from each group in turn. If  $k = 5$ , the resulting transmission pixel sequence would be  $\{\mathbf{r}_1, \mathbf{r}_6, \mathbf{r}_{11}, \dots, \mathbf{r}_2, \mathbf{r}_7, \mathbf{r}_{12}, \dots\}$ , as shown in Figure 3b. As a result, the transmitted pixels are the samples uniformly picked from the pixel set. Theoretically, using such sequence would make the PSP-BS converge more quickly, because in this case fewer received pixels can represent the image better.

### 3.4.3. Block Sequence

Since the step method only considers image sampling in the horizontal direction, we further consider another method, which can consider both vertical and horizontal directions simultaneously, called the block method. The block method first splits the image into several  $b \times b$  square blocks, as shown in Figure 3c. Then the new pixel is picked from each block in turn. In such a design, the sampling process is performed on the full spatial domain. Theoretically, it would shorten convergence time of PSP-BS, too.



**Figure 3.** Illustration of the pixel transmission sequence of a  $9 \times 9$  hyperspectral image formed by different methods. (a) A band-interleaved-by-sample/pixel (BIS/BIP) sequence. (b) Step method with  $k = 5$ . (c) The block method with  $b = 3$ .

In the later sections, we call the PSP-OMPBS implemented with original BIS/BIP sequence, step sequence, and block sequence PSP-OMPBS, S-PSP-OMPBS, and B-PSP-OMPBS, respectively.

Finally, we summarize the distinct features of OMPBS and the proposed PSP-OMPBS in Table 1.

**Table 1.** The basic properties of orthogonal matching pursuit-based BS (OMPBS) and the proposed PSP-OMPBS.

	OMPBS	PSP-OMPBS (proposed)
Description	A traditional BS method which is implemented on a pre-obtained image data.	A PSP-BS method. It is used in the moment of spectral sample transmission or collecting.
Core algorithm	OMP-BS [37]	Re-OMPBS (Algorithm 2)
Use of past information	No	Yes
Use of PTS	No	Yes (Original/Step/Block)
Least square estimator used in the optimization	$\mathbf{Q}_j = (\mathbf{P}_j^T \mathbf{P}_j)^{-1} \mathbf{P}_j^T \mathbf{B}$	Recursive Equation (9)
Residual term $\mathbf{R}_j^T \mathbf{R}_j$ calculation	Direct computation	Recursive Equations (11)–(14)
Capable for progressive processing	No, the computing time will increase with the data volume.	Yes



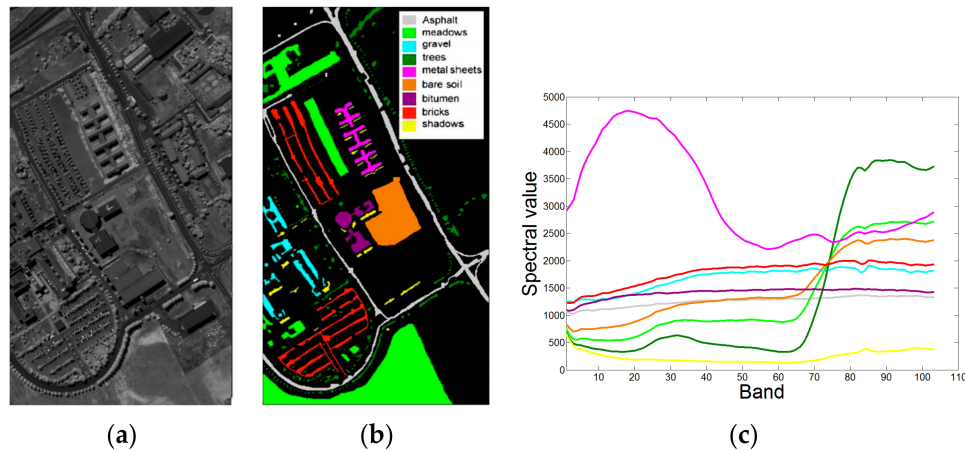
## 4. Experiments

In this section, the proposed PSP-OMPBS is tested on two publicly-available real hyperspectral images. To evaluate the performance of PSP-OMPBS, we conduct three different studies, including BS accuracy analysis, land cover classification, and computational efficiency.

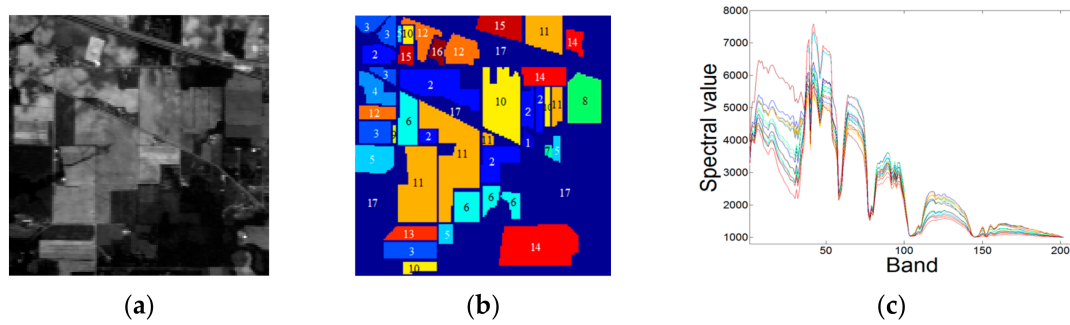
### 4.1. Hyperspectral Dataset and Experimental Setting

The first dataset used in the experiments was a real hyperspectral image, which was collected by the reflective optics system imaging spectrometer (ROSIS) optical sensor over an urban area of the University of Pavia. The Pavia image measures  $610 \times 340$ , with very high spatial resolution of about 1.3 m per ground pixel. The original data contains 115 spectral bands, with a spectral range from 0.43 to  $0.86 \mu\text{m}$ . After removing 12 noisy bands, the remaining 103 bands were used for the experiments. Figure 4a–c respectively show the image scene of the 50th band, the geometric locations of all target classes, and the corresponding spectral signatures. According to the ground truth in Figure 5b, there are 9 classes in this image scene, consisting of several urban targets, such as vegetation, soil, and roads.

The second dataset used in our experiments is another real hyperspectral image data, Purdue's Indiana Indian Pine test site, which was collected by the airborne visible-infrared imaging spectrometer (AVIRIS) system. It has been extensively studied in the literature and provides a good candidate for those who are interested in algorithm design and analysis. It has a 20 m spatial resolution and a 10 nm spectral resolution in the range of  $0.4\text{--}2.5 \mu\text{m}$  with size  $145 \times 145$  pixel vectors, taken from an area of mixed agriculture and forestry in Northwestern Indiana, U.S. It was recorded in June 1992 with 220 bands, among which bands 104–108 and 150–162 were removed, whereas the remaining 202 bands were retained. Figure 5a,b shows the image of band 20 and the ground truth map, respectively. The ground truth map contains 16 crops classes and one background class.



**Figure 4.** ROSIS image scene: University of Pavia. (a) Band 80. (b) Ground truth map for nine classes. (c) Spectral signatures of nine classes.



**Figure 5.** AVIRIS image scene: Purdue Indiana Pine test site. (a) Band 20. (b) Ground truth map. (c) Spectral signatures of 17 classes.

According to the introduction, the development of PSP-BS is not for achieving outstanding BS for a particular data analysis, such as image classification or spectral unmixing, but for the real-time BS monitoring during data transmission. In this case, the experiment is conducted based on self-comparison, instead of comparing with state-of-the-art methods. To evaluate the BS accuracy and efficiency of PSP-OMPBS, we still can adopt OMPBS as the compared method (baseline).

In the following progressive experiments, we set  $n = 100$ –207,400 for the Pavia data, and  $n = 100$ –21,025 for the Purdue data. For the  $p$  value, we set 16 for the Pavia data and 34 for the Purdue data, according to the virtual dimensionality (VD) algorithm [39] with false alarm 0.01. Three PTS methods mentioned in Section 3.4 were adopted for PSP-OMPBS methods. We empirically set  $k$  as 200 and 100 and  $b$  as 50 and 25 for the Pavia and Purdue data, respectively. Table 2 lists the parameters used in the experiment.

**Table 2.** The parameters used in the experiments.

Parameter	Pavia Data	Purdue Data
$N$ (number of total pixel)	207,400	21,025
$p$ (number of selected bands)	16	34
$k$ (step size of step PTS)	200	100
$b$ (block size of block PTS)	50	25

#### 4.2. BS Accuracy Analysis

The accuracy of the instant BS result during data transmission is an index to evaluate the effectiveness of PSP-OMPBS. Theoretically, the BS would gradually converge to the final BS result (i.e., the result of OMPBS performed on the complete image cube) over time. To observe this phenomenon, we use the results of OMPBS implemented on both images, as the ground truths. For instance, Pavia's ground truth is  $\Omega_{\text{OMPBS}}^{\text{Pavia}}(N) = \{91, 62, 16, 1, 34, 3, 73, 105, 5, 46, 85, 8, 83, 2, 11, 78\}$ . To evaluate BS correctness, the accuracy index (ACC) is defined by

$$\text{ACC}(n) = \frac{||\Omega_{\text{PSP-OMPBS}}(n) \cap \Omega_{\text{OMPBS}}(n)||_0}{p}. \quad (15)$$

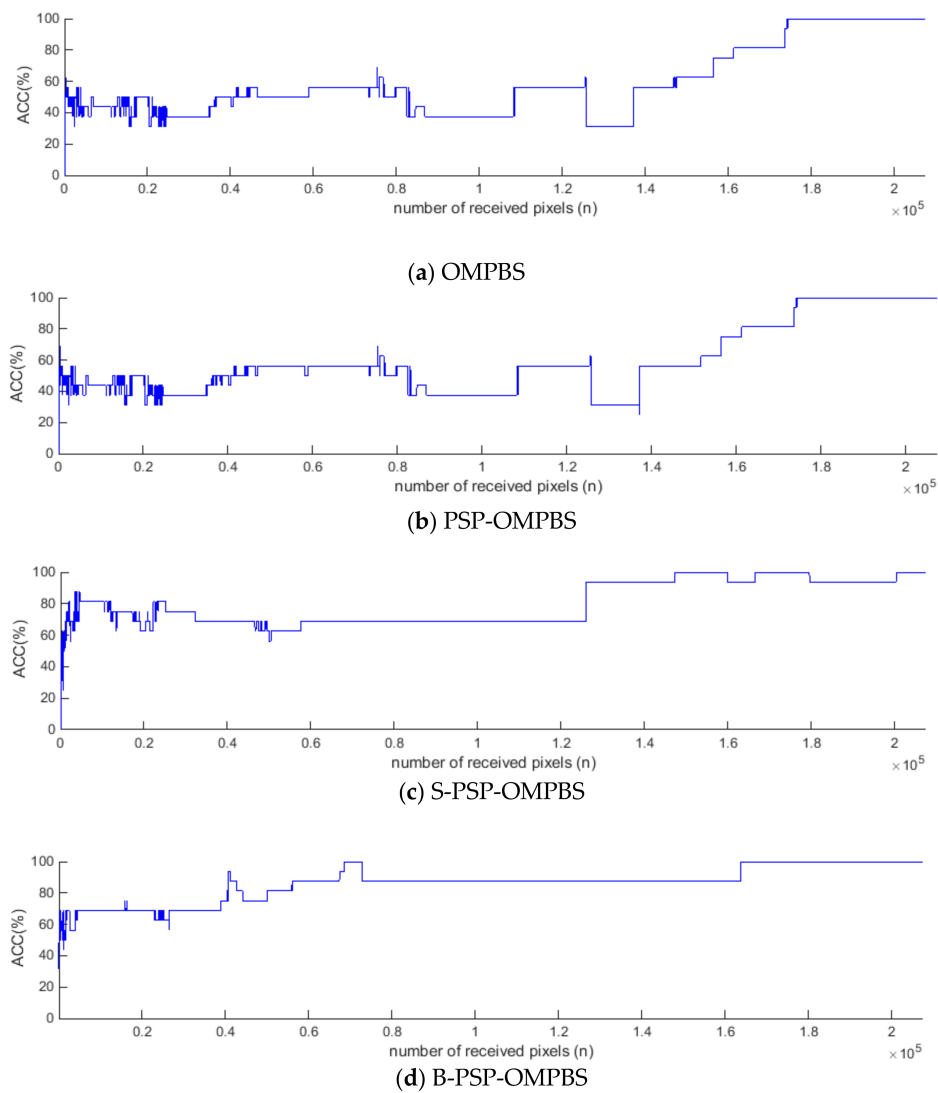
Base on Equation (15),  $\text{ACC}(n)$  indicates the ratio of the target bands in overall  $p$ -selected bands, when the number of received pixels is  $n$ . Higher ACC values mean higher BS correctness. Figure 6a–d plots the ACC curves of OMPBS, PSP-OMPBS, S-PSP-OMPBS, and B-PSP-OMPBS, all implemented on the Pavia data. Several observations can be seen:

1. Comparing Figure 6a with Figure 6b, the tendency of the ACC curves of OMPBS and PSP-OMPBS are nearly the same. This implies that the derivation of Re-OMPBS is correct. In fact, these two curves are still slightly different in some regions. Based on our study, the difference

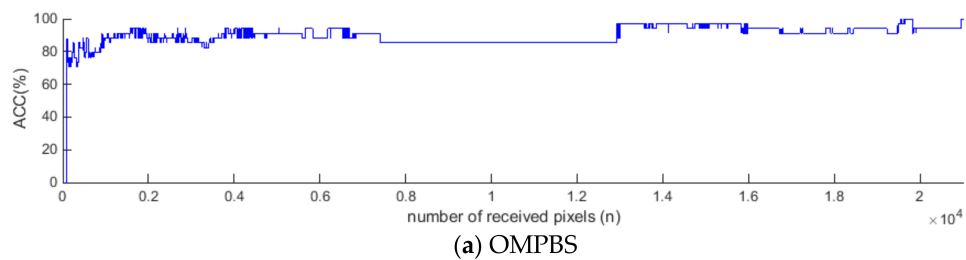
was caused by the numerical errors produced by using the recursive equations derived from Woodbury's Identity.

2. In the OMPBS and PSP-OMPBS curves, the ACC values kept at 40~60% when  $n$  is less than  $1.5 \times 10^5$ . This is mainly due to the fact that these values were generated by using the original BIS/BIP PTS. The BS results could not stabilize quickly.
3. The ACC results of using step sequence are shown in Figure 6c. We find that the ACC values located between 60~80% at  $n \in [2000, 125,000]$ . After receiving  $1.25 \times 10^5$  pixels, the ACCs increased to 90%. In other words, using the PTS formed by uniform sampling drastically accelerated the speed of convergence. It suggests that using full pixels is not necessary to obtain the correct BS result using PSP-OMPBS. The nearly-correct BS result could be obtained during transmission.
4. Using a block pixel transmission sequence also accelerated the BS convergence. Figure 6d shows the corresponding ACC results of B-PSP-OMPBS. Similar conclusions can be drawn. It only requires 40,000 pixels to reach 65~70% accuracy. After receiving 60,000 pixels, the ACC grew to over 80%. The overall ACC performance is undoubtedly better than OMPBS, PSP-OMPBS, and even S-PSP-OMPBS.
5. In Figure 6a–d, it can be seen that the ACC curves remain flat in some segments. In those regions, the BS results kept the same. That is, the new incoming pixels did not change the BS results.
6. All the curves reached 100% at  $n = N = 207,400$ .

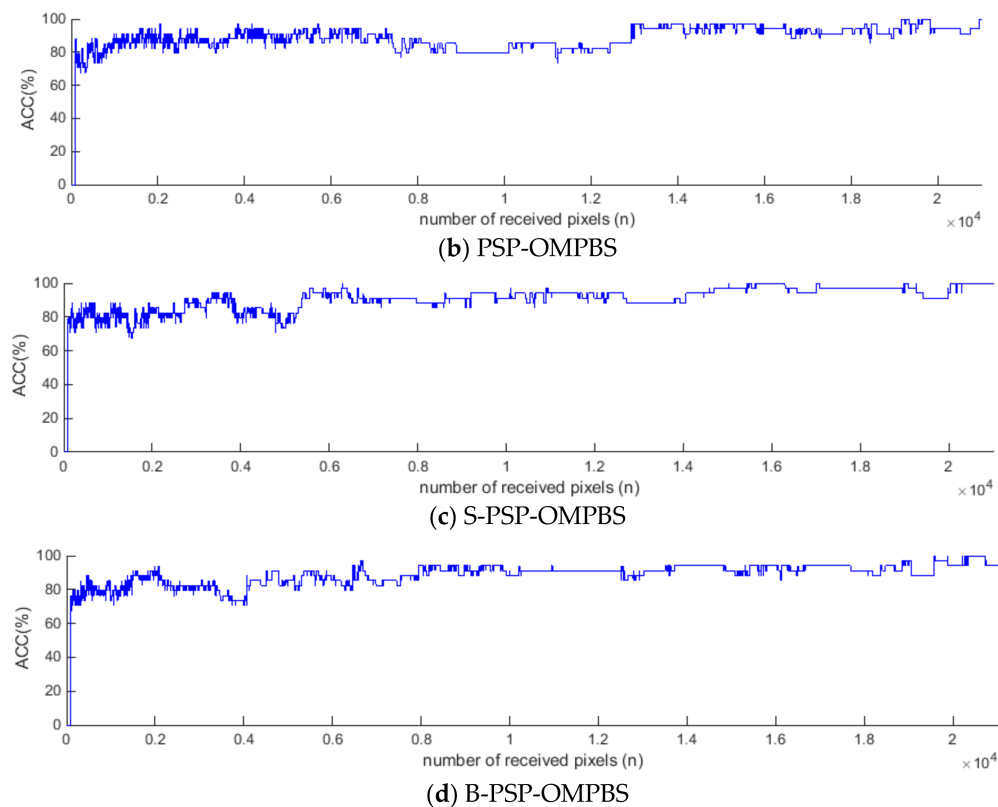
In the remote sensing community, the Purdue image was considered as a tough image for algorithm evaluation because of its noise and heavily-mixed properties. Figure 7a–d plots the ACC curves of OMPBS, PSP-OMPBS, S-PSP-OMPBS, and B-PSP-OMPBS, implemented on the Purdue data, respectively. Theoretically, the larger  $p$  and the special properties of the Purdue image, may lead to different consequences. Compared with the Pavia case, we have several interesting findings. Firstly, the oscillation of the four ACC curves is more notable. This is probably due to the noise property of the Purdue image. Secondly, the tendency of OMPBS and PSP-OMPBS curves are the same. This is in accordance with our expectations, because both methods are essentially the same. However, their ACC values are slightly different at some places. The PSP-OMPBS curve of the Purdue image is more unstable and lower at middle region. We think such a phenomenon is caused by the larger setting of  $p$ . The greater  $p$  is set, the more numerical errors will be accumulated while selecting later bands at each  $n$ . Thirdly, and most importantly, we found that the advantages of using sampled sequences (i.e., step and block) is not obvious in the Purdue experiment. We think the strange phenomenon is caused by the homogeneity of spectral profiles of the 16 ground classes. In this case, using any kind of PTS may obtain analogous pixel sets, so that the BS results are similar. From this point of view, in the case of the Purdue data, the OMPBS selects bands based more on spectral variation, instead of spatial/geographical variation. Despite this issue, it is still observed that S-PSP-OMPBS and B-PSP-OMPBS outperformed OMPBS a little bit with regard to ACC stability. The ACC trends of S-PSP-OMPBS and B-PSP-OMPBS did not drop after they reached over 80%, particularly at  $n \in [6000, 13,000]$ . In conclusion, the ACC performance of PSP-OMPBS methods varies with the properties of images. Using PSP-OMPBS on the images with lower noise seemed to produce more stable ACC curves. The sampled pixel sequences were more suitable for the images with larger heterogeneity in the spatial domain. The spectral similarity of ground classes was another important factor. Finally, we can find the curve of B-PSP-OMPBS did not end at 100% when  $n = N = 21,025$ . This was caused by the numerical error.



**Figure 6.** Accuracy index (ACC) curves of different methods implemented on ROSIS Pavia dataset: (a) OMPBS, (b) PSP-OMPBS, (c) step sequence PSP-OMPBS (S-PSP-OMPBS), and (d) block sequence PSP-OMPBS (B-PSP-OMPBS).



**Figure 7.** Cont.



**Figure 7.** ACC curves of different methods implemented on the AVIRIS Purdue dataset: (a) OMPBS, (b) PSP-OMPBS, (c) step sequence PSP-OMPBS (S-PSP-OMPBS), and (d) block sequence PSP-OMPBS (B-PSP-OMPBS).

Tables 3 and 4 list the corresponding BS results of the Pavia and Purdue experiments, where  $n$  is selected by 50% of  $N$  for each case. The last row shows the BS ground truth. It can be seen that the BS accuracies of PSP-OMPBSs are better than OMPBS at the 50% middle of transmission.

**Table 3.** List of the 16 bands, selected by four different methods, implemented on the Pavia data at  $n = 103,700$ .

Method	Selected Bands	ACC
OMPBS	86,20,63,1,103,3,45,73,6,32,83,9,4,55,12,2	37.5
PSP-OMPBS	86,20,63,1,103,3,45,73,6,32,83,9,4,55,12,2	37.5
S-PSP-OMPBS	91,62,15,1,33,3,73,103,5,46,85,7,2,83,10,22	68.75
B-PSP-OMPBS	91,62,16,1,34,3,73,103,5,47,46,48,85,8,83,2	87.5
OMPBS (full pixels)	91,62,16,1,34,3,73,103,5,46,85,8,83,2,11,78	N/A

**Table 4.** List of the 34 bands, selected by four different methods, implemented on the Purdue data at  $n = 10,513$ .

Method	Selected Bands	ACC
OMPBS	42,29,2,89,6,35,1,53,117,3,4,15,38,5,32,72,41,104,7,39,34,8,37,43,9,44,97,63,45,10,33,36,46,40	85.31
PSP-OMPBS	42,29,2,89,6,35,1,53,117,3,4,15,16,38,5,32,72,41,104,7,39,34,8,37,43,44,45,97,9,63,33,10,46,36	85.29
S-PSP-OMPBS	42,29,3,89,35,1,2,70,9,118,4,38,5,32,14,44,16,6,144,43,39,97,34,7,45,37,63,8,41,48,52,47,50,10	88.24



Table 4. Cont.

Method	Selected Bands	ACC
B-PSP-OMPBS	42,29,2,89,35,6,1,70,118,3,38,4,14,5,32,43,144,44, 7,39,34,97,8,45,11,48,37,63,41,100,46,9,33,17	91.18
OMPBS (full pixels)	42,29,3,89,35,1,2,70,9,118,4,38,5,32,15,48,6,144, 41,34,39,97,7,37,8,47,45,63,44,10,43,33,46,36	N/A

#### 4.3. Land Cover Classification

Image classification is a common procedure to evaluate the effectiveness of a BS approach. For each dataset, we implemented supervised classification using a PSP-OMPBS-selected band  $\Omega_P(n)$ , with a support vector machine (SVM) classifier [40]. The radial basis function (RBF) kernel was adopted for SVM with the selected parameter  $[\sigma/4, \sigma/2, \sigma, 2\sigma, 4\sigma]$  where  $r$  is calculated by the average pairwise distance among training data  $\sigma = E||x_i - x_j||$ . The training samples were obtained by randomly selecting 10% of data samples in each class, according to the ground truth maps shown in Figures 4b and 5b. The other 90% of samples were used as the test samples.

For measuring the classification performance, average accuracy (AA), overall accuracy (OA), and Cohen's kappa coefficient were used as the performance metrics. Since PSP-OMPBS provided 207,400 and 21,025 BS results for two datasets respectively, we simply chose the BS results using 20%, 40%, 60%, 80%, and 100% of  $N$  for the experiment. Three PSP-OMPBS algorithms were considered. The classification result of using full bands was also used for the comparison.

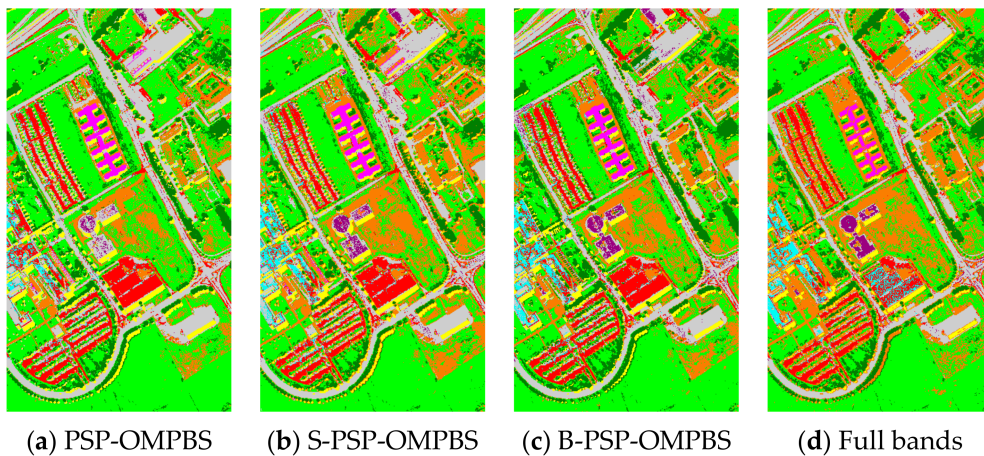
Table 5 lists all the AA, OA, and kappa coefficient values of the SVM results performed on PSP-OMPBS selected bands for both the Pavia and Purdue datasets. We observed that using the selected bands of PSP-OMPBS preserved sufficient spectral information to achieve comparative classification performance, compared to the results of using full bands. The highest accuracies of the AA, OA, and kappa coefficient values reached were 0.907, 0.868, and 0.874, respectively, which is close to the full bands results of 0.914, 0.880, and 0.884, respectively. This simply implies that the bands selected by PSP-OMPBSs are informatively complementary.

On the other hand, in the Purdue case, the most accurate results for the AA, OA, and kappa coefficient values were 0.736, 0.600, and 0.693, respectively, and were slightly worse than the full bands results of 0.756, 0.586, and 0.716. This might be simply because only using 34 bands is not enough for the Purdue image classification. This issue can be resolved by increasing  $p$ , fine-tuning kernel parameters, or use other types of SVM classifiers that utilize spectral-spatial information to improve overall performance. Such a study is beyond the scope of this paper, and thus is not included.

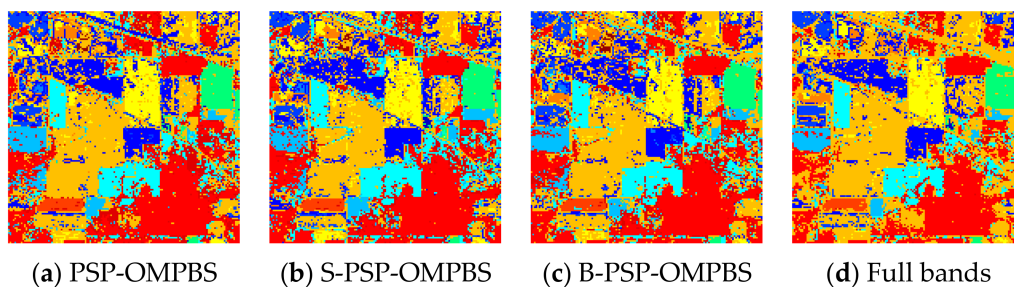
To evaluate classification performance in visual assessment, Figures 8 and 9 plot the classification maps of SVM using the bands selected by PSP-OMPBS, S-PSP-OMPBS, and B-PSP-OMPBS at  $n = 60\%$  of  $N$  for the Pavia and Purdue data, respectively. The maps using full bands are included for comparison. In the Pavia case, it can be seen that using PSP-OMPBS-selected bands could generate nearly the same classification maps compared to those obtained by using full bands. In the Purdue case, the quality of the PSP-OMPBS-generated maps are a little worse than the full bands map, due to the insufficient number of selected bands.

**Table 5.** Classification performance, in the form of the overall accuracy (OA)/average accuracy (AA)/kappa coefficient, of support vector machine (SVM) performed on the PSP-OMPBS-selected bands  $\Omega_p(n)$  in different  $n$  stages.

	Pavia Data (16 Selected Bands)			Purdue Data (34 Selected Bands)		
	PSP-OMPBS	S-PSP-OMPBS	B-PSP-OMPBS	PSP-OMPBS	S-PSP-OMPBS	B-PSP-OMPBS
$n = 20\%$ of $N$	0.904/0.863/ 0.872	0.906/ <b>0.868</b> / <b>0.874</b>	0.891/0.845/ 0.853	0.724/0.560/ 0.678	0.729/0.592/ 0.686	0.721/0.573/ 0.677
$n = 40\%$ of $N$	0.904/0.861/ 0.870	0.899/0.852/ 0.864	0.899/0.853/ 0.865	<b>0.736/0.600</b> / <b>0.693</b>	0.725/0.570/ 0.681	0.719/0.564/ 0.673
$n = 60\%$ of $N$	0.897/0.854/ 0.861	0.901/0.853/ 0.867	0.900/0.860/ 0.866	0.713/0.551/ 0.666	0.711/0.537/ 0.665	0.724/0.550/ 0.680
$n = 80\%$ of $N$	0.898/0.853/ 0.862	0.902/0.859/ 0.868	0.899/0.853/ 0.864	0.724/0.589/ 0.679	0.708/0.573/ 0.659	0.712/0.579/ 0.664
$n = 100\%$ of $N$ (equivalent to OMPBS)	0.904/0.861/ 0.871	<b>0.907</b> /0.865/ <b>0.874</b>	0.901/0.856/ 0.866	0.711/0.533/ 0.663	0.728/0.549/ 0.684	0.735/0.545/ 0.680
<b>Full bands</b>	0.914/0.880/0.884			0.756/0.586/0.716		



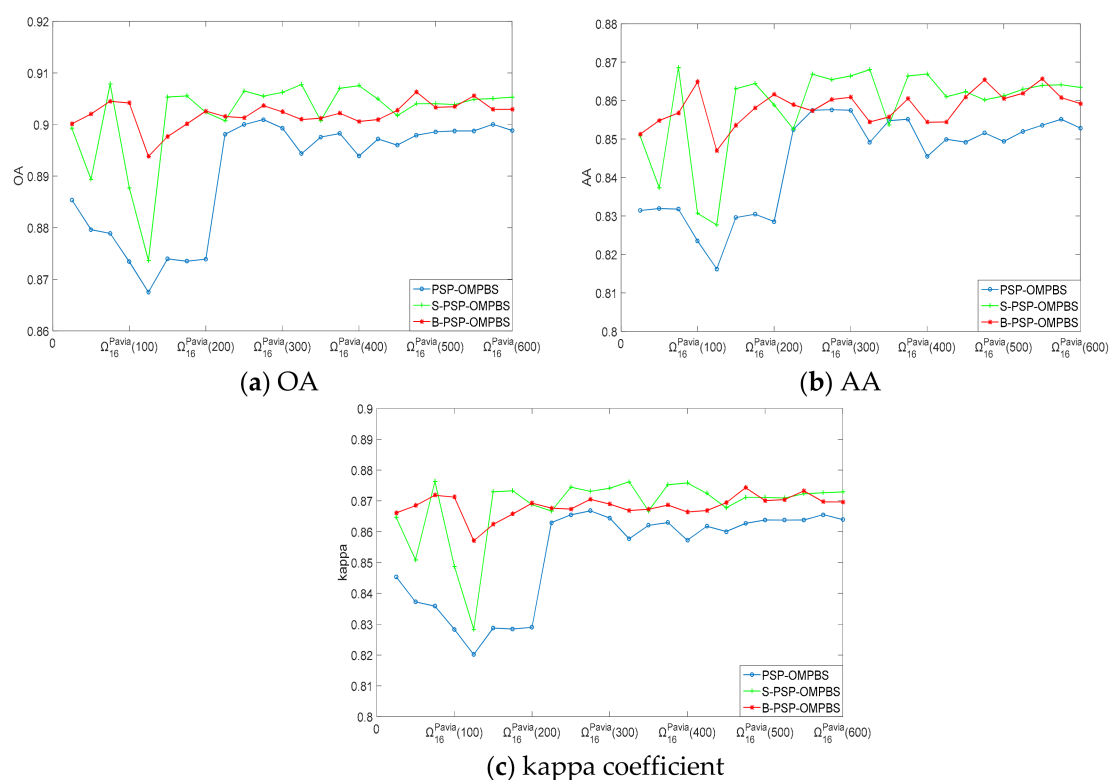
**Figure 8.** Examples of Pavia classification maps of SVM (RBF) performed on the 16 bands selected by (a) PSP-OMPBS, (b) S-PSP-OMPBS, (c) B-PSP-OMPBS, and (d) full 103 bands. For three PSP-OMPBSs,  $n$  is set at 50% of  $N$ . The corresponding OA values are 0.86, 0.89, 0.90, and 0.91.



**Figure 9.** Examples of Purdue classification maps of SVM (RBF) performed on the 34 bands selected by (a) PSP-OMPBS, (b) S-PSP-OMPBS, (c) B-PSP-OMPBS, and (d) full 202 bands. For three PSP-OMPBSs,  $n$  is set at 50% of  $N$ . The corresponding OA values are 0.72, 0.73, 0.73, and 0.76, respectively.

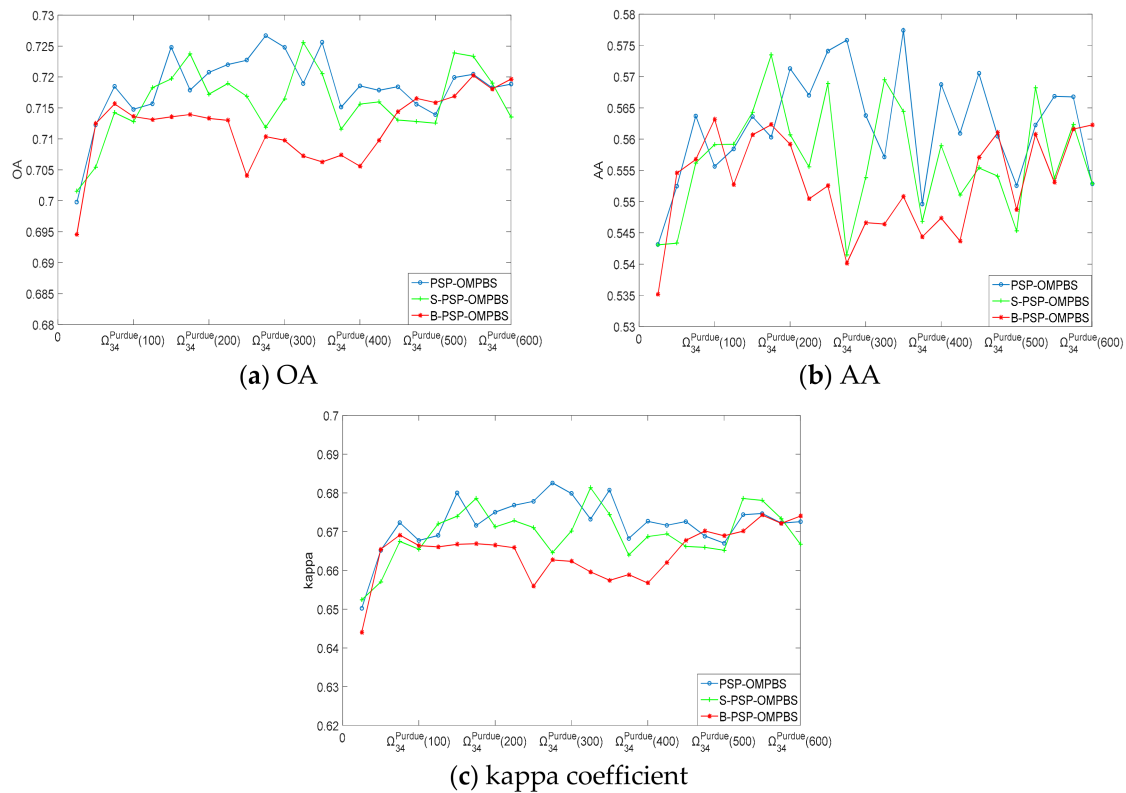
To analyze the classification performance in the progressive manner, Figure 10 plots the kappa OA/AA/kappa coefficient curves of the SVM classification using uniformly selected BS results,

performed by PSP-OMPBSs on the  $n$ -axis with interval 25 for the Pavia dataset. In those figures, the  $x$ -axis ( $n$ ) indicates band set  $\Omega_p(n)$ , and the  $y$ -axis denotes the corresponding accuracy metrics of the classification. Several observations can be found. First, all the metric curves of PSP-OMPBS and S-PSP-OMPBS are not stable in the beginning  $n \in [1,200]$  because of the poor BS quality that occurs with a low number of received pixels. With too few pixels, PSP-OMPBS could not select correct bands to fulfill the complementary spectral information. This resulted in unstable and lower classification performance. In contrast, the curves of B-PSP-OMPBS could be consistent because the transmitted pixels were uniformly sampled from the image. Second, the overall averaged performances of S-PSP-OMPBS and B-PSP-OMPBS are slightly better than OMP-BS for these three criteria. Third, when  $n$  is greater than 250, the curves of all three methods tended to be consistent. According to our extended study, all the curves will stay roughly at the same level, with extremely low deviation in the future time  $n \in [600,N]$ .



**Figure 10.** SVM classification of using the bands selected by three PSP-OMPBS methods at  $n = [1,600]$ : (a) OA, (b) AA, and (c) kappa coefficient, for the ROSIS Pavia dataset.

Similarly, Figure 11 shows the OA, AA, and kappa coefficient curves for the Purdue experiment. Similarly, there are some interesting findings. First, the classification performance is low at the beginning  $n \in [1,100]$ . Second, the Purdue image seemed to require more pixels to stabilize the BS quality. We can find that the curves of PSP-OMPBS, S-PSP-OMPBS, and B-PSP-OMPBS could not converge at  $n \in [100,600]$  in each figure, particular Figure 10b. This is probably because of the heavily mixed and noisy properties of the Purdue image. Each newly incoming pixel may easily disturb the result of band selection. According to our investigation, this phenomenon will last up to  $n = 4000$ . Finally, the S-PSP-OMPBS and B-PSP-OMPBS did not outperform PSP-OMPBS in the Purdue case. We think this is caused by the homogeneity of spectral profiles of the 16 ground classes. In this case, using a sampled transmission sequence could not significantly improve the collecting of spectral information.

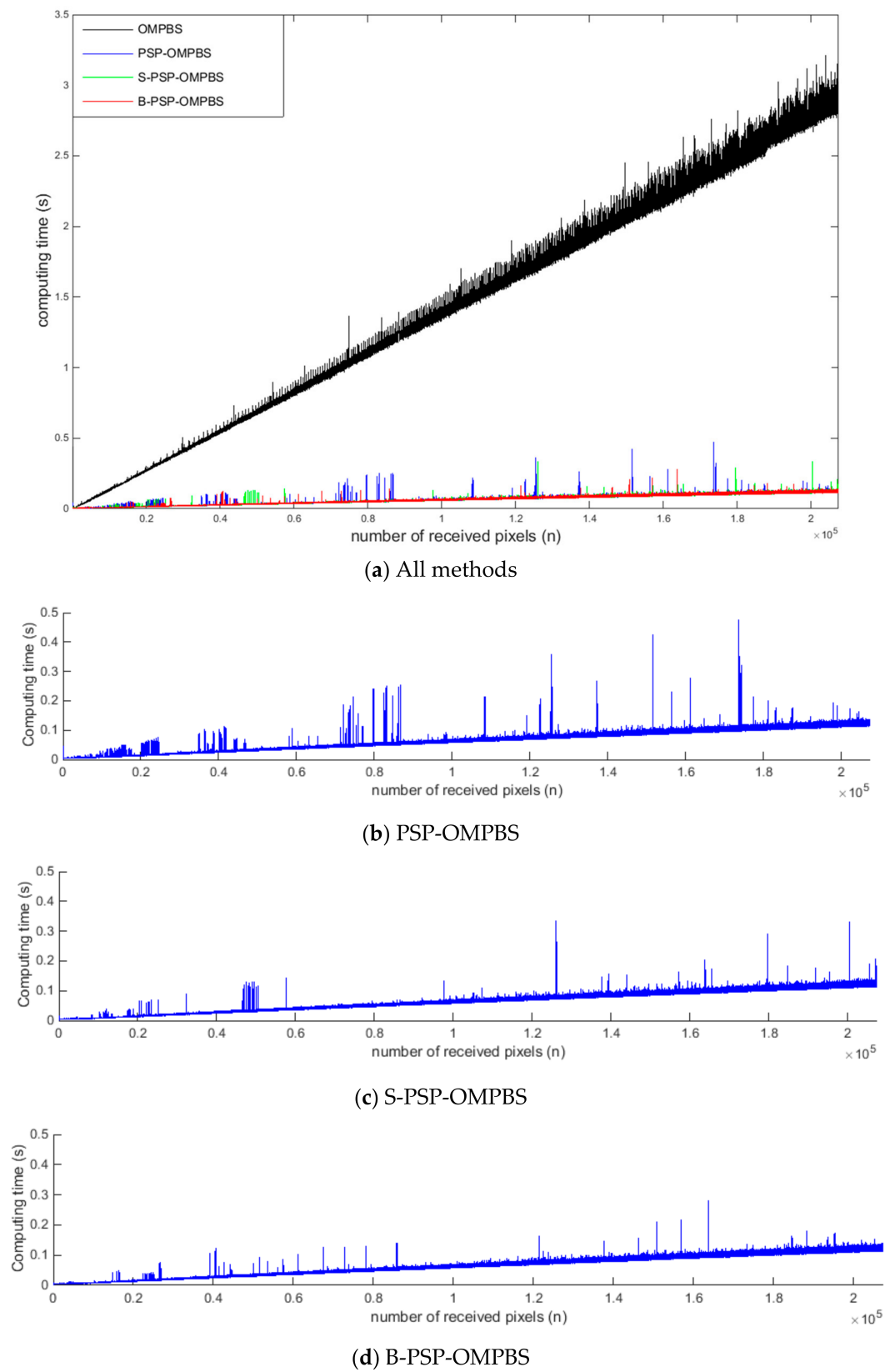


**Figure 11.** SVM classification of using the bands selected by three PSP-OMPBS methods at  $n = [1,600]$ : (a) OA, (b) AA, and (c) kappa coefficient, for AVIRIS Purdue dataset.

#### 4.4. Computing Time

To validate the real-time capability in BIS/BIP transmission, we first compare three PSP-OMPBSs with the OMPBS performed in a progressive manner. The number of transmitted pixels,  $n$ , was set from 1 to  $N$  for each image. The experiments were conducted on a computer with Intel i7-4790 3.6 GHz CPU, 16 GB RAM, Windows 7 and Matlab 2015. The values of computing time are reported by the average of ten random runs.

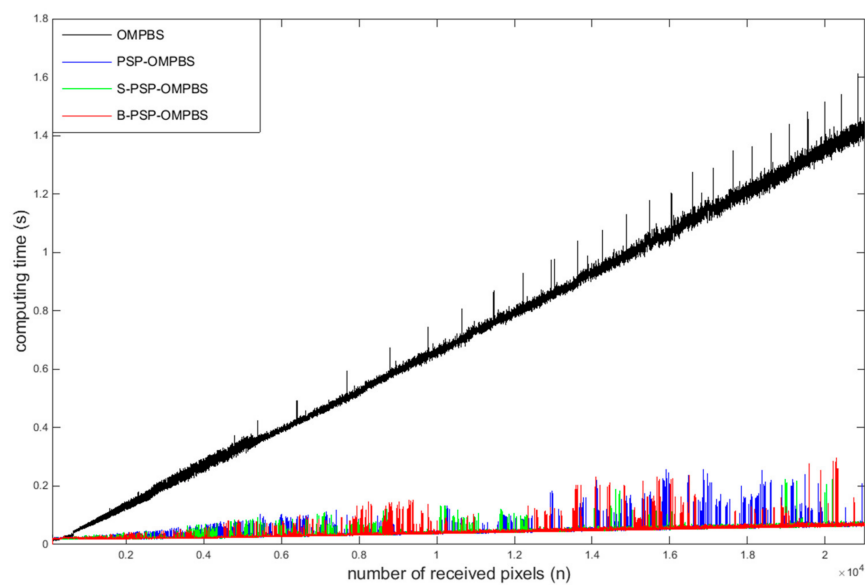
Figure 12a shows the required computing time of progressive OMPBS, PSP-OMPBS, S-PSP-OMPBS, and B-PSP-OMPBS, implemented on the Pavia dataset, where the  $x$ -axis presents the  $n$  value and the  $y$ -axis presents the corresponding computing time. It can be observed that the computing time of OMPBS significantly increases when  $n$  increases. This is due to the increase in size of  $\mathbf{B}$ , which results in greater computational complexity for OMPBS. The variation of computing time increases with  $n$ , too. On the contrary, by virtue of recursive processing, the PSP-BSs produced almost flat curves. The computing time is stably under 0.1 s in most  $n$  indices. This implies that PSP-BS has the potential to be run in a real-time manner during transmission. Figure 12b–d further shows the individual time curve for PSP-OMPBS, S-PSP-OMPBS, and B-PSP-OMPBS, respectively. It could be seen that the computing times of most of the  $n$  regions stick on the straight trend line. In those cases, the BS results remained the same, so that the recursive equations could be applied continuously. On the other hand, we can find some of the “peak phenomenon” shown in the curves. Those peaks occurred when the updating condition was not reached. Thus, the optimization problems were solved by non-recursive equations instead. The extra computing time is required.



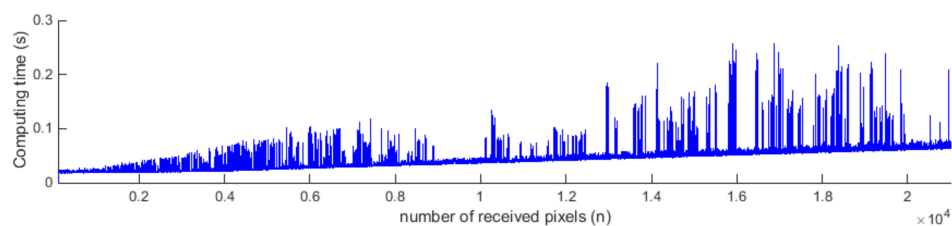
**Figure 12.** Plots showing the computing time required for different progressive BS methods implemented on the ROSIS Pavia dataset: (a) all methods, (b) PSP-OMPBS, (c) step sequence PSP-OMPBS (S-PSP-OMPBS), and (d) block sequence PSP-OMPBS (B-PSP-OMPBS).



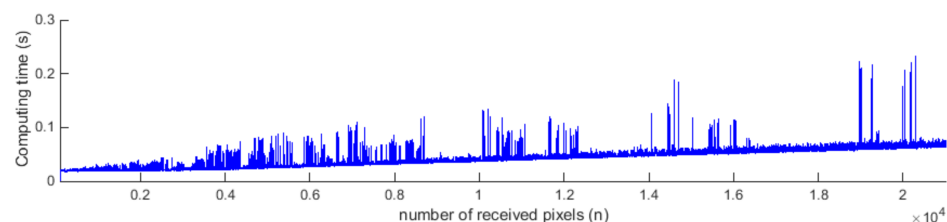
Similarly, Figure 13a shows the computing time of progressive OMPBS, PSP-OMPBS, S-PSP-OMPBS, and B-PSP-OMPBS implemented on the Purdue dataset, and Figure 13b–d further shows the individual curves of the three PSP-OMPBSs. Similarly, the three PSP-OMPBS methods required significantly less computing time than OMPBS. The required time for PSP-OMPBSs is at less than 0.07 s from  $n = 1$ –21,025, while OMPBS needs over 1 s when  $n$  is larger than 14,000. Again, this implies the superior computational efficiency provided by PSP-OMPBS. Comparing Figure 13 with Figure 12, we can interestingly find that the peak phenomenon occurred more frequently in the Purdue case. This is probably due to the noisy property of the Purdue image. Under the circumstances, the BS result varies easily if the new incoming pixel is a noisy pixel. This results in the instability of BS on the  $n$ -axis, and thus reduces the opportunity of using recursive equations. Fortunately, the peak phenomenon disappeared gradually after sufficient pixels were received, since in later periods of transmission, the received pixels represented the image well. In other words, the BS results tended to be more stable.



(a) All methods

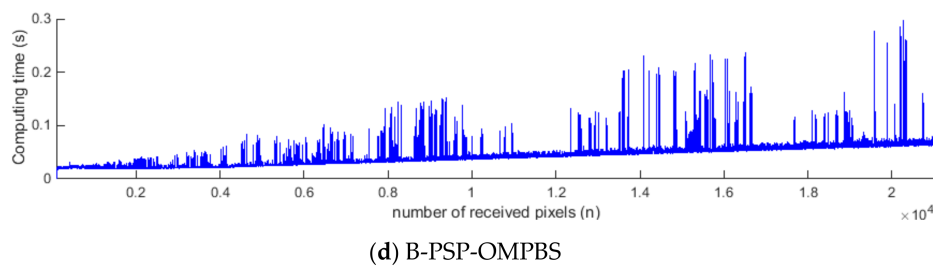


(b) PSP-OMPBS



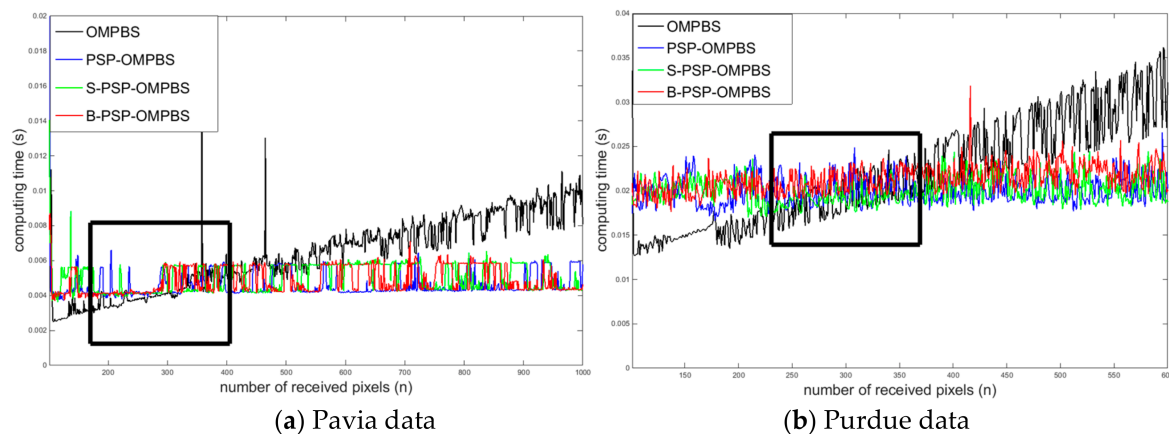
(c) S-PSP-OMPBS

Figure 13. Cont.



**Figure 13.** Plots of computing time required for different progressive BS methods implemented on the AVIRIS Purdue dataset: (a) all methods, (b) PSP-BS OMPBS, (c) step sequence PSP-OMPBS (S-PSP-OMPBS), and (d) block sequence PSP-OMPBS (B-PSP-OMPBS).

It is worth mentioning that PSP-OMPBSs required a little more computing time compared to OMPBS at the beginning of transmission. Figure 14a,b shows the zoom-in computing time plots of the Pavia and Purdue datasets at  $n = 100$ – $1000$  and  $n = 100$ – $600$ . There is a noticeable intersection between the OMPBS and PSP-BS curves at roughly  $n = 300$ . When  $n$  is less than 300, PSP-OMPBS requires more computing time because it needs to perform an additional logic operation (i.e., Step 4 in PSP-OMPBS algorithm), which leads to an increase of overall computing time. As  $n$  grows, the proportion of computing logic operations decreases, and this additional burden is relatively diminished.



**Figure 14.** Zoom-in plots of computing time at the beginning of progressive process: (a) the Pavia data, and (b) the Purdue data.

Finally, Table 6 lists the overall accumulative computing time of the progressive experiments of Figures 12 and 13. It was found that using PSP-BS could significantly reduce the amount of calculation. In addition, there was no significance difference between three PSP-OMPBS methods.

**Table 6.** The overall required computing time (in seconds) in the experiments.

Method	Pavia Data	Purdue Data
OMPBS	296,540	14,750
PSP-OMPBS	13,031	915
S-PSP-OMPBS	13,002	893
B-PSP-OMPBS	13,006	906

#### 4.5. Graphical User Intervace Design

In order to analyze the relationship between the ground location of the received pixels, the produced values of quantitative indexes, and the BS results of progressive processing, a Matlab

graphical user interface (GUI) was developed, as shown in Figure 15. It allows users to load different image data, input  $p$  values, and choose different PTS methods (original, step, block) with the corresponding parameters. Once all the inputs are loaded and the stage button is pressed, the PSP-OMPBS starts to do real-time BS simulation. The red square in the left will show the image scene, where the red dots present the locations of the received pixels. In the top-right corner, the green square will record the statistics, including the number of received pixels ( $n$ ), the processing time, and ACC value for the current  $n$ . The time curve of a short period is drawn. In the bottom-right corner, the yellow square shows the BS results of the previous stage and the current stage, in which the red bins denote the instant results of PSP-OMPBS and the black bins denote the BS ground truth.

For an easy illustration, the experiment shown in Figure 15 was performed by B-PSP-OMPBS on the Pavia dataset. The parameters were set by  $p = 10$  and  $b = 10$ . Over time, we can observe that the image gradually filled with red dots (transmitted sample pixels), and the ACC increased until it reached 100%. As a result, the BS can be fully monitored in the whole progressive process.

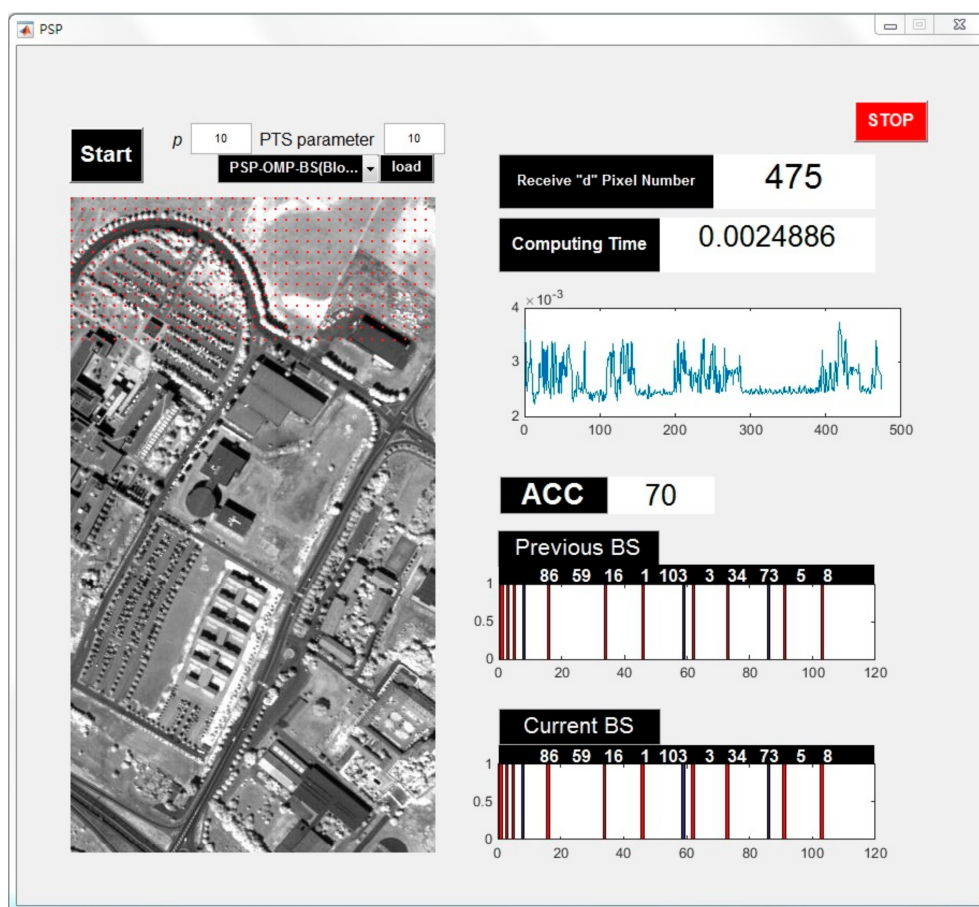


Figure 15. Graphical user interface (GUI) design for PSP-OMPBS.

## 5. Discussion

With the perspective of HSI application, there are several advantages of using PSP-BS. First, we can observe all the BS results in the whole transmission, where the local BS results (i.e., the BS results in a particular time segment) would not be missed. This is one of the most attractive features in progressive processing [34,35]. For instance, some “weaker bands” would appear during the transmission and disappear later on. Those bands might be very important for analyzing some specific types of ground materials, and should be considered to be selected. Besides, by virtue of progressive processing, the PSP-BS can easily be implemented on specific spatial regions (i.e., sub-image) within the image

scene. Analyzing the partial data is also important. The second advantage of PSP-BS is the saving of storage volume. After PSP-BS is done, the redundant bands can be immediately removed, without waiting to apply the BS algorithm again, which saves time and storage space. The third advantage of PSP-BS is the saving of transmission bandwidth. Once the BS results converge in the early stage of transmission, we can ask the transmitter to re-transmit the rest of the image data only with the selected spectral dimensions, in order to save bandwidth. This is particularly important if the transmission bandwidth is limited, or the volume of original data cube is extremely large.

Similar to PBP, that PSP-BS obeys the principle that the time of processing one band must be less than the time of transmitting one band; in PSP-BS, the processing time at each stage must be less than the time to receive a new spectral pixel. In fact, both PBP and PSP methods suffer from the computation complexity issue when the image size or spectral dimension increases. So a balance between computation capability and transmission bandwidth is a prerequisite for the real-time process. This issue has not been emphasized in the related literature. Unlike the PBP works [33–35], the PSP-OMPBS algorithm is further involved with the iterations related to sequential search. The computing time may increase with the drastic increase of  $p$ . Thus, the computation time must be further reduced to satisfy the prerequisite. Fortunately, the calculation of PSP-OMPBS can be accelerated by parallel processing. For instance, the task of Step 2, and any terms about matrix multiplication in other steps, can be allocated to different cores of the central processing unit (CPU) to compute. The calculation of the matrix inverse can also be reduced by using a graphics processing unit (GPU). If the processing time cannot be reduced to fulfill the real-time process, we still can perform BS with larger  $n$  interval. In this case, deriving new recursive equations is necessary. We leave that to our future work.

## 6. Conclusions

This paper presents an instant BS method based on progressive sample processing (PSP), called PSP-OMPBS. PSP-OMPBS uses a recursive algorithm to efficiently accelerate computation. It processes BS pixel-by-pixel according to the BIS/BIP format, or by re-arranged pixel transmission sequences. Unlike traditional BS methods, which must re-implement the total received data, PSP-OMPBS can immediately obtain BS results when receiving a new pixel by referring to past information. The experiments conducted on two hyperspectral datasets show that PSP-OMPBS can instantly output the BS of each stage with very low computing time. Besides, by adopting different types of pixel transmission sequences, it is proved that using sampled sequences can significantly accelerate the BS convergence speed. Those advantages have allowed PSP-OMPBS to be applied effectively in a real-time manner during pixel transmission, and have potential for real-time BS monitoring. Our future work includes reducing the computational complexity by parallel processing, minimizing the numerical errors, considering other pixel transmission strategies, fusing different BS algorithms with PSP, and extending PSP to other topics in the field of hyperspectral image processing.

**Acknowledgments:** This work was supported by the Ministry of Science and Technology in Grants No.: MOST 105-2221-E-110-085 and MOST 105-2221-E-224-054.

**Author Contributions:** Keng-Hao Liu proposed the idea, derived the theoretical part, designed the experiments, and wrote the paper; Hung-Chang Chien and Meng-Han Lu built the Matlab codes, performed the experiments, and designed the Matlab GUI; Shih-Yu Chen reviewed the paper, inspected the theoretical part, and gave constructive advice.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Maisel, P.W.; Kramber, W.J.; Lee, J.K. Optimum band selection for supervised classification of multispectral data. *Photogramm. Eng. Remote Sens.* **1990**, *56*, 55–60.
2. Keshava, N. Distance metrics and band selection in hyperspectral processing with applications to material identification and spectral libraries. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1552–1565. [[CrossRef](#)]

3. Du, Q.; Yang, H. Similarity-based unsupervised band selection for Hyperspectral Image Analysis. *IEEE Geosci. Remote Sens. Lett.* **2008**, *5*, 564–568. [\[CrossRef\]](#)
4. Chang, C.-I.; Wang, S. Constrained band selection for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 1575–1585. [\[CrossRef\]](#)
5. Martínez-Usó, A.; Pla, F.; Sotoca, J.M.; García-Sevilla, P. Clustering-Based Hyperspectral Band Selection Using Information Measures. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 4158–4171. [\[CrossRef\]](#)
6. Guo, B.; Gunn, S.; Damper, B.; Nelson, J. Adaptive band selection for hyperspectral image fusion using mutual information. In Proceedings of the 8th IEEE International Conference on Information Fusion, Philadelphia, PA, USA, 25–28 July 2005; p. 8.
7. Feng, J.; Jiao, L.C.; Zhang, X.; Sun, T. Hyperspectral Band Selection Based on Trivariate Mutual Information and Clonal Selection. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 4092–4105. [\[CrossRef\]](#)
8. Feng, J.; Jiao, L.; Liu, F.; Sun, T.; Zhang, X. Mutual-Information-Based Semi-Supervised Hyperspectral Band Selection With High Discrimination, High Information, and Low Redundancy. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 2956–2969. [\[CrossRef\]](#)
9. Sun, W.; Zhang, L.; Du, B. A Sparse Self-Representation Method for Band Selection in Hyperspectral Imagery Classification. In Proceedings of the 7th IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (Whispers), Tokyo, Japan, 2–5 June 2015; IEEE: New York, NY, USA; pp. 1–4.
10. Sun, W.; Zhang, L.; Zhang, L.; Lai, Y.M. A Dissimilarity-Weighted Sparse Self-Representation Method for Band Selection in Hyperspectral Imagery Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 4374–4388. [\[CrossRef\]](#)
11. Sun, W.; Jiang, M.; Li, W.; Liu, Y. A Symmetric Sparse Representation Based Band Selection Method for Hyperspectral Imagery Classification. *Remote Sens.* **2016**, *8*, 238. [\[CrossRef\]](#)
12. Lai, C.H.; Chen, C.S.; Chen, S.Y.; Liu, K.H. Sequential band selection method based on orthogonal matching pursuit. In Proceedings of the 8th IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Los Angeles, CA, USA, 21–24 August 2016; IEEE: New York, NY, USA; pp. 1–4.
13. Fisher, K.; Chang, C.-I. Progressive band selection for satellite hyperspectral data compression and transmission. *J. Appl. Remote Sens.* **2010**, *4*, 041770.
14. Chang, C.-I.; Liu, K.-H. Progressive Band Selection of Spectral Unmixing for Hyperspectral Imagery. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 2002–2017. [\[CrossRef\]](#)
15. Liu, S.; Du, Q.; Tong, X.; Samat, A.; Pan, H.; Ma, X. Band Selection-Based Dimensionality Reduction for Change Detection in Multi-Temporal Hyperspectral Images. *Remote Sens.* **2017**, *9*, 1008. [\[CrossRef\]](#)
16. Geng, X.; Sun, K.; Ji, L.; Zhao, Y. A Fast Volume-Gradient-Based Band Selection Method for Hyperspectral Image. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 7111–7119. [\[CrossRef\]](#)
17. Iordache, M.D.; Bioucas-Dias, J.M.; Plaza, A. Potential and limitations of band selection and library pruning in sparse hyperspectral unmixing. Proceedings of 2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Tokyo, Japan, 2–5 June 2015; IEEE: New York, NY, USA; pp. 1–4.
18. Chang, Y.L.; Chang, L.; Fang, J.P. Particle swarm optimization/impurity function class overlapping scheme based on multiple attribute decision making model for hyperspectral band selection. Proceedings of 2015 IEEE International CONFERENCE Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015; IEEE: New York, NY, USA; pp. 441–444.
19. Huber-Lerner, M.; Hadar, O.; Rotman, S.R.; Huber-Shalem, R. Hyperspectral Band Selection for Anomaly Detection: The Role of Data Gaussianity. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 732–743. [\[CrossRef\]](#)
20. Cao, X.; Xiong, T.; Jiao, L. Supervised Band Selection Using Local Spatial Information for Hyperspectral Image. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 329–333. [\[CrossRef\]](#)
21. Zhan, Y.; Hu, D.; Xing, H. Hyperspectral Band Selection Based on Deep Convolutional Neural Network and Distance Density. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2365–2369. [\[CrossRef\]](#)
22. Wang, L.; Li, H.C.; Xue, B.; Chang, C.I. Constrained Band Subset Selection for Hyperspectral Imagery. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2032–2036. [\[CrossRef\]](#)
23. Zhu, G.; Huang, Y.; Li, S.; Tang, J.; Liang, D. Hyperspectral Band Selection via Rank Minimization. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2320–2324. [\[CrossRef\]](#)



24. Park, H.; Choi, J.; Park, N.; Choi, S. Sharpening the VNIR and SWIR Bands of Sentinel-2A Imagery through Modified Selected and Synthesized Band Schemes. *Remote Sens.* **2017**, *9*, 1080. [CrossRef]
25. Yang, C.; Tan, Y.; Bruzzone, L.; Lu, L. Discriminative Feature Metric Learning in the Affinity Propagation Model for Band Selection in Hyperspectral Images. *Remote Sens.* **2017**, *9*, 782. [CrossRef]
26. Hihara, H.; Moritani, K.; Inoue, M.; Hoshi, Y.; Iwasaki, A.; Takada, J.; Inada, H.; Suzuki, M.; Seki, T.; Ichikawa, S.; Tani, J. On board Image Processing System for Hyperspectral Sensor. *Sensors* **2015**, *15*, 24926–24944. [CrossRef] [PubMed]
27. Conoscenti, M.; Coppola, R.; Magli, E. Constant SNR, Rate Control, and Entropy Coding for Predictive Lossy Hyperspectral Image Compression. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7431–7441. [CrossRef]
28. Giordano, R.; Lombardi, A.; Guccione, P. Efficient clustering and on-board ROI-based compression for Hyperspectral Radar. In Proceedings of the IARIA Conference, Lisbon, Portugal, 26–30 June 2016; pp. 33–38.
29. Giordano, R.; Guccione, P. ROI-Based On-Board Compression for Hyperspectral Remote Sensing Images on GPU. *Sensors* **2017**, *17*, 1160. [CrossRef] [PubMed]
30. Quesada-Barriuso, P.; Argüello, F.; Heras, D.B. Computing Efficiently Spectral-Spatial Classification of Hyperspectral Images on Commodity GPUs. In *Recent Advances in Knowledge-based Paradigms and Applications. Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2014.
31. Ma, N.; Wang, S.; Ali, S.M.; Cui, X.; Peng, Y. High Efficiency On-Board Hyperspectral Image Classification with Zynq SoC. Proceedings of 2016 7th International Conference on Mechatronics and Manufacturing (ICMM 2016), Harbin, China, 16 March 2016; Volume 45. [CrossRef]
32. Chang, C.I. Overview and Introduction. In *Real-Time Progressive Hyperspectral Image Processing*; Springer-Verlag: New York, NY, USA, 2016; pp. 1–32.
33. Chang, C.I.; Wu, C.C.; Liu, K.H.; Chen, H.M.; Chen, C.C.C.; Wen, C.H. Progressive Band Processing of Linear Spectral Unmixing for Hyperspectral Imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2583–2597. [CrossRef]
34. Chang, C.I.; Li, Y.; Hobbs, M.C.; Schultz, R.C.; Liu, W.M. Progressive Band Processing of Anomaly Detection in Hyperspectral Imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 3558–3571. [CrossRef]
35. Chang, C.I.; Schultz, R.C.; Hobbs, M.C.; Chen, S.Y.; Wang, Y.; Liu, C. Progressive Band Processing of Constrained Energy Minimization for Subpixel Detection. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 1626–1637. [CrossRef]
36. Elhamifar, E.; Sapiro, G.; Vidal, R. See all by looking at a few: Sparse modeling for finding representative objects. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; IEEE: New York, NY, USA; pp. 1600–1607.
37. Tropp, J.A.; Gilbert, A.C. Signal Recovery from Random Measurements via Orthogonal Matching Pursuit. *IEEE Trans. Inf. Theory* **2007**, *53*, 4655–4666. [CrossRef]
38. Woodbury Matrix Identity. Available online: [https://en.wikipedia.org/wiki/Woodbury\\_matrix\\_identity](https://en.wikipedia.org/wiki/Woodbury_matrix_identity) (accessed on 10 December 2017).
39. Chang, C.-I.; Du, Q. Estimation of number of spectrally distinct signal sources in hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 608–619. [CrossRef]
40. LibSVM. Available online: <https://www.csie.ntu.edu.tw/~cjlin/libsvm/> (accessed on 18 January 2018).

