APPROVAL SHEET

Title of Dissertation: Recursive Band Processing Algorithms for Finding Unknown Targets in Hyperspectral Imagery

Name of Candidate: Yao Li
Doctor of Philosophy, 2016

Dissertation and Abstract Approved: Chen-I Chang
Chein-I Chang
Professor
Department of Computer Science and Electrical Engineering

Date Approved: November 23, 2016

Abstract

Title of Document: RECURSIVE BAND PROCESSING ALGORITHMS FOR FINDING UNKNOWN TARGETS IN HYPERSPECTRAL IMAGERY

Yao Li, Doctor of Philosophy, 2016

Directed By: Professor Chein-I Chang
Department of Computer Science and Electrical Engineering

Hyperspectral imaging has emerged as a remote sensing image processing technique in many applications. The reason that hyperspectral data is called hyperspectral is mainly because hundreds of contiguous spectral bands provide the massive spectral information used for data analysis. Such high spectral resolution leads to a challenging issue in how to effectively utilize vast amount of spectral information. In order to resolve this dilemma one general approach is to use Band Selection (BS) to judiciously select a desired band subset that can well represent the data. Unfortunately, two prime issues, (1) the number of bands need to be selected *p*, and (2) how to select appropriate bands, need to be addressed. Recently, an alternative approach, Progressive Band Selection (PBS) was proposed which can process data band by band without knowing the value of *p*.

This dissertation looks into PBS and develops a rather different approach according to Band SeQuential (BSQ) data acquisition format, called Recursive Band Processing (RBP) which focuses on PBS specified by applications. With the custom-designed algorithmic architectures RBP can be carried out by PBS recursively in a similar

manner that a Kalman filter does. Since the utility of RBP must be realized by applications, three applications of interest are investigated: (1) unsupervised active hyperspectral target detection where the well-known Automatic Target Generation Process (ATGP) is extended to RBP-ATGP; (2) unsupervised target identification where one of most popular algorithms Pixel Purity Index (PPI) is re-derived as RBP-PPI; (3) endmember finding where a recently developed Orthogonal Projection (OP) based Simplex Growing Algorithm (OPSGA) is further extended to RBP-OPSGA.

Several advantages result from the proposed RBP-based algorithms. First it is specifically designed based on BSQ data acquisition format and it is particularly suitable for hyperspectral data communication and transmission with limited bandwidth. Second, RBP provides spectral profiles of changes in target detection which allow users to screen preliminary results while data collection is ongoing without waiting for completion of full data set. Most importantly, the recursive nature in RBP can facilitate the hardware design which can significantly reduce computational complexity in chip design.

RECURSIVE BAND PROCESSING ALGORITHMS FOR FINDING UNKNOWN TARGETS IN HYPERSPECTRAL IMAGERY

By

Yao Li

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy,
Electrical Engineering
2016

To my parents

Acknowledgement

I would like to express my gratitude to my research advisor Prof. Chein-I Chang. His work ethic, motivation and dedication to the success of his students are unparalleled. Selecting him as my mentor is definitely one of the best choice in my life. Without his encouragement, support and guidance, it is impossible for me to finish my dissertation after so many years and thousands miles away.

I would like also to thank my defense committee members: Prof. Jian Chen, Prof. Ting Zhu, Dr. Damon Bradley and Dr. Haleh Safavi. For their time, patience and advise on my dissertation.

There are many people in our RSSIPL lab, whose general supports have allowed me to continue my journey in research. In addition, they helped me out in many dilemmas, joined me many discussions of some key issues related to my research topic and served as audiences for my presentations. Without their great support, I would never have been able to figure out many original ideas, which lead me to my today's achievements. These people include Prof. Meiping Song, Prof. Lin Wang, Dr. Shih-Yu Chen, Dr. Yulei Wang, Dr. Hsiao-Chi Li, Dr. Cheng Gao, Li-Chien Lee and Chien-Yu Lin.

I would also like to sincere thank Dr. Peter Hu for his constant support and his offering of research opportunities in the Patient Safety Research lab in the Shock Trauma Center in University of Maryland Baltimore. I really appreciate Dr. Hu's trust, and respect his diligence and devotion to work. In the Patient Safety lab, I received a lot of help from Dr. Colin Mackenzie, Dr. Lynn Stansbury, Dr. Hegang Chen and Dr. Shiming Yang.

Finally, I would like to express my greatest appreciation to my parents for their understanding, encouragement ad unwavering support. It is my parents now living in China who have given me countless love and encouragement that keep me moving forward. It is also my dearest parents who have promised me to take good care of themselves so that I can concentrate on my research without distraction.

Table of Content

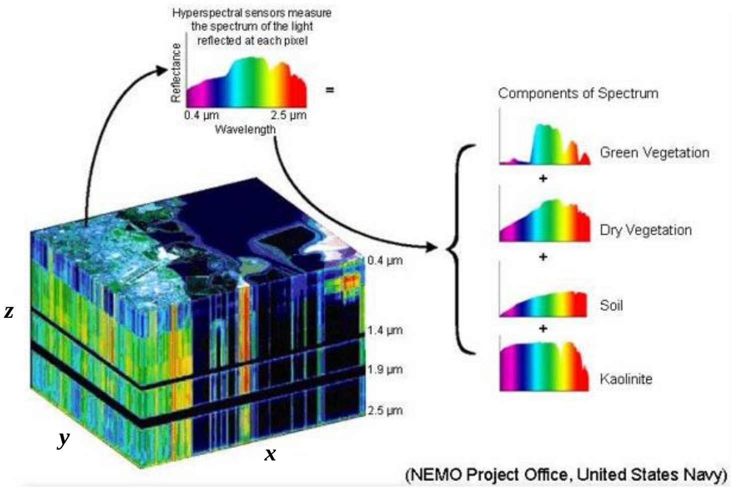
List of Figures

List of Tables

Chapter 1: Introduction

1.1 Hyperspectral Image Processing

Hyperspectral image processing analyzes information from across hundreds of contiguous electromagnetic spectrum for each pixel in an image to detect objects of interest, which are often difficult to identify with regular color images. Because of its advantages in capturing rich information, hyperspectral imaging has received broader attention in its applications in agriculture, mineralogy, reconnaissance and surveillance, and biomedicine with the fast advancement in sensor technique and computational technologies. Like a color image may have 3 RGB bands and a multispectral image may have tens of discrete bands, a hyperspectral image typically has hundreds of bands. Fig. 1.1 presents an example of a hyperspectral image cube where *x*-axis and *y*-axis denotes the rows and columns of the image, and *z*-axis is specified by spectral dimensionality, λ.



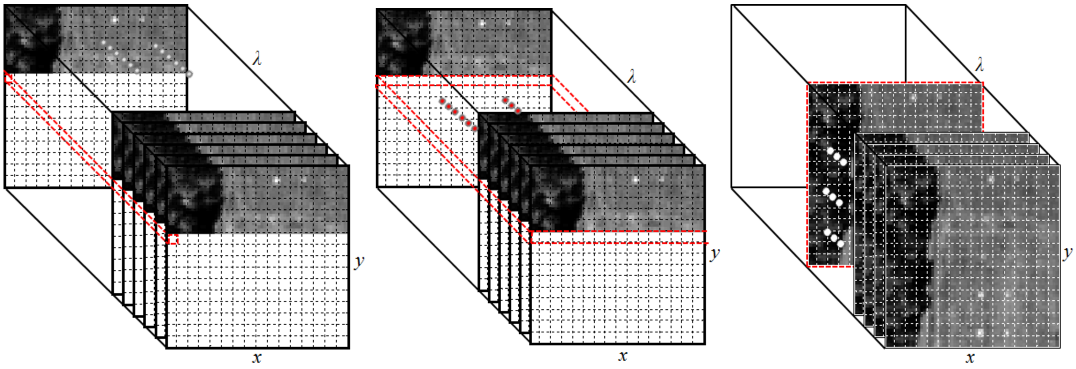
(NEMO Project Office, United States Navy)

Figure 1.1. An example of hyperspectral image cube

With increased spectral resolution in conjunction with the increased number of bands hyperspectral imaging distinguishes itself from traditional spatial-domain based image processing techniques by exploring spectral properties of data rather than spatial information. This leads to a significant difference in designing hyperspectral imaging algorithms where direct extensions or expansions from multispectral imaging techniques are generally not effective. One major strength resulting from hyperspectral imaging is target detection which can be categorized into active target detection with application in reconnaissance where the desired target information is provided and passive target detection with applications in surveillance where no target knowledge is available. This dissertation looks into three major algorithms: (1) ATGP for active target detection, (2) PPI and OPSGA for passive target detection and further derive their RBP versions.

1.2 Acquisition Formats Used by Hyperspectral Imaging

In general, there are two data acquisition format generally used by hyperspectral imaging sensors (Schowengerdt, 1997). The first type is referred to as Band-Interleaved-by-Pixel (BIP)/ Band-Interleaved-by-Sample (BIS), or Band-Interleaved-by-Line (BIL) process shown in Fig. 1.2(a) and Fig. 1.2(b) respectively. In Fig. 1.2, the coordinates (x, y) represent the spatial location of a particular data sample or pixel, λ is a wavelength parameter used to specify the spectral bands, the data currently being collected is highlighted by red dashed lines. BIP/ BIS format, as shown in Fig. 1.2(a), at each time instant, one pixel with full bands information is collected. While a line of pixels with full bands of information is collected for BIL format displayed in Fig. 1.2(b). The second type is referred to as Band-Sequential (BSQ) date acquisition format shown in Fig. 1.2(c), which collects one band with full image pixels at a time.



(a) BIP/BIs (b) BIL (c) BSQ

Figure 1.2. Hyperspectral image data acquisition format (a) BIP/BIS (b) BIL (c) BSQ

1.3 Progressive Processing in Hyperspectral Imaging (PHSI)

Progressive processing is a sequence of data processing so that each process can be used to update and improve its previous processes by subsequent processes in terms of performance. In other words, a progressive process can effectively and intelligently utilize its information stage by stage in a succession of stage processes to achieve its best performance. In order to implement progressive processing, two types of PHSI can be designed, sample-wise Progressive Sample Processing (PSP) and band-wise Progressive Band Processing (PBP).

1.3.1 Sample-wise PHSI

Consider an AVIRIS hyperspectral image cube of 100 lines and 100 samples per line with 224 contiguous spectral bands per sample. A sample can be considered as a pixel vector with each component corresponding to a pixel collected by a particular wavelength. The idea of PSP is to process the data one sample vector at a time with all 224 spectral bands being collected. Each successive sample that is received and processed by PSP according to the data acquisition formt BIP/BIS. When each new sample vector, the result is updated by previous results. The same process is repeated over and over again until all sample vectors have been processed. With this interpretation, real time processing can be considered as a version of PSP.

1.3.2 Band-wise PHSI

By contrast, PBP is a version of band-wise PHSI which processes band data, referred to as band image, one at time band by band progressively. It first receives the 1st band image and in the meantime it also processes the band image without waiting for the 2nd band image being collected. When it receives the 2nd band, it then updates results by the results obtained from the 1st band image and the newly received 2nd band image. The same process is then repeated over and over again until it reaches the last or final band images. However, such process can be also terminated earlier if data analysts do not need any more band data for their analysis. Two advantages can be gained from PBP. One is that PBP can be carried out simultaneously by different users using different sets of bands. Another is that PBP can significantly reduce computing time because it reduces processing 3D image cubes to 2D images

1.4 Dissertation Outline

Contributions of this study include: (1) a thorough study on the concept of RBP and making RBP applicable to hyperspectral data exploitation in finding targets of interest; (2) development of RBP-ATGP algorithm which can find additional targets of interest comparing with the original ATGP, speed-up the process 30% comparing PBP-ATGP; (3) development of RBP-PPI on the basis of PPI; (4) development of RBP-FIPPI which process the data in RBP manner, expedite the process 10% and (5) development of RBP-OPSGA and RBP-GSGA without calculating the matrix determinant.

There are many advantages that RBP has but cannot be offered by most hyperspectral data processing techniques. One is to provide users with preliminary results progressively band by band so that users can terminate data processing anytime by aborting the process. Second, it reduces the tremendous amount of storage and computing time by only processing innovations information without re-processing previously processed data. Both advantages are crucial in satellite data processing when data can be communicated and processed band by band progressively between transmitting and receiving ends. Most importantly, it can be realized in real time so that a timely decision can be made while data processing is taking place. Finally, it paves the way for implementation in hardware devices such as a Field Programmable Gate Array (FPGA) or Application Specific Integrated Circuit (ASIC).

The thesis is organized as follows:

Chapter 2 reviews background knowledge that is necessary to understand the work in this dissertation. It includes a detailed description of the data sets to be used for experiments as well as descriptions of algorithms to be developed for RBP in this dissertation.

Chapter 3 develops Recursive Band Processing of Automatic Target Generation Process (RBP-ATGP) for ATGP which can implement ATGP band by band progressively.

Chapter 4 introduces and presents a recursive band processing version of a well-known endmember candidate finding algorithm, Pixel Purity Index (PPI) referred to as RBP-PPI. In addition, RBP-Progressive-PPI (RBP-P-PPI) and RBP-Causal-PPI (RBP-C-PPI) are further proposed to address the issues in finding an appropriate number of skewers.

Chapter 5 further derives a recursive band processing version of Fast Iterative Pixel Purity Index (RBP-FIPPI) for extracting endmembers without the need of determining the number of skewers and selecting a threshold value, which are the major two drawbacks in PPI.

Chapter 6 introduces and presents a recursive-band-processing version of a well-known endmember finding algorithm, Simplex Growing Algorithm (SGA) referred to as PBP-SGA. SGA was previously developed by Chang et al. to ease the computational complexity of one of the two most widely used endmember finding algorithms in remote sensing community, N-finder algorithm (N-FINDR) developed by Winter. PBP-SGA allows users to find endmembers by growing simplexes band by band without waiting for receipt of all bands.

Finally, chapter 7 summarizes the results obtained in this dissertation and presents some applications and extensions for future work.

Chapter 2: Preliminaries

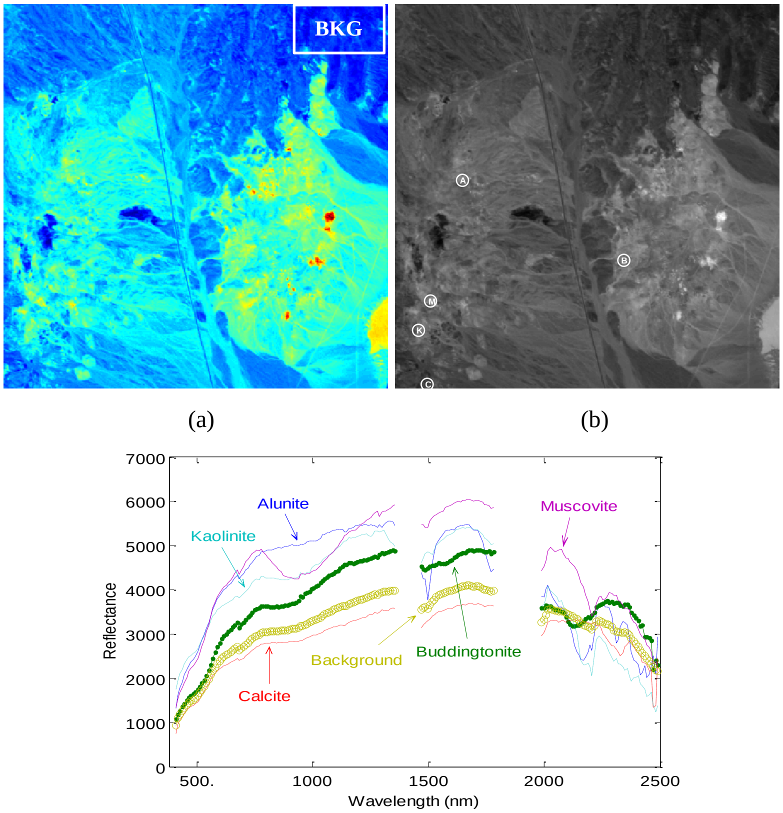
This chapter provides the background information needed to fully understand the history and applications of the proposed work in this dissertation. It begins by introducing hyperspectral data and concludes by presenting the algorithms to be studied in this dissertation, that are Automatic Target Generation Process (ATGP), Pixel Purity Index (PPI), Fast Iterative Pixel Purity Index (FIPPI), and Simplex Growing Algorithm (SGA) for unsupervised targets finding and endmember extraction.

2.1 Hyperspectral Data

Hyperspectral images are distinguished from color or multispectral images by the number of spectral bands and the resolution of the spectral bands. That said, it is important to understand that there is no clear standard definition for what hyperspectral data is. The term hyperspectral was defined in (HYMSMO Program Office, 1995) as “high spectral resolution sensor, having hundreds of spectral bands, each with a resolution between 0.1 and 10nm.” The same document defined multispectral as “moderate spectral resolution sensors, having tens of image bands, and spectral resolution greater than 10nm.” For the purposes of this dissertation, a hyperspectral image is defined as an image having greater than 100 spectral bands with an average spectral resolution less than or equal to 10nm. The average spectral resolution is given since the spectral resolution for each band varies depending on wavelength. Two kinds of data sets are used throughout this work to demonstrate the application of the algorithms developed. The first kind is synthetic images custom-designed and simulated based on Cuprite image data, which is available on the USGS website http://aviris.jpl.nasa.gov/. Synthetic images are simulated which provides complete knowledge to conduct quantitative analysis for performance evaluation. The second kind is real hyperspectral image scene. Proposed algorithms are applying on HYDICE scene, which is collected from an airborne platform with ground truth carefully recorded.

2.1.1 Synthetic Data

In order to substantiate and validate the proposed Recursive Band Processing (RBP) approach, first, a real image scene is used to custom design synthetic images for experiments where a set of controllable parameters was used to obtain complete ground truth. The image scene to be used was a real Cuprite image data shown in Fig. 2.1(a) which is available at the USGS website http://aviris.jpl.nasa.gov/. This scene is a 224-band image with size of 350 × 350 pixels and was collected over the Cuprite mining site, Nevada, in 1997. It is well understood mineralogically. As a result, a total of 189 bands were used for experiments where bands 1-3, 105-115 and 150-170 have been removed prior to the analysis due to water absorption and low SNR in those bands. Although there are more than five minerals in the data set, the ground truth available for this region provides only the locations of the pure pixels: Alunite (A), Buddingtonite (B), Calcite (C), Kaolinite (K) and Muscovite (M). The locations of these five pure minerals are labeled by A, B, C, K and M respectively and shown in Fig. 2.1(b). Available from the image scene is a set of these reflectance spectra shown in Fig. 2.1(c) that will be used to simulate synthetic images.



(a)

(b)

(c)

Figure 2.1. (a) Cuprite AVIRIS image scene (b) spatial positions of five pure pixels corresponding to minerals: alunite (A), buddingtonite (B), calcite (C), kaolinite (K) and muscovite (M); (c) Five mineral reflectance spectra and background signature (b) - which is the sample mean of the image in (a)

Since real images generally do not have complete ground truth about the endmembers, we must rely on synthetic images which are simulated by complete knowledge to conduct quantitative analysis for performance evaluation of various scenarios. Fig. 2.2 shows a synthetic image simulated by 25 panels using five mineral signatures in Fig. 2(c).

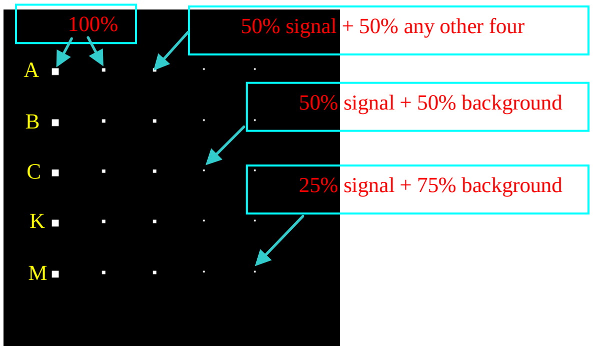
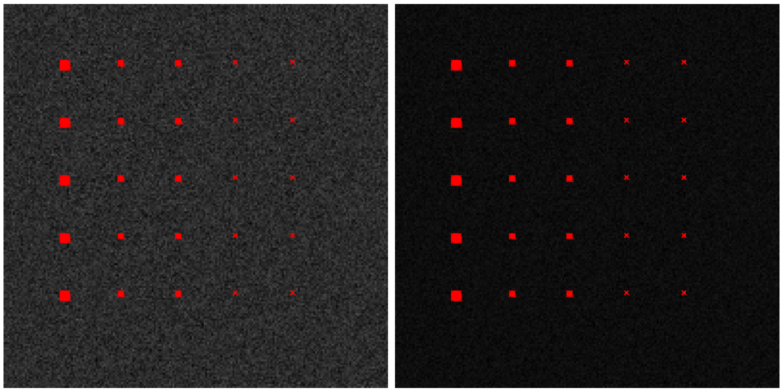

100%
50% signal + 50% any other four
50% signal + 50% background
25% signal + 75% background
A
B
C
K
M

Figure 2.2. A set of 25 panels simulated by A,B,C,K,M

Among 25 panels are five 4 × 4 pure-pixel panels for each row in the 1st column and five 2 × 2 pure-pixel panels for each row in the 2nd column, the five 2 × 2 mixed pixel panels for each row in the 3rd column and both the five 1 × 1 subpixel panels for each row in the 4th column and the 5th column where the mixed and subpanel pixels were simulated according to legends in Fig. 2.2. So, a total of 100 pure pixels (80 in the 1st column and 20 in the 2nd column), referred to as endmember pixels were simulated in the data by the five endmembers, A,B,C,K,M. An area marked by “BKG” at the upper right corner of Fig. 2.1(a) was selected to find its sample mean, i.e., the average of all pixel vectors within the area “BKG”, denoted by **b** and plotted in Fig. 2.1(c), to be used to simulate the background for image scene in Fig. 2.2. The reason for this background selection is empirical since the selected area “BKG” seemed more homogeneous than other regions. Nevertheless, other areas can be also selected for the same purpose. This **b**-simulated image background was further corrupted by an additive noise to achieve a certain signal-to-noise ratio (SNR) which was defined as 50% signature (i.e., reflectance/radiance) divided by the standard deviation of the noise in (Du Q. and Ren H., 2003).

Once target pixels and background are simulated, two types of target insertion can be designed to simulate experiments for various applications. The first type of target insertion is Target Implantation (TI) which can be simulated by inserting clean target panels into the clean image background plus additive Gaussian noise by replacing their corresponding background pixels. As a result, the implanted target pixels satisfy both abundance sum-to-one constraint (ASC) and abundance non-negativity constraint (ANC). The second type of target insertion is Target Embeddedness (TE) which can be also simulated by embedding clean target panels into the clean image background plus additive Gaussian noise by superimposing target pixels over the background pixels in which case the embedded target pixels violate the ASC but still satisfy ANC. Fig. 2.3 (a, b) shows the TI scene and TE scene correspondingly. As shown in Fig. 2.3, the ground truth pixels were marked by red cross.



(a) (b)

Figure 2.3. Synthetic Scene (a) TI Scene (b) TE Scene

2.1.2 Real Data (HYDICE Scene)A real image scene collected by HYperspectral Digital Imagery Collection Experiments (HYDICE) is used for experiments throughout this dissertation to demonstrate the utility of the recursive band processing approaches proposed. The image scene to be studied is a subset of the Forest Radiance I (FR-I) standardized hyperspectral imagery data set which was collected using the HYDICE sensor in 1996 (Topographic Technology Laboratory Topographic Engineering Center, 1995). An example image scene is shown in Figure 2.4(a) which has a size of 64 lines with 64 samples per line. Each sample was originally acquired by 210 spectral bands with spectral coverage from 0.4 µm to 2.5 µm. Low signal/high noise bands: bands 1-3 and bands 202-210; and water vapor absorption bands: bands 101-112 and bands 137-153 were subsequently removed for a total of 169 bands used in the experiments. The FR-I dataset captured multiple hyperspectral images where materials of different types were arranged within the image scene. Figure 2.4(a) is an image that contains 15 panels in the scene and the ground truth map is shown in Figure 2.4(b). Note that the ground truth map contains two colors associated with each panel. A memo associated with the original investigation (Resmini, Ron, HYMSMO, 1997) defined two heuristics:

1) Detection of targets highlighted by Red;

2) Hits of target highlighted by yellow mask areas.

The red and yellow ground truth map colors correspond to the red and yellow masks mentioned above. The spatial resolution varies based on the altitude of the HYDICE sensor and is 1.56m for the images shown in Figure 2.4. The spectral resolution varies with wavelength, ranges from 3nm to 15nm and has an average of 10nm. More information on the HYDICE sensor can be found in (Nischan, Kerekes, & Baum, 1999). Within the scene in Figure 2.4, there is a large field that to dominates the image and a forest on the left edge. A small road divides the forest and the field.

There are 15 panels distributed into five rows and three columns inside the field. Each panel will be denoted by p_{ij} with rows indexed by i and columns indexed by j. For each row $i = 1, 2, \ldots, 5$, there are three panels, p_{i1}, p_{i2}, p_{i3}, composed of the same material but with three different sizes. The sizes of the panels in the first, second and third columns are 3m × 3m, 2m × 2m and 1m × 1m respectively. Since the size of the panels in the third column is 1m × 1m, they cannot clearly be seen in Figure 2.4(a) due to the fact that the panel size is less than the 1.56m pixel resolution of the sensor. For each column $j = 1, 2, 3$, the 5 panels, p_{1j}, p_{2j}, p_{3j}, p_{4j}, p_{5j} have the same size but are composed of different paint and material combinations. A list of the materials used is given in Table 2.1.

Table 2.1: List of materials used for target panel pixels marked by red and the corresponding panel number

Panel Pixels	**Materials**
p_{11}, p_{12}, p_{13}	Nomex Kevlar (woodland)
p_{211}, p_{221}, p_{22}, p_{23}	Light olive parachute
p_{311}, p_{312}, p_{32}, p_{33}	Dark olive parachute
p_{411}, p_{412}, p_{42}, p_{43}	Spectral tag on green fabric
p_{511}, p_{521}, p_{52}, p_{53}	Green fabric with no spectral tag

It should be noted that the panels in rows 2 and 3 were made with the similar material but different camouflage. This is also the case for the panels in rows 4 and 5. The 1.56m spatial resolution of the image scene suggests that most of the 15 panels are one pixel in size except that p_{21}, p_{31}, p_{41}, p_{51} which are two-pixel panels, denoted by p_{211}, p_{212}, p_{311}, p_{312}, p_{411}, p_{412}, p_{511}, p_{512}. Figure 2.4(b) shows the precise spatial locations of these 15 panels. Figure 2.4(c) plots the 5 panel spectral signatures p_i for $i = 1, 2, \ldots, 5$ obtained by averaging the red pixels of the 3m × 3m and 2m × 2m panels in row i of Figure 2.4(b). The red pixels of the 1m × 1m panels are not used to calculate the spectral signatures since they are not pure pixels, mainly due to that fact that the spatial resolution of the panels is smaller than the pixel resolution of the sensor in which case the ratio of the panel area to a single pixel is approximately $(1.56)^{-2} \approx 0.4$. These panel signatures in Fig. 2.4(c) were used as prior target knowledge for experiments.

[1] HYMSMO is Hyperspectral MASINT Support to Military Operations

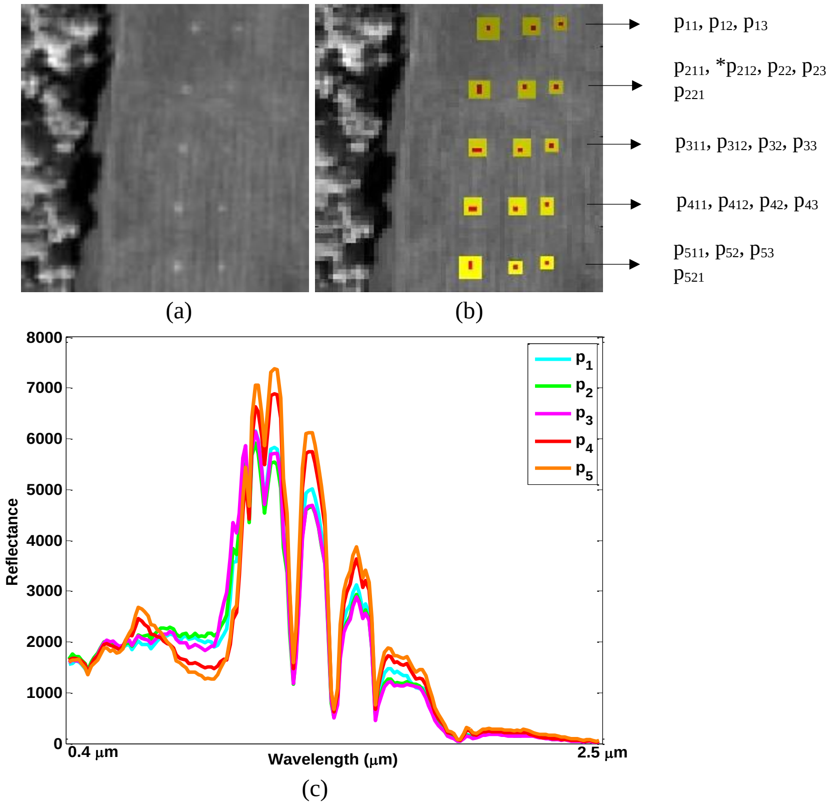


Figure 2.4. (a) A HYDICE panel scene with 9 signatures identified by prior knowledge via the ground truth given in (b) which contains 15 panels with ground truth map of spatial locations of the 15 panels; (c) the spectral profile of the five panel signatures p_1, p_2, p_3, p_4, and p_5

2.2 Automatic Target Generation Process (ATGP)

The development of ATGP primarily comes from a need of finding targets of interest in image data when no prior knowledge is available. It implements a sequence of orthogonal complement subspaces from which a succession of targets of interest can be found by finding maximal orthogonal projections. Interestingly, as shown in (Chang, 2013, Chang, 2013, and Greg, 2010), most of such ATGP-generated target pixels turned out to be endmembers. This is certainly not a coincidence because the concept behind ATGP is actually the same as pixel purity index (PPI) except two key differences. PPI requires a very large number of random vectors, called skewers, to find maximal/minimal orthogonal projection (OP) compared to ATGP that finds targets of interest from a sequence of orthogonal projection subspaces with maximal OP. As a result, PPI simultaneously extracts all endmembers, whereas ATGP extracts targets sequentially one at a time. Additionally, PPI takes advantage of random nature in skewers to uncover all possible endmembers as opposed to ATGP which searches for possible target candidates by finding the maximal OP. Nonetheless, both PPI and ATGP use the same principle, OP, in two different ways.

ATGP was originally called automatic target detection and classification algorithm (ATDCA) in (Chang, 2003). It repeatedly implements an orthogonal subspace projector defined by

$$P_{\mathbf{U}}^{\perp}=\mathbf{I}-\mathbf{U}\left(\mathbf{U}^{T}\mathbf{U}\right)^{-1}\mathbf{U}^{T} \tag{2.1}$$

to find targets of interest directly from the data without prior knowledge as follows.

Automatic Target Generation Process (ATGP)

1) Initial condition:

   Select an initial target pixel vector $\mathbf{t}_0=\arg\left\{\max_{\mathbf{r}}\mathbf{r}^{T}\mathbf{r}\right\}$ and an error threshold ε. Set $p=1$ and $\mathbf{U}_0=\left[\mathbf{t}_0\right]$

2) At the pth iteration, apply $P_{\mathbf{t}_0}^{\perp}$ via (1) to all image pixels $\mathbf{r}$ in the image and find the nth target $\mathbf{t}_n$ satisfying

$$\mathbf{t}_p=\arg\left\{\max_{\mathbf{r}}\left[\left(P_{\mathbf{U}_{p-1}}^{\perp}\mathbf{r}\right)^{T}\left(P_{\mathbf{U}_{p-1}}^{\perp}\mathbf{r}\right)\right]\right\} \tag{2.2}$$

   where $\mathbf{U}_{p-1}=\left[\mathbf{t}_1\mathbf{t}_2\cdots\mathbf{t}_{p-1}\right]$ is the target matrix generated at the (p-1)st stage.

   It should be pointed out that (2.2) can be carried out by the following procedure:

   a. Assume that $\{\mathbf{r}_i\}_{i=1}^{N}$ are the set of total data sample vectors. Let i = 1 and $\mathbf{r}_p^{\max}\leftarrow\mathbf{r}_1$.

   b. If $i>1$ calculate $\left(P_{\mathbf{U}_{p-1}}^{\perp}\mathbf{r}_i\right)^{T}\left(P_{\mathbf{U}_{p-1}}^{\perp}\mathbf{r}_i\right)=||P_{\mathbf{U}_{p-1}}^{\perp}\mathbf{r}_i||^2$ and $||P_{\mathbf{U}_{p-1}}^{\perp}\mathbf{r}_i||>||P_{\mathbf{U}_{p-1}}^{\perp}\mathbf{r}_p^{\max}||$ then $\mathbf{r}_p^{\max}\leftarrow\mathbf{r}_i$ and go to step (c). Otherwise, continue.

   c. If $i<N$, the go to step 2. Otherwise, continue.

   d. $\mathbf{t}_p\leftarrow\mathbf{r}_p^{\max}$ and go to step 3.

3). Stopping rule:

   If $\mathbf{t}_p^{T}P_{\mathbf{U}_{p-1}}^{\perp}\mathbf{t}_p>\varepsilon$ (2.3)

   then let $\mathbf{U}_p=\left[\mathbf{U}_{p-1}\mathbf{t}_p\right]=\left[\mathbf{t}_1\mathbf{t}_2\cdots\mathbf{t}_p\right]$ be the p-th target matrix, go to step 2. Otherwise continue.

4). At this stage, ATGP is terminated and the final set of produced target pixel vectors comprises p target pixel vectors, $\left\{\mathbf{t}_0,\mathbf{t}_1,\mathbf{t}_2,\cdots,\mathbf{t}_{p-1}\right\}=\left\{\mathbf{t}_0\right\}\cup\left\{\mathbf{t}_1,\mathbf{t}_2,\cdots,\mathbf{t}_{p-1}\right\}$

It is worth noting that replacing $\mathbf{t}_p$ in (2.3) with $\mathbf{t}_0$ yields

$$\mathbf{t}_0^{T}P_{\mathbf{U}_{p-1}}^{\perp}\mathbf{t}_0 \tag{2.4}$$

which is exactly the orthogonal projection correlation index (OPCI) defined in (Chang, 2003).

2.3 *Pixel Purity Index (PPI)*

The PPI has been widely used for endmember finding due to availability of the popular ENVI software system originally developed by Analytical Imaging and Geophysics (AIG). While the details of the specific steps to implement ENVI's PPI are not available in the literature, a MATLAB version of the PPI described below is only based on the limited published results and our own interpretation. Nevertheless, our algorithm was verified and validated by the PPI with ENVI 3.6, both of which produce the same results.

MATLAB PPI Algorithm

1) *Initialization:*
   (i) Let K be the number of skewers which is a pre-assumed sufficiently large positive integer.
   (ii) Randomly generate a set of K unit vectors called "skewers" $\{\mathbf{skewer}_k\}_{k=1}^{K}$.
2) *PPI count calculation:*
   For each **skewer**$_k$, all the data sample vectors are projected onto **skewer**$_k$ to find sample vectors at its extreme positions and form an extrema set for this particular **skewer**$_k$, denoted by $S_{extrema}(\mathbf{skewer}_k)$. Despite the fact that a different **skewer**$_k$ generates a different extrema set $S_{extrema}(\mathbf{skewer}_k)$, it is very likely that some sample vectors may appear in more than one extrema set. Define an indicator function of a set S, $I_S(\mathbf{r})$ by

$$I_S(\mathbf{r}) = \begin{cases} 1; \text{if } \mathbf{r} \in S \\ 0; \text{if } \mathbf{r} \notin S \end{cases} \text{ and } N_{\mathrm{PPI}}(\mathbf{r}) = \sum\nolimits_k I_{S_{extrema}(\mathbf{skewer}_k)}(\mathbf{r}) \tag{2.5}$$

   where $N_{PPI}(\mathbf{r})$ is defined as the PPI count of sample vector **r**.
3) *Candidate selection:*
   Find the PPI counts $N_{PPI}(\mathbf{r})$ for all the sample vectors defined by (2.5).
4) *Endmember finding:*
   Let t be a threshold value set for the PPI count. Extract all the sample vectors with $N_{PPI}(\mathbf{r}) \geq t$

2.4 *Fast Iterative Pixel Purity Index (FIPPI)*

As noted in Section 2.4, there are three major issues in implementing PPI. One is "how many skewers are needed for PPI to work effectively?" Of course, in order to cover as many directions of interestingness as possible, we would like to have the number of skewers K sufficiently large. But what value of K is considered to be large enough? So far, there is no guideline available for determining the value of K. Another is "what value of the threshold should be selected as a cut-off value for PPI counts to extract endmembers?" Theoretically speaking, the higher the PPI count of a data sample, the more likely the data sample to be an endmember. Unfortunately, according to our extensive experiments, this may not be always true. However, it is generally true that an endmember must have its PPI count at least greater than 0. A third issue is inconsistency caused by skewers that are randomly generated. In other words, the PPI counts obtained for data sample vectors are not reproducible because such PPI counts can vary quite differently if different sets of skewers are used. More precisely, PPI counts produced by running PPI at different times or by different users at the same time are generally not the same due to randomly generated different sets of skewers. Under such circumstances, Fast Iterative Pixel Purity Index (FIPPI) is proposed. First of all, it takes the advantage of Virtual Dimensionality (VD) in (Chang, 2003) to estimate the number of endmembers *p* in the data set. FIPPI starts using the *p* endmembers generated by ATGP, and processes iteratively until it converge. FIPPI uses the value of *p* determined by VD, which solved the problem of *K* in the original PPI. Meanwhile, *p* endmembers is a smaller set of skewers comparing with *K* skewers, the computational complexity drops a lot. Then, FIPPI identifies the pixels that with the maximum of the minimum projection value instead of finding the pixels with PPI Count value greater than threshold *t*, in other words there is no need to determine the value of *t*. Moreover, FIPPI using the pixels found by ATGP as initial condition, so there is no randomness involved in the process.

Fast Pixel Purity Index Algorithm

1) *Initialization:* Find the VD using the Harsanyi–Farrand–Chang (HFC) method in (Harsanyi et al., 1994), and let it be *p*, the number of endmembers required to generate.
2) *Dimensionality Reduction:* Apply the MNF transform for dimensionality reduction and retain the first *p* components. Let $\left\{\mathbf{skewer}_j^{(0)}\right\}_{j=1}^{p}$ be an initial set of skewers generated by selecting those pixels that correspond to target pixels generated by ATGP in (Ren & Chang, 2003).
3) *Iterative Rule:* At iteration $n \geq 0$, for each $\mathbf{skewer}_j^{(n)}$ all the sample vectors are projected onto this particular $\mathbf{skewer}_j^{(n)}$ to find those which are at its extreme positions to form an extrema set, denoted by $S_{extrema}(\mathbf{skewer}_j^{(n)})$. Find the sample vectors that produce the largest $N_{PPI}(\mathbf{r}_j^{(k)})$ defined by (2.5) and let them be denoted by $\left\{\mathbf{r}_j^{(k)}\right\}$.

4) *Stopping Rule:* Form the joint set, $\left\{\textbf{skewer}_j^{(k+1)}\right\}=\left\{\mathbf{r}_j^{(k)}\right\}_{N_{PPI}(\mathbf{r}_j^{(k)})>0}\cup\left\{\textbf{skewer}_j^{(k)}\right\}$. If $\left\{\textbf{skewer}_j^{(k+1)}\right\}=\left\{\textbf{skewer}_j^{(k)}\right\}$, then no new endmembers are added to the skewer set. In this case, the algorithm is terminated. Otherwise, let $k\leftarrow k+1$ and go to step 2.

2.5 Simplex Growing Algorithm (SGA)

In this section, we first briefly review the simplex growing algorithm (SGA) developed by (Chang et al., 2006) starts with one vertex and then begins to grow a simplex by one vertex at a time until it reaches *p* vertices.

A key to making SGA work hinges on how to appropriately select new vertices to augment growing simplexes. According to N-FINDR for a given positive integer *p* a simplex formed by *p* endmembers is one that produces the maximum volume among all possible simplexes formed by any set of *p* data sample vectors. Using this as a criterion SGA grows the current *k*-vertex simplex $S(\mathbf{e}^{(0)},\mathbf{e}^{(1)},\cdots,\mathbf{e}^{(k-1)})$ to a (*k*+1)–vertex simplex $S(\mathbf{e}^{(0)},\mathbf{e}^{(1)},\cdots,\mathbf{e}^{(k-1)},\mathbf{e}^{(k)})$ by finding a new (*k*+1)st vertex $\mathbf{e}^{(k)}$ so that the new (*k*+1)-vertex simplex $S(\mathbf{e}^{(0)},\mathbf{e}^{(1)},\cdots,\mathbf{e}^{(k-1)},\mathbf{e}^{(k)})$ produces its volume no less than volumes of all possible (*k*+1)–vertex simplexes $S(\mathbf{e}^{(0)},\mathbf{e}^{(1)},\cdots,\mathbf{e}^{(k-1)},\mathbf{r})$ augmented by any other data sample vector **r**. The detailed implementation of the above growing simplex process is summarized as follows.

Simplex Growing Algorithm (SGA)

1. Initialization:

(a) Let *p* be the number of endmembers to be generated.

(b) There are two ways to generate random initial endmembers for SGA.

(i) Randomly select a data sample vector as an initial endmember $\mathbf{e}^{(0)}$ and set $k=0$. In this case, the SGA is referred to as 1-SGA.

(ii) Randomly select a pair of two data sample vectors $(\mathbf{e}^{(0)},\mathbf{e}^{(1)})$ to form as a random degenerate 2-dimesnional simplex which is a line segment connecting $\mathbf{e}^{(0)}$ and $\mathbf{e}^{(1)}$. Set $k=1$. In this case, the SGA is referred to as 2-SGA.

2. At $k\geq 0$ and for each sample vector **r**, we calculate $V(\mathbf{e}^{(0)},\cdots,\mathbf{e}^{(k)},\mathbf{r})$ defined by

$$V(\mathbf{e}^{(0)},\cdots,\mathbf{e}^{(k)},\mathbf{r})=\frac{\mathrm{Det}\left(\begin{bmatrix}1 & 1 & \ldots & 1 & 1\\ \mathbf{e}^{(0)} & \mathbf{e}^{(1)} & \ldots & \mathbf{e}^{(k)} & \mathbf{r}\end{bmatrix}\right)}{(k+1)!} \tag{2.6}$$

which is the volume of the simplex $S(\mathbf{e}^{(0)}, \mathbf{e}^{(1)}, \cdots, \mathbf{e}^{(k)}, \mathbf{r})$ specified by vertices $\mathbf{e}^{(0)}, \mathbf{e}^{(1)}, \cdots, \mathbf{e}^{(k)}, \mathbf{r}$ and the definition of the matrix determinant in (2.6) can be found in any linear algebra book. Since the matrix $\begin{bmatrix} 1 & 1 & \ldots & 1 & 1 \\ \mathbf{e}^{(0)} & \mathbf{e}^{(1)} & \ldots & \mathbf{e}^{(k)} & \mathbf{r} \end{bmatrix}$ in (2.6) is not necessarily a square matrix, a DR technique such as PCA or MNF is required to reduce the original data dimensionality L to the dimension k+1.

3. Find $\mathbf{e}^{(k+1)}$ that yields the maximum of (1), that is,

$$\mathbf{e}^{(k+1)} = \arg\left\{\max_{\mathbf{r}}\left[V(\mathbf{e}^{(0)}, \cdots, \mathbf{e}^{(k)}, \mathbf{r})\right]\right\} \tag{2.7}$$

4. Stopping rule:

If $k < p-1$, then $k \leftarrow k+1$ and go step 2. Otherwise, the final set of $\left\{\mathbf{e}^{(0)}, \mathbf{e}^{(1)}, \cdots, \mathbf{e}^{(p-1)}\right\}$ is the desired p endmembers.

Chapter 3: Recursive Band Processing of Automatic Target Generation Process (RBP-ATGP)

3.1 Introduction

Automatic Target Generation Process (ATGP) is an unsupervised target detection technique which has been widely used in hyperspectral target detection (Chang, 2003). It implements a succession of orthogonal subspace projections to find targets which have maximal leaking residuals from subspaces generated by previous targets. Generally speaking, there are three types of unsupervised targets are of interest in the literature. One is endmembers whose spectral signatures are pure. There are two challenging issues; signature purity and number of endmembers need to be addressed. A second type is anomalies. In the past anomaly detection has been studied extensively (Reed and Yu, 1990). It makes use of data sample correlation/covariance to capture targets whose spectral profiles are different their surrounding data sample vectors. In analogy with endmembers, there are also two challenging issues involved in anomaly detection which have not received much attention in the past. One is how to discriminate its detected anomalies one from another. The other is how many distinct anomalies are present in the data. A third type is unknown targets whose signatures are spectrally distinct. In order to differentiate the detected unknown targets a general approach is to design a target finding process which can detect different distinct targets in a sequence of target detection. Due to its simplicity and easy implementation, ATGP receives a great interest in many applications. However, the original ATGP cannot process hyperspectral data progressively, this chapter takes an innovative look of ATGP, which implement target detection using ATGP band by band, to be called Recursive band processing of ATGP (RBP-ATGP). The work of RBP-ATGP is inspired by one of the hyperspectral data acquisition format – BSQ, noted in the previous chapters. Several advantages can be gained from RBP-ATGP. Many advantages can be gained from RBP-ATGP. First of all, RBP-ATGP can be implemented whenever bands are available without waiting for full bands of data information being collected according to the BSQ data acquisition format. Second, a spectral profile of RBP-ATGP detected abundance can be provided. Such target detection profiles allow users to see changes in target detection maps by ATGP in each band which are not available in one-shot ATGP using full band information. Third, using progressive target detection map profiles enables users to find weak targets which may be captured by certain bands and later be overwhelmed and dominated by strong targets subsequently detected by ATGP. The targets of this type such as moving targets are generally not shown in the final ATGP-detection map, i.e., detection map produced by ATGP using full bands. Fourth, with the help of RBP-ATGP, progressive band selection can be conducted without the need of a priori, the number of bands need to be selected. The hint and clue of choosing the bands with essential information can be gained at the same time. In addition, the RBP-ATGP is specifically designed for BSQ data acquisition format, and data processing can be conducted with limited bandwidth.

3.2 <u>Derivations of RBP-ATGP</u>

In order for ATGP to be processed band by band a key issue is to implement the succession of orthogonal projection progressively band by band. Let

$$\begin{aligned}\mathbf{U}_{l\times p} &= \left[\mathbf{t}_1\mathbf{t}_2\cdots\mathbf{t}_p\right]\\ &= \begin{bmatrix} t_{11} & t_{21} & \cdots & t_{(p-1)1} & t_{p1}\\ t_{12} & t_{22} & & t_{(p-1)2} & t_{p2}\\ \vdots & \ddots & \ddots & & \\ t_{1(l-1)} & t_{2(l-1)} & \cdots & t_{(p-1)(l-1)} & t_{p(l-1)}\\ t_{1l} & t_{2l} & \cdots & t_{(p-1)l} & t_{pl}\end{bmatrix} = \begin{bmatrix}\mathbf{U}_{(l-1)\times p}\\ \mathbf{t}^T(l)\end{bmatrix}\end{aligned} \tag{3.1}$$

Then

$$\begin{aligned}\left[\mathbf{U}_{l\times p}^T\mathbf{U}_{l\times p}\right]^{-1} &= \left(\left[\mathbf{U}_{(l-1)\times p}^T \quad \mathbf{t}(l)\right]\begin{bmatrix}\mathbf{U}_{(l-1)\times p}\\ \mathbf{t}^T(l)\end{bmatrix}\right)^{-1}\\ &= \left(\mathbf{U}_{(l-1)\times p}^T\mathbf{U}_{(l-1)\times p} + \mathbf{t}(l)\mathbf{t}^T(l)\right)^{-1}\end{aligned} \tag{3.2}$$

where $\mathbf{U}_{(l-1)\times p} = \begin{bmatrix} t_{11} & t_{21} & \cdots & t_{(p-1)1} & t_{p1}\\ t_{12} & t_{22} & \vdots & t_{(p-1)2} & t_{p2}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ t_{1(l-1)} & t_{2(l-1)} & \cdots & t_{(p-1)(l-1)} & t_{p((l-1)}\end{bmatrix}$ is a $(l-1)\times p$ matrix and $\mathbf{t}(l) = \left(t_{1l}, t_{2l}, \ldots, t_{pl}\right)^T$ is a *p*-dimensional vector. Applying Woodbury's identity

$$\left[\mathbf{A}+\mathbf{u}\mathbf{v}^T\right]^{-\mathbf{1}} = \mathbf{A}^{-1} - \frac{\left[\mathbf{A}^{-1}\mathbf{u}\right]\left[\mathbf{v}^T\mathbf{A}^{-1}\right]}{1+\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \tag{3.3}$$

to $\left[\mathbf{U}_{l\times p}^T\mathbf{U}_{l\times p}\right]^{-1}$ in (3.2) yields

$$\left[\mathbf{U}_{l\times p}^T\mathbf{U}_{l\times p}\right]^{-1} = \left[\mathbf{U}_{(l-1)\times p}^T\mathbf{U}_{(l-1)\times p}\right]^{-1} - \frac{\left[\left(\mathbf{U}^T_{(l-1)\times p}\mathbf{U}_{(l-1)\times p}\right)^{-1}\mathbf{t}(l)\right]\left[\mathbf{t}^T(l)\left(\mathbf{U}^T_{(l-1)\times p}\mathbf{U}_{(l-1)\times p}\right)^{-1}\right]}{1+\mathbf{t}^T(l)\left(\mathbf{U}^T_{(l-1)\times p}\mathbf{U}_{(l-1)\times p}\right)^{-1}\mathbf{t}(l)} \tag{3.4}$$

$$\begin{aligned}
\mathbf{U}^{\#}_{l\times p} &= \left[\mathbf{U}^T_{l\times p}\mathbf{U}_{l\times p}\right]^{-1}\mathbf{U}^T_{l\times p} \\
&= \left[\mathbf{U}^T_{(l-1)\times p}\mathbf{U}_{(l-1)\times p}\right]^{-1}\left[\mathbf{U}^T_{(l-1)\times p} \quad \mathbf{t}(l)\right] - \left\{\frac{\left[\left(\mathbf{U}^T_{(l-1)\times p}\mathbf{U}_{(l-1)\times p}\right)^{-1}\mathbf{t}(l)\right]\left[\mathbf{t}^T(l)\left(\mathbf{U}^T_{(l-1)\times p}\mathbf{U}_{(l-1)\times p}\right)^{-1}\right]}{1+\mathbf{t}^T(l)\left(\mathbf{U}^T_{(l-1)\times p}\mathbf{U}_{(l-1)\times p}\right)^{-1}\mathbf{t}(l)}\right\}\left[\mathbf{U}^T_{(l-1)\times p} \quad \mathbf{t}(l)\right] \\
&= \left[\left(\mathbf{U}^T_{(l-1)\times p}\mathbf{U}_{(l-1)\times p}\right)^{-1}\mathbf{U}^T_{(l-1)\times p} \quad \boldsymbol{\rho}_{l|(l-1)}\right] - \frac{1}{1+\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}}\left[\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{U}^T_{(l-1)\times p} \quad \boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{t}(l)\right]
\end{aligned} \tag{3.5}$$

where $\boldsymbol{\rho}_{l|(l-1)} = \left[\left(\mathbf{U}^T_{(l-1)\times p}\mathbf{U}_{(l-1)\times p}\right)^{-1}\mathbf{t}(l)\right]$ is a *p*-dimensional vector.

$$\begin{aligned}
P^{\perp}_{\mathbf{U}_{l\times p}} &= \mathbf{I}_{l\times l} - \mathbf{U}_{l\times p}\mathbf{U}^{\#}_{l\times p} \\
&= \mathbf{I}_{l\times l} - \begin{bmatrix} \mathbf{U}_{(l-1)\times p} \\ \mathbf{t}^T(l) \end{bmatrix}\left(\left[\left(\mathbf{U}^T_{(l-1)\times p}\mathbf{U}_{(l-1)\times p}\right)^{-1}\mathbf{U}^T_{(l-1)\times p} \quad \boldsymbol{\rho}_{l|(l-1)}\right] - \frac{1}{1+\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}}\left[\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{U}^T_{(l-1)\times p} \quad \boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{t}(l)\right]\right) \\
&= \begin{bmatrix} \mathbf{I}_{(l-1)\times(l-1)} & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \mathbf{U}_{(l-1)\times p}\left(\mathbf{U}^T_{(l-1)\times p}\mathbf{U}_{(l-1)\times p}\right)^{-1}\mathbf{U}^T_{(l-1)\times p} & \mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)} \\ \boldsymbol{\rho}^T_{l\|(l-1)}\mathbf{U}^T_{(l-1)\times p} & \mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)} \end{bmatrix} \\
&+ \frac{1}{1+\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}}\begin{bmatrix} \mathbf{U}_{(l-1)\times p} \\ \mathbf{t}^T(l) \end{bmatrix}\left[\boldsymbol{\rho}_{l\backslash(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{U}^T_{(l-1)\times p} \quad \boldsymbol{\rho}_{l\backslash(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{t}(l)\right] \\
&= \begin{bmatrix} \mathbf{I}_{(l-1)\times(l-1)} & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \mathbf{U}_{(l-1)\times p}\mathbf{U}^{\#}_{(l-1)\times p} & \mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)} \\ \boldsymbol{\rho}^T_{l\|(l-1)}\mathbf{U}^T_{(l-1)\times p} & \mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)} \end{bmatrix} \\
&+ \frac{1}{1+\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}}\begin{bmatrix} \mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{U}^T_{(l-1)\times p} & \mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{t}(l) \\ \mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{U}^T_{(l-1)\times p} & \mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{t}(l) \end{bmatrix} \\
&= \begin{bmatrix} P^{\perp}_{\mathbf{U}_{(l-1)\times p}} & -\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)} \\ -\left[\mathbf{U}_{(l-1)\times p}\left(\mathbf{U}^T_{(l-1)\times p}\mathbf{U}_{(l-1)\times p}\right)^{-1}\mathbf{t}(l)\right]^T & 1-\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)} \end{bmatrix} \\
&+ \frac{1}{1+\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}}\begin{bmatrix} \mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{U}^T_{(l-1)\times p} & \mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{t}(l) \\ \mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{U}^T_{(l-1)\times p} & \mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}^T_{l|(l-1)}\mathbf{t}(l) \end{bmatrix}
\end{aligned} \tag{3.6}$$

Let $\mathbf{r}_l = \left(r_1, r_2, \cdots, r_{l-1}, r_l\right)^T = \left(\mathbf{r}^T_{l-1}, r_l\right)$ Then

$$
\begin{aligned}
\mathbf{r}_l^T P_{\mathbf{U}_{l\times p}}^{\perp}\mathbf{r}_l &= \begin{bmatrix}\mathbf{r}_{l-1}^T & r_l\end{bmatrix} P_{\mathbf{U}_{l\times p}}^{\perp}\begin{bmatrix}\mathbf{r}_{l-1}\\ r_l\end{bmatrix}\\
&= \begin{bmatrix}\mathbf{r}_{l-1}^T & r_l\end{bmatrix}\begin{bmatrix} P_{\mathbf{U}_{(l-1)\times p}}^{\perp} & -\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\\ -\left[\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\right]^T & 1-\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\end{bmatrix}\begin{bmatrix}\mathbf{r}_{l-1}\\ r_l\end{bmatrix}\\
&+\frac{1}{1+\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}}\begin{bmatrix}\mathbf{r}_{l-1}^T & r_l\end{bmatrix}\begin{bmatrix}\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{U}_{(l-1)\times p}^T & \mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{t}(l)\\ \mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{U}_{(l-1)\times p}^T & \mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{t}(l)\end{bmatrix}\begin{bmatrix}\mathbf{r}_{l-1}\\ r_l\end{bmatrix}\\
&= \begin{bmatrix}\mathbf{r}_{l-1}^T P_{\mathbf{U}_{(l-1)\times p}}^{\perp} - r_l\left[\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\right]^T & -\mathbf{r}_{l-1}^T\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)} + r_l\left(1-\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\right)\end{bmatrix}\begin{bmatrix}\mathbf{r}_{l-1}\\ r_l\end{bmatrix}\\
&+\frac{1}{1+\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}}\begin{bmatrix}\mathbf{r}_{l-1}^T\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{U}_{(l-1)\times p}^T + r_l\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{U}_{(l-1)\times p}^T & \mathbf{r}_{l-1}^T\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{t}(l) + r_l\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{t}(l)\end{bmatrix}\begin{bmatrix}\mathbf{r}_{l-1}\\ r_l\end{bmatrix}\\
&= \mathbf{r}_{l-1}^T P_{\mathbf{U}_{(l-1)\times p}}^{\perp}\mathbf{r}_{l-1} - r_l\left[\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\right]^T\mathbf{r}_{l-1} - \mathbf{r}_{l-1}^T\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}r_l + r_l\left(1-\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\right)r_l\\
&+\frac{1}{1+\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}}\left\{\mathbf{r}_{l-1}^T\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{U}_{(l-1)\times p}^T\mathbf{r}_{l-1} + r_l\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{U}_{(l-1)\times p}^T\mathbf{r}_{l-1}\right\}\\
&+\frac{1}{1+\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}}\left\{\mathbf{r}_{l-1}^T\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{t}(l)r_l + r_l\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{t}(l)r_l\right\}\\
&= \mathbf{r}_{l-1}^T P_{\mathbf{U}_{(l-1)\times p}}^{\perp}\mathbf{r}_{l-1} - 2r_l\left[\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\right]^T\mathbf{r}_{l-1} + r_l\left(1-\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\right)r_l\\
&+\frac{1}{1+\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}}\left(\mathbf{r}_{l-1}^T\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)} + \boldsymbol{\rho}_{l|(l-1)}^T\mathbf{t}(l)r_l\right)\left(\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{U}_{(l-1)\times p}^T\mathbf{r}_{l-1} + \boldsymbol{\rho}_{l|(l-1)}^T\mathbf{t}(l)r_l\right)
\end{aligned}
\tag{3.7}
$$

$$
\begin{aligned}
\mathbf{r}_l^T P_{\mathbf{U}_{l\times p}}^{\perp}\mathbf{r}_l &= \mathbf{r}_{l-1}^T P_{\mathbf{U}_{(l-1)\times p}}^{\perp}\mathbf{r}_{l-1} - 2r_l\left[\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\right]^T\mathbf{r}_{l-1} + r_l\left(1-\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}\right)r_l\\
&+\frac{1}{1+\mathbf{t}^T(l)\boldsymbol{\rho}_{l|(l-1)}}\left(\mathbf{r}_{l-1}^T\mathbf{U}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)} + \boldsymbol{\rho}_{l|(l-1)}^T\mathbf{t}(l)r_l\right)\left(\boldsymbol{\rho}_{l|(l-1)}^T\mathbf{U}_{(l-1)\times p}^T\mathbf{r}_{l-1} + \boldsymbol{\rho}_{l|(l-1)}^T\mathbf{t}(l)r_l\right)
\end{aligned}
\tag{3.8}
$$

where $\boldsymbol{\rho}_{l|(l-1)} = \left[\left(\mathbf{U}_{(l-1)\times p}^T\mathbf{U}_{(l-1)\times p}\right)^{-1}\mathbf{t}(l)\right]$ is a p-dimensional vector. According to (3.8) $\mathbf{r}^T P_{\mathbf{U}_{(l-1)\times p}}^{\perp}\mathbf{r}$ is identical to $\left(P_{\mathbf{U}_{(l-1)\times p}}^{\perp}\mathbf{r}\right)^T\left(P_{\mathbf{U}_{(l-1)\times p}}^{\perp}\mathbf{r}\right)$ due to the fact that $\left(P_{\mathbf{U}_{(l-1)\times p}}^{\perp}\right)^T = P_{\mathbf{U}_{(l-1)\times p}}^{\perp}$ and $P_{\mathbf{U}_{p-1}}^{\perp}$ is idempotent. That is, $\left(P_{\mathbf{U}_{(l-1)\times p}}^{\perp}\mathbf{r}\right)^T\left(P_{\mathbf{U}_{(l-1)\times p}}^{\perp}\mathbf{r}\right) = \mathbf{r}^T\left(P_{\mathbf{U}_{(l-1)\times p}}^{\perp}\right)^T P_{\mathbf{U}_{(l-1)\times p}}^{\perp}\mathbf{r} = \mathbf{r}^T P_{\mathbf{U}_{(l-1)\times p}}^{\perp}\mathbf{r}$.

3.3 Detailed RBP-ATGP Algorithm

According to Eq. (3.6), $P_{\mathbf{U}_{l\times p}}^{\perp}$ in (2.1) carried out by ATGP can be updated by $P_{\mathbf{U}_{(l-1)\times p}}^{\perp}$ without re-calculating $P_{\mathbf{U}_{l\times p}}^{\perp}$; $\mathbf{t}_p^T P_{\mathbf{U}_{l\times p}}^{\perp}\mathbf{t}_p$ in (2.5) can be updated by (3.8) via $\mathbf{t}_p^T P_{\mathbf{U}_{(l-1)\times p}}^{\perp}\mathbf{t}_p$. By virtue of (3.8) a recursive band processing of ATGP (RBP-ATGP) can be derived as follows.

Recursive Band Processing of ATGP (RBP-ATGP) Algorithm

Outer Loop indexed by *l* from VD to *L*

Inner Loop indexed by *p* from 1 to VD

1. Initial condition:

Let $l \geq p$ to avoid the issue of matrix singularity. Find an initial target pixel vector $\mathbf{t}_p^{(l)} = \arg\left\{\max_{\mathbf{r}_l} \mathbf{r}_l^T \mathbf{r}_l\right\}$. Set $\mathbf{U}_p^{(l)} = \left[\mathbf{t}_1^{(l)}\mathbf{t}_2^{(l)} \cdots \mathbf{t}_p^{(l)}\right]$ and calculate $P_{\mathbf{U}_p^{(l)}}^{\perp}$.

2. At the lth iteration, find $\mathbf{t}_p^{(l)}$ by maximizing $\mathbf{r}^T P_{\mathbf{U}_{l\times p}}^{\perp}\mathbf{r}$ via (3.8) over all data sample vectors, $\left\{\mathbf{r}_l\ (i)\right\}_{i=1}^{N}$. This can be done by the following
    a. Initial conditions: Let $\max_l^p = \left(\mathbf{r}_l\ (i)\right)^T P_{\mathbf{U}_{l\times p}}^{\perp}\mathbf{r}_l\ (i)$ and $\mathbf{t}_p^{(l)} \leftarrow \mathbf{r}_l\ (i)$. Set i =2.
    b. At the ith iteration, calculate $\left(\mathbf{r}_l(i)\right)^T P_{\mathbf{U}_{l\times p}}^{\perp}\mathbf{r}_l(i)$
    c. If $\left(\mathbf{r}_l(i)\right)^T P_{\mathbf{U}_{l\times p}}^{\perp}\mathbf{r}_l(i) > \max_l^p$, then $\max_l^p = \left(\mathbf{r}_l(i)\right)^T P_{\mathbf{U}_{l\times p}}^{\perp}\mathbf{r}_l(i)$ and $\mathbf{t}_p^{(l)} \leftarrow \mathbf{r}_l(i)$. Otherwise continue.
    d. If $i < N$, then $i \leftarrow i+1$ and go step 1.b. Otherwise, the algorithm is terminated and $\mathbf{t}_l^{(p)}$ is already found.
3. Let $l \leftarrow l+1$ and use (3.5) and (3.6) to update $\mathbf{U}_{l\times p}^{\#}$ and $P_{\mathbf{U}_{l\times p}}^{\perp}$ via previously calculated $\mathbf{U}_{(l-1)\times p}^{\#}$ and $P_{\mathbf{U}_{(l-1)\times p}}^{\perp}$.
4. End (**Inner Loop**)

Go to step 2 until $l = L$.

End (**Outer Loop**)

From the above designed recursive algorithm RBP-ATGP can generate an $L \times L$ target matrix, TM denoted by

$$\mathrm{TM} = \mathbf{M}^{\mathrm{ATGP}} = \left[\mathbf{t}_p^{(l)}\right]_{L\times L} = \begin{bmatrix} \mathbf{t}_1^{(1)} & \mathbf{t}_1^{(2)} & \cdots & \mathbf{t}_1^{(L-1)} & \mathbf{t}_1^{(L)} \\ \mathbf{t}_2^{(1)} & \mathbf{t}_2^{(2)} & \cdots & \mathbf{t}_2^{(L-1)} & \mathbf{t}_2^{(L)} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{t}_{L-1}^{(1)} & \vdots & \ddots & \mathbf{t}_{L-1}^{(L-1)} & \mathbf{t}_{L-1}^{(L)} \\ \mathbf{t}_L^{(1)} & \mathbf{t}_L^{(2)} & \cdots & \mathbf{t}_L^{(L-1)} & \mathbf{t}_L^{(L)} \end{bmatrix} \tag{3.9}$$

with $\mathbf{t}_p^{(l)}$ being the pth target generated by ATGP using the first l bands as shown in Fig. 3.1 where x and y axes denote the number of bands, n_{B}, used to perform ATGP and $\mathbf{s}_p$ is the pth target generated by ATGP. So, when ATGP uses all full bands $\mathbf{t}_p^{(L)} = \mathbf{t}_p$ is the pth target as it was originally designed in (Chang, 2003). That is, ATGP generates a set of target signal vectors $\left\{\mathbf{t}_j^{(L)}\right\}_{j=1}^{p}$ which can be arranged as a target vector, TV = $\mathbf{V}^{\text{target}}$

$$\mathrm{TV} = \mathbf{V}^{\mathrm{ATGP}} = \left(\mathbf{t}_1^{(L)}, \mathbf{t}_2^{(L)}, \cdots, \mathbf{t}_p^{(L)}\right) \tag{3.10}$$

corresponding to the last L^{th} column in $\mathbf{M}^{\mathrm{ATGP}}$ in Fig. 3.1.

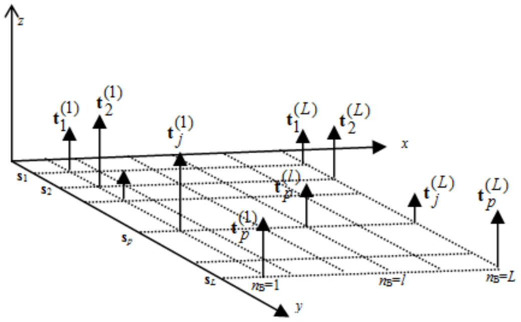


Figure 3.1. L^2 target pixels, $\left\{\mathbf{t}_p^{\,l}\right\}_{p=1,l=1}^{L,L}$ generated by RBP-ATGP

It is the $\mathbf{M}^{\mathrm{target}}$ in (3.9) generated by RBP-ATGP distinguishes itself from the target vector $\mathbf{V}^{\mathrm{ATGP}}$ in (3.10) generated by ATGP. This is because for each j target ATGP generates only one single target, which is $\mathbf{t}_j^{(L)}$, compared to PBP-ATGP which can generate at most L different targets, $\left\{\mathbf{t}_j^{(l)}\right\}_{l=1}^{L} = \left\{\mathbf{t}_j^{(1)}, \mathbf{t}_j^{(2)}, \cdots, \mathbf{t}_j^{(L)}\right\}$ where $\mathbf{t}_j^{(i)}$ may not be same as $\mathbf{t}_j^{(k)}$ if $i \neq k$ for each j.

3.4 *Experiments*

3.4.1 Synthetic Data

A. Synthetic TI Data Set

Fig. 3.2 shows progressive detection profiles of these five mineral signatures, A,B,C,K,M for TI where x, y and z axes denote found signatures, the number of the first bands being used for data processing, denoted n_l and RBP-ATGP detected abundance fractions respectively. As shown in Fig. 3.2 the signatures s_1, s_2, s_3, s_4 and s_5 in x-axis are the first five targets found by RBP-ATGP and the five mineral signatures, A,B,C,K,M are represented by the panel pixels, A(1,1), K(1,1), M(1,1), B(1,1) and C(1,1) correspond to the first panel pixel of the five 4×4 panels in five rows, each of which is located at the upper left corner in each row in the first column in Fig. 2.2.

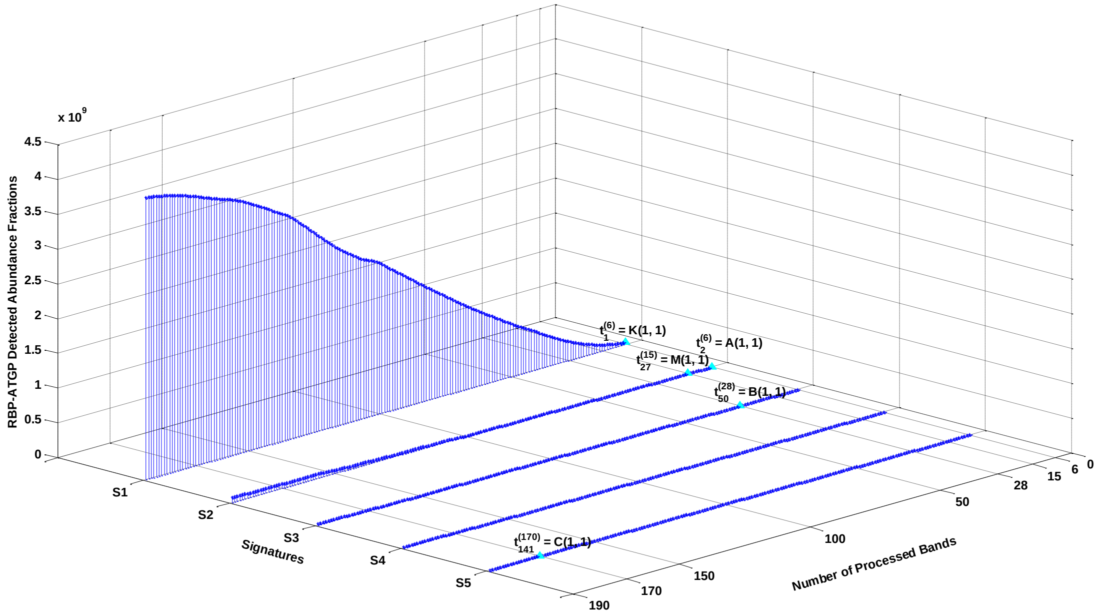
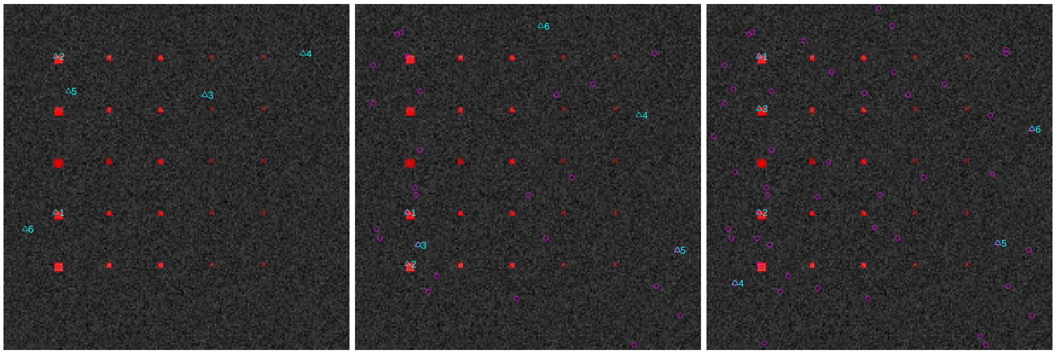


Figure 3.2. 3D plot of detection progressive maps of five signatures versus n_l for TI

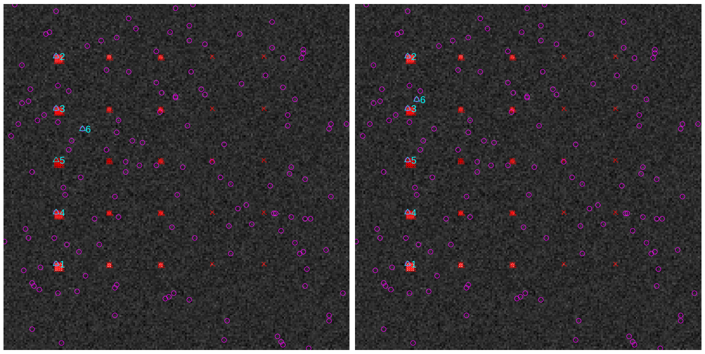
Fig. 3.3 also shows five RBP-ATGP found target maps using different numbers of bands starting with n_l = 6 in Fig. 3.3 (a) progressively increasing to n_l =15 in Fig. 3.3 (b), n_l = 28 in Fig. 3.3 (c), n_l =170 in Fig. 3.3 (d) and finally reaching the full bands, n_l =189 in Fig. 3.3 (e), each of which shows the targets found in particular transition bands specified by Fig. 3.2. More specifically, the bands underneath the figures in Fig. 3.3 were those actually picked up panel pixels where the red crosses indicate ground truth pixels and triangles highlight the targets found in the current band with the numbers next to the triangles indicating the order of the targets being found. Also, the targets found in the previous bands are also marked by the magenta color.



(a) 1st 6 bands

(b) 1st 15 bands

(c) 1st 28 bands



(d) 1st 170 bands (e) 1st 189 bands

Figure 3.3. Progressive target detection maps by RBP-ATGP with different numbers of processed bands: (a) n_l = 6 (b) n_l = 15 (c) n_l = 28 (d) n_l = 170 (e) n_l = 189

Table 3.1 summarizes 6 target pixels found by RBP-ATGP when n_l varies from 6 bands, 15 bands, 28 bands, 170 bands to full 189 bands, to extract panel pixels corresponding to five mineral signatures, A, B, C, K and M. The panel pixels $\mathbf{t}_j^{(l)}$ in the 2nd column of the table were identified as the jth target among the 6 target pixels found by ATGP using the first l bands. The value of j indicates the jth order of the target found by ATGP when the particular lth band was the new band added to process ATGP.

Table 3.1. Summary of panel pixels being identified by RBP-ATGP versus n_l for TI

n_l	Panel pixels found by RBP-ATGP
6	$\mathrm{K} = \mathbf{t}_1^{(6)}$, $\mathrm{A} = \mathbf{t}_2^{(6)}$
15	$\mathrm{K} = \mathbf{t}_1^{(15)}$, $\mathrm{M} = \mathbf{t}_2^{(15)}$
28	$\mathrm{A} = \mathbf{t}_1^{(28)}$, $\mathrm{K} = \mathbf{t}_2^{(28)}$, $\mathrm{B} = \mathbf{t}_3^{(28)}$
170, 189	$\mathrm{M} = \mathbf{t}_1^{(170)}$, $\mathrm{A} = \mathbf{t}_2^{(170)}$, $\mathrm{B} = \mathbf{t}_3^{(170)}$, $\mathrm{K} = \mathbf{t}_4^{(170)}$, $\mathrm{C} = \mathbf{t}_5^{(170)}$

Table 3.2 also tabulates all panel pixels that were found by RBP-ATGP corresponding to the five mineral signatures with using the minimal n_l where the value of the subscript "j" in $\mathbf{t}_j^{(l)}$ listed in the 3rd column is its order appearing in the entire sequence of RBP-ATGP-generated target pixels and the order of the five panel pixels to be found by RBP-ATGP in the 4th column by their particular bands where order of the signature found among 6 ATGP-found targets when the total number of bands up to the particular band were being used to process ATGP.

For example, the panel pixel B(1,1) found to be the 3rd ATGP target in Fig. 3.3(c), $\mathrm{B} = \mathbf{t}_3^{(28)}$ when n_l = 28 was used. But if we include the previous RBP-ATGP generated targets found by using the number pf processed bands, n_l < 28, the panel pixel B(1,1) is actually found as the 50th target by RBP-ATGP, i.e., $\mathrm{B} = \mathbf{t}_{50}^{(28)}$. Similarly, C(1,1) was picked up as the

5th target by RBP-ATGP in Fig. 3.3(d) when the minimal n_l =170, but it was actually found as 141st ATGP target by counting all previous generated RBP-ATGP targets.

Table 3.2. Summary of minimal n_l to identify signatures and their orders found in their particular bands for TI

Signatures found by RBP-ATGP	Minimal n_l identifying signatures	Order of signatures found by RBP-ATGP	Order of signatures found using bands up to its particular band
$A = \mathbf{t}_2^{(6)}$	6	2	2
$B = \mathbf{t}_{50}^{(28)}$	28	50	3
$C = \mathbf{t}_{141}^{(170)}$	170	141	5
$K = \mathbf{t}_1^{(6)}$	6	1	1
$M = \mathbf{t}_{27}^{(15)}$	15	27	2

As noted, the n_l is defined as the number of l bands used to process ATGP where these l bands starts with the 1st band and ends with the lth band and the "l" is used to as an index to track the lth band. So, if two different targets, $\mathbf{t}_j^{(l)}$ and $\mathbf{t}_k^{(m)}$ with n_l, n_m, used to process ATGP then these targets will be first ranked by n_l. If $l < m$, then $\mathbf{t}_j^{(l)}$ is more significant than $\mathbf{t}_k^{(m)}$ because $\mathbf{t}_j^{(l)}$ can be found by ATGP with smaller n_l than that used to extract $\mathbf{t}_k^{(m)}$. If both $\mathbf{t}_j^{(l)}$ and $\mathbf{t}_k^{(m)}$ are found with the same $n_l = n_m$ and $j < k$, then $\mathbf{t}_j^{(l)}$ will be ranked higher than $\mathbf{t}_k^{(m)}$ since $\mathbf{t}_j^{(l)}$ is found ahead of $\mathbf{t}_k^{(l)}$. Therefore the target set, $\left\{\mathbf{t}_p^{(l)}\right\}$ found by RBP-ATGP can be ranked by their significance according to two indices, superscript "l" and the subscript "p". More specifically, the value of "l" is first to rank targets. If the values of l are the same, the subscript "p" will be then used to prioritize their orders. In other words, when two targets found by the same "n_l" and their significance should be ranked by their found order, "p".

For a given n_l ATGP is implemented to find p targets. In this case, for two targets $\mathbf{t}_j^{(l)}$ and $\mathbf{t}_k^{(l)}$ with $j < k$ found by ATGP using the same n_l it implies that $\mathbf{t}_j^{(l)}$ is found prior to $\mathbf{t}_k^{(m)}$. This indicates that $\mathbf{t}_j^{(l)}$ is more significant than $\mathbf{t}_k^{(m)}$. So, for example, according to Table 4.1 the two mineral signatures A and K were identified by $\mathrm{A}(1{,}1) = \mathbf{t}_2^{(6)} = \mathrm{K}(1{,}1) = \mathbf{t}_1^{(6)}$ and both A and K were found by RBP-ATGP using the same n_l = 6. Now if "n_l" is increased from 6 to 15, then the two mineral signatures K and M were identified by $\mathrm{K}(1{,}1) = \mathbf{t}_1^{(15)}, \mathrm{M}(1{,}1) = \mathbf{t}_2^{(15)}$ where K once again was picked up by ATGP using the first 15 bands and M was the first time to be found by ATGP as the 2nd ATGP target. In this case, the minimal n_l for A, K and M were 6, 6 and 15 respectively, but their found orders by RBP-ATGP were actually 1, 2 and 27 as tabulated in Table 3.2. Tables 3.1 and 3.2 offer such advantage of RBP-ATGP over ATGP in the sense that RBP-ATGP keeps track of the targets it detects and also record these targets in its data base.

B. Synthetic TE Data Set

Same experiments conducted for TI were also performed for TE. Fig. 3.4 shows progressive detection profiles of these five mineral signatures, A, B, C, K, M for TE where *x*, *y* and *z* axes denote found signatures, n_l and RBP-ATGP detected abundance fractions respectively. Similar to Fig. 3.2, Fig. 3.4 also shows the five mineral signatures, A, B, C, K, M were found by the panel pixels, A(1,1), K(1,1), M(1,1), B(1,1) and C(1,1) correspond to the first panel pixel of the five 4×4 panels in five rows, each of which is located at the upper left corner in each row in the first column in Fig. 2.2.

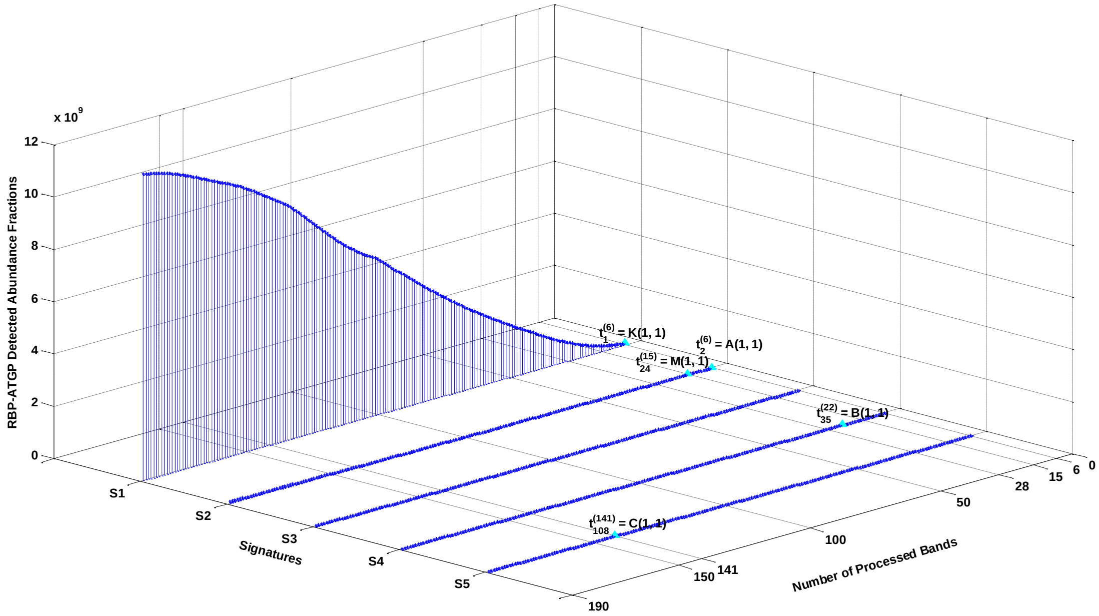
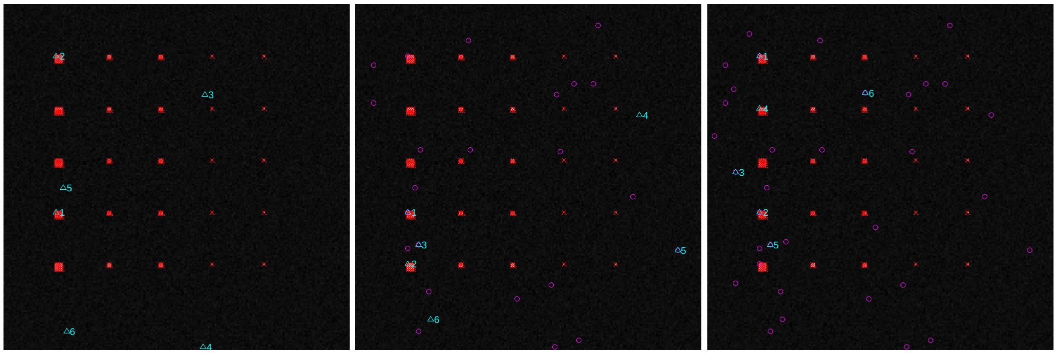
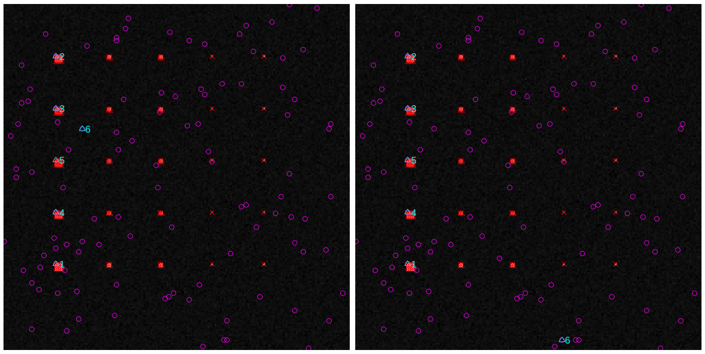


Figure 3.4. 3D plot of detection progressive maps of five signatures versus n_l for TE



(a) 1st 6 bands (b) 1st 15 bands (c) 1st 22 bands



(d) 1st 141 bands (e) 1st 189 bands

Figure 3.5. Progressive target detection maps by RBP-ATGP with different numbers of processed bands: (a) n_l = 6 (b) n_l = 15 (c) n_l = 22 (d) n_l = 141 (e) n_l = 189

Fig. 3.5 further shows five RBP-ATGP found target maps using different sets of bands starting with n_l = 6 in Fig. 3.5(a) progressively increasing to n_l = 15 in Fig. 3.5(b), n_l = 22 in Fig. 3.5(c), n_l = 141 in Fig. 3.5(d) and finally reaching the full bands, n_l = 189 in Fig. 3.5(e) where each detection map shows the targets found in the transition bands in 4 ranges in Fig. 3.5(b). Specifically, the bands underneath the figures were those actually picked up panel pixels where the red crosses indicate ground truth pixels and triangles highlight the targets found in the current band with the numbers next to the triangles indicating the order of the targets being found. The targets found in the previous bands are also marked by the magenta color.

According to Fig. 3.5 Table 3.3 summarizes 6 target pixels found by RBP-ATGP as n_l varies where the number of processed bands starting from the n_l = 6, 15, 28, 170 to full 189 bands, to extract panel pixels corresponding to five panel signatures. The panel pixels $\mathbf{t}_j^{(l)}$ in the 2nd column of the table are identified as the *j*th target among these 6 target pixels with the first *l* bands being used to process ATGP.

Table 3.3. Summary of panel pixels being identified by RBP-ATGP n_l for TE

n_l	Panel pixels found by RBP-ATGP
6	$\mathrm{K} = \mathbf{t}_1^{(6)}$, $\mathrm{A} = \mathbf{t}_2^{(6)}$
15	$\mathrm{K} = \mathbf{t}_1^{(15)}$, $\mathrm{M} = \mathbf{t}_2^{(15)}$
22	$\mathrm{A} = \mathbf{t}_1^{(22)}$, $\mathrm{K} = \mathbf{t}_2^{(22)}$, $\mathrm{B} = \mathbf{t}_4^{(22)}$
141, 189	$\mathrm{M} = \mathbf{t}_1^{(141)}$, $\mathrm{A} = \mathbf{t}_2^{(141)}$, $\mathrm{B} = \mathbf{t}_3^{(141)}$, $\mathrm{K} = \mathbf{t}_4^{(141)}$, $\mathrm{C} = \mathbf{t}_5^{(141)}$

Table 3.4 also tabulates all panel pixels that were found by RBP-ATGP corresponding to the five mineral signatures with using the minimal n_l where the value of the subscript *j* in $\mathbf{t}_j^{(l)}$ listed in the 3rd column is its order appearing in the entire sequence of RBP-ATGP-generated target pixels and the order of the five panel pixels to be found by RBP-ATGP in the 4th

column with the order of the signature found among 6 ATGP-found targets as the total number of bands up to the particular band were being used to process ATGP.

Table 3.4. Summary of minimal n_l to identify signatures and their orders found in their particular bands for TE

Signatures found by RBP-ATGP	Minimal n_l identifying signatures	Order of signatures found by RBP-ATGP	Order of signatures found using bands up to its particular band
$\mathrm{A} = \mathbf{t}_{2}^{(6)}$	6	2	2
$\mathrm{B} = \mathbf{t}_{35}^{(22)}$	22	35	4
$\mathrm{C} = \mathbf{t}_{108}^{(141)}$	141	108	5
$\mathrm{K} = \mathbf{t}_{1}^{(6)}$	6	1	1
$\mathrm{M} = \mathbf{t}_{24}^{(15)}$	15	24	2

For example, the panel pixel B(1,1) found to be the 4th ATGP target in Fig. 3.5(c), $\mathrm{B} = \mathbf{t}_{4}^{(22)}$ when n_l = 22 was used. But if we include the previous RBP-ATGP generated targets found by using the processed bands, $n_l < 22$, the panel pixel B(1,1) is actually found as the 35th target by RBP-ATGP, i.e., $\mathrm{B} = \mathbf{t}_{35}^{(22)}$. Similarly, C(1,1) was picked up as the 5th target by RBP-ATGP in Fig. 3.5(d) when the minimal n_l =141, but it was actually found as 108th ATGP target, $\mathrm{C} = \mathbf{t}_{108}^{(141)}$ by counting all previous generated ATGP targets.

3.4.2 Real Data (HYDICE Scene)

Fig. 3.6 also plots 3D detected abundance fractions by PBP-ATGP in *z*-axis where *x* and *y* axes denote the number of bands from 1 to 169 being processed and 20 panel pixels including 19 R panel pixels plus the yellow panel pixel, p_{212} in Fig. 2.4(b). There are a total of 6 R-panel pixels, p_{11}, p_{221}, p_{311}, p_{312}, p_{411}, p_{521} and 1 Y-panel pixel, p_{212} being identified during the entire RBP-ATGP process. Table 3.5 summarizes the number of bands processed that a signature first identified and the order of this signature found in the particular band for the 7 ground truth pixels identified by RBP-ATGP.

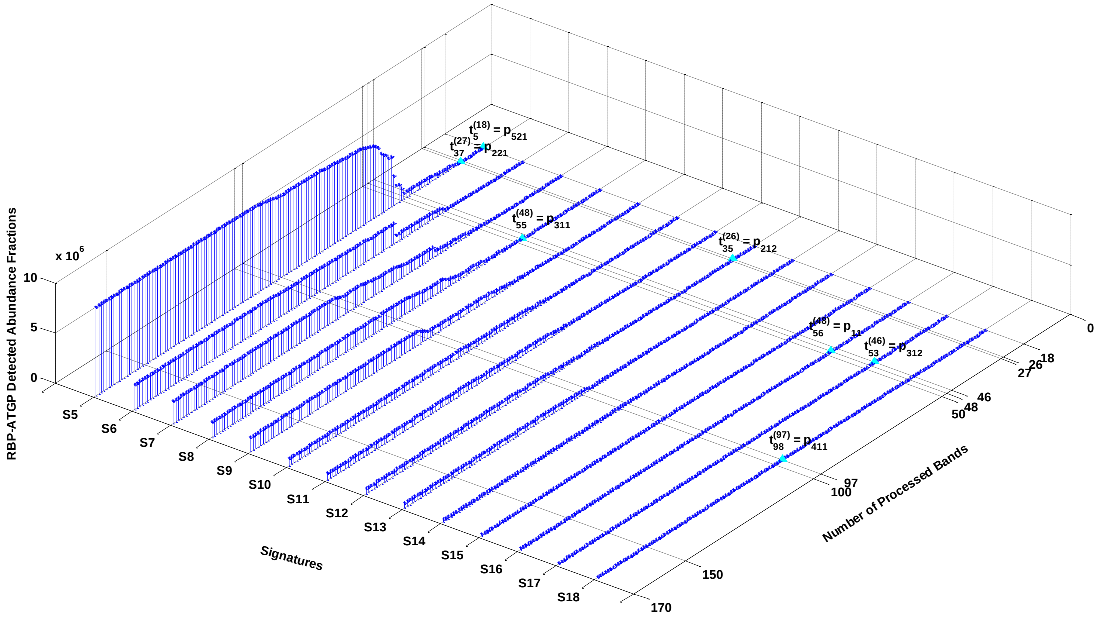


Figure 3.6. 3D detection map plot of the 5th to the 18th targets versus the number of bands being processed

Table 3.5. Summary of minimal number of bands processed to identify signatures and order of signatures to be found in particular bands

Signatures found by RBP-ATGP	Minimal n_l identifying signatures	Order of signatures to be found	Order of signatures found in the particular band
$p_{11} = \mathbf{t}_{56}^{(48)}$	48	56	16
$p_{221} = \mathbf{t}_{37}^{(27)}$	27	37	5
$p_{212} = \mathbf{t}_{35}^{(26)}$	26	35	12
$p_{311} = \mathbf{t}_{55}^{(48)}$	48	55	8
$p_{312} = \mathbf{t}_{53}^{(46)}$	46	53	17
$p_{411} = \mathbf{t}_{98}^{(97)}$	97	98	18
$p_{521} = \mathbf{t}_{5}^{(18)}$	18	5	5

The cyan arrow in the graph indicates a ground truth pixel is first identified. To compare the results in Fig. 3.6, Fig. 3.7 plots 18 targets generated by ATGP using full bands with the number of targets to be generated, p = 18 determined by the work (Chang, 2003) where the red crosses indicate the spatial locations of the 19 R panel pixels, the yellow cross highlights the location of the Y-panel pixel and the cyan upper triangles show the targets found by ATGP. As we can see, only 5 R panel pixels in five different rows were identified during this process. That is, $p_{11} = \mathbf{t}_6^{(169)}$, $p_{212} = \mathbf{t}_{17}^{(169)}$, $p_{312} = \mathbf{t}_5^{(169)}$, $p_{411} = \mathbf{t}_{18}^{(169)}$ and $p_{521} = \mathbf{t}_3^{(169)}$. It should be noted that the target pixel found by $\mathbf{t}_{17}^{(169)}$ is the yellow panel pixel p_{212} not the ground truth R panel pixel p_{221}.

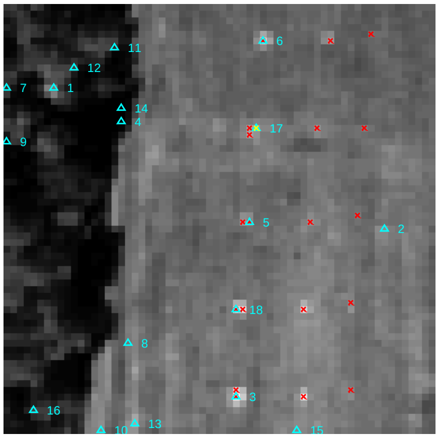
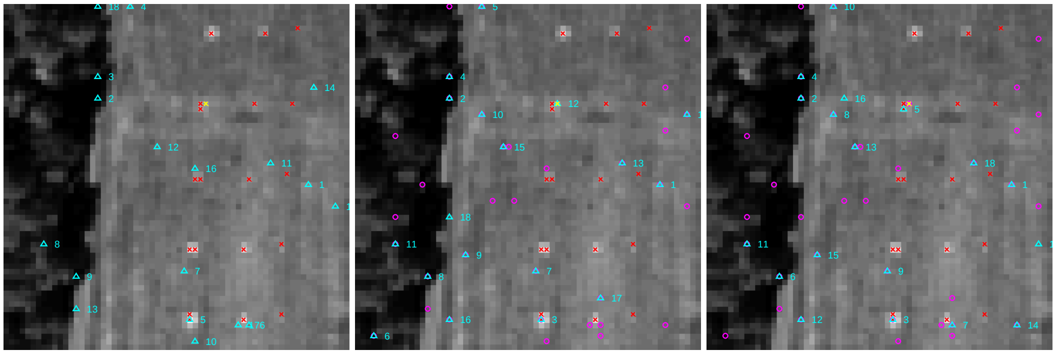
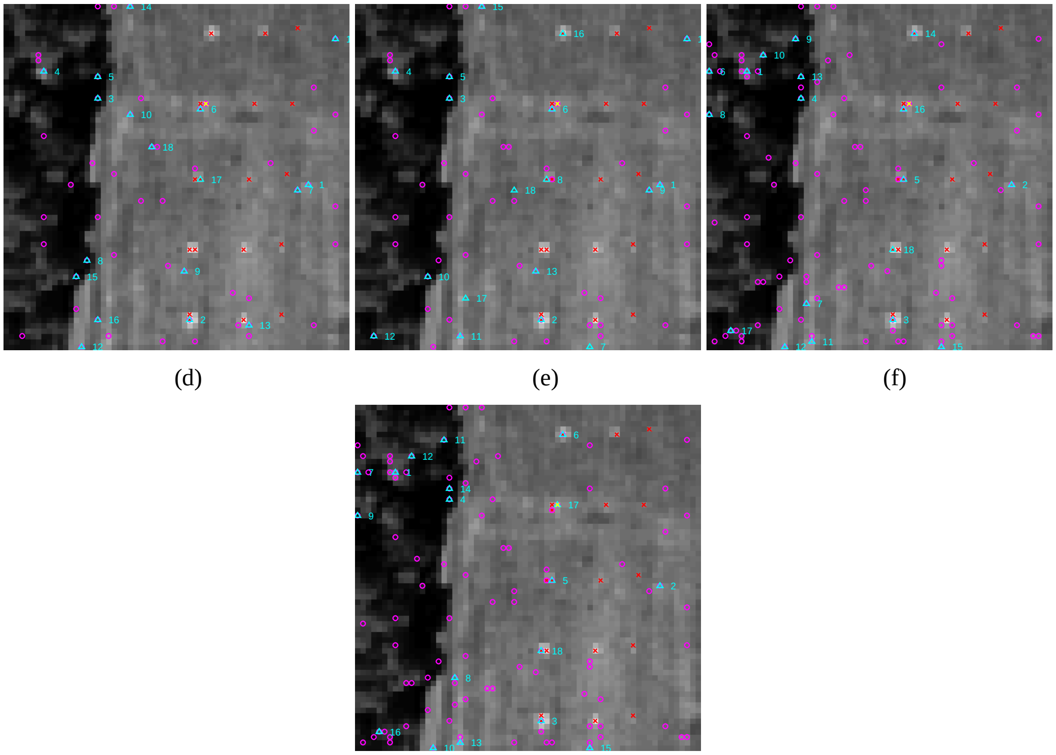


Figure 3.7. ATGP Results with p = 18

As shown in Fig. 3.6 there were 6 R panel pixels, p_{11}, p_{311}, p_{312}, p_{411}, p_{42}, p_{521} and one yellow panel pixel p_{212} found by RBP-ATGP. In order to further dictate the targets found by RBP-ATGP in the transition bands Fig. 3.8 shows the targets picked up by transition bands where the red cross indicates the spatial location of the 19 R-panel pixels, the yellow cross denotes the location of the Y-panel pixel p_{212}, the cyan upper triangles highlight the targets found in the current band, and the numbers next to the triangles reflect the order of the targets being found by RBP-ATGP. In addition, the bands underneath each of figures were the one actually to pick up panel pixels. The targets found in the previous bands are also marked by magenta circles.



(a) (b) (c)



(d) (e) (f)

(g)

Figure 3.8. RBP-ATGP Target Detection Results with different number of bands processed, n_l: (a) n_l = 18 (b) n_l = 26 (c) n_l = 27 (d) n_l = 46 (e) n_l = 48 (f) n_l = 97 (g) n_l = 169

As shown in Fig. 3.8, the 5 panel pixels in Fig. 3.7 can be found by RBP-ATGP using the first 97 bands. Comparing the results available in (Chang, 2003) two more panel pixels, p_{212} and p_{42} were found in Fig. 3.8(f) which showed the advantage of using RBP-ATGP.

Theoretically speaking, there would be the maximal number of targets, $p \times L$ that can be found by RBP-ATGP band by band progressively. That is, for each different band ATGP can generate a different target. However, practically, the size of the ATGP-generated target set is smaller than $p \times L$, since some of the targets may be picked up by ATGP more than once. For example, for the HYDICE data set, p = 18, $p \times L = 3042$. In reality the number of distinct targets found by RBP-ATGP was 98 which was much smaller than 3042. In the meantime, the number of targets found by RBP-ATGP tended to be stabilized as more bands were added into the progressive process and the generated RBP-ATGP targets stopped varying after 97 bands were processed as shown in Fig. 3.9.

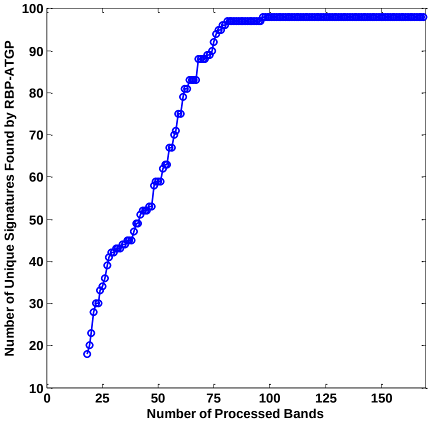


Figure 3.9. The plot of the number of unique signatures found by RBP-ATGP versus number of processed bands

Furthermore, Fig. 3.10 demonstrates a 3 dimensional histogram to show how frequently a particular pixel is picked up by RBP-ATGP as a target. The *x*-axis and *y*-axis correspond to the columns and rows of the HYDICE scene. The *z*-axis indicates the number of times a given pixel was detected as a target by RBP-ATGP after processing all bands. So, for HYDICE scene with 169 bands, the maximum value for *z*-axis is 169. As shown in the figure, the magenta arrows indicate the ground truth panel pixel.

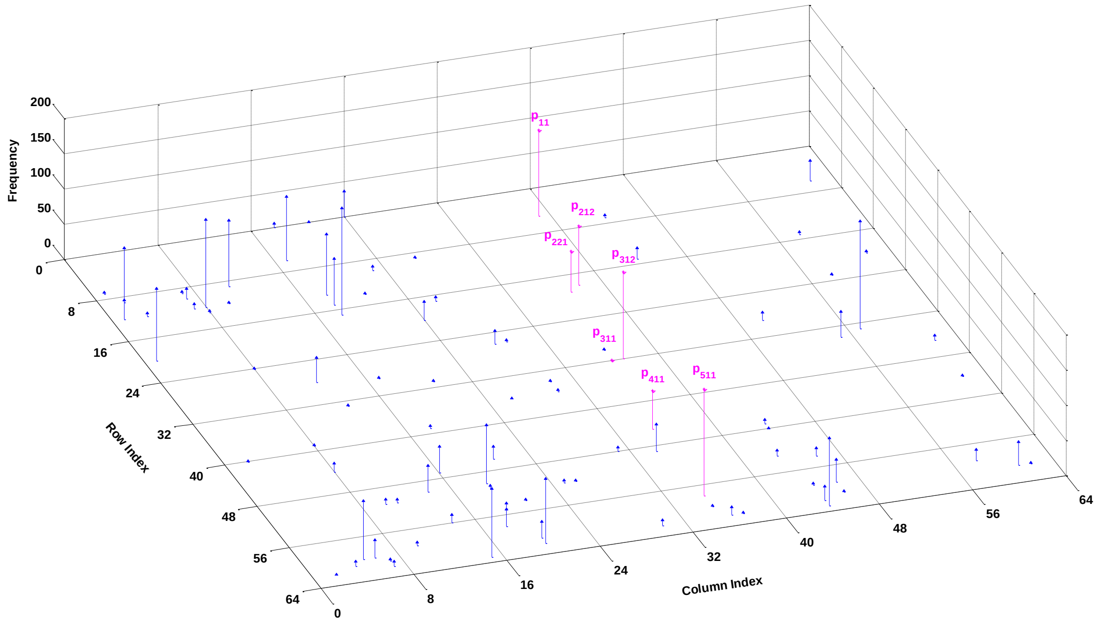
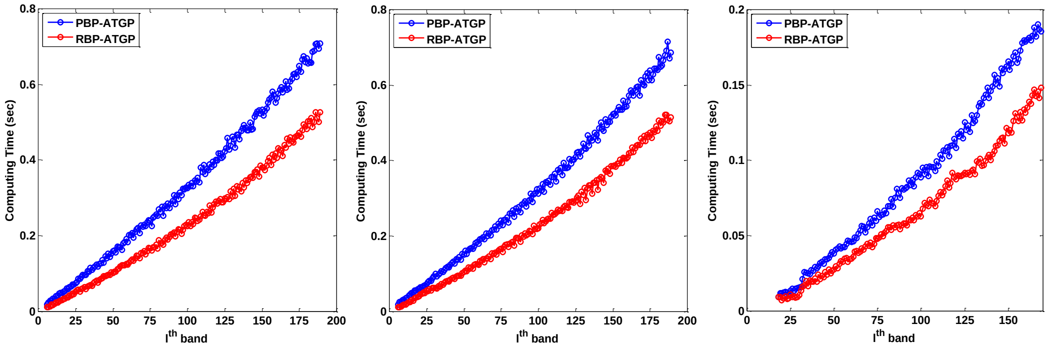


Figure 3.10. 3D Histogram of the pixels picked up by RBP-ATGP as targets

3.5 Computational Complexity

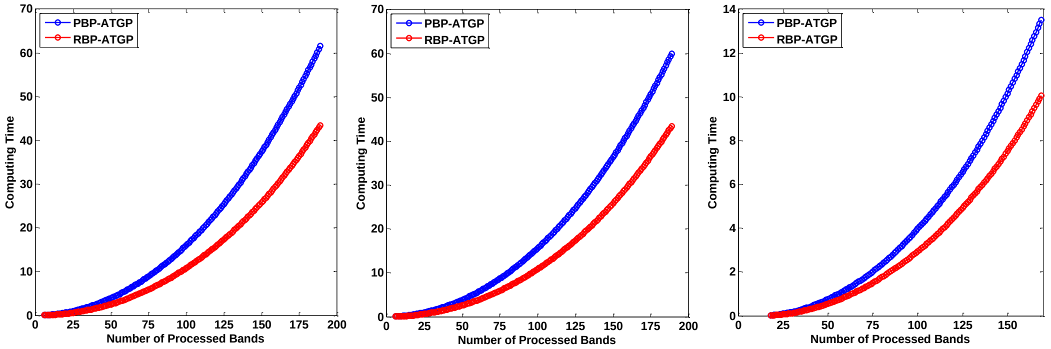
In order to evaluate computational efficiency of RBP-ATGP in data processing time, we calculate the computing time of RBP-ATGP using the recursive equations specified by (3.8) to update the results and PBP-ATGP without using recursive equations but rather repeatedly implementing (2.5) with **U** augmented band by band as every time the l^{th} band is a new band coming in. Both RBP-ATGP and PBP-ATGP were run and executed in MATLAB R2012B with an Intel Core i7 – 3770 running at 3.40 GHz with 16GB of RAM on three data sets ten times to produce an average computing time. Fig. 3.11 (a-c) plots computing times required for RBP-ATGP and PBP-ATGP without using recursive equations (3.8) to process TI, TE and HYDICE data respectively where the *y*-axis is computer processing time in seconds and *x*-axis is the l^{th} band as new band coming in. As can be seen from these figures the processing time required for RBP-ATGP and PBP-ATGP to run each new individual band is nearly linear where RBP-ATGP required less time than PBP-ATGP did. It is also worth noting that the fluctuations of plots were resulting from numerical computations by computer implementation.



(a) TI (b) TE (c) HYDICE

Figure 3.11. Computing time versus the l^{th} band by RBP-ATGP (a) TI (b) TE (c) HYDICE

For a fair comparison Fig. 3.12(a-c) further plots computing time versus the number of processed bands, n_l, (i.e., the number of the first l bands to be used for processing) required for RBP-ATGP with using recursive equations (3.8) and PBP-ATGP without using recursive equation (3.8) for TI, TE and HYDICE data respectively.



(a) TI (b) TE (c) HYDICE

Figure 3.12. Computing time versus n_l by ATGP and RBP-ATGP (a) TI (b) TE (c) HYDICE

It should be noted that the plots of RBP-ATGP is the accumulative computer processing time by RBP-ATGP, which is obtained by summing all the computing times in Fig. 3.11 up to the l^{th} band. As we can see RBP-ATGP ran faster than PBP-ATGP which was implemented without using recursive equations (3.8). In the latter case, PBP-ATGP repeatedly used (2.5) to directly calculate band varying $P_{\mathbf{U}_{l\times p}}^{\perp}$ when each new band, i.e., the l^{th} band came in.

Table 3.6. Comparison of computing time in seconds required by RBP-ATGP and PBP-ATGP

	TI (sec)	TE (sec)	HYDICE (sec)
ATGP	0.5794	0.5895	0.1854
RBP-ATGP	43.4537	43.4818	10.0628
PBP-ATGP	61.5242	60.0149	13.5124

Table 3.6 calculated the final computer times required for RBP-ATGP and PBP-ATGP to complete the entire data sets where RBP-ATGP achieved around 30% efficiency compared to PBP-ATGP. However, we believe that this saving would be tremendous and pronounced once RBP-ATGP is implemented in hardware due to its recursive structure.

3.6 GUI Design

A graphical user interface (GUI) with a screenshot shown in Fig. 3.13 was developed using Matlab's guide to aid algorithm performance analysis. The GUI allows the user to load different data sources and also to choose different numbers of signatures for analysis. The three images displayed on top of the windows show a color image of the scene, a gray scale image of the current band being processed, and the result of subpixel detection after the current band is processed. At the bottom a window shows changes of the detected abundance fraction values in real time with the cyan upward arrow at the top to highlight the results of current band being processed. Once the data is loaded, the users can start processing by clicking the start button. When each band is received at each iteration RBP-ATGP processes this new received to produce results with the current band and the resulting image updated to allow the users to observe the changes in target detection by RBP-ATGP.

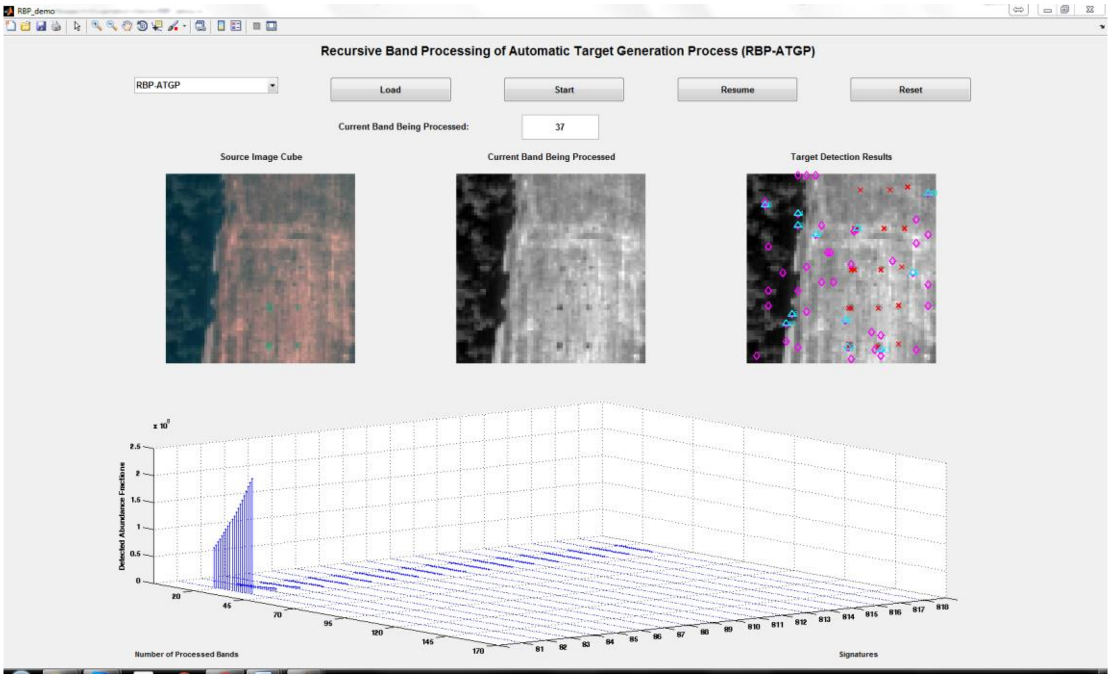

RBP_demo
Recursive Band Processing of Automatic Target Generation Process (RBP-ATGP)
RBP-ATGP
Load
Start
Resume
Reset
Current Band Being Processed:
37
Source Image Cube
Current Band Being Processed
Target Detection Results
x 10^7
Detected Abundance Fractions
2.5
2
1.5
1
0.5
0
20
45
70
95
120
145
170
Number of Processed Bands
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
Signatures

Figure 3.13. GUI interface for RBP-ATGP

3.7 Conclusion

ATGP is an unsupervised hyperspectral target detection which has been shown to be very effective in many applications such as subpixel target detection, endmembers, anomalies, man-made targets, etc. Its field programmable gate array (FPGA) implementation were also studied (Chang, 2003). Interestingly, the idea of implementing ATGP according to BSQ band by band has not been investigated in the past. This paper develops such an approach, called RBP-ATGP which allows image analysts to dictate ATGP-detected targets from band to band to capture the effect of inter-band correlation on progressive changes in target detection by ATGP. Eventually, RBP-ATGP produces a set of L^2 targets which provides more target information band to band compared to L targets for each ATGP-generated by ATGP using entire L bands. This additional piece of information is significant to find weak targets which may be dominated by ATGP-generated target using full bands.

Chapter 4: Recursive Band Processing of Pixel Purity Index (RBP-PPI)

4.1 Introduction

Pixel purity index (PPI) has been widely studied for finding endmembers (Chang, 2003; Chang 2013). However, its utility has been shown more than endmemmber finding, for example, finding seed training samples for multispectral image classification in (Chen, Lin, Chen, Wen, Chen, Ouyang and Chang, 2013), finding feasible sets for potential endmember candidates (Xiong et al., 2011). This chapter presents a new application of PPI in band processing, which is called recursive band processing of PPI (RBP-PPI). The idea of RBP-PPI is derived from Band SeQuential (BSQ) data acquisition format where band is acquired band by band. Implementing RBP-PPI is much harder than it looks, since there are 4 variables involved in the process, that are the number of bands processed, the number of pixels processed, the number of skewers required and the proper threshold for PPI Count value. The key of RBP-PPI is how to calculate PPI counts for data sample vectors recursively band by band. However, this is very closely related to the number of skewers, *K* to be used by PPI. Technically speaking, a skewer is a randomly generated unit vector which points out a direction which data samples can be aligned with. How to effectively determine an appropriate value for *K* still remains unsolved and is a very challenging issue needed to be addressed although there was one developed in (Chang and Plaza, 2006). Interestingly, when it comes to RBP-PPI, how to process skewers band by band become a major issue, which is not encountered in full-band processed PPI. This is due to the fact that skewers not only are changing band by band but also need to be normalized to unity again once a new band is fed in. In order to address this issue one approach is developed in this dissertation, that is using a fix set of skewers for all bands and skewers are only updated by its most recent band information provided by the incoming band.

Some benefits can be gained from RBP-PPI process. First of all, progressive changes in PPI counts of data samples as more bands are included for data processing can be captured from RBP-PPI process. This allows image analysts to keep track of significance of data samples. Second is to help find crucial bands for data processing according to progressive changes in PPI counts. Intuitively, RBP-PPI can be considered as a band-to-band slow motion version of PPI which dictates slow varying changes in spectral variation that cannot be provided by PPI which is basically a one-shot operation of full band data processing. Moreover, RBP-PPI is designed for BSQ data acquisition format, data process can be done with limited bandwidth.

4.2 Derivation RBP-PPI

Let L = total number of bands, N = the total number of data samples, $\mathbf{r}_i(l)$ = the i^{th} image pixel acquired by the l^{th} spectral band where $1 \leq i \leq N$ and $1 \leq l \leq L$. Then the set of all data sample is denoted by $\{\mathbf{r}_i(l)\}_{i=1}^{N}$. $\mathbf{r}_i(l)$ can be represented by:

$$\mathbf{r}_i(l) = (r_{i1}, \cdots r_{i(l-1)}, r_{il})^T = \left(\mathbf{r}_i^T(l-1), r_{il}\right)^T \tag{4.1}$$

Also suppose that there is a given set of K skewers denoted by $\{\mathbf{s}_k(l)\}_{k=1}^{K}$:

$$\mathbf{s}_k(l) = (s_{k1}, \cdots s_{k(l-1)}, s_{kl})^T = \left(\mathbf{s}_k^T(l-1), s_{kl}\right)^T \tag{4.2}$$

Then, all the data sample vectors are projected onto $\mathbf{s}_k(l)$, the projected value can be calculated by:

$$\begin{aligned}\mathbf{r}_i^T(l)\mathbf{s}_k(l) &= \left(\mathbf{r}_i^T(l-1), r_{il}\right)\left(\mathbf{s}_k^T(l-1), s_{kl}\right)^T \\ &= \mathbf{r}_i^T(l-1)\mathbf{s}_k(l-1) + r_{il}s_{kl}\end{aligned} \tag{4.3}$$

The extrema set can be defined as

$$S_{extrema}(\hat{\mathbf{s}}_k(l)) = \arg\left\{\max_{\mathbf{r}_i;1\leq i\leq N}(\mathbf{r}_i(l)\cdot\hat{\mathbf{s}}_k(l))\right\} \cup \arg\left\{\min_{\mathbf{r}_i;1\leq i\leq N}(\mathbf{r}_i(l)\cdot\hat{\mathbf{s}}_k(l))\right\} \tag{4.4}$$

Then, the indication function can be expressed by

$$I_{S_{extrema}(\hat{\mathbf{s}}_k(l))}(\mathbf{r}_i(l)) = \begin{cases} 1; \text{ if } \mathbf{r}_i(l) \in S_{extrema}(\hat{\mathbf{s}}_k(l)) \\ 0; \text{ otherwise} \end{cases} \tag{4.5}$$

4.3 Detailed RBP-PPI Algorithm

As noted in the previous sections, using a fix set of skewers is proposed to address the issue that skewers not only are changing band by band but also need to be normalized to unity once a new band is fed in, that is skewers are only updated by its most recent band information provided by the incoming band. In this section, various versions of RBP-PPI algorithms using a fix set of skewers are introduced. For the first band l = 1, a set of K skewers, $\{\mathbf{skewer}_k(1)\}_{k=1}^{K}$ is randomly generated, where $\mathbf{skewer}_k(1) = (s_{k1})$ is actually a scalar s_{kl}. These K scalar skewers will be then used to update skewers across all bands. In other words, when the band number is increased to l, the K-skewer set used for l is $\{\mathbf{skewer}_k(l)\}_{k=1}^{K}$ where each $\mathbf{skewer}_k(l)$ is actually made up by $\mathbf{skewer}_k(l) = \left(\mathbf{skewer}_k^T(l-1), s_{kl}\right)^T$. More specifically, once skewer set is generated at band one, l = 1, the locations of skewers are fixed and used for all following bands by adding the new band information $\{s_{kl}\}_{k=1}^{K}$ provided by the incoming band, the l^{th} band for $\{\mathbf{skewer}_k(l)\}_{k=1}^{K}$. In the later discussions, a general version of RBP-PPI is proposed. However, RBP-PPI is neither a causal nor a real time process since it requires to find $\max_l(k)$ and $\min_l(k)$ via (4.3 – 4.4) for all the data samples $\{\mathbf{r}_i(l)\}_{i=1}^{N}$. Under such circumstances, two versions of implementing RBP-PPI are further proposed. There are three loops for these two newly proposed algorithms, that are the outer loop, the middle loop and the inner loop. The outer loop for both of these two versions of algorithms is the total number of bands. While one version implements skewers in the middle loop while data samples in the inner loop the resulting PPI is called RBP-Progressive-PPI (RBP-P-PPI). On the other hand, swapping the inner and middle loops results in RBP-Causal-PPI (RBP-C-PPI).

Algorithm for Implementing General RBP-PPI Using A Fix Set of Skewers

```
• Initialization: $N_{\text{PPI}}^{(k)}(\mathbf{r}(l)) = 0$
• Loop from l = 1 to L
    • Input the skewer set, $\{\mathbf{s}_k(l)\}_{k=1}^{K}$ by adding the $l^{\text{th}}$ band data information.
    • $\{\hat{\mathbf{s}}_k(l)\}_{k=1}^{K}$ are obtained by normalizing $\{\mathbf{s}_k(l)\}_{k=1}^{K}$ to unit random vectors.
    • Calculate the projected value according to equation (4.3), then identify the extrema
      set $S_{extrema}(\hat{\mathbf{s}}_k(l))$ based on eq. (4.4).
        A. If $I_S^{l-1}(\mathbf{r}(l)) = 0$ and $I_S^{l}(\mathbf{r}(l)) = 0$, then $N_{\text{PPI}}^{(k)}(\mathbf{r}(l)) = 0$. Otherwise, continue.
        B. If $I_S^{l-1}(\mathbf{r}(l)) = 0$ and $I_S^{l}(\mathbf{r}(l)) = 1$, then $N_{\text{PPI}}^{(k)}(\mathbf{r}(l)) = 1$. Otherwise, continue.
        C. If $I_S^{l-1}(\mathbf{r}(l)) = 1$ and $I_S^{l}(\mathbf{r}(l)) = 0$, then $N_{\text{PPI}}^{(k)}(\mathbf{r}(l)) = 0$. Otherwise, continue.
        D. If $I_S^{l-1}(\mathbf{r}(l)) = 1$ and $I_S^{l}(\mathbf{r}(l)) = 1$, then $N_{\text{PPI}}^{(k)}(\mathbf{r}(l)) = 0$. Otherwise, continue.
    • Calculate $N_{\text{PPI}}(\mathbf{r}_i(l)) = \sum_{k=1}^{K} N_{\text{PPI}}^{(k)}(\mathbf{r}_i(l)$.
    • Let $l \leftarrow l+1$ until $l$ = L.
```

- **End loop.**

4.4 Experiments

4.4.1 Synthetic Data

A. TI Data Set

The image scene from Figure 2.3(a) was processed by proposed RBP-PPI. Fig. 4.1 shows the plots of PPI count value versus the different number of bands processed for the 5 mineral signatures A, B, C, K, M in TI scene, where x-axis denotes the number of bands is being processed, y-axis indicates the different signatures, and z-axis presents the PPI-count value respectively. In this figure, 500 skewers are generated and utilized in the experiments. Since all of the 20 pure pixels for each signature on the first two columns can be found simultaneously, only the top left pixels on the first column for each signature are used for demonstrations and marked as A(1, 1), B(1, 1), C(1, 1), K(1, 1), M(1, 1). As shown in Fig. 4.1, the magenta arrow states the first band that a particular signature is identified as endmember candidate. For instance, signature B(1, 1) is first identified as endmember candidate with 27 bands collected and processed.

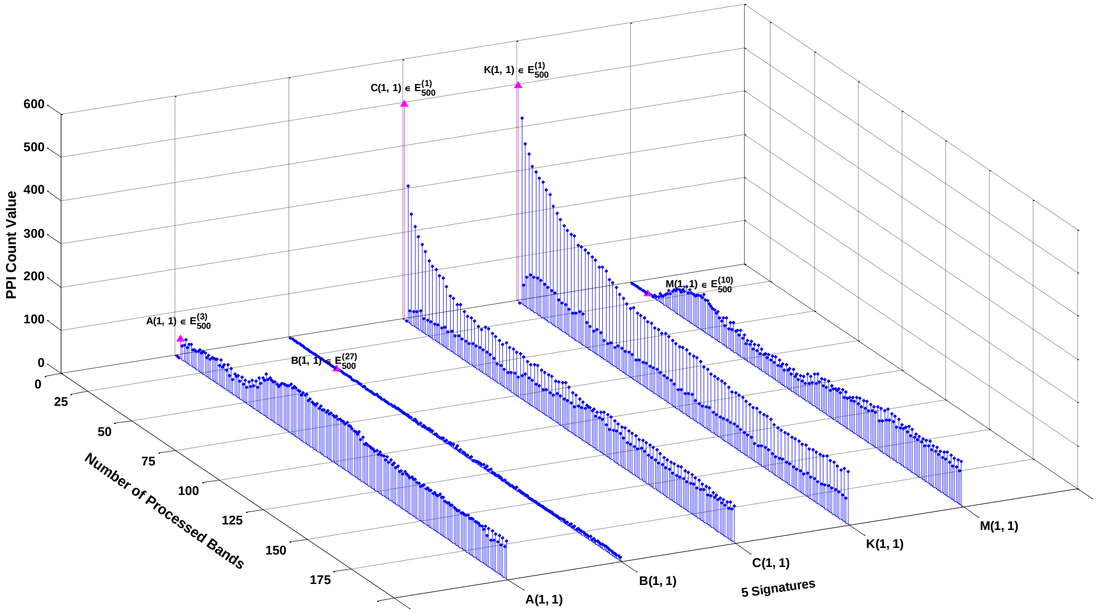
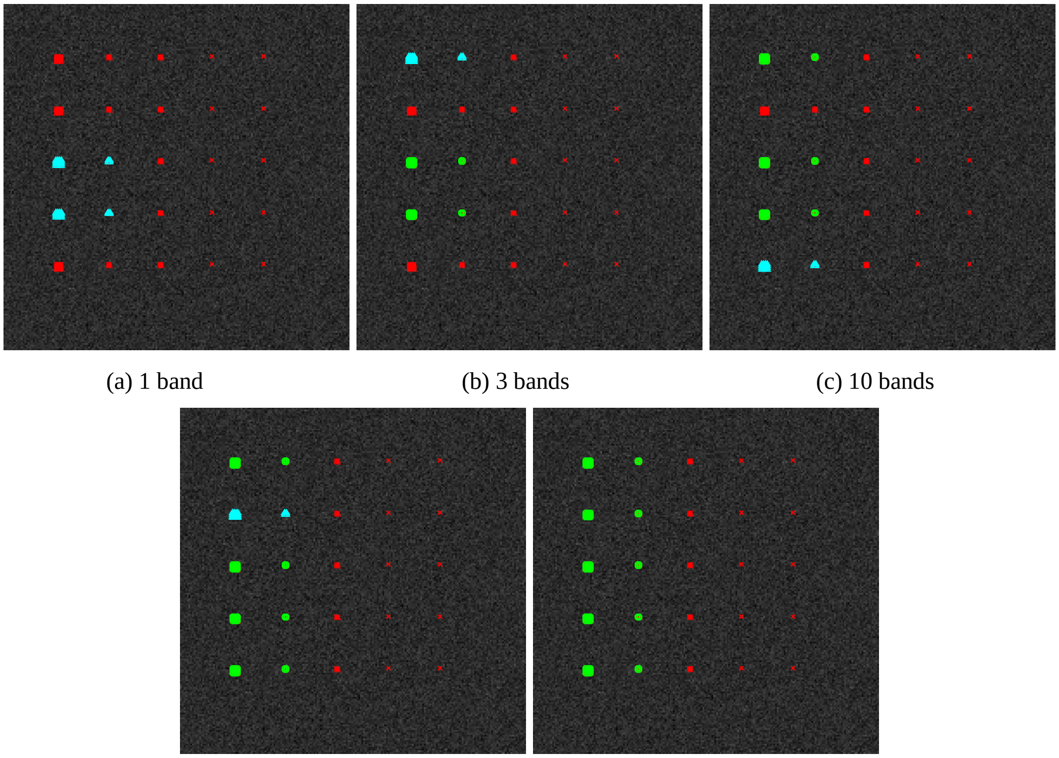


Figure 4.1. 3D plots of PPI counts versus the number of processed bands for the 5 different signatures

Fig. 4.2 shows the spatial location of the endmember candidates identified by RBP-PPI with different number of bands collected and processed. Fig. 4.2(a) starts with only 1 band, then progressively increases to 3 bands in Fig. 4.2(b), 10 bands in Fig. 4.2(c), 27 bands in Fig.

4.2(d), and reaches the full bands 189 bands in Fig. 4.2(e) in the end, each of which shows the endmember candidates finding results by RBP-PPI in the particular transition bands specified by Fig. 4.1. In Fig. 4.2, the red cross indicates the ground truth pixels in TI scene, the cyan upper triangle highlights the endmember candidates found in the current band, and the endmember candidates identified in the previous bands are marked by green circles as well.



(a) 1 band (b) 3 bands (c) 10 bands

(d) 27 bands (e) 189 bands (full bands)

Figure 4.2. Endmember candidates finding results by RBP-PPI with different number of bands collected and processed: (a) 1 band (b) 3 bands (c) 10 bands (d) 27 bands (e) 189 bands (full bands)

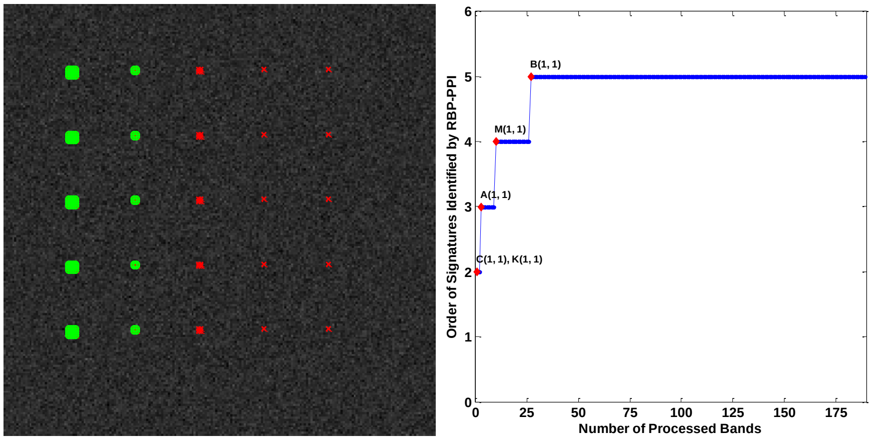
Table 4.1 summarizes the 5 mineral signatures A, B, C, K, M among the endmember candidates identified by RBP-PPI when n_l varies from 1 band, 3 bands, 10 bands, 27 bands to full bands 189 bands. As demonstrated in the table, $E^{(l)}_{N_{skewers}}$ in the 2nd column was identified using $N_{skewers}$ skewer sets with first *l* bands collected and processed.

Table 4.1. Summary of the mineral signatures being identified by RBP-PPI versus n_l for TI scene

n_l	Signatures identified by RBP-PPI
1	$\mathrm{C}(1,1) \in E^{(1)}_{500},\ \mathrm{K}(1,1) \in E^{(1)}_{500}$
3	$\mathrm{A}(1,1) \in E^{(3)}_{500}$
10	$\mathrm{M}(1,1) \in E^{(10)}_{500}$
27	$\mathrm{B}(1,1) \in E^{(27)}_{500}$

189	$\mathrm{A}(1,1) \in E_{500}^{(189)}, \mathrm{B}(1,1) \in E_{500}^{(189)}, \mathrm{C}(1,1) \in E_{500}^{(189)}, \mathrm{K}(1,1) \in E_{500}^{(189)}, \mathrm{M}(1,1) \in E_{500}^{(189)}$

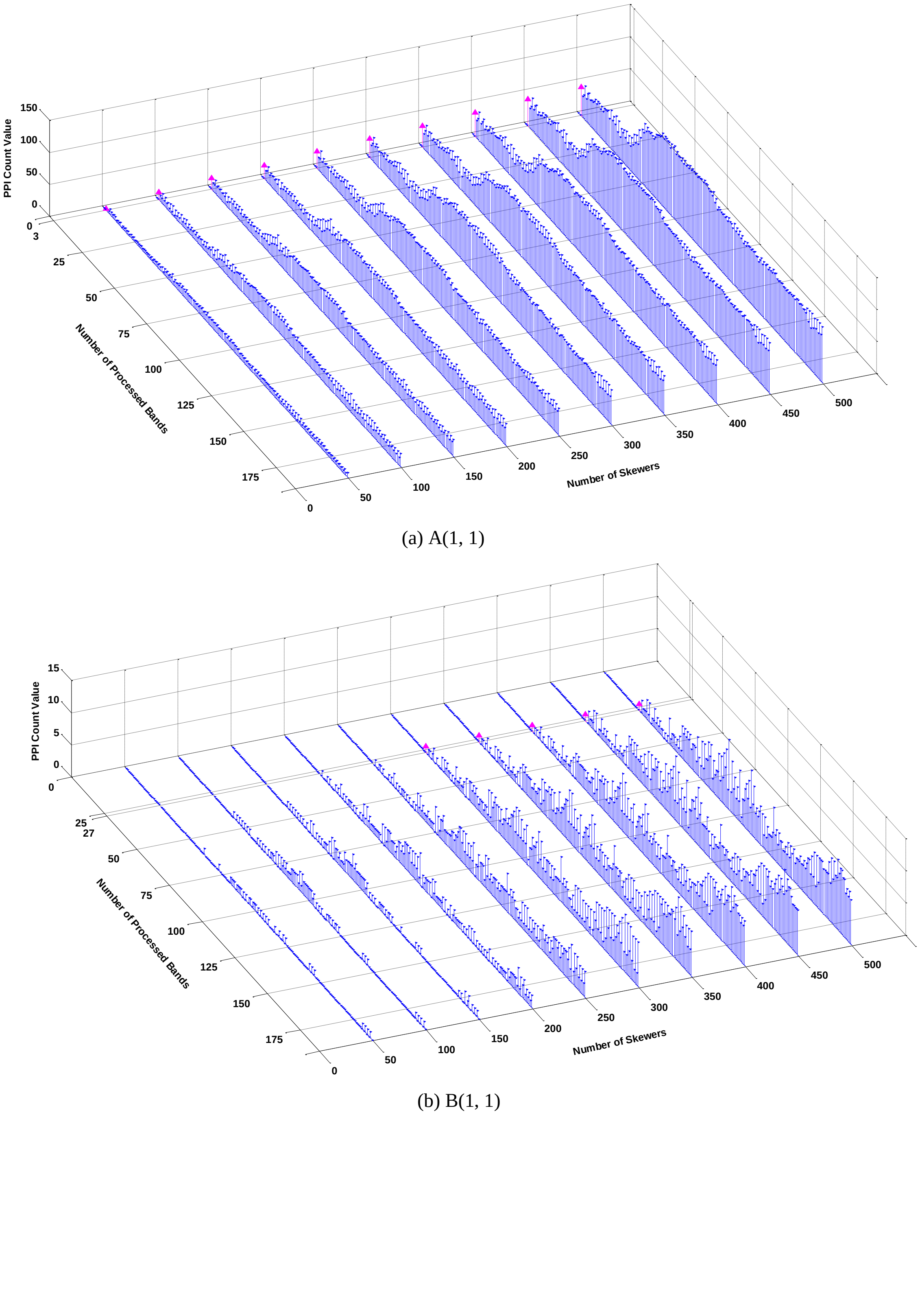
Furthermore, Fig. 4.3(a) displays the endmember candidates identified by PPI for TI scene using 500 skewers with all the bands received and processed. While the same results can be achieved with only 27 bands collected and processed with RBP-PPI. Moreover, Fig. 4.3(b) plots the order of each mineral signature A, B, C, K and M been identified as endmember candidates versus the number of bands processed. It is clear to see that with only 27 bands available, all the 5 signatures have been found.



(a) (b)

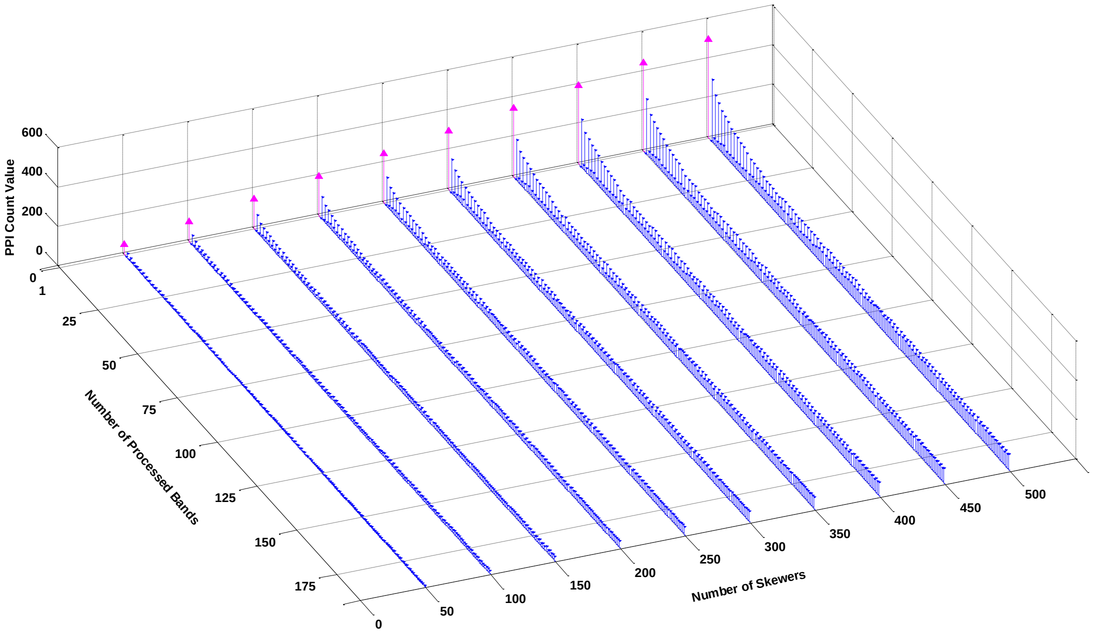
Figure 4.3. Comparative study between PPI and RBP-PPI (a) PPI identified signatures (b) The order of signatures identified by RBP-PPI

As stated in the previous sections, 500 skewers are generated and used in the RBP-PPI process for TI scene. However, choosing a proper number of skewers remains an unsolved problem in the literature. RBP-P-PPI is further proposed when it comes to address the issue that how many skewers need to be generated for process. Fig. 4.4 plots the variation of PPI Count versus the number of processed bands as the number of skewers increases from 50 to 500 with step size 50 for the 5 mineral signatures, where x-axis denotes the number of bands collected and processed, y-axis displays the number of skewers generated for RBP-P-PPI process, and z-axis presents the value of PPI Count correspondingly.

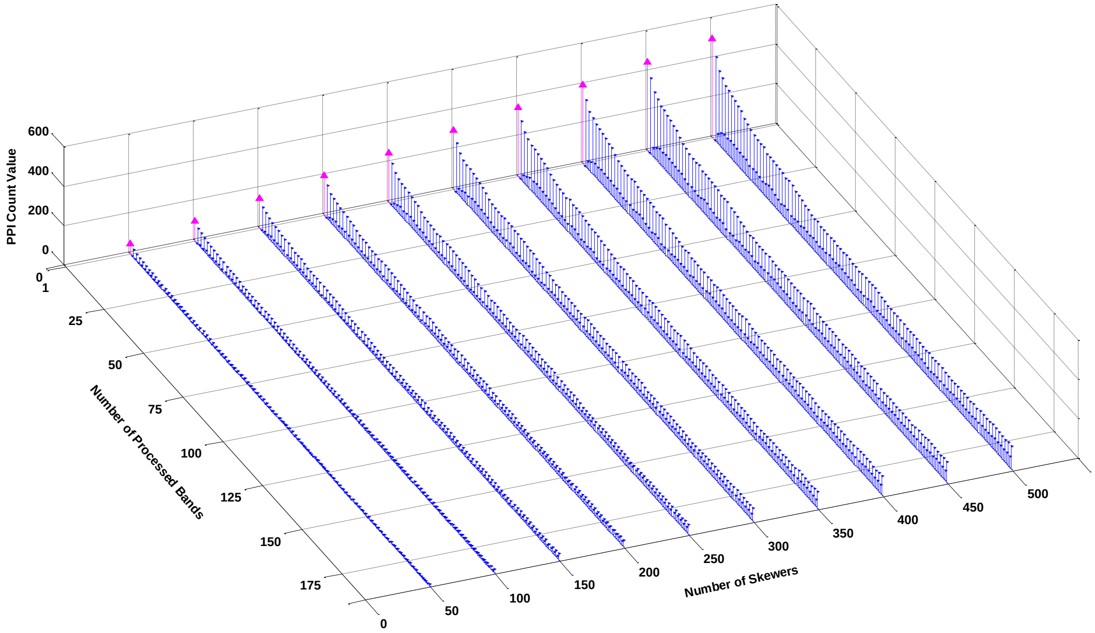

PPI Count Value
150
100
50
0
0
3
25
50
75
100
125
150
175
Number of Processed Bands
0
50
100
150
200
250
300
350
400
450
500
Number of Skewers
PPI Count Value
15
10
5
0
0
25
27
50
75
100
125
150
175
Number of Processed Bands
0
50
100
150
200
250
300
350
400
450
500
Number of Skewers

(a) A(1, 1)

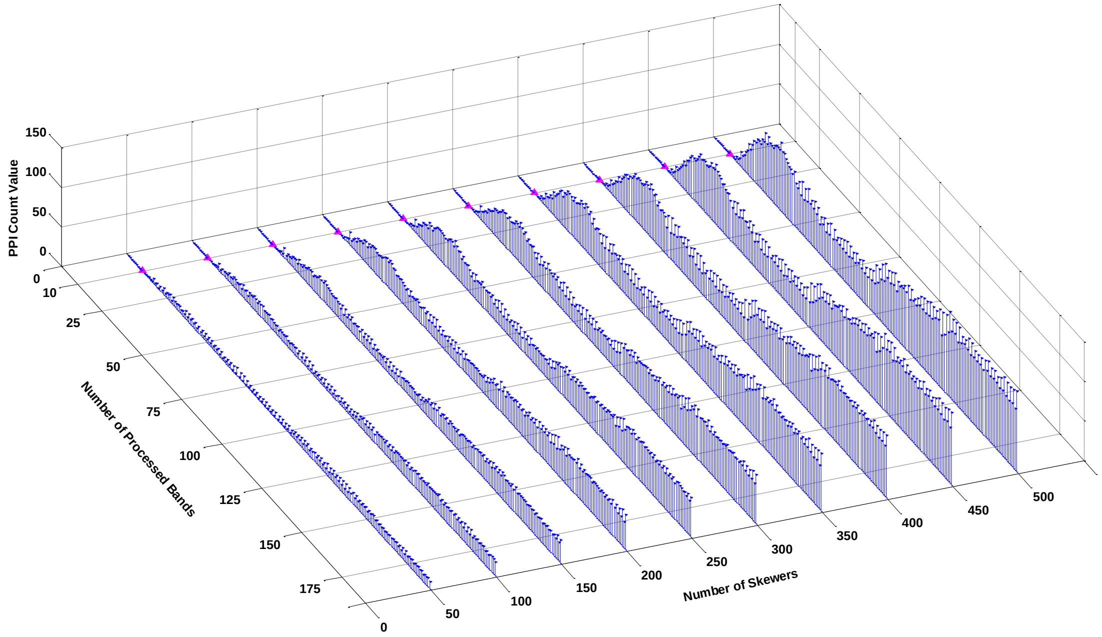
(b) B(1, 1)


PPI Count Value
600
400
200
0
0
1
25
50
75
100
125
150
175
Number of Processed Bands
0
50
100
150
200
250
300
350
400
450
500
Number of Skewers

(c) C(1, 1)


PPI Count Value
600
400
200
0
0
1
25
50
75
100
125
150
175
Number of Processed Bands
0
50
100
150
200
250
300
350
400
450
500
Number of Skewers

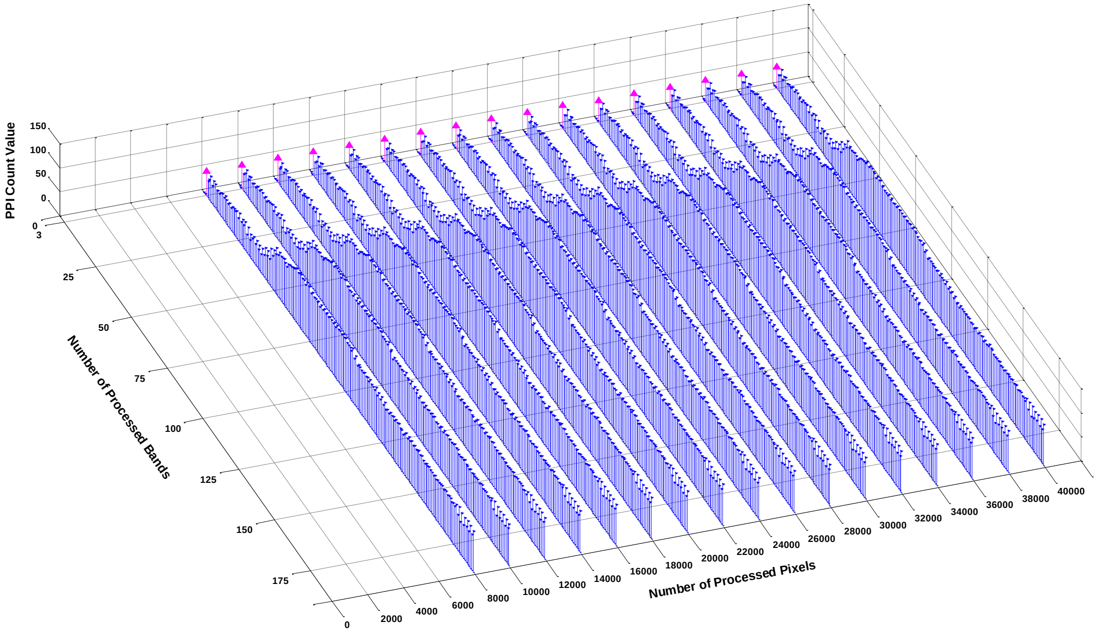
(d) K(1, 1)



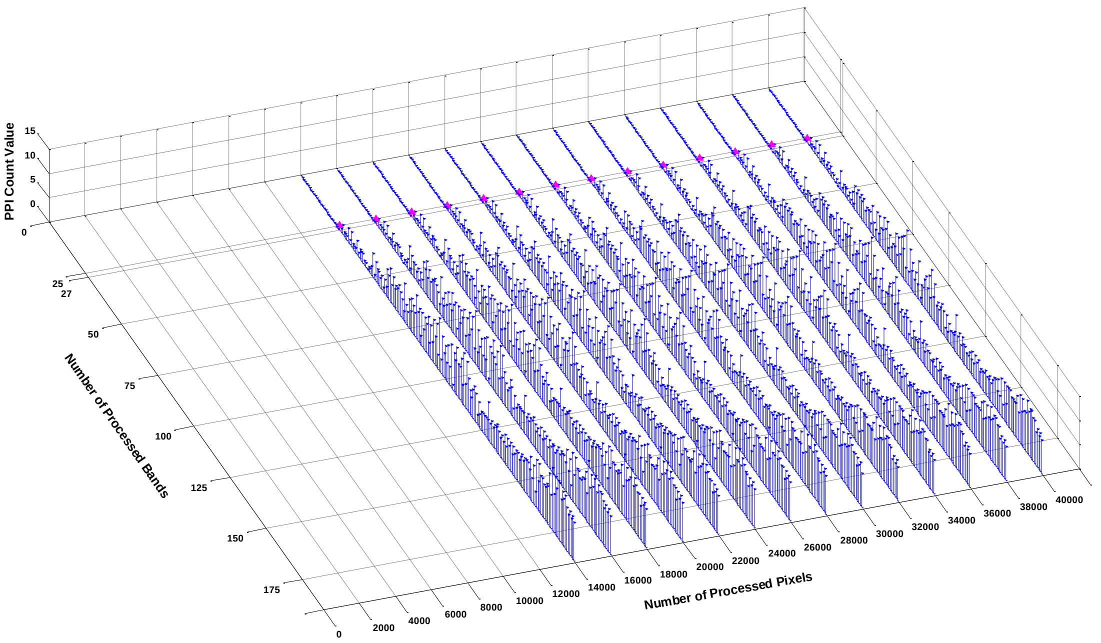
(e) M(1, 1)

Figure 4.4. PPI Count variations versus the number of processed bands and the number of skewers for each mineral signatures (a) A(1, 1) (b) B(1, 1) (c) C(1, 1) (d) K(1, 1) (e) M(1, 1)

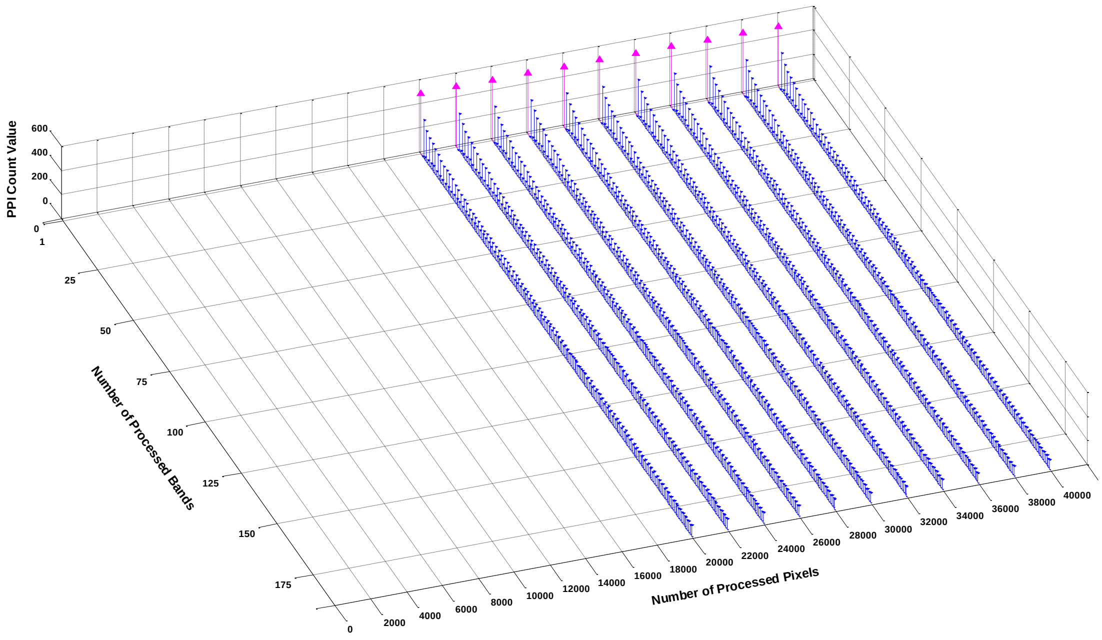
As shown in the figure, the height of each bar represents the value of PPI Count for a particular combination of the number of processed bands and the number of skewers generated for process. The magenta upward arrow indicates that one material begins to be identified as endmember candidates in the band that it first identified as an endmember candidate in RBP-PPI. For instance, signature B(1, 1) is first identified as endmember candidates with 27 bands available in RBP-PPI with 500 skewers. In the RBP-P-PPI process, it can be identified as endmember candidates with only 300 skewers generated. The less skewers generated in the process could reduce the computational complexity.


PPI Count Value
150
100
50
0
0
3
25
50
75
100
125
150
175
Number of Processed Bands
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
22000
24000
26000
28000
30000
32000
34000
36000
38000
40000
Number of Processed Pixels

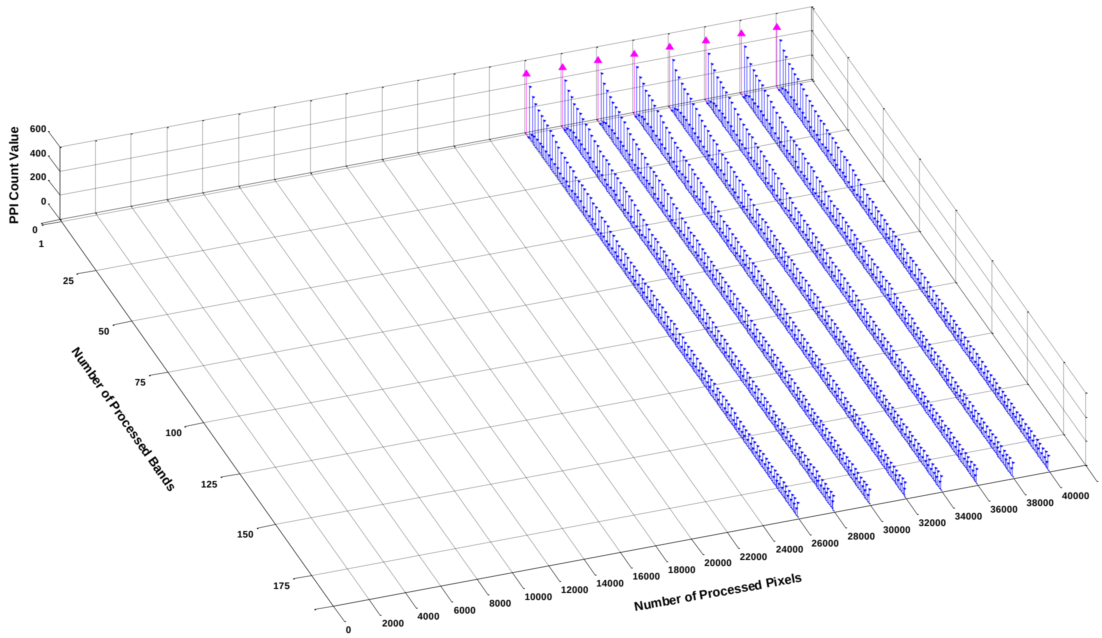
(a) A(1, 1)


PPI Count Value
15
10
5
0
0
25
27
50
75
100
125
150
175
Number of Processed Bands
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
22000
24000
26000
28000
30000
32000
34000
36000
38000
40000
Number of Processed Pixels

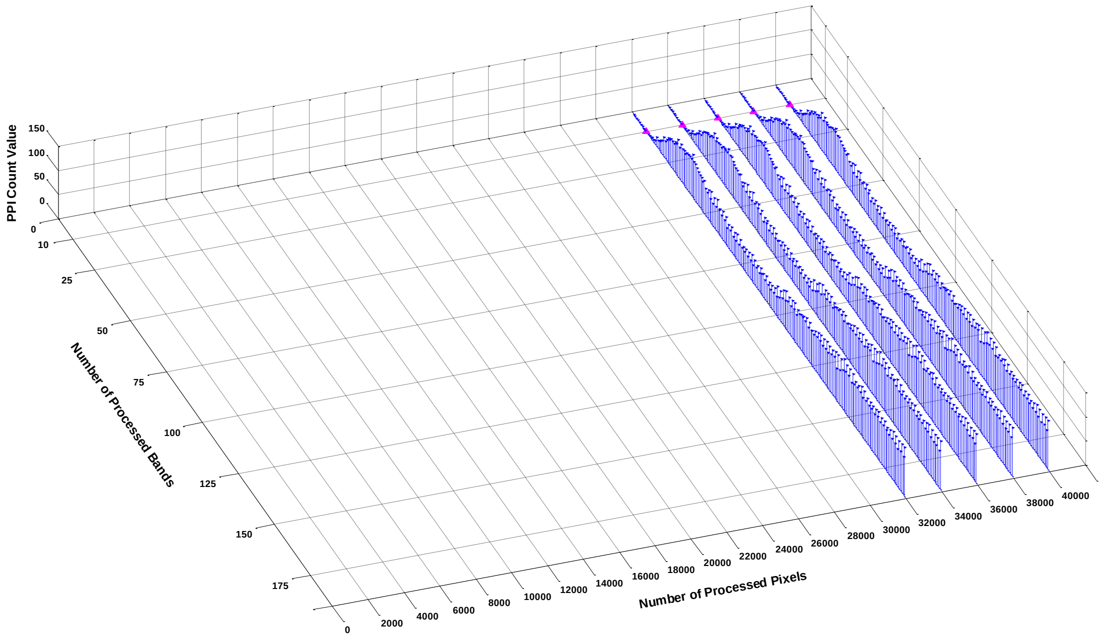
(b) B(1, 1)


PPI Count Value
600
400
200
0
0
1
25
50
75
100
125
150
175
Number of Processed Bands
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
22000
24000
26000
28000
30000
32000
34000
36000
38000
40000
Number of Processed Pixels

(c) C(1, 1)


PPI Count Value
600
400
200
0
0
1
25
50
75
100
125
150
175
Number of Processed Bands
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
22000
24000
26000
28000
30000
32000
34000
36000
38000
40000
Number of Processed Pixels

(d) K(1, 1)

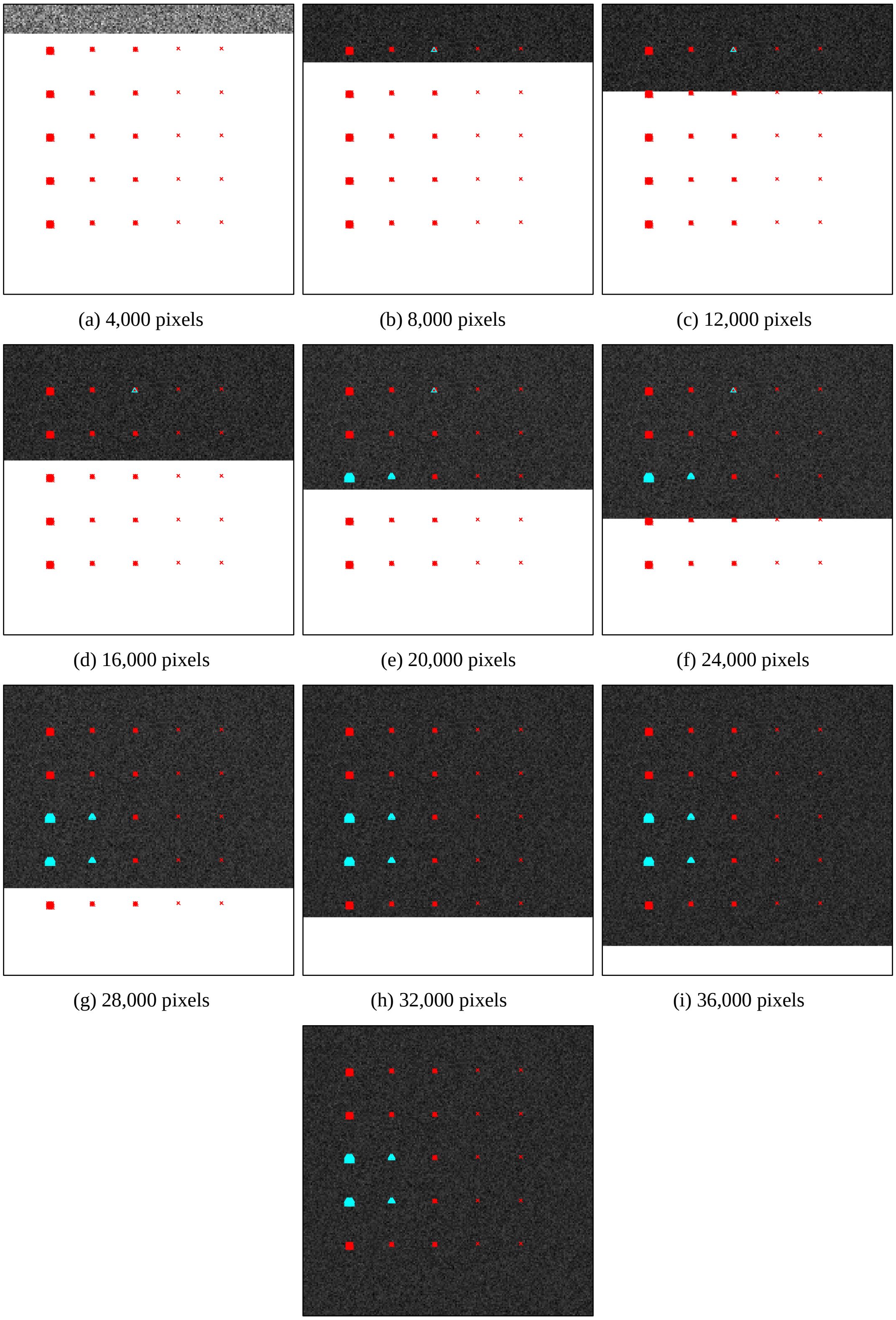


(e) M(1, 1)

Figure 4.5. PPI Count variations versus the number of processed bands and the number of processed pixels for each mineral signatures (a) A(1, 1) (b) B(1, 1) (c) C(1, 1) (d) K(1, 1) (e) M(1, 1)

RBP-C-PPI is further proposed to address the issue that partial pixels are available in the data collection process. Fig. 4.5 plots the changes of PPI Count with different number of processed bands as the number of processed pixels increases from 2,000 to 40,000 with step size 2,000 for the 5 mineral signatures, where x-axis denotes the number of bands collected and processed, y-axis displays the number of pixels included in the process, and z-axis presents the value of PPI Count correspondingly. As shown in the Fig. 4.5, the height of each bar represents the value of PPI Count for a particular combination of the number of processed bands and the number of processed bands. The magenta upward arrow indicates that one material begins to be identified as endmember candidates in the band that it first identified as an endmember candidate in RBP-PPI. For instance, signature B(1, 1) is first identified as endmember candidates with 27 bands available in RBP-PPI with 500 skewers. In the RBP-C-PPI process, it can be identified as endmember candidates with 14,000 pixels collected.

Moreover, Fig. 4.6 shows a RBP-C-PPI process with the number of processed pixels range from 4,000 to 40,000 with step size 4,000 in band 1. As displayed in the figure, the red cross indicates the ground truth location of the TI scene, the cyan upper triangle highlights the ground truth pixels that been identified as endmember candidates. The white area in the scene indicates the pixels in this area are not available yet. The beauty of RBP-C-PPI process is to extract the moving targets in the scene. For instance, sub-pixel material A on the third column can be identified as endmember candidates with 8,000 to 24,000 pixels available, and it vanished after that. RBP-P-PPI catches this characteristic properly.



(a) 4,000 pixels

(b) 8,000 pixels

(c) 12,000 pixels

(d) 16,000 pixels

(e) 20,000 pixels

(f) 24,000 pixels

(g) 28,000 pixels

(h) 32,000 pixels

(i) 36,000 pixels

(j) 40,000 pixels (full pixels)

Figure 4.6. An example of RBP-C-PPI process with different number of pixels processed in band 1. (a) 4,000 pixels (b) 8,000 pixels (c) 12,000 pixels (d) 16,000 pixels (e) 20,000 pixels (f) 24,000 pixels (g) 28,000 pixels (h) 32,000 pixels (i) 36,000 pixels (j) 40,000 pixels (full pixels)

B. Synthetic TE Data

Fig. 4.7 shows the plots of PPI count value versus the different number of bands processed for the 5 mineral signatures A, B, C, K, M in TE scene, where x-axis denotes the number of bands is being processed, y-axis indicates the different signatures, and z-axis presents the PPI-count value respectively. In this figure, 500 skewers are generated and utilized in the experiments. Since all of the 20 pure pixels for each signature on the first two columns can be found simultaneously, only the top left pixels on the first column for each signature are used for demonstrations and marked as A(1, 1), B(1, 1), C(1, 1), K(1, 1), M(1, 1). As shown in Fig. 4.7, the magenta arrow states the first band that a particular signature is identified as endmember candidate. For instance, signature B(1, 1) is first identified as endmember candidate with 21 bands collected and processed.

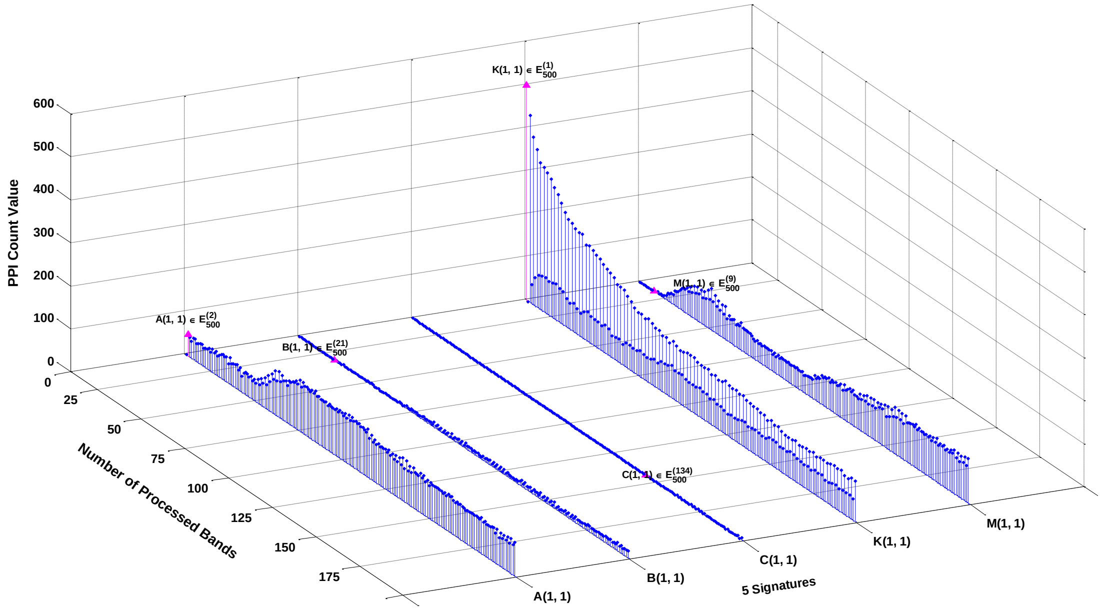
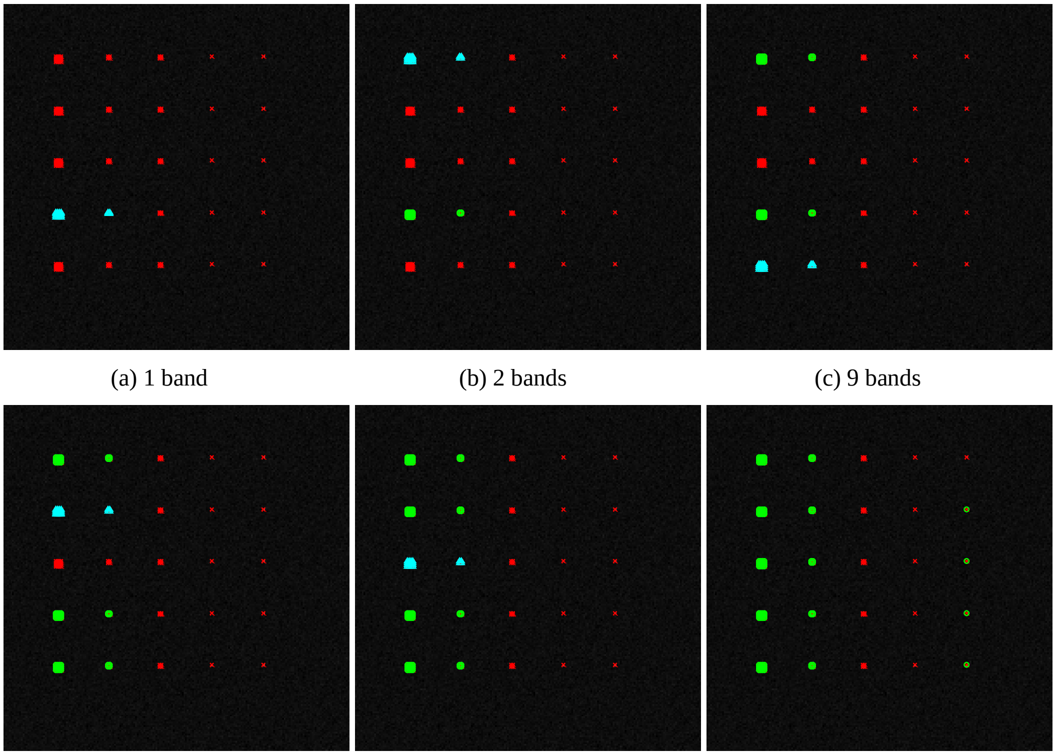


Figure 4.7. 3D plots of PPI counts versus the number of processed bands for the 5 different signatures

Fig. 4.8 shows the spatial location of the endmember candidates identified by RBP-PPI with different number of bands collected and processed. Fig. 4.8(a) starts with only 1 band, then progressively increases to 2 bands in Fig. 4.8(b), 9 bands in Fig. 4.8(c), 21 bands in Fig. 4.8(d), 134 bands in Fig. 4.8(e), and reaches the full bands 189 bands in Fig. 4.8(f) in the end, each of which shows the endmember candidates finding results by RBP-PPI in the particular transition bands specified by Fig. 4.7. In Fig. 4.8, the red cross indicates the ground truth pixels in TE scene, the cyan upper triangle highlights the endmember candidates found in the current band, and the endmember candidates identified in the previous bands are marked by green circles as well.



(a) 1 band (b) 2 bands (c) 9 bands

(d) 21 bands (e) 134 bands (f) 189 bands (full bands)

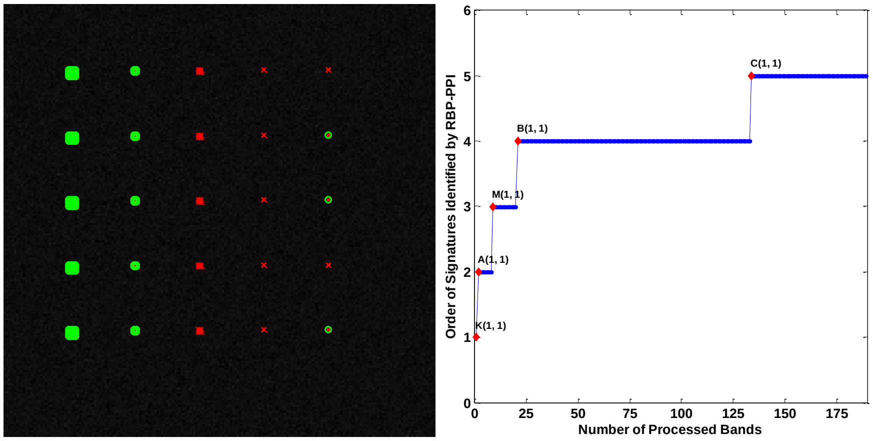
Figure 4.8. Endmember candidates finding results by RBP-PPI with different number of bands collected and processed: (a) 1 band (b) 2 bands (c) 9 bands (d) 21 bands (e) 134 bands (f) 189 bands (full bands)

Table 4.2 summarizes the 5 mineral signatures A, B, C, K, M among the endmember candidates identified by RBP-PPI when n_l varies from 1 band, 2 bands, 9 bands, 21 bands, 134 bands to full bands 189 bands. As demonstrated in the table, $E^{(l)}_{N_{skewers}}$ in the 2nd column was identified using $N_{skewers}$ skewer sets with first *l* bands collected and processed.

Table 4.2. Summary of the mineral signatures being identified by RBP-PPI versus n_l for TE scene

n_l	Signatures identified by RBP-PPI
1	$\mathrm{K}(1,1) \in E^{(1)}_{500}$
2	$\mathrm{A}(1,1) \in E^{(2)}_{500}$
9	$\mathrm{M}(1,1) \in E^{(9)}_{500}$
21	$\mathrm{B}(1,1) \in E^{(21)}_{500}$
134	$\mathrm{C}(1,1) \in E^{(134)}_{500}$
189	$\mathrm{A}(1,1) \in E^{(189)}_{500}, \mathrm{B}(1,1) \in E^{(189)}_{500}, \mathrm{C}(1,1) \in E^{(189)}_{500}, \mathrm{K}(1,1) \in E^{(189)}_{500}, \mathrm{M}(1,1) \in E^{(189)}_{500}$

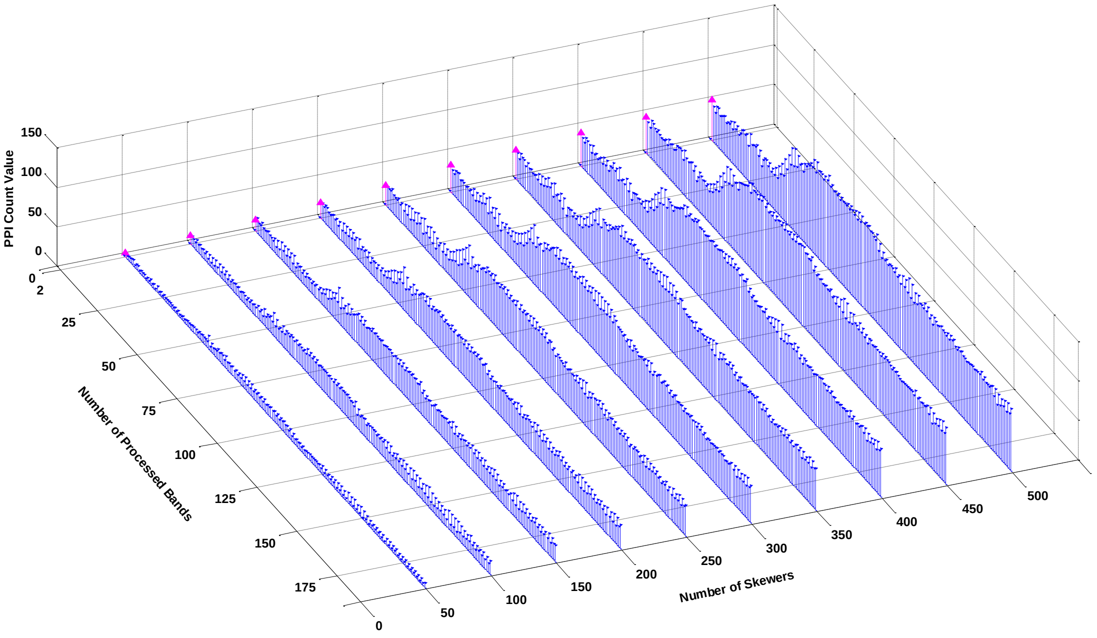
Furthermore, Fig. 4.9(a) displays the endmember candidates identified by PPI for TE scene using 500 skewers with all the bands received and processed. As shown in Fig. 4.9(a), 5 mineral signatures and 3 sub-pixel can be found as endmember candidates. However, with only 134 bands collected and processed all the five signatures can be found with RBP-PPI. In addition, 4 sub-pixel can be found as endmember candidates by apply RBP-PPI onto TE scene, one additional sub-pixel can be discovered comparing with the original PPI. Moreover, Fig. 4.9(b) plots the order of each mineral signature A, B, C, K and M been identified as endmember candidates versus the number of bands processed. It is clear to see that with only 134 bands available, all the 5 signatures have been found.



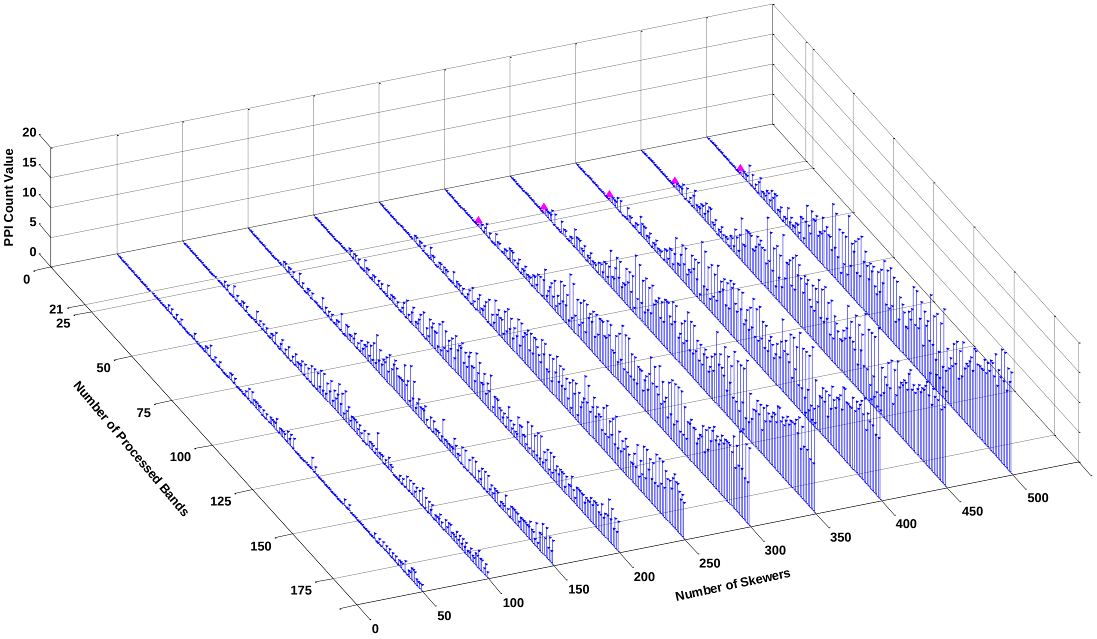
(a) (b)

Figure 4.9. Comparative study between PPI and RBP-PPI (a) PPI identified signatures (b) The order of signatures identified by RBP-PPI

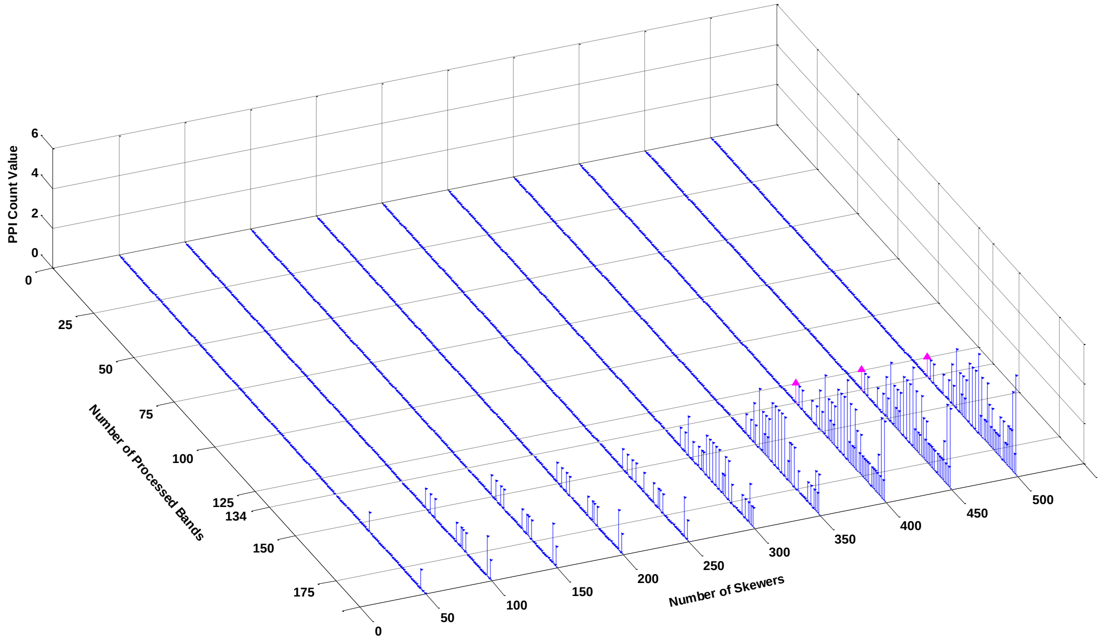
As stated in the previous sections, 500 skewers are generated and used in the RBP-PPI process for TE scene. However, choosing a proper number of skewers remains an unsolved problem in the literature. RBP-P-PPI is further proposed when it comes to address the issue that how many skewers need to be generated for process. Fig. 4.10 plots the variation of PPI Count versus the number of processed bands as the number of skewers increases from 50 to 500 with step size 50 for the 5 mineral signatures, where x-axis denotes the number of bands collected and processed, y-axis displays the number of skewers generated for RBP-P-PPI process, and z-axis presents the value of PPI Count correspondingly.


PPI Count Value
150
100
50
0
0
2
25
50
75
100
125
150
175
Number of Processed Bands
0
50
100
150
200
250
300
350
400
450
500
Number of Skewers

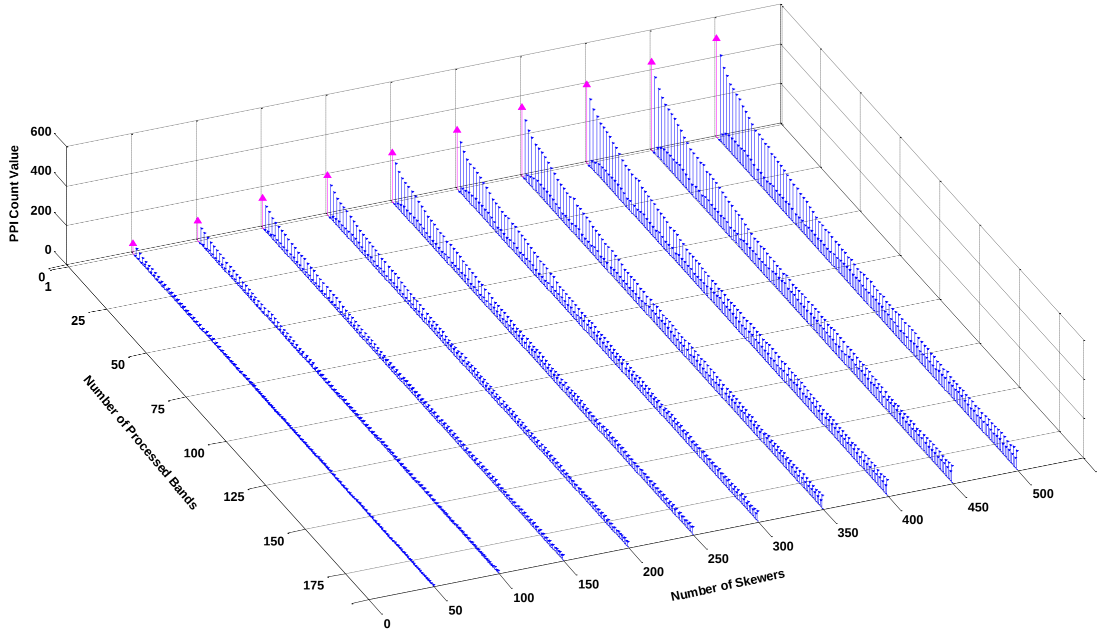
(a) A(1, 1)


PPI Count Value
20
15
10
5
0
0
21
25
50
75
100
125
150
175
Number of Processed Bands
0
50
100
150
200
250
300
350
400
450
500
Number of Skewers

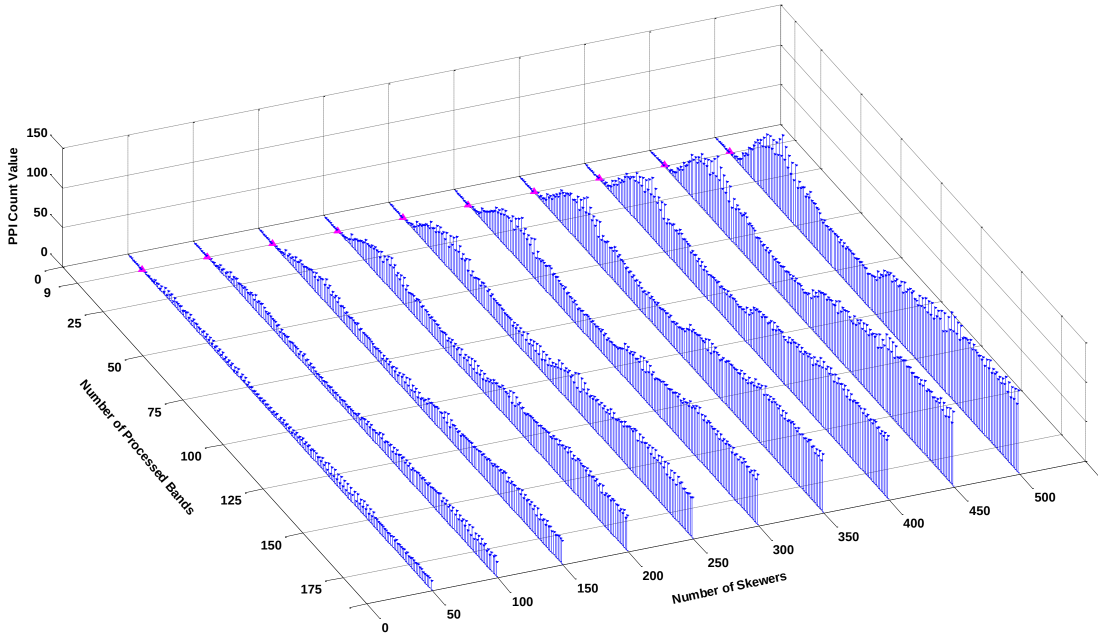
(b) B(1, 1)


PPI Count Value
0
2
4
6
0
25
50
75
100
125
134
150
175
Number of Processed Bands
0
50
100
150
200
250
300
350
400
450
500
Number of Skewers

(c) C(1, 1)


PPI Count Value
0
200
400
600
0
1
25
50
75
100
125
150
175
Number of Processed Bands
0
50
100
150
200
250
300
350
400
450
500
Number of Skewers

(d) K(1, 1)

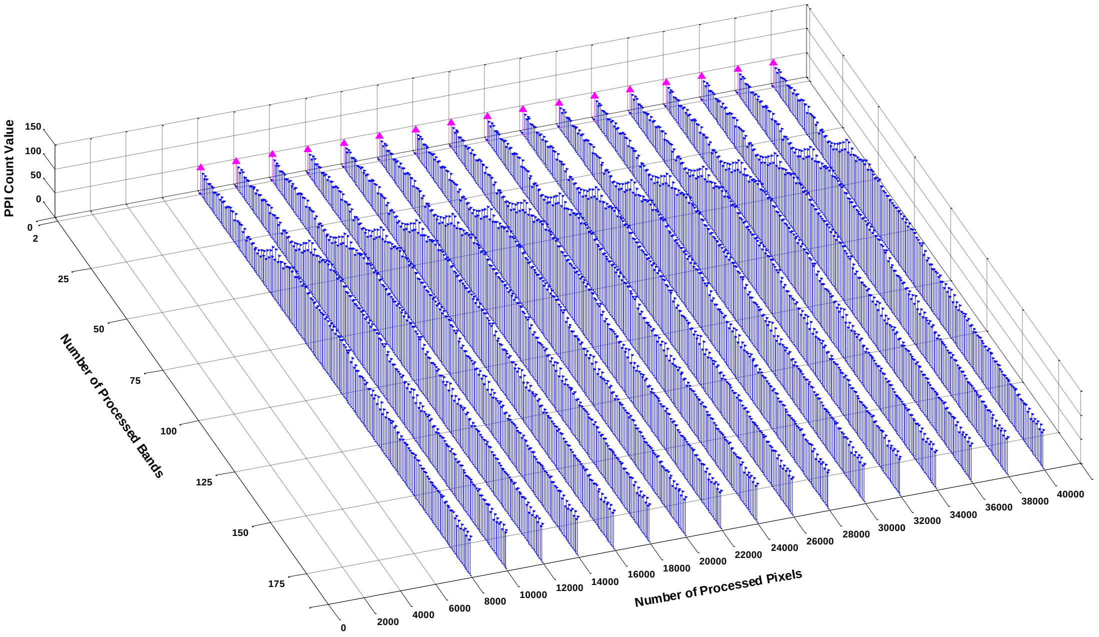


(e) M(1, 1)

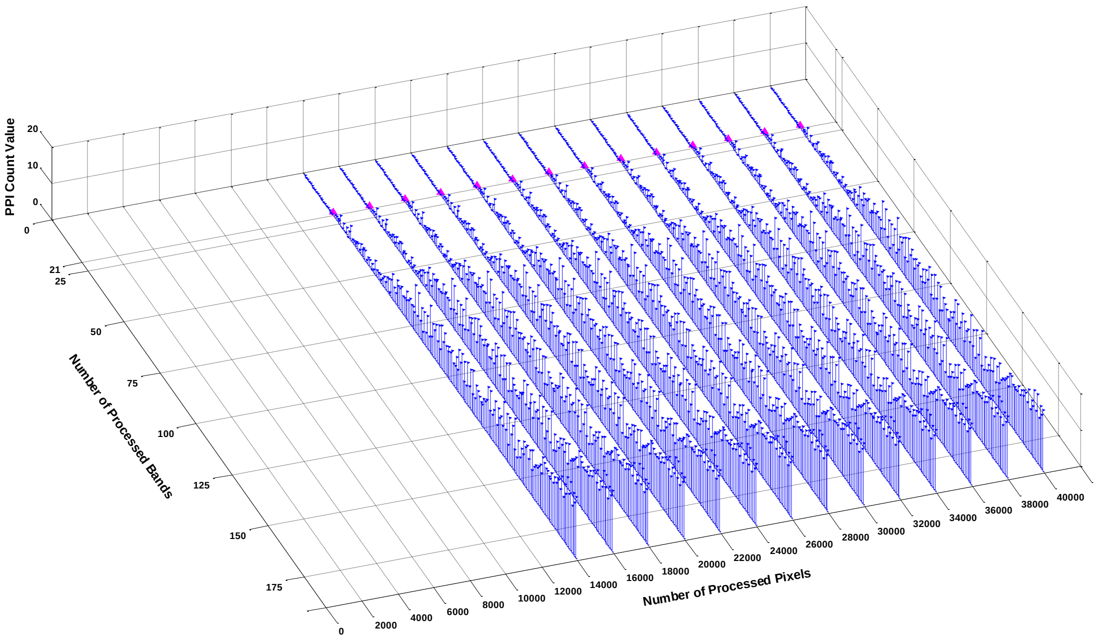
Figure 4.10. PPI Count variations versus the number of processed bands and the number of skewers for each mineral signatures (a) A(1, 1) (b) B(1, 1) (c) C(1, 1) (d) K(1, 1) (e) M(1, 1)

As shown in the figure, the height of each bar represents the value of PPI Count for a particular combination of the number of processed bands and the number of skewers generated for process. The magenta upward arrow indicates that one material begins to be identified as endmember candidates in the band that it first identified as an endmember candidate in RBP-PPI. For instance, signature B(1, 1) is first identified as endmember candidates with 21 bands available in RBP-PPI with 500 skewers. In the RBP-P-PPI process, it can be identified as endmember candidates with only 300 skewers generated. The less skewers generated in the process could reduce the computational complexity. Based on our extensive experiments, the spectrum of material C is very similar to background, and it is difficult to identify. This is the reason that it is the last material be discovered and it can be detected with 134 bands collected and processed, which requires the largest number of skewers as well.

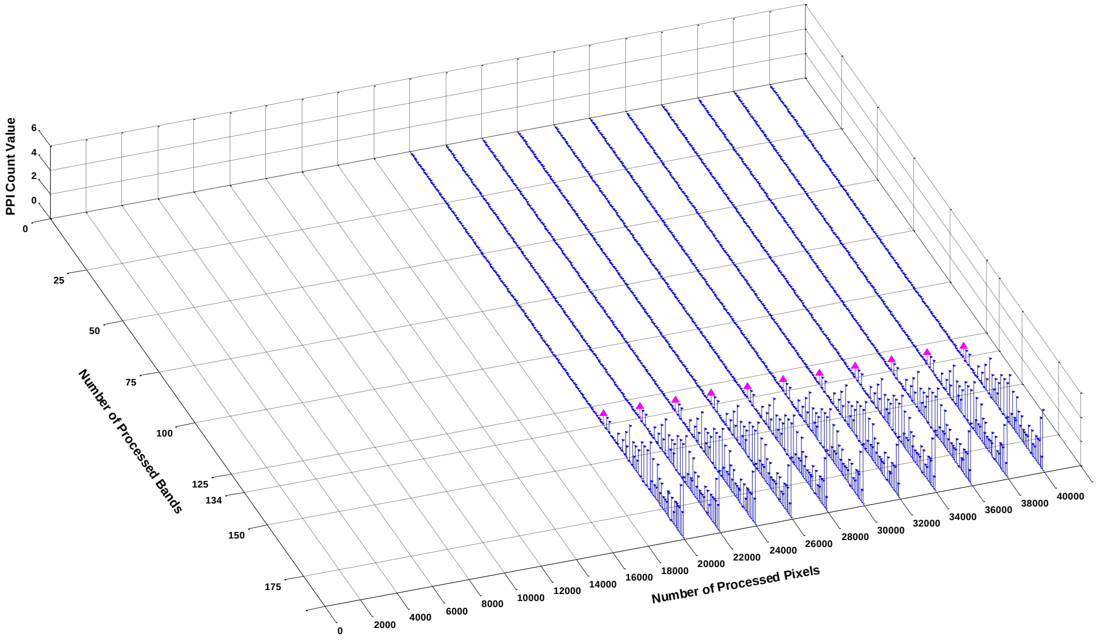
RBP-C-PPI is further proposed to address the issue that partial pixels are available in the data collection process. Fig. 4.11 plots the changes of PPI Count with different number of processed bands as the number of processed pixels increases from 2,000 to 40,000 with step size 2,000 for the 5 mineral signatures in TE scene, where x-axis denotes the number of bands collected and processed, y-axis displays the number of pixels included in the process, and z-axis presents the value of PPI Count correspondingly.


PPI Count Value
150
100
50
0
0
2
25
50
75
100
125
150
175
Number of Processed Bands
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
22000
24000
26000
28000
30000
32000
34000
36000
38000
40000
Number of Processed Pixels

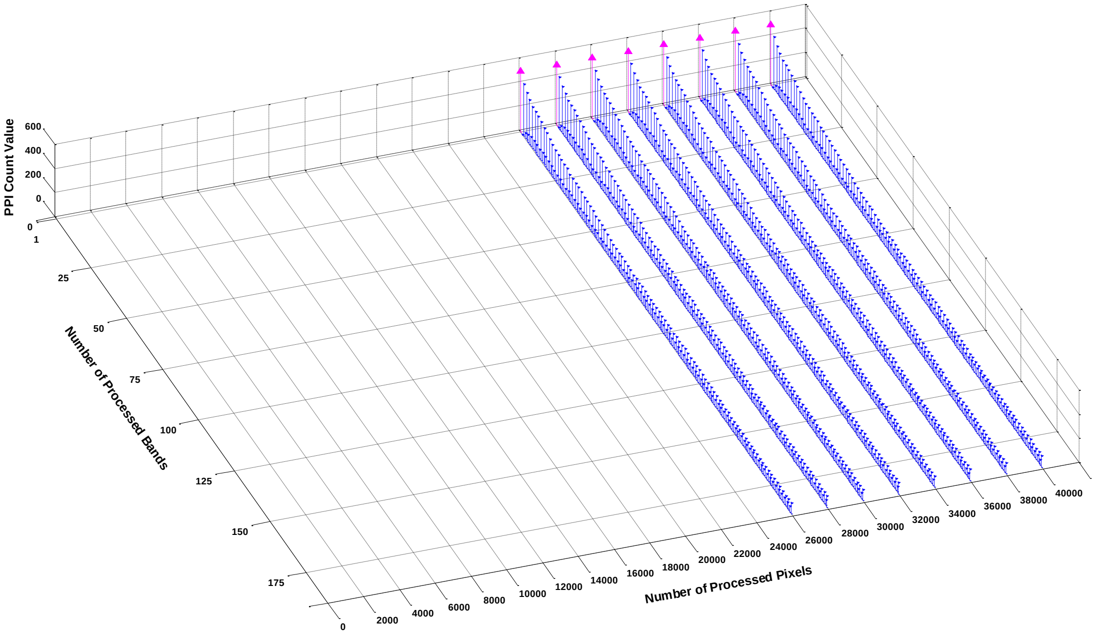
(a) A(1, 1)


PPI Count Value
20
10
0
0
21
25
50
75
100
125
150
175
Number of Processed Bands
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
22000
24000
26000
28000
30000
32000
34000
36000
38000
40000
Number of Processed Pixels

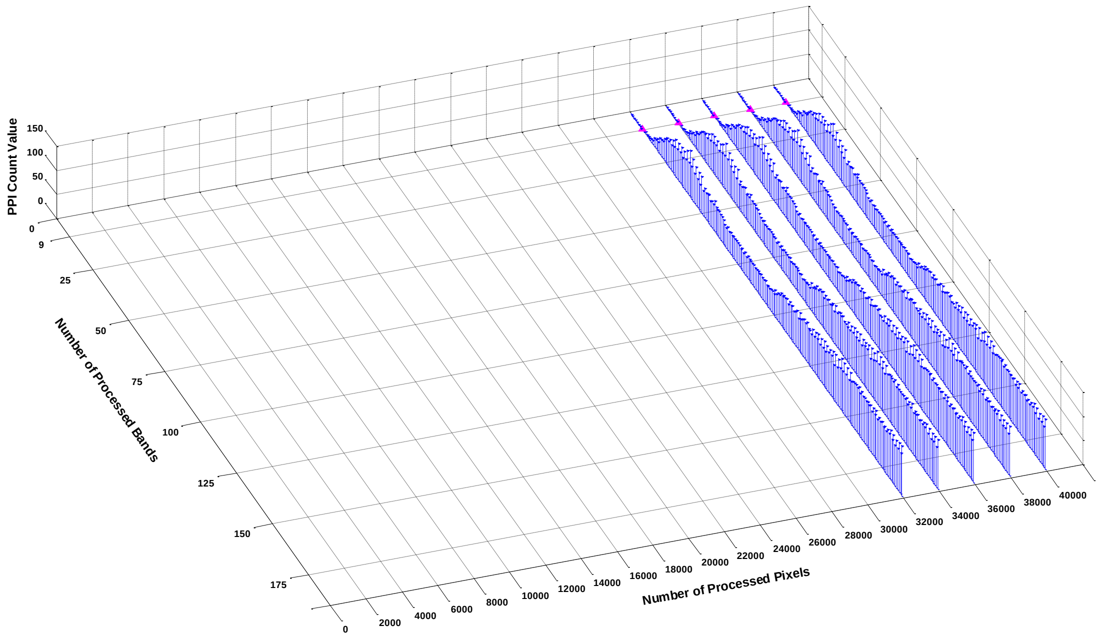
(b) B(1, 1)


PPI Count Value
6
4
2
0
0
25
50
75
100
125
134
150
175
Number of Processed Bands
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
22000
24000
26000
28000
30000
32000
34000
36000
38000
40000
Number of Processed Pixels

(c) C(1, 1)


PPI Count Value
600
400
200
0
0
1
25
50
75
100
125
150
175
Number of Processed Bands
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
22000
24000
26000
28000
30000
32000
34000
36000
38000
40000
Number of Processed Pixels

(d) K(1, 1)

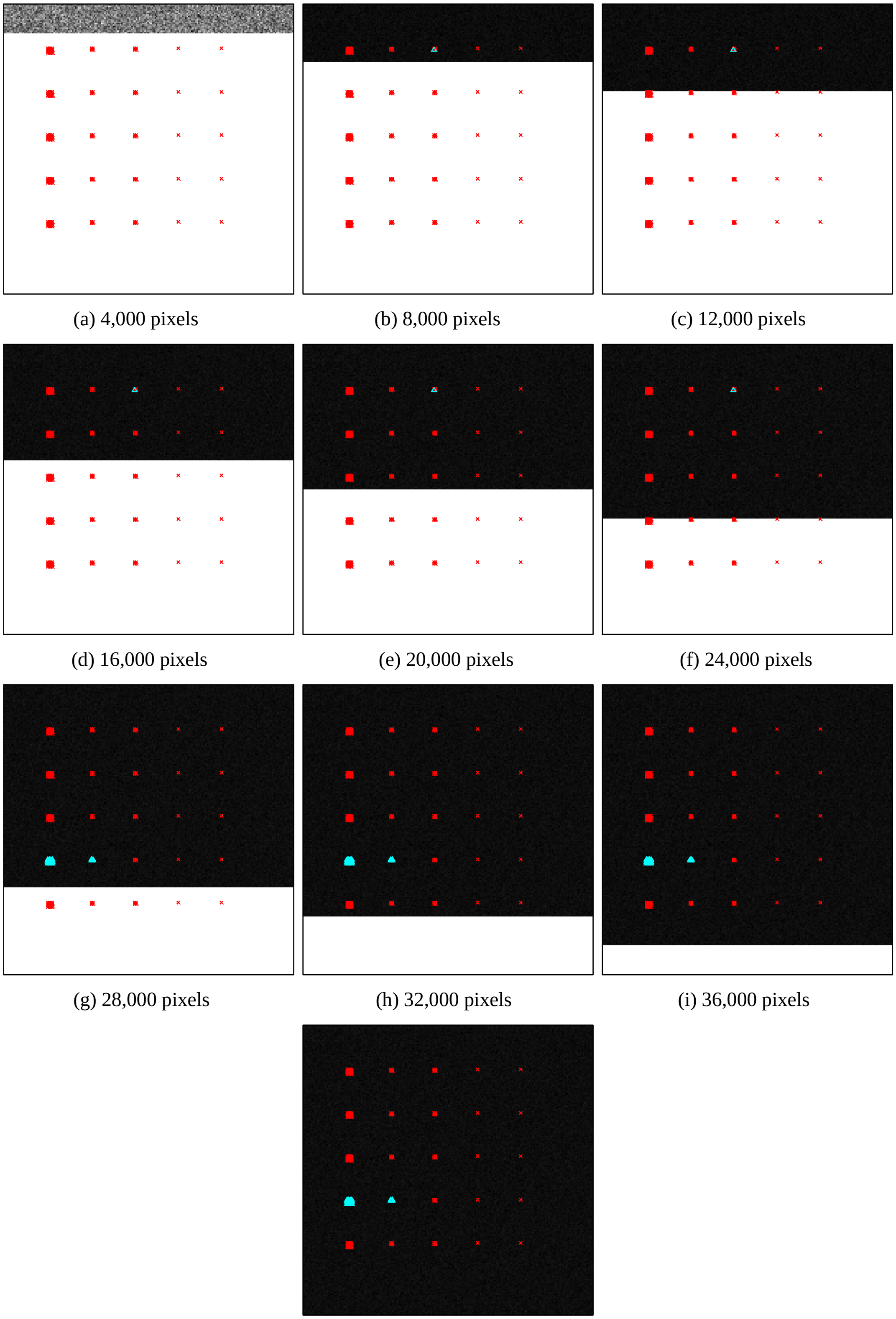


(e) M(1 1)

Figure 4.11. PPI Count variations versus the number of processed bands and the number of processed pixels for each mineral signatures (a) A(1, 1) (b) B(1, 1) (c) C(1, 1) (d) K(1, 1) (e) M(1, 1)

As shown in the Fig. 4.11, the height of each bar represents the value of PPI Count for a particular combination of the number of processed bands and the number of processed bands. The magenta upward arrow indicates that one material begins to be identified as endmember candidates in the band that it first identified as an endmember candidate in RBP-PPI. For instance, signature B(1, 1) is first identified as endmember candidates with 21 bands available in RBP-PPI with 500 skewers. In the RBP-C-PPI process, it can be identified as endmember candidates with 14,000 pixels collected.

Moreover, Fig. 4.12 shows a RBP-C-PPI process with the number of processed pixels range from 4,000 to 40,000 with step size 4,000 in band 1. As displayed in the figure, the red cross indicates the ground truth location of the TI scene, the cyan upper triangle highlights the ground truth pixels that been identified as endmember candidates. The white area in the scene indicates the pixels in this area are not available yet. The beauty of RBP-C-PPI process is to extract the moving targets in the scene. For instance, sub-pixel material A on the third column can be identified as endmember candidates with 8,000 to 24,000 pixels available, and it vanished after that. RBP-P-PPI catches this characteristic properly.



(a) 4,000 pixels

(b) 8,000 pixels

(c) 12,000 pixels

(d) 16,000 pixels

(e) 20,000 pixels

(f) 24,000 pixels

(g) 28,000 pixels

(h) 32,000 pixels

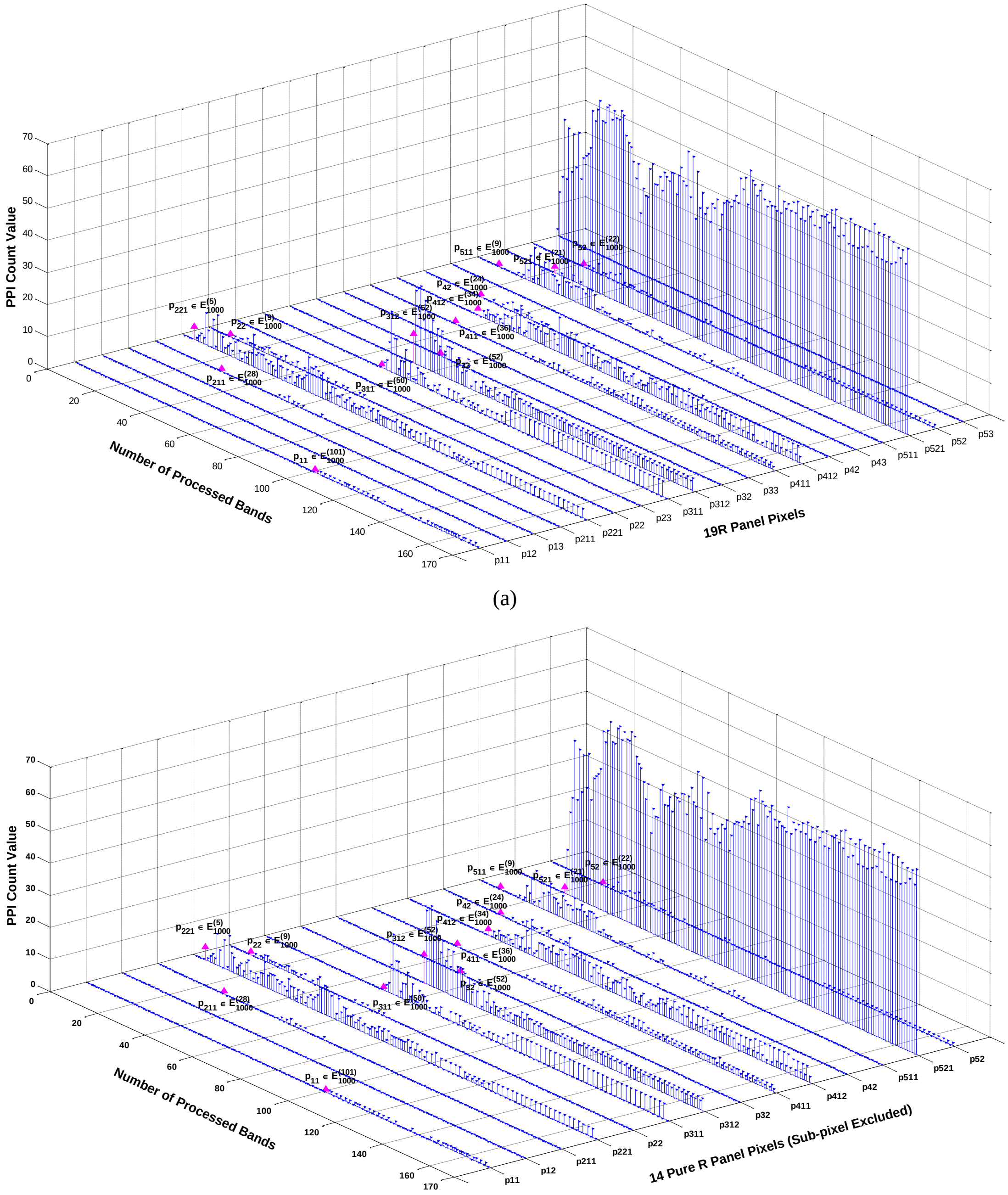
(i) 36,000 pixels

(j) 40,000 pixels (full pixels)

Figure 4.12. An example of RBP-C-PPI process with different number of pixels processed in band 1. (a) 4,000 pixels (b) 8,000 pixels (c) 12,000 pixels (d) 16,000 pixels (e) 20,000 pixels (f) 24,000 pixels (g) 28,000 pixels (h) 32,000 pixels (i) 36,000 pixels (j) 40,000 pixels (full pixels)

4.4.2 Real Data (HYDICE Scene)

Fig. 4.13(a-b) shows the plots of PPI count value versus the different number of bands processed for the different 19 R-panel pixels with 1,000 skewers being used for PPI where the *x*-axis is the number of bands processed , *y*-axis is the coordinates of 19 R-panel pixels according to the order of row by row, i.e., p_{11}, p_{12}, p_{13}, p_{211}, p_{221}, p_{22}, p_{23}, p_{311}, p_{312}, p_{32}, p_{33}, p_{411}, p_{412}, p_{42}, p_{43}, p_{511}, p_{521}, p_{52}, p_{53}, and *z*-axis is the corresponding PPI count value. Fig. 4.13 (a) plots the variation of the PPI count for all of the 19 R-panel pixels, while Fig. 4.13 (b) only plots the changes of PPI count for the 14 pure R-panel pixels and excludes the 5 sub-pixels which make the 3D graph clearer. In these two graphs, the height of the lines indicates the value of PPI count. The magenta arrows in the graph illustrate the transition bands, which present the PPI count of a particular panel pixels increases from zero to a non-zero value and have been first found as an endmember candidate in the current band being processed. For instance, the PPI count value of panel pixel p_{11} jumps from 0 to 1 and is first identified as an endmember candidate after 101 bands have been received and processed. As we can see in the figure, the PPI count of the panel pixels on the 1st column is stronger than the PPI count of the one-pixel-size panel pixels on the 2nd column. Meanwhile, PPI count of the subpixel panels on the 3rd column stay 0 during the entire process, in other words, sub-pixel cannot be found by PBP-PPI. It is clear to see that the PPI count of both p_{511} and p_{521} is stronger than p_{52}, and PPI count of p_{53} is 0 during the entire process and cannot be identified. PPI count of the panel pixels tend to stabilized as more bands are added into the process. Without PBP-PPI we would not be able to evidence such valuable information which was compromised by subsequent bands.

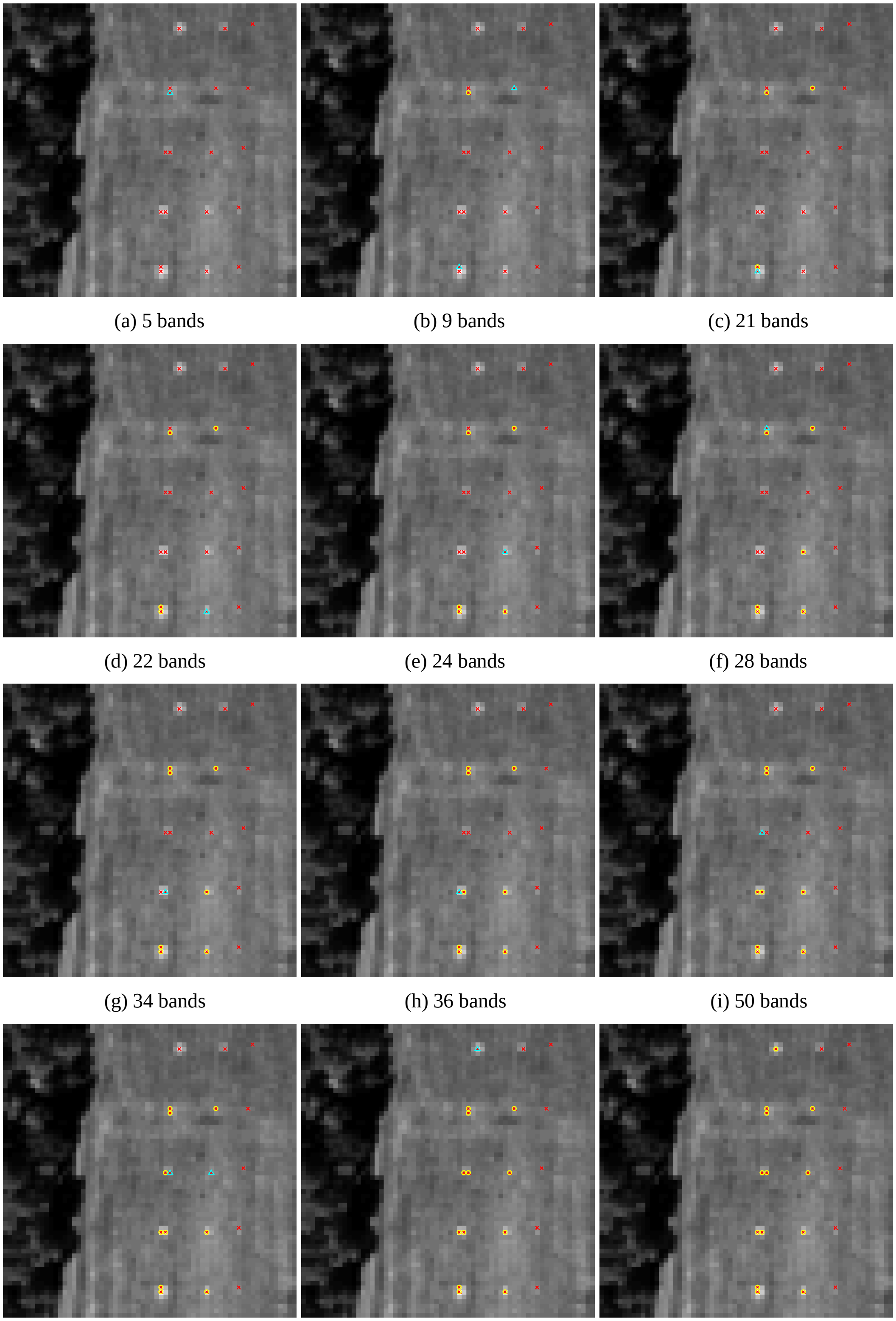


(a)

(b)

Figure 4.13. 3D plots of PPI counts versus the number of processed bands for different R-panel pixels (a) PPI count variation for the 19 R-panel pixels (b) PPI count variation for the 14 pure R-panel pixels

Furthermore, Fig. 4.14 highlights the panel pixels with PPI count greater than 0 in the transition bands. In the following figures, the red cross indicates the position of the R-panel pixels, the cyan upper triangle displays the panel pixels with PPI count greater than 0 found in current band. The yellow circle represents the panel pixels have been found in the previous bands processed. As we can see, after band 101, 13 out of 19 panel pixels have been identified as endmember candidate.



(a) 5 bands (b) 9 bands (c) 21 bands

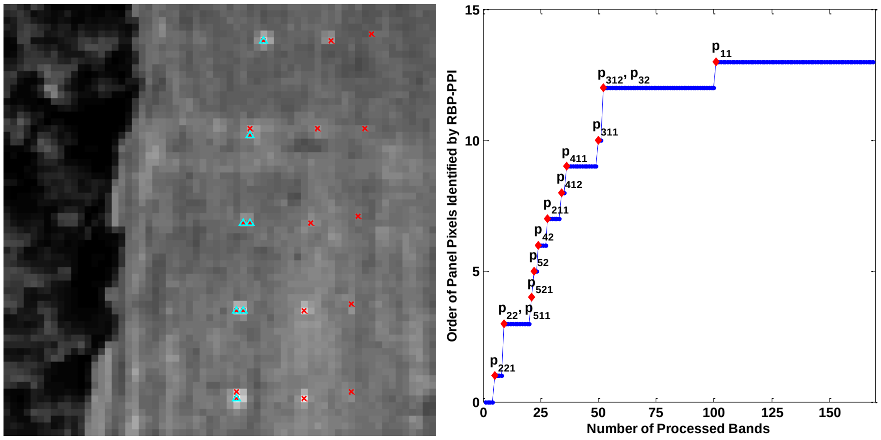
(d) 22 bands (e) 24 bands (f) 28 bands

(g) 34 bands (h) 36 bands (i) 50 bands

(j) 52 bands (k) 101 bands (l) 169 bands (full bands)

Figure 4.14. Panel pixels with PPI count greater than 0 after different number of bands processed. (a) 5 bands (b) 9 bands (c) 21 bands (d) 22 bands (e) 24 bands (f) 28 bands (g) 34 bands (h) 36 bands (i) 50 bands (j) 52 bands (k) 101 bands (l) 169 bands (full bands)

Figure 4.15 conducts a comparative study between the original PPI and the proposed RBP-PPI applying on HYDICE scene. Fig. 4.15(a) displays the ground truth pixels identified as endmember candidates by PPI. In Fig. 4.15(a), the red cross indicates the spatial location of the R-panel pixel in HYDICE scene, the cyan upper triangle indicates the ground truth pixels identified as endmember candidates. There are a total of 7 R-panel pixels been identified as endmember candidates. However, according to Fig. 4.13(a) and Fig. 4.14, there are a total of 13 R-panel pixels were identified as endmember candidates by RBP-PPI, p_{11}, p_{211}, p_{221}, p_{22}, p_{311}, p_{312}, p_{32}, p_{411}, p_{412}, p_{42}, p_{511}, p_{521}, p_{52} being identified by PBP-PPI. Fig. 4.15(b) further shows the order of these 13 panel pixels found by PBP-PPI where *x*-axis denotes the number of bands being processed and *y*-axis represents the order of panel pixels being identified. For instance, p_{221} is the 1st panel pixels being identified with only 5 bands collected, p_{22} and p_{521} is the 2nd and 3rd panel pixels identified after collected 9 bands. After 101 bands were processed, the last panel pixel p_{11} was found.



(a) (b)

Figure 4.15. Comparative study between PPI and RBP-PPI (a) PPI identified signatures (b) The order of signatures identified by RBP-PPI

As noted in the previous discussions, RBP-PPI will identify a set of endmember candidate band-by-band progressively. Figure 4.16 plots the number of distinct endmember candidates found by RBP-PPI as the number of processed bands gradually increasing with 1000 skewers on HYDICE data set. RBP-PPI will identify a total of 373 endmember candidates for the next step process. At the same time, the number of distinct endmember candidate found by RBP-PPI is stabilized as more bands are included in the process. After 106 bands processed, the number of distinct endmember candidate stops varying and achieves the same results as full bands added into process.

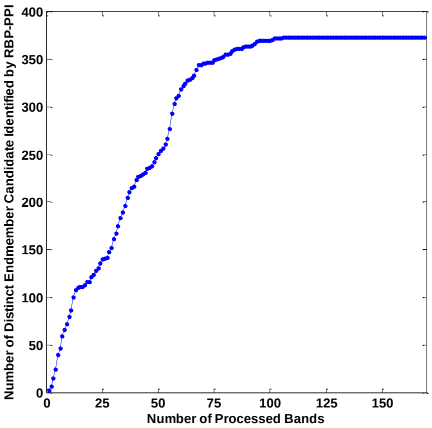


Figure 4.16. The plot of number of distinct endmember candidates versus number of processed bands for HYDICE data

Furthermore, Fig. 4.17 plots a 3-dimensional histogram to show how frequently a particular pixel was picked up by RBP-PPI as an endmember candidate where the x-axis and y-axis corresponding to the columns and rows of special location of HYDICE scene. It should be noted that the z-axis in Fig. 4.17 indicates the number of times a given pixel at a particular special location was identified as an endmember candidate. For HYDICE scene with 169 bands the maximum value for z-axis is 169. As shown in Fig 4.17, the magenta upward arrows indicate the spatial location of the 13 R-panel pixels.

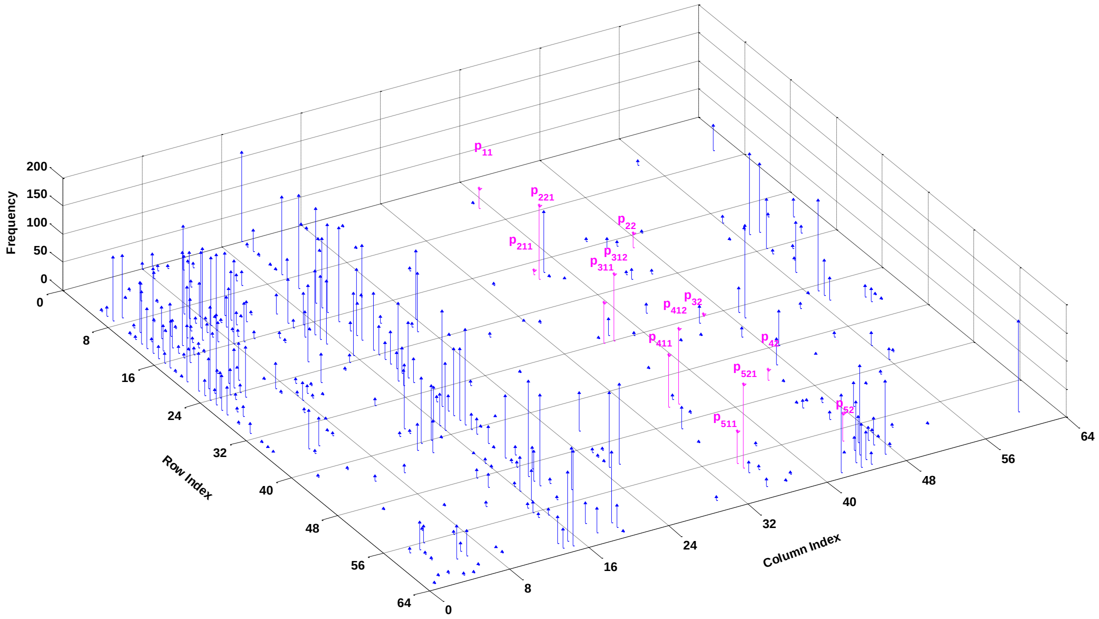
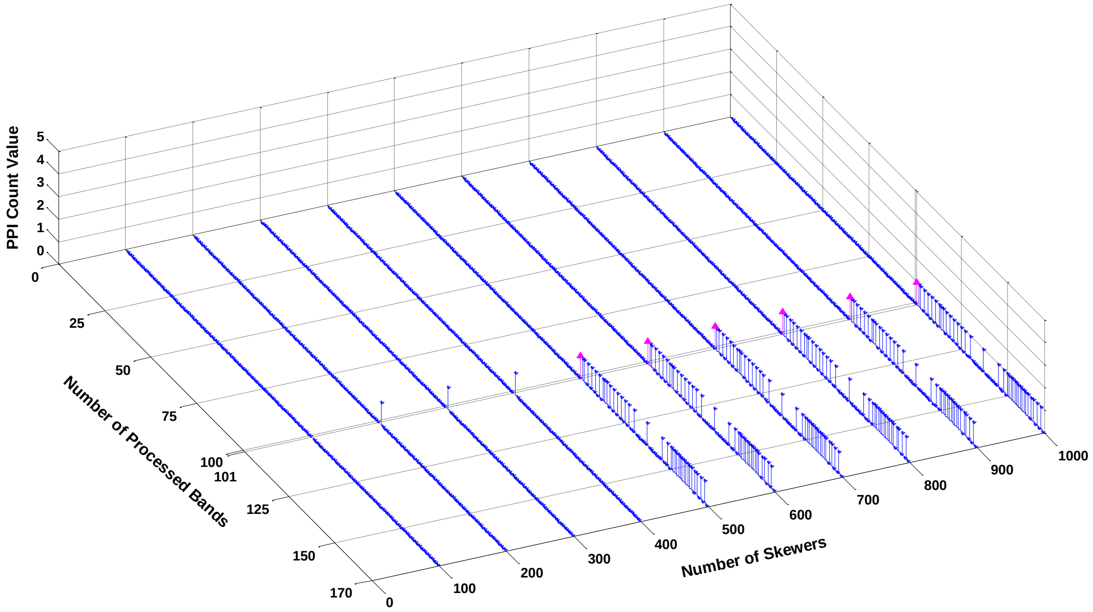
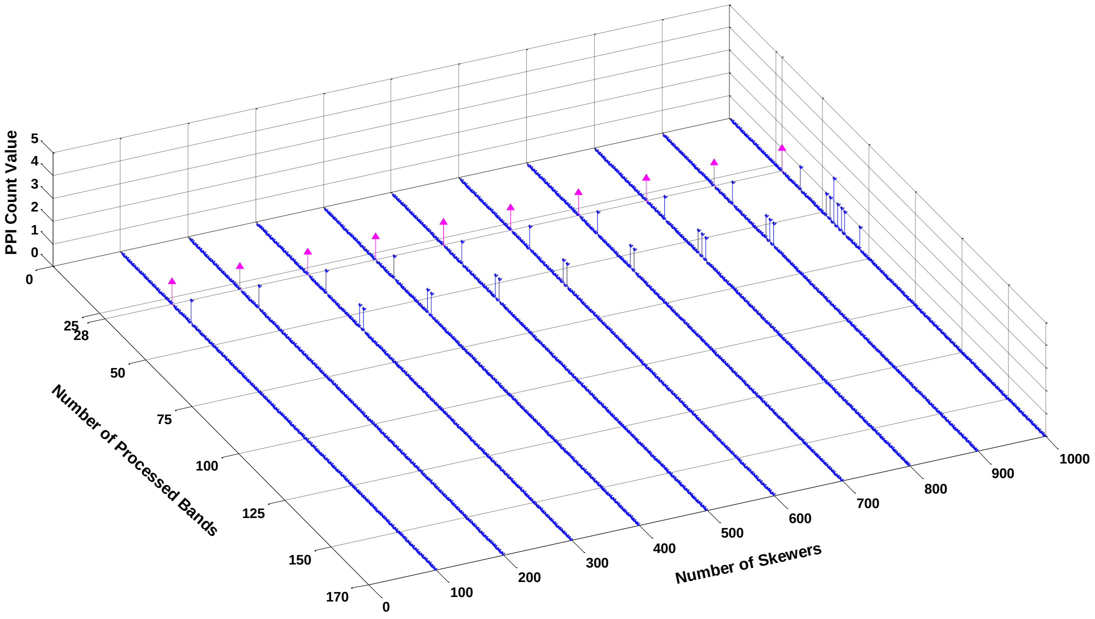


Figure 4.17. 3D Histogram of the pixels picked up by RBP-PPI as endmember candidates

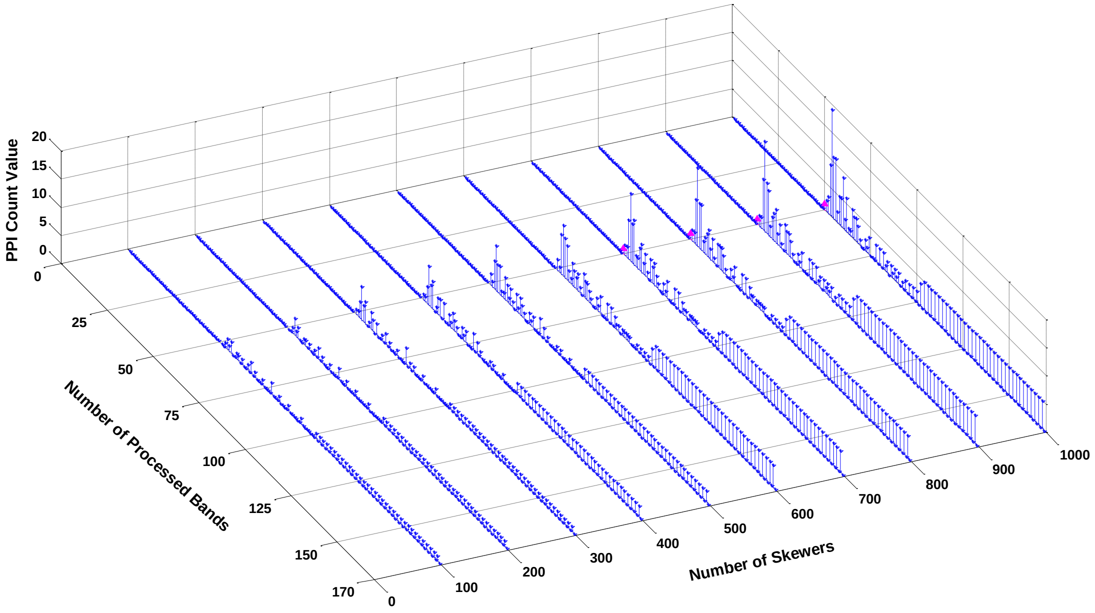
As stated in the previous sections, 1000 skewers are generated and used in the RBP-PPI process for HYDICE scene. However, choosing a proper number of skewers remains an unsolved problem in the literature. RBP-P-PPI is further proposed when it comes to address the issue that how many skewers need to be generated for process. Fig. 4.18 plots the variation of PPI Count versus the number of processed bands as the number of skewers increases from 100 to 1000 with step size 100 for R-panel pixels p_{11}, p_{211}, p_{311}, p_{411}, and p_{511}, where x-axis denotes the number of bands collected and processed, y-axis displays the number of skewers generated for RBP-P-PPI process, and z-axis presents the value of PPI Count correspondingly.



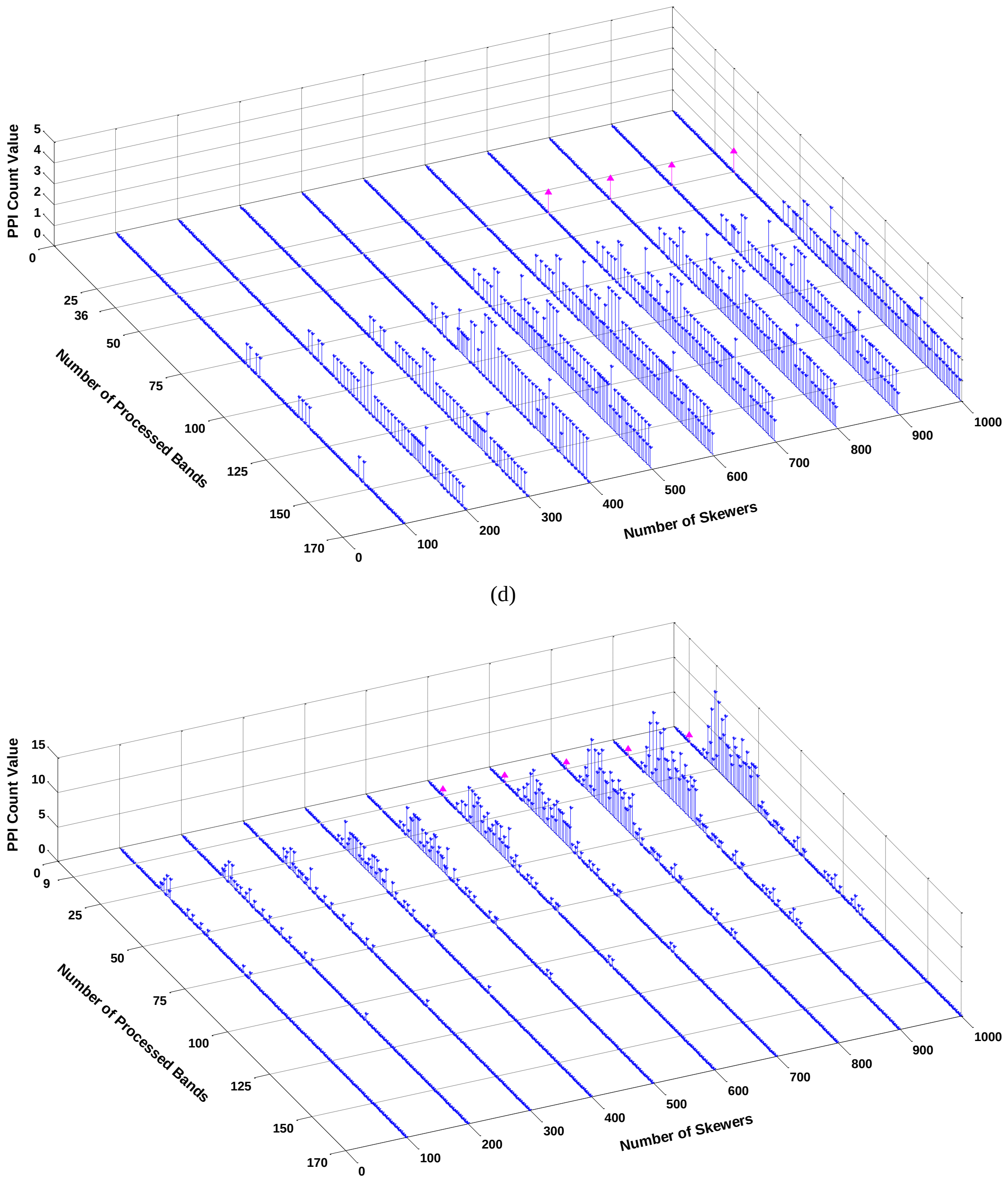
(a)


PPI Count Value
5
4
3
2
1
0
0
25
28
50
75
100
125
150
170
Number of Processed Bands
0
100
200
300
400
500
600
700
800
900
1000
Number of Skewers

(b)


PPI Count Value
20
15
10
5
0
0
25
50
75
100
125
150
170
Number of Processed Bands
0
100
200
300
400
500
600
700
800
900
1000
Number of Skewers

(c)



(d)

(e)

Figure 4.18. PPI Count value versus the different number of processed bands and different number of skewers (a) p_{11} (b) p_{211} (c) p_{311} (d) p_{411} (e) p_{511}

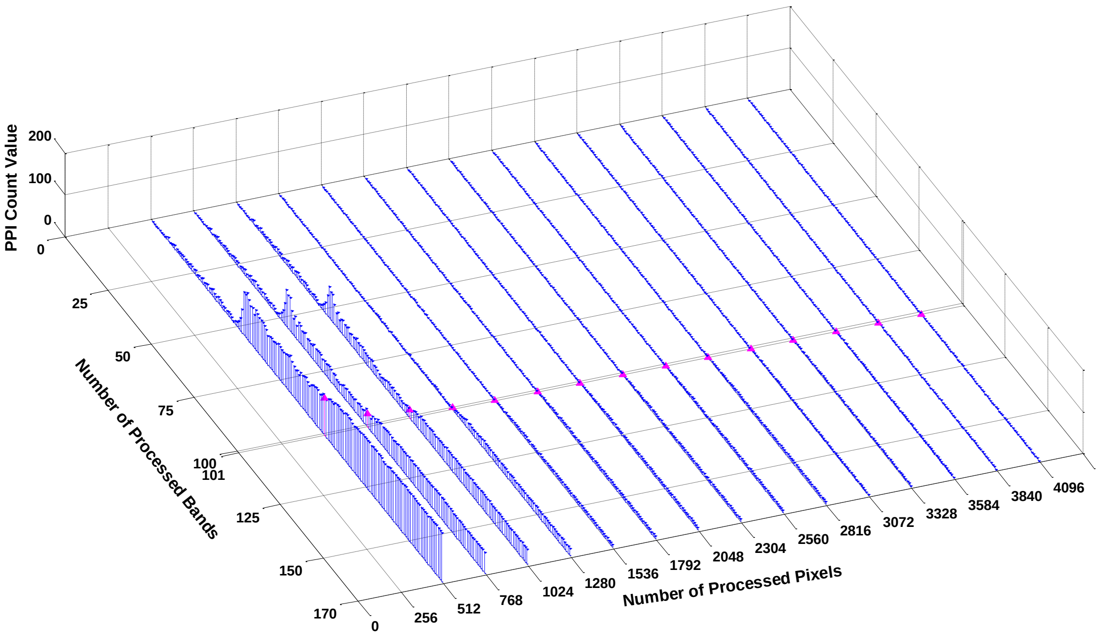
As shown in the figure, the height of each bin represents the value of PPI Count for a particular combination of the number of processed bands and the number of skewers generated for process. The magenta upward arrow indicates that one material begins to be identified as endmember candidates in the band that it first identified as an endmember candidate in RBP-PPI. For instance, R-panel pixel p_{211} is first identified as endmember candidates with 28 bands available in RBP-PPI with 100 skewers. However, in the RBP-P-

PPI process, it can be identified as endmember candidates with only 100 skewers generated. Table 4.3 summarizes the number of bands processed that a particular pixel is first found as endmember candidates and the minimum number of skewers required to extract the particular pixel for all the R-panel pixels on the 1st column been identified as endmember candidate.

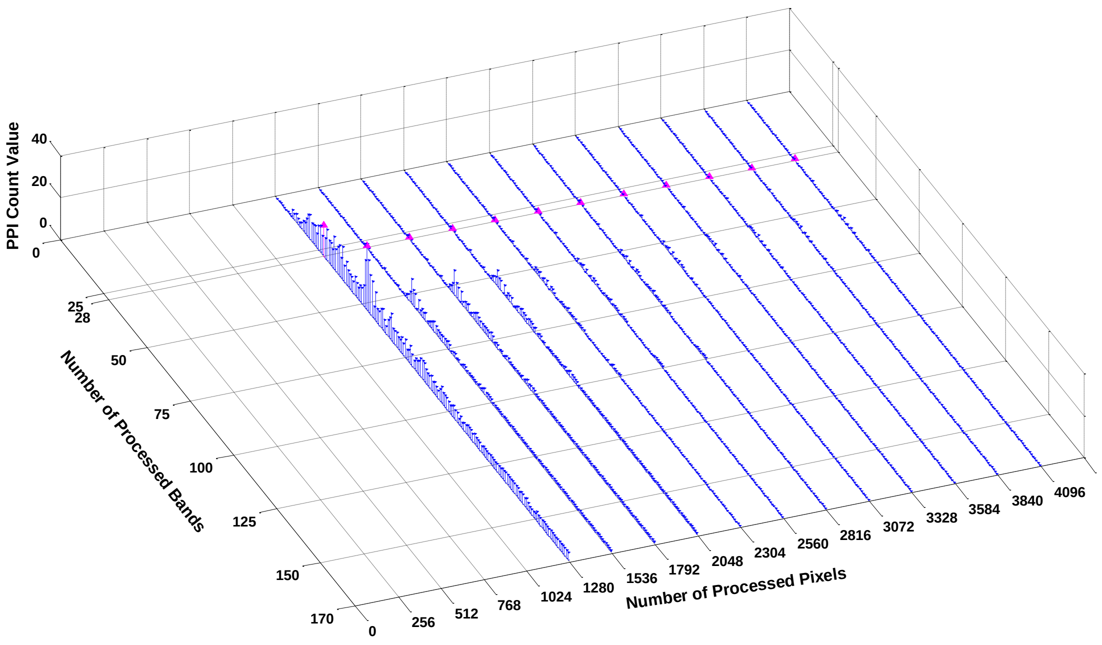
Table 4.3. Summary for all the R-panel pixels on the 1st column in RBP-P-PPI process

R-panel pixels on the 1st column	Number of bands processed that a pixel is first found as an endmember candidate	Minimum number of skewers required to extract the particular pixel
p_{11}	101	500
p_{211}	28	100
p_{221}	5	500
p_{311}	50	700
p_{312}	52	100
p_{411}	36	700
p_{412}	34	400
p_{511}	9	600
p_{521}	21	700

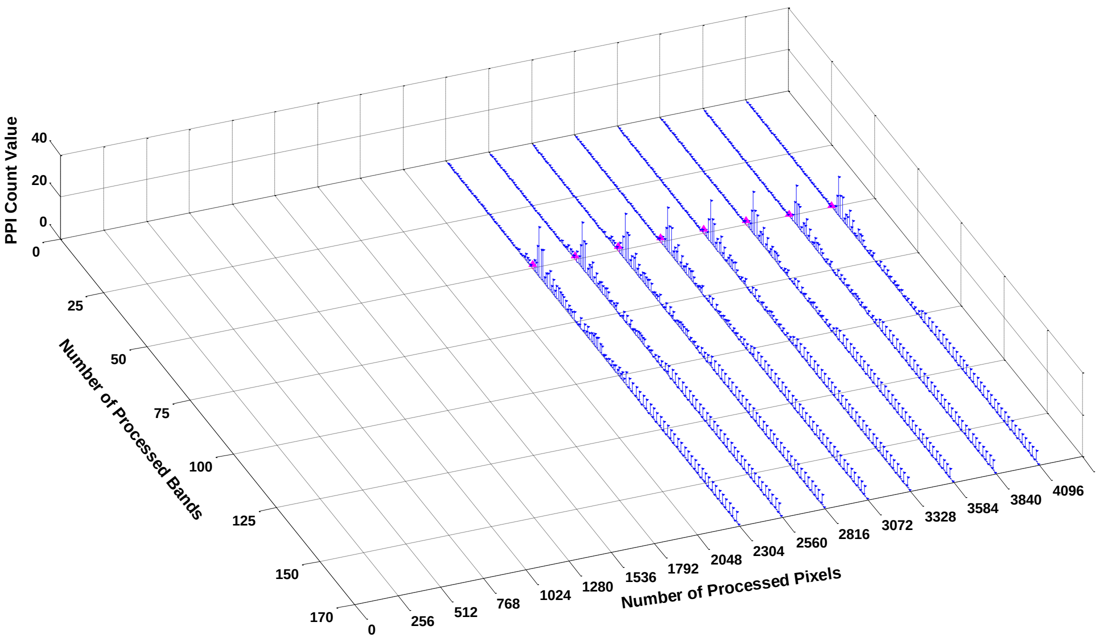
As displayed in the table, in order to extract all the R-panel pixels on the 1st column, a minimum of 700 skewers is required. The less skewers generated in the process could reduce the computational complexity.



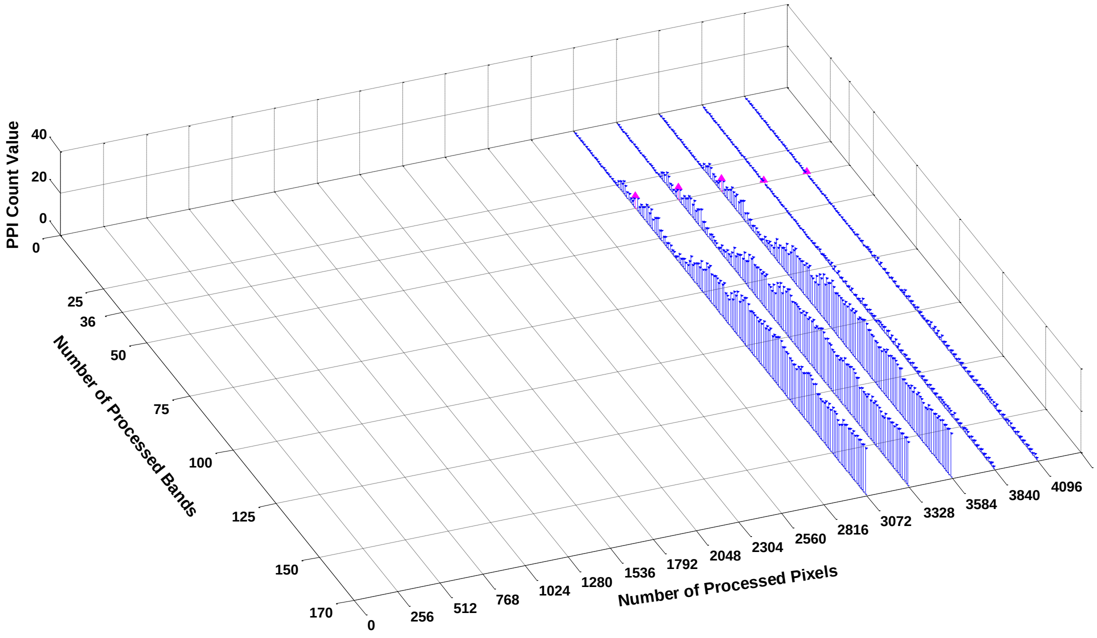
(a)


PPI Count Value
40
20
0
0
25
28
50
75
100
125
150
170
Number of Processed Bands
0
256
512
768
1024
1280
1536
1792
2048
2304
2560
2816
3072
3328
3584
3840
4096
Number of Processed Pixels

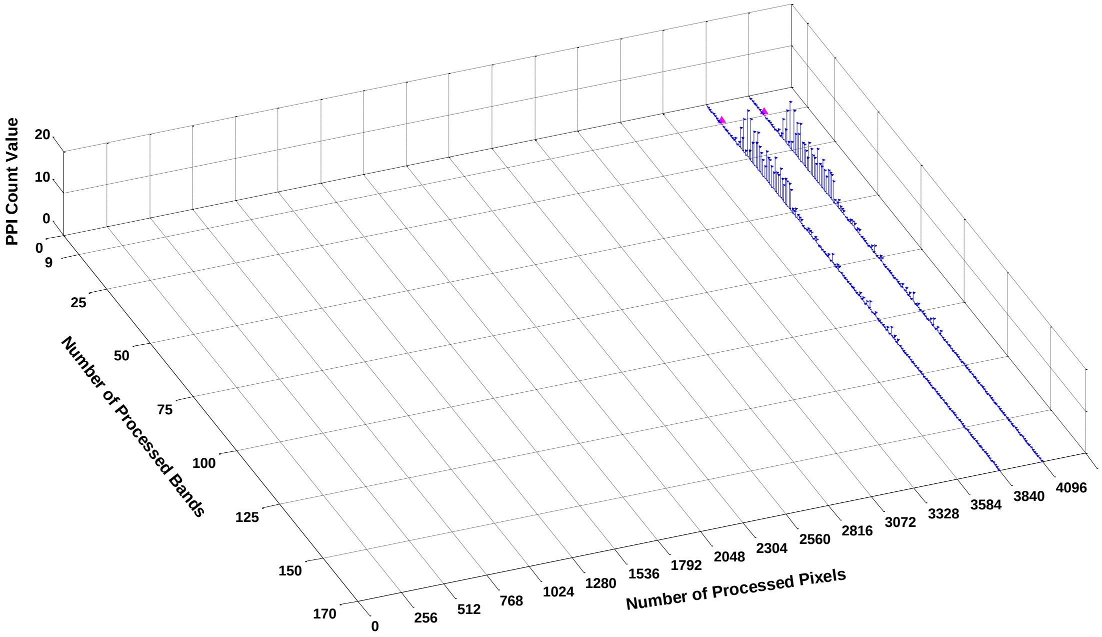
(b)


PPI Count Value
40
20
0
0
25
50
75
100
125
150
170
Number of Processed Bands
0
256
512
768
1024
1280
1536
1792
2048
2304
2560
2816
3072
3328
3584
3840
4096
Number of Processed Pixels

(c)



(d)

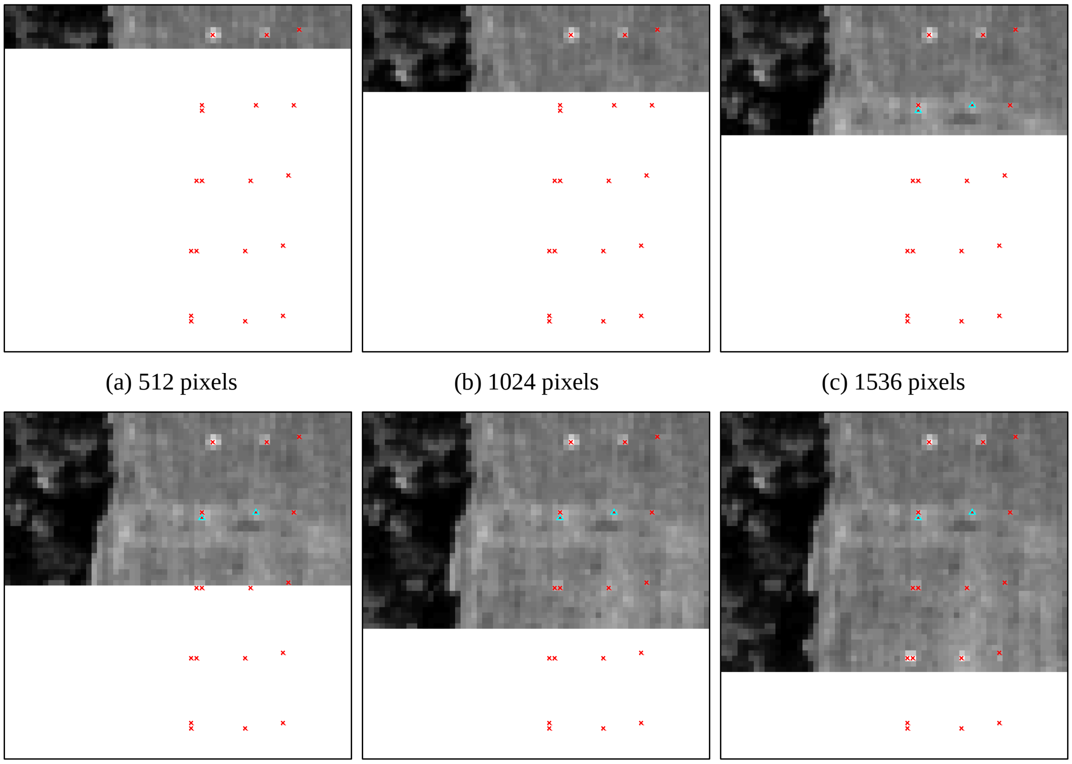


(e)

Figure 4.19. PPI Count variations versus the number of processed bands and the number of processed pixels for each R-panel pixel (a) p_{11} (b) p_{211} (c) p_{311} (d) p_{411} (e) p_{511}

RBP-C-PPI is further proposed to address the issue that partial pixels are available in the data collection process. Fig. 4.19 plots the changes of PPI Count with different number of processed bands as the number of processed pixels increases from 256 to 4096 with step size 256 for the 5 R-panel pixels on the 1st column in HYDICE scene, where x-axis denotes the number of bands collected and processed, y-axis displays the number of pixels included in the process, and z-axis presents the value of PPI Count correspondingly. As shown in the Fig. 4.19, the height of each bar represents the value of PPI Count for a particular combination of the number of processed bands and the number of processed bands. The magenta upward arrow indicates that one material begins to be identified as endmember candidates in the band that it first identified as an endmember candidate in RBP-PPI. For instance, R-panel pixel p_{411} is first identified as endmember candidates with 36 bands available in RBP-PPI with 1000 skewers. In the RBP-C-PPI process, it can be identified as endmember candidates with 3,072 pixels collected.

Moreover, Fig. 4.20 shows a RBP-C-PPI process with the number of processed pixels range from 512 to 4096 with step size 512 in band 5. As displayed in the figure, the red cross indicates the ground truth location of the HYDICE scene, the cyan upper triangle highlights the ground truth pixels that been identified as endmember candidates. The white area in the scene indicates the pixels in this area are not available yet. The beauty of RBP-C-PPI process is to extract the moving targets in the scene. For instance, R-panel pixel p_{22} on the 2nd column can be identified as endmember candidates with 1536 to 3072 pixels available, and it vanished after that. RBP-P-PPI catches this characteristic properly.



(a) 512 pixels

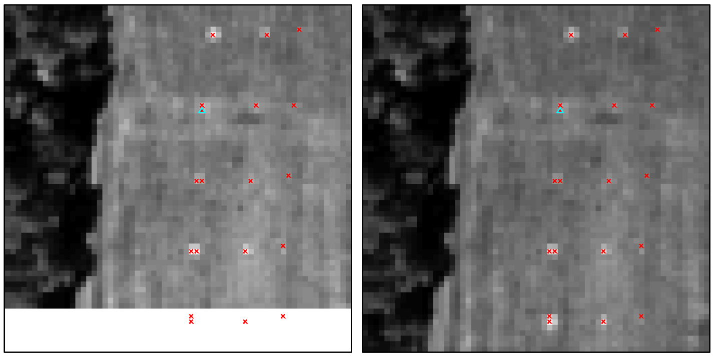
(b) 1024 pixels

(c) 1536 pixels

(d) 2048 pixels

(e) 2560 pixels

(f) 3072 pixels



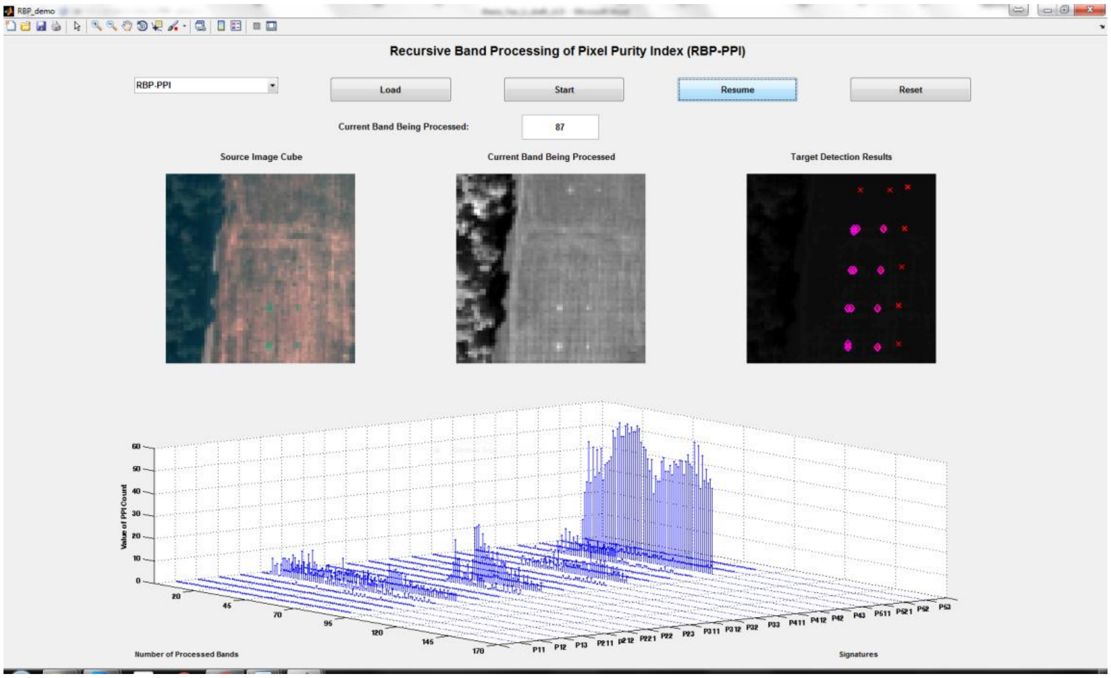
(g) 3584 pixels

(h) 4096 pixels (full pixels)

Figure 4.20. An example of RBP-C-PPI process with different number of pixels processed in band 5. (a) 512 pixels (b) 1024 pixels (c) 1536 pixels (d) 2048 pixels (e) 2560 pixels (f) 3072 pixels (g) 3584 pixels (h) 4096 pixels (full pixels)

4.5 GUI Design

A graphical user interface (GUI) with a screenshot shown in Fig. 4.21 was developed using Matlab's GUIDE to aid in algorithm performance analysis. The three images displayed on top of the window show a color image of the scene, a gray scale image of the current band being processed, and the result after the current band is processed. At the bottom a window shows the variations of the PPI counts as more bands being collected for the different signatures. Once the data is loaded, the user can start the process by clicking the start button. At each iteration each band is received and being processed. Upon completion of processing, the current band and the resulting image are updated to allow the user to observe the results. A final note on Fig. 4.21 is worthwhile. The plots shown at the bottom window is a real-time progressive version of Fig. 4.1/Fig. 4.7/Fig. 4.13. Unfortunately, this nice feature can be only demonstrated in real time process.



Current Band Being Processed: 87

Figure 4.21. GUI Design for RBP-PPI

4.6 *Conclusions*

This chapter derives a new version of PPI in recursive-band-processing (RBP), referred to as RBP-PPI, other than endmember finding as commonly being used in the literature. The use of RBP-PPI enables us to find and dictate spectral variation from band to band so that missing details in spectral characterization of data samples can be captured for data analysis. This is particularly useful and crucial when it comes to find weak targets with only small changes in certain specific bands with lower PPI counts which can be overwhelmed or dominated by subsequent strong targets found by PPI with higher PPI counts. This is mainly because PPI using full bands can only show the final PPI counts of data samples but cannot provide any information of changes in PPI counts of data samples from band to band. In addition, the potential of RBP-PPI in finding significant bands in changes in PPI counts is also demonstrated by experiments. To further study RBP-PPI we can extend RBP-PPI to real time implementation for future hardware design.

Chapter 5: Recursive Band Processing of Fast Iterative Pixel Purity Index (RBP-FIPPI)

5.1 Introduction

PPI is widely used for endmember finding (Chang, 2013). However, there are three major issues in implementing PPI. One is "how many skewers are needed for PPI to work effectively?" Of course, in order to cover as many directions of interestingness as possible, we would like to have the number of skewers *K* sufficiently large. But what value of *K* is considered to be large enough? So far, there is no guideline available for determining the value of *K*. Another is "what value of the threshold should be selected as a cut-off value for PPI counts to extract endmembers?" Theoretically speaking, the higher the PPI count of a data sample, the more likely the data sample to be an endmember. Unfortunately, according to our extensive experiments, this may not be always true. However, it is generally true that an endmember must have its PPI count at least greater than 0. A third issue is inconsistency caused by skewers that are randomly generated. In other words, the PPI counts obtained for data sample vectors are not reproducible because such PPI counts can vary quite differently if different sets of skewers are used. More precisely, PPI counts produced by running PPI at different times or by different users at the same time are generally not the same due to randomly generated different sets of skewers.

Under such circumstances, Fast Iterative Pixel Purity Index (FIPPI) is proposed in (Chang and Plaza, 2006). It has several significant advantages over the PPI. First of all, it makes use of a recently developed concept, virtual dimensionality (VD), in (Chang, 2003) to estimate the number of endmembers required to be generated. The VD allows us to replace the two parameters and used in the PPI so that the algorithm's sensitivity to these parameters can be resolved. Second, the FIPPI takes advantage of the automatic target generation process (ATGP) in (Chang, 2003) to generate an appropriate set of initial endmembers that can thus speed up the algorithm considerably. Third, the FIPPI is an iterative algorithm that converges very rapidly with tremendous savings in computation time. Most importantly, the PPI requires a visualization tool to manually select a final set of endmembers. Such a problem is avoided by the FIPPI because the FIPPI is automatic and the final set of FIPPI-generated endmembers is always the same, regardless of who is a user of the FIPPI. This is considered to be one of the most significant advantages of the FIPPI over the PPI.

This chapter introduces FIPPI in an innovative manner, which implements FIPPI band by band from a specific data acquisition point of view. The idea of RBP-FIPPI is derived from data acquisition according to band sequential (BSQ) format which is different from another data acquisition format, band-interleaved-pixel (BIP) or band-interleaved-line (BIL) implemented data sample by sample or line by line respectively. The difference between these two can be explained by how an image is processed progressively or sequentially. A progressive image processing processes each pixel of an 8-bit image bit by bit. As a result, each image pixel is revisited and processed 8 times. One such example is bit plane coding for image enhancement and compression. On the other hand, Pulse Code Modulation (PCM) implements an 8-bit quantizer to encode an image using 256 gray level values which can fully process each of its pixels sequentially pixel by pixel with no need of revisiting pixels again. Downloading images from website make use of PCM. In context of progressive and sequential image processing described above the PPI using the BSQ format can be considered as progressive hyperspectral imaging technique while the PPI using the BIP format can be viewed as sequential hyperspectral imaging technique. With this interpretation the total number of bands *L*, used to acquire a hyperspectral image can be interpreted as the total number of bits used to encode an image. The spectral resolution is progressively and gradually improved by adding more bands which is similar to spatial resolution is progressively and gradually improved by adding more bits.

Several advantages can be benefited from RBP-FIPPI. First, progressive changes in PPI counts of data samples as more bands are included for data processing can be resulted from RBP-FIPPI. This allows image analysts to keep track of significance of data samples. Second, is to help find crucial bands for data processing according to progressive changes in PPI counts. Third, hyperspectral data collection and process can be done simultaneously without waiting the full bands information are available.

5.2 Detailed RBP-FIPPI Algorithm

Recursive-Band-Processing of FIPPI (RBP-FIPPI) Algorithm

```
Initial Conditions: Find the VD using the the Harsanyi-Farrand-Chang (HFC) method and let it be $p$.
Outer loop from $l = p$ to $L$
    Inner loop
```

1. Initial Condition: Let $\left\{\mathbf{skewer}_{l,j}^{(0)}\right\}_{j=1}^{p}$ be an initial set of p skewers generated by selecting those pixels that correspond to target pixels generated by ATGP.

2. Normalize $\left\{\mathbf{skewer}_{l,j}^{(0)}\right\}_{j=1}^{p}$ into a unit vector.

3. Iterative rule: At iteration *k*, for each $\mathbf{skewer}_{l,j}^{(k)}$, all the sample vectors are projected onto this particular $\mathbf{skewer}_{l,j}^{(k)}$ to find those which are at its extreme positions to form an extrema set, denoted by $S_{extrema}(\mathbf{skewer}_{l,j}^{(k)})$. Find the sample vectors that produce the largest $N_{PPI}(\mathbf{r}_{l,j}^{(k)})$ and let them be denoted by $\left\{\mathbf{r}_{l,j}^{(k)}\right\}$.

4. Form the joint set $\left\{\mathbf{skewer}_{l,j}^{(k+1)}\right\} = \left\{\mathbf{r}_{l,j}^{(k)}\right\}_{N_{PPI}(\mathbf{r}_{l,j}^{(k)})>0} \cup \left\{\mathbf{skewer}_{l,j}^{(k)}\right\}$.

5. If $\left\{\mathbf{skewer}_{l,j}^{(k+1)}\right\} = \left\{\mathbf{skewer}_{l,j}^{(k)}\right\}$, then no new endmembers are added to the skewer set. In this case, the algorithm is terminated, break the inner loop. Otherwise, let $k \leftarrow k+1$, and go to step 2.

End Inner loop

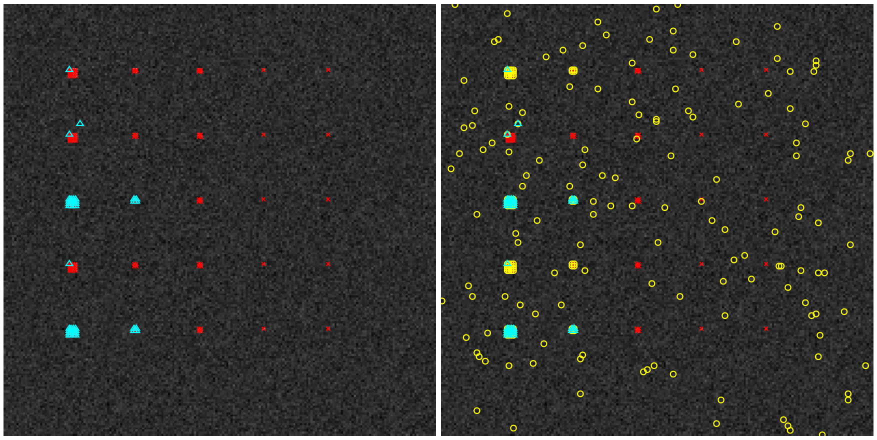
End Outer loop

5.3 Experiments

5.3.1 Synthetic Data

A. TI Data Set

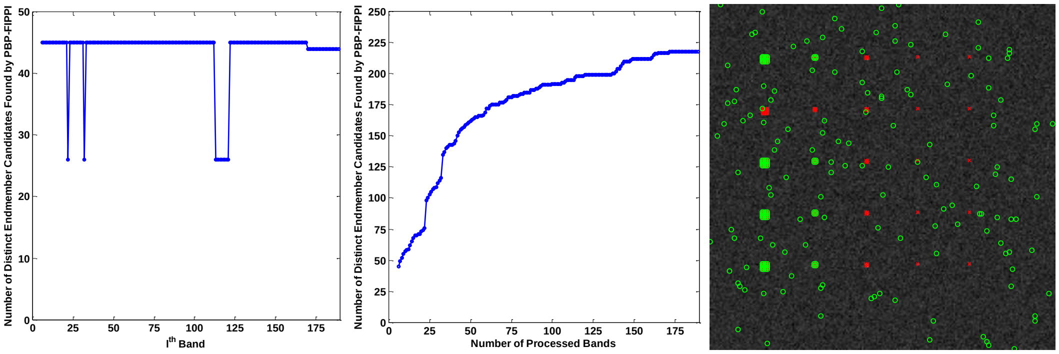
Fig. 5.1 conducts a comparative study between FIPPI and PBP-FIPPI. Fig. 5.1(a) shows the results of original FIPPI applying on Synthetic TI scene with VD = 6, Fig. 5.1(b) displays the spatial location of PBP-FIPPI results with the same value of VD. As shown in Fig. 5.1, the red cross indicates the spatial location of the ground truth pixels, the yellow circle denotes the endmember candidates found in the previous bands, the cyan upper triangle highlights the endmember candidates in the current band. PBP-FIPPI acquires the same results as FIPPI in the last iteration, which verified the correction of the algorithm. FIPPI is a one-shot process, while PBP-FIPPI is not, and some moving targets can be discovered during the process, the pixels circled by yellow color presents the additional endmember candidates picked up by PBP-FIPPI. Furthermore, a total of 81 ground truth pixels can be discovered by PBP-FIPPI comparing with only 43 ground truth pixels picked up by the original FIPPI.



(a) (b)

Figure 5.1. Comparative study between FIPPI and PBP-FIPPI applying on synthetic TI data set. (a) FIPPI experimental results with VD = 6 (b) PBP-FIPPI experimental results with VD = 6

Furthermore, Fig. 5.2(a) plots the number of skewers identified in each band, Fig. 5.2(b) displays the total number of distinct skewers as the number of processed bands is increasing, Fig. 5.2(c) plots the spatial location of the endmembers found by PBP-FIPPI in TI scene. As shown in Fig. 5.2(b), there are a total of 218 distinct skewers are found as endmembers. After 172 bands processed, no additional endmember candidates are found. In Fig. 5.2(c), the ground truth pixels are marked as red crosses, the 218 endmembers are highlighted by green circle.



(a) (b) (c)

Figure 5.2. Result of PBP-FIPPI Algorithm. (a) The number of skewers found by PBP-FIPPI in each band (b) The number of distinct skewers identified by PBP-FIPPI versus the number of processed bands (c) Spatial location of the PBP-FIPPI results on TI scene

As mentioned in the previous sections, there are 81 ground truth pixels are identified as endmember candidates by PBP-FIPPI. Fig. 5.3 displays the order of the ground truth pixels been discovered versus the number of bands processed. It is clear shown in the graph that, material A, C and K can be discovered with 6 bands collected and processed. After 28 bands collected and processed, all the 5 materials have been found as endmember candidates.

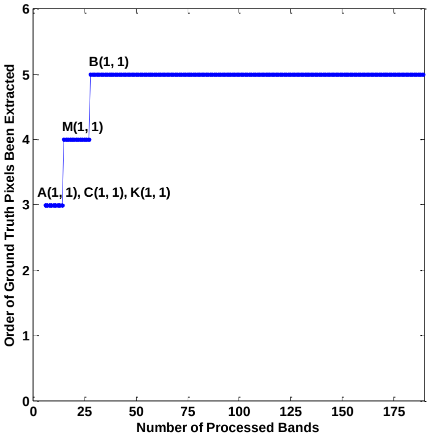
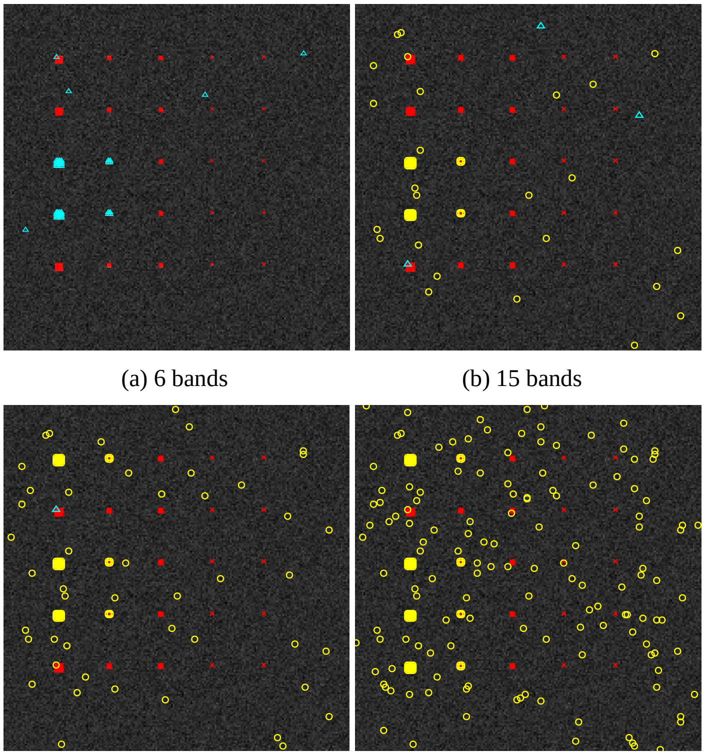


Figure 5.3. Order of the ground truth pixels been discovered versus the number of bands processed

Fig. 5.4 further shows the spatial location of the endmember candidates identified by PBP-FIPPI with different number of bands processed starting from 6 bands. In these figures, the red cross indicates the spatial location of the ground truth pixels, the yellow circle denotes the endmember candidates found in the previous bands, the cyan upper triangle highlights the endmember candidates in the current band. With 6 bands processed, materials A, C, and K can be discovered, followed by material M with 15 bands collected and processed, material B is the last material been identified with 28 band received. After 28 bands, all the five materials in TI scene are detected.



(a) 6 bands

(b) 15 bands

(c) 28 bands

(d) 189 bands (full bands)

Figure 5.4. Endmember candidates identified by PBP-FIPPI with different number of bands processed (a) 6 bands (b) 15 bands (c) 28 bands (d) 169 bands (full bands)

In addition, Fig. 5.5 demonstrates a 3 dimensional histogram to show how frequently a particular pixel is picked up by PBP-FIPPI as an endmember. The x-axis and y-axis correspond to the columns and rows of the TI scene. The z-axis indicates the number of times a given pixel was detected as an endmember by PBP-FIPPI after processing all bands. So, for TI scene with 189 bands, the maximum value for z-axis is 189. As shown in the figure, the magenta arrows indicate the ground truth pixels.

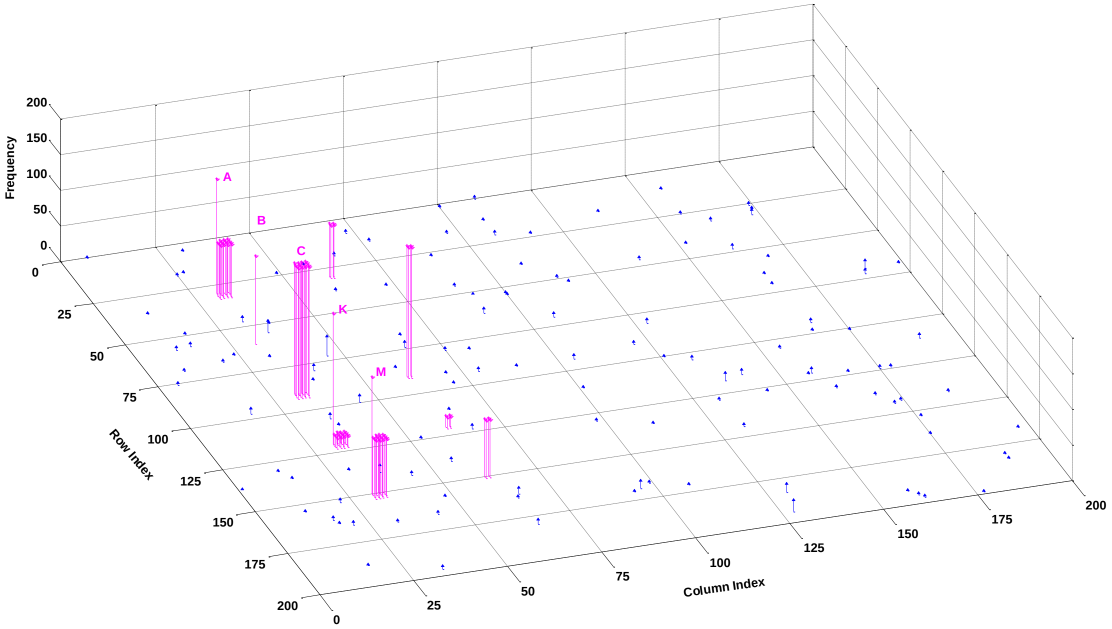
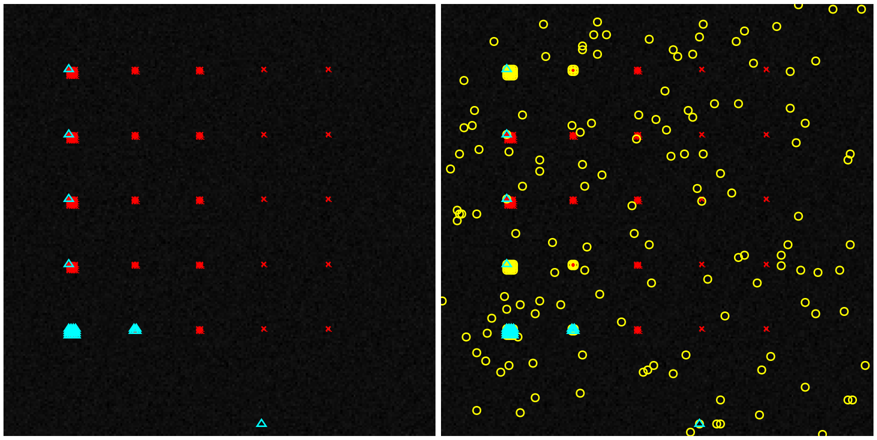


Figure 5.5. 3D Histogram of the pixels picked up by PBP-FIPPI as targets

B. TE Experiments

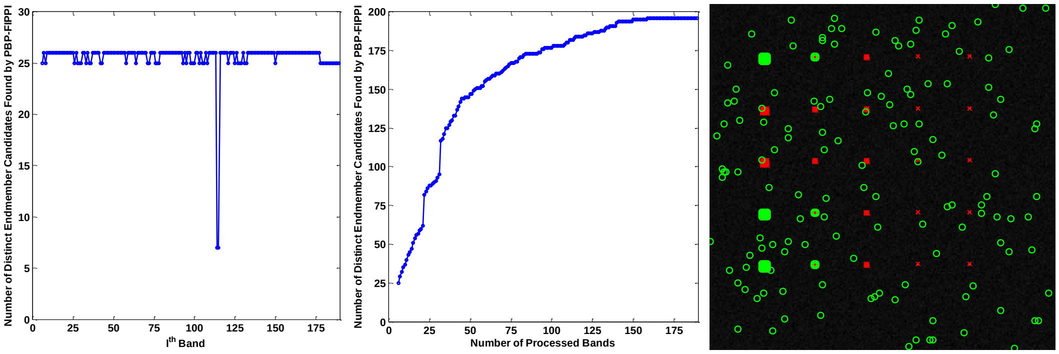
Fig. 5.6 conducts a comparative study between FIPPI and PBP-FIPPI. Fig. 5.6(a) shows the results of original FIPPI applying on Synthetic TE scene with VD = 6, Fig. 5.6(b) displays the spatial location of PBP-FIPPI results with the same value of VD. As shown in Fig. 5.6, the red cross indicates the spatial location of the ground truth pixels, the yellow circle denotes the endmember candidates found in the previous bands, the cyan upper triangle highlights the endmember candidates in the current band. PBP-FIPPI acquires the same results as FIPPI in the last iteration, which verified the correction of the algorithm. FIPPI is a one-shot process, while PBP-FIPPI is not, and some moving targets can be discovered during the process, the pixels circled by yellow color presents the additional endmember candidates picked up by PBP-FIPPI. Furthermore, a total of 62 ground truth pixels can be discovered by PBP-FIPPI comparing with only 24 ground truth pixels picked up by the original FIPPI.



(a) (b)

Figure 5.6. Comparative study between FIPPI and PBP-FIPPI applying on synthetic TE data set. (a) FIPPI experimental results with VD = 6 (b) PBP-FIPPI experimental results with VD = 6

Furthermore, Fig. 5.7(a) plots the number of skewers identified in each band, Fig. 5.7(b) displays the total number of distinct skewers as the number of processed bands is increasing. As shown in Fig. 5.7(b) a total of 196 distinct skewers are found as endmember candidates, and Fig. 5.7(c) plots the spatial location of the 196 endmember candidates in TE scene. After 159 bands processed, no additional endmember candidates are found.



(a) (b) (c)

Figure 5.7. Result of PBP-FIPPI Algorithm. (a) The number of skewers found by PBP-FIPPI in each band (b) The number of distinct skewers identified by PBP-FIPPI versus the number of processed bands (c) Spatial location of the PBP-FIPPI results on TE scene

As mentioned in the previous sections, there are 62 ground truth pixels are identified as endmember candidates by PBP-FIPPI. Fig. 5.8 displays the order of the ground truth pixels been discovered versus the number of bands processed. It is clear shown in the graph that, material A, and K can be discovered with 6 bands collected and processed. After 141 bands collected and processed, all the 5 materials have been found as endmember candidates.

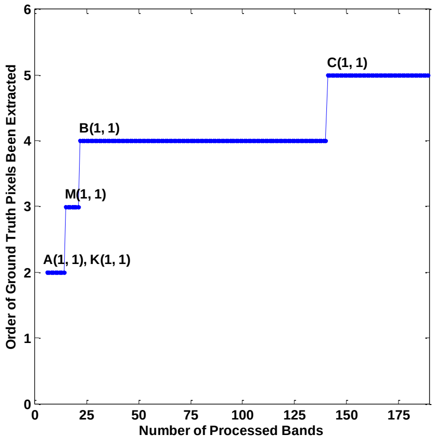


Figure 5.8. Order of the ground truth pixels been discovered versus the number of bands processed

Figure 5.9 further shows the spatial location of the endmember candidates identified by PBP-FIPPI with different number of bands processed starting from 6 bands. In these figures, the red cross indicates the spatial location of the ground truth pixels, the yellow circle denotes the endmember candidates found in the previous bands, the cyan upper triangle highlights the endmember candidates in the current band. With 6 bands collected, material A and K are discovered. Followed by material M with 15 bands collected, material B with 22 bands received. Material C is the last material been identified with 141 bands processed.

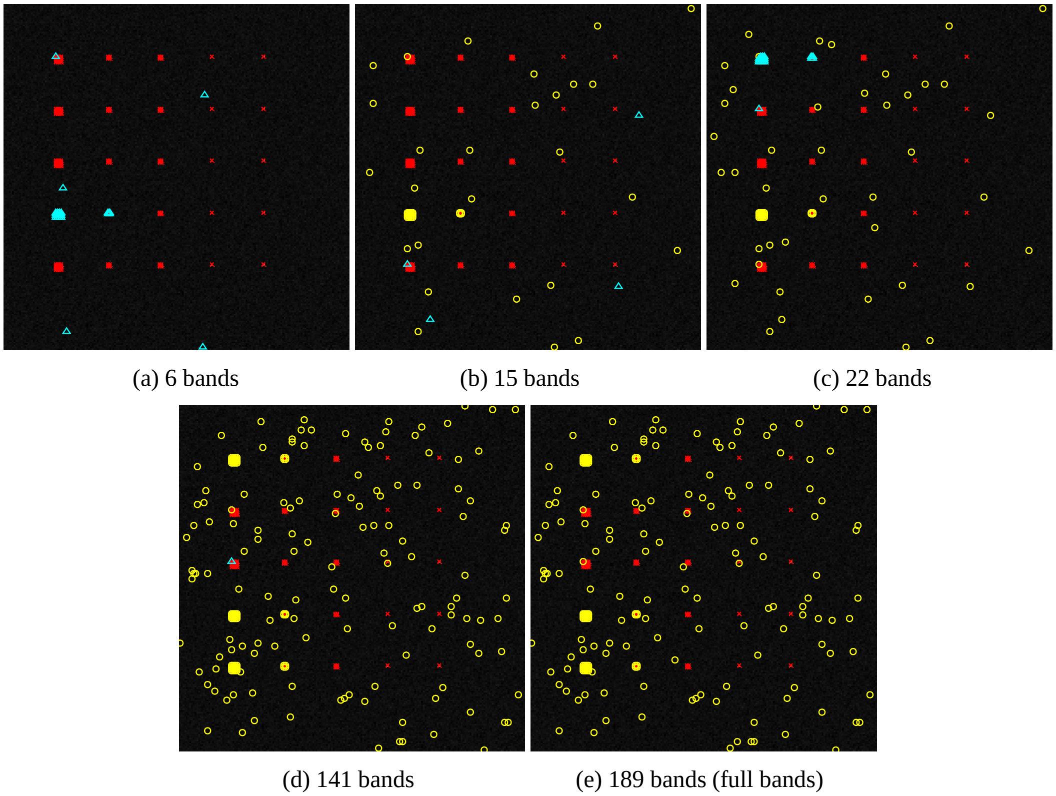

(a) 6 bands
(b) 15 bands
(c) 22 bands
(d) 141 bands
(e) 189 bands (full bands)

Figure 5.9. Endmember candidates identified by PBP-FIPPI with different number of bands processed (a) 6 bands (b) 15 bands (c) 28 bands (d) 141 bands (e) 169 bands (full bands)

In addition, Fig. 5.10 demonstrates a 3 dimensional histogram to show how frequently a particular pixel is picked up by PBP-FIPPI as an endmember. The x-axis and y-axis correspond to the columns and rows of the TE scene. The z-axis indicates the number of times a given pixel was detected as an endmember by PBP-FIPPI after processing all bands. So, for TE scene with 189 bands, the maximum value for z-axis is 189. As shown in the figure, the magenta arrows indicate the ground truth pixels.

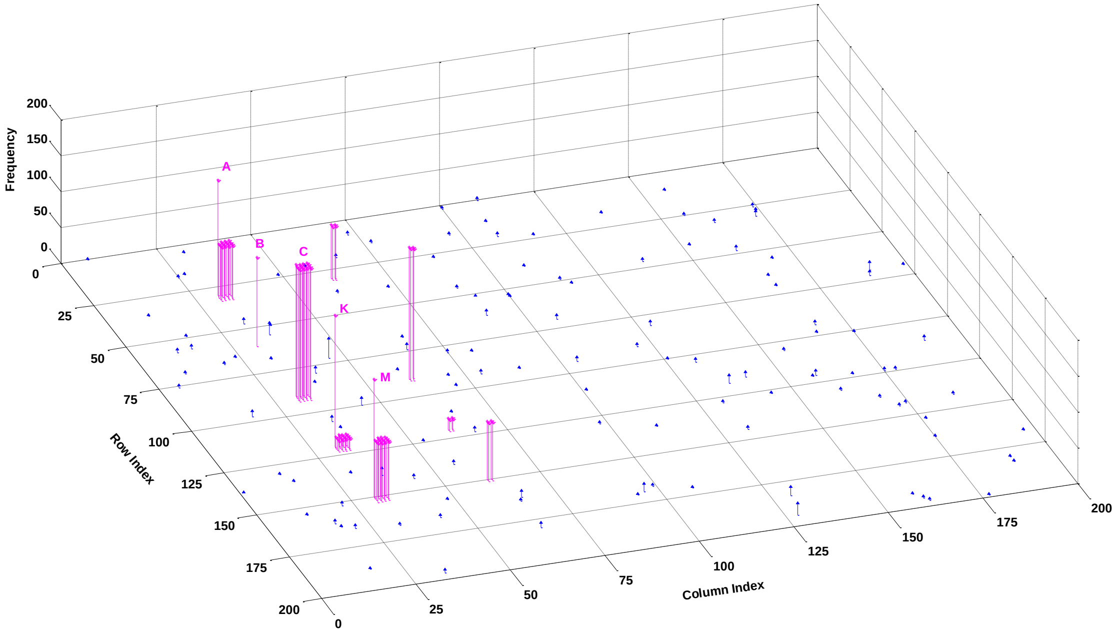
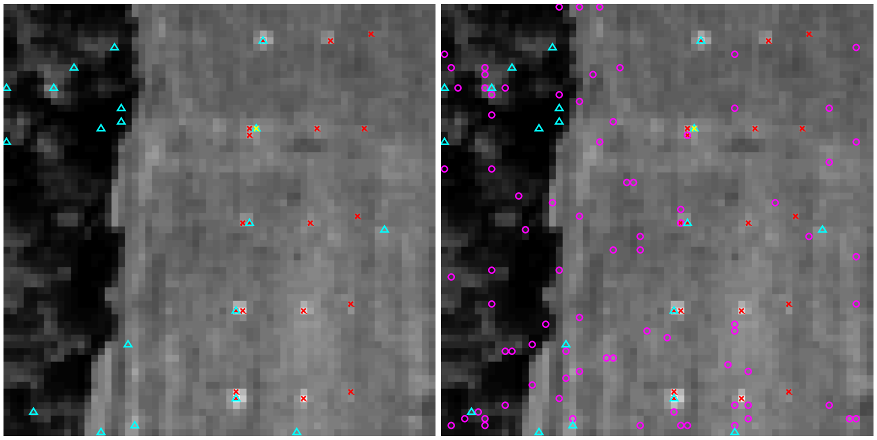


Figure 5.10. 3D Histogram of the pixels picked up by PBP-FIPPI as targets (TE Scene)

5.3.2 Real Data (HYDICE Scene)

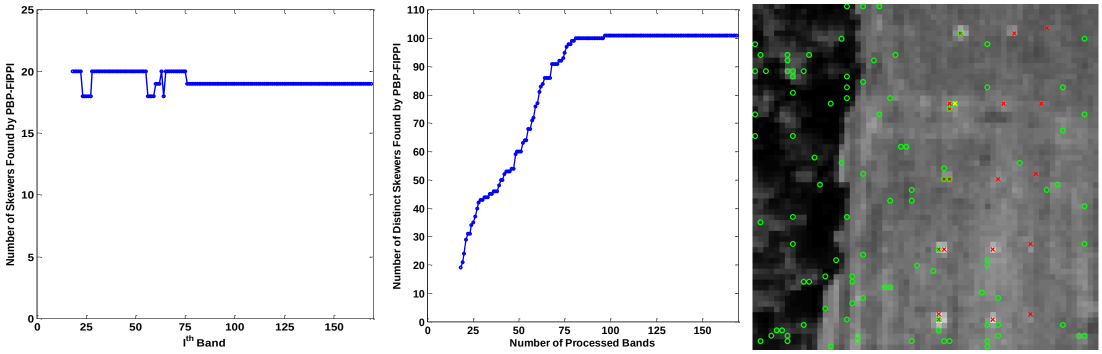
Fig. 5.11 conducts a comparative study between FIPPI and PBP-FIPPI. Fig. 5.11(a) shows the results of original FIPPI applying on HYDICE scene with VD = 18, Fig. 5.11(b) displays the spatial location of PBP-FIPPI results with the same value of VD. As shown in Fig. 5.11, the red cross indicates the spatial location of the R-panel pixels and the yellow cross indicates the positon of the Y-panel pixel, the magenta circle denotes the endmember candidates found in the previous bands, the cyan upper triangle highlights the endmember candidates in the current band. PBP-FIPPI acquires the same results as FIPPI in the last iteration, which verified the correction of the algorithm. 6 R-panel pixels and 1 Y-panel pixel are identified by PBP-FIPPI, comparing with the original FIPPI, additional 2 R-panel pixels are found.



(a) (b)

Figure 5.11. Comparative study between FIPPI and PBP-FIPPI. (a) FIPPI experimental results with VD = 18 (b) PBP-FIPPI experimental results with VD = 18

Furthermore, Fig. 5.12(a) plots the number of skewers identified in each band, Fig. 5.12(b) displays the total number of distinct skewers as the number of processed bands is increasing. As shown in Fig. 5.12(b) a total of 101 distinct skewers are found as endmember candidates, and Fig. 5.12(c) plots the spatial location of the 101 endmember candidates in HYDICE scene. After 97 bands processed, no additional endmember candidates are found.



(a) (b) (c)

Figure 5.12. Result of PBP-FIPPI Algorithm. (a) The number of skewers found by PBP-FIPPI in each band (b) The number of distinct skewers identified by PBP-FIPPI versus the number of processed bands (c) Spatial location of the PBP-FIPPI results on HYDICE scene

As mentioned in the previous sections, 6 R-panel pixels and 1 Y-panel pixel are identified as endmember candidates by PBP-FIPPI. Fig. 5.13 displays the order of the panel pixels been discovered versus the number of bands processed. It is clear shown in the graph that, p_{521} is the first panel pixel been discovered with only 18 bands collected and processed. After 97 bands collected and processed, all the 7 panel pixels have been found as endmember candidates.

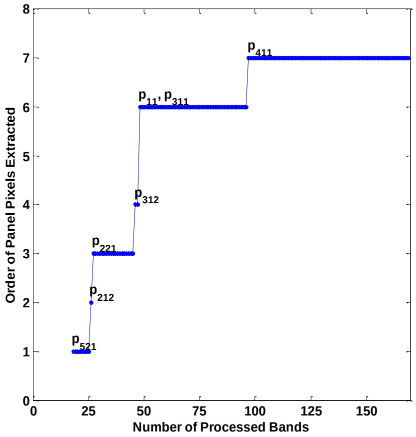
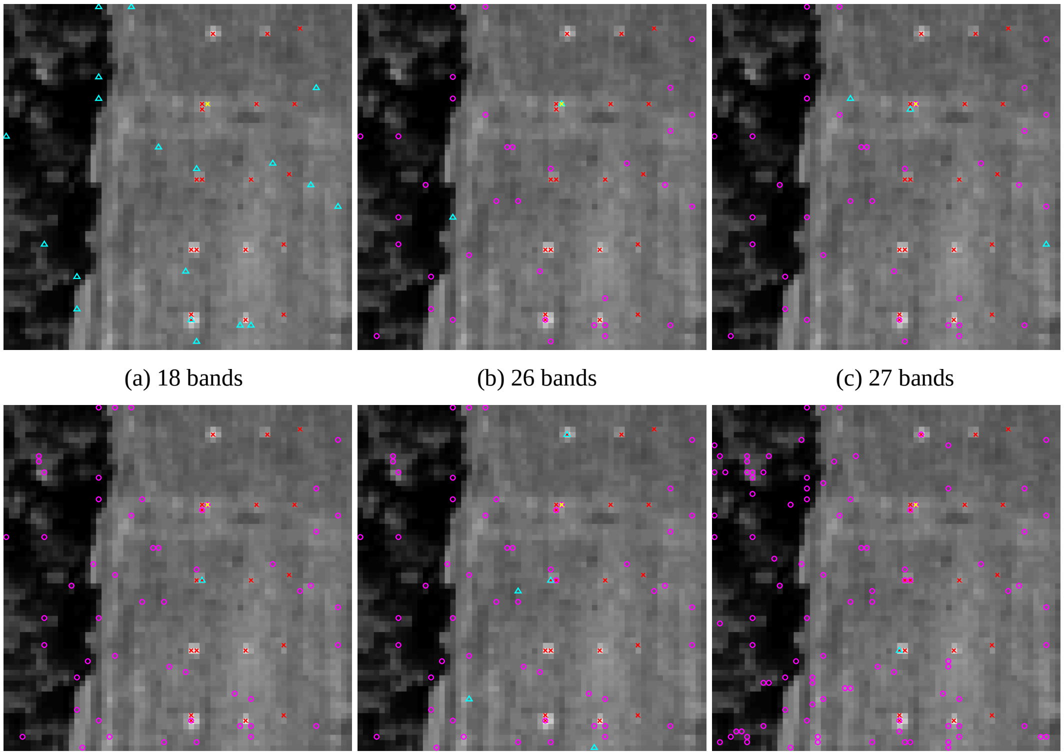


Figure 5.13. Order of the panel pixels been discovered versus the number of bands processed

Figure 5.14 further shows the spatial location of the endmember candidates identified by PBP-FIPPI with different number of bands processed starting from 18 bands. In these figures, the red cross indicates the spatial location of the R-panel pixels and the yellow cross indicates the positon of the Y-panel pixel, the magenta circle denotes the endmember candidates found in the previous bands, the cyan upper triangle highlights the endmember candidates in the current band. With 18 bands processed, panel pixel p_{521} is detected, followed by p_{212} with 26 bands processed. After 97 bands, no additional panel pixels are detected.



(a) 18 bands

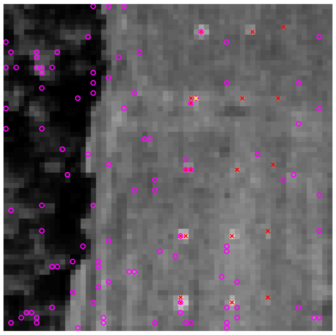
(b) 26 bands

(c) 27 bands

(d) 46 bands

(e) 48 bands

(f) 97 bands



(g) 169 bands (full bands)

Figure 5.14. Endmember candidates identified by PBP-FIPPI with different number of bands processed (a) 18 bands (b) 26 bands (c) 27 bands (d) 46 bands (e) 48 bands (f) 97 bands (g) 169 bands (full bands)

Moreover, Fig. 5.15 demonstrates a 3 dimensional histogram to show how frequently a particular pixel is picked up by PBP-FIPPI as an endmember. The x-axis and y-axis correspond to the columns and rows of the HYDICE scene. The z-axis indicates the number of times a given pixel was detected as an endmember by PBP-FIPPI after processing all bands. So, for HYDICE scene with 169 bands, the maximum value for z-axis is 169. As shown in the figure, the magenta arrows indicate the panel pixels.

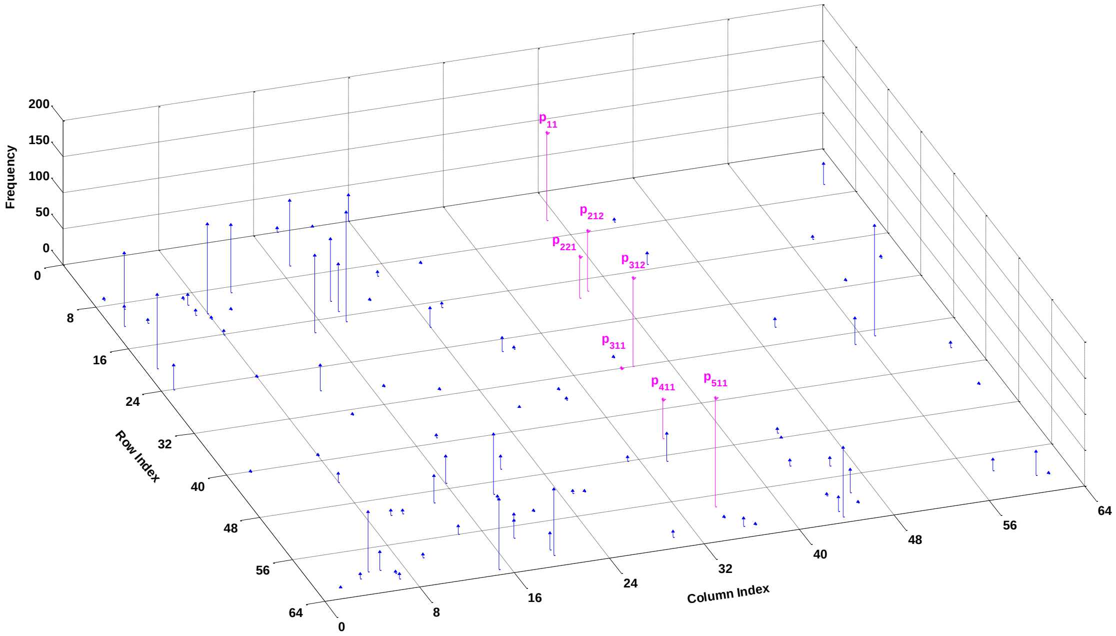
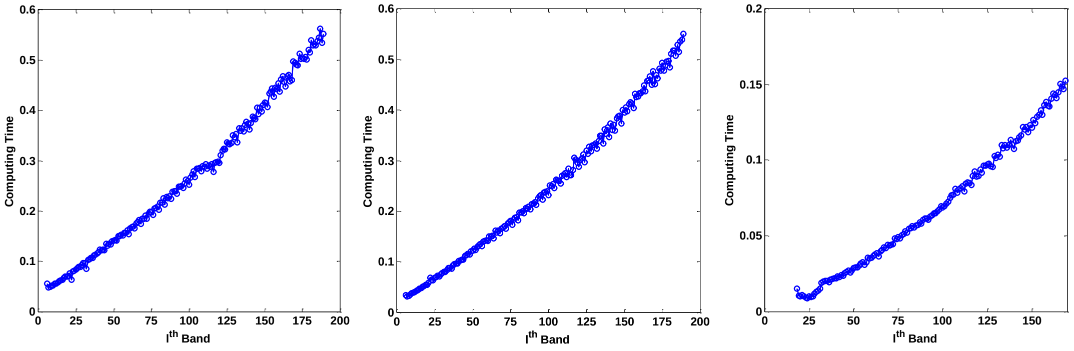


Figure 5.15. 3D Histogram of the pixels picked up by PBP-FIPPI as targets (HYDICE Scene)

5.4 Computational Complexity

The computing time of PBP-FIPPI was compared to the original FIPPI. Each algorithm was run and executed in MATLAB R2012B with an Intel Core i7 – 3770 running at 3.40

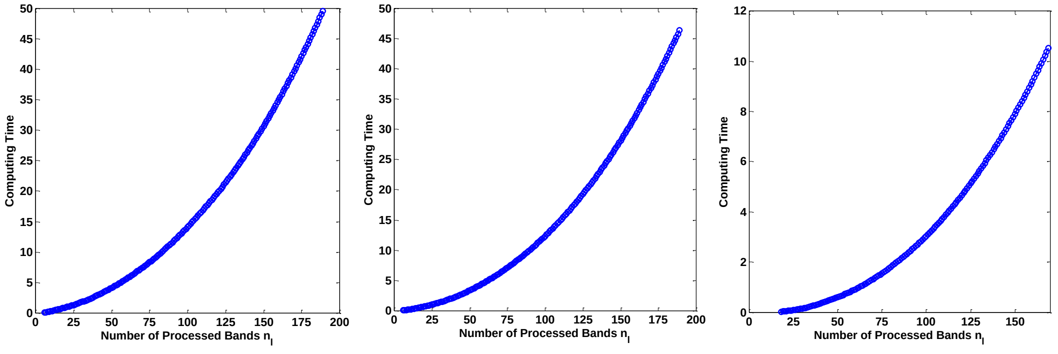
GHz with 16GB of RAM on three data sets ten times to produce an average computing time. Fig. 5.16(a-c) plots computing time required for PBP-FIPPI applying on TI, TE and HYDICE data resepectively where the *y*-axis is computer processing time in seconds and *x*-axis is the l^{th} band as new band coming in. As can be seen from these figures the processing time required for PBP-FIPPI to run each new individaul band is nearly linear. The computing time increasing as more bands are collected and received. It is also worth noting that the fluctuations of plots were resulting from numerical computations by computer implementation.



(a) TI (b) TE (c) HYDICE

Figure 5.16. Computing time versus the l^{th} band by PBP-FIPPI (a) TI scene (b) TE scene (c) HYDICE scene

For a fair comparison Fig. 5.17(a-c) further plots computing time versus the number of processed bands, n_l, (i.e., the number of the first *l* bands to be used for processing) required for PBP-FIPPI for TI, TE and HYDICE data respectively. Table 5.1 further compared the computing time between FIPPI and PBP-FIPPI in detail.



(a) TI (b) TE (c) HYDICE

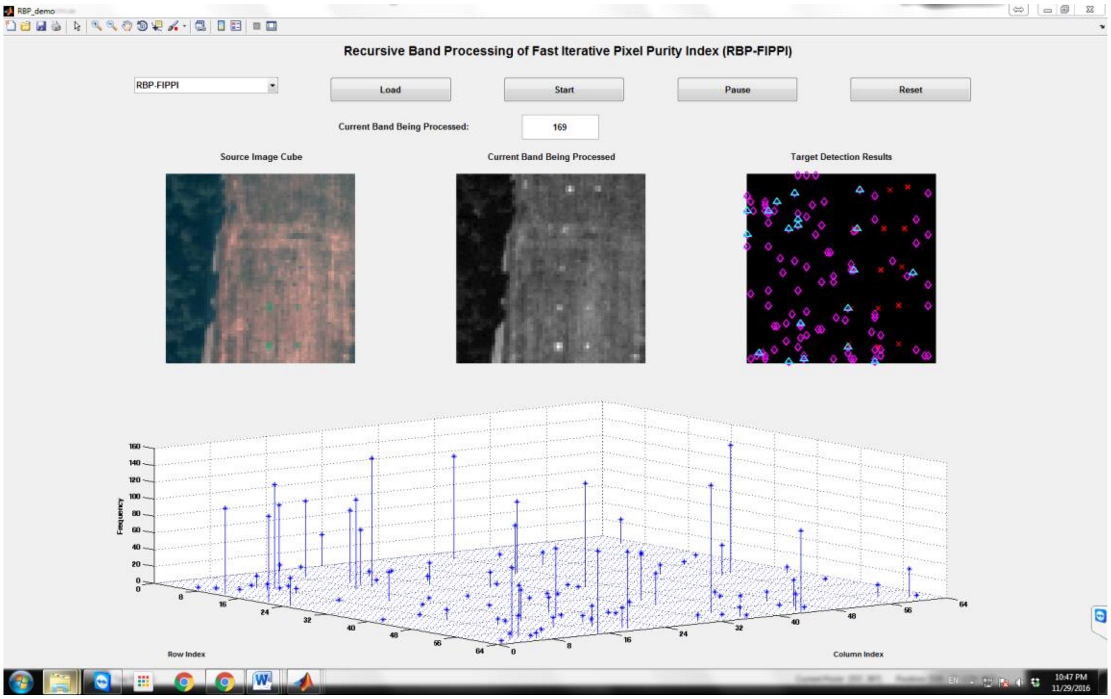
Figure 5.17. Computing time versus n_l by PBP-FIPPI (a) TI scene (b) TE scene (c) HYDICE scene

Table 5.1. Comparison of computing time in seconds required by PBP-FIPPI

	TI (sec)	TE (sec)	HYDICE (sec)
FIPPI	0.6372	0.6061	0.1756
PBP-FIPPI	0.5523	0.5514	0.1524

5.5 GUI Design

A graphical user interface (GUI) with a screenshot shown in Fig. 5.18 was developed using Matlab's GUIDE to aid in algorithm performance analysis. The three images displayed on top of the window show a color image of the scene, a gray scale image of the current band being processed, and the result after the current band is processed. At the bottom a window shows the frequency of a particular pixel is discovered as endmember in a 3D view. Once the data is loaded, the user can start the process by clicking the start button. At each iteration each band is received and being processed. Upon completion of processing, the current band and the resulting image are updated to allow the user to observe the results. A final note on Fig. 5.18 is worthwhile. The plots shown at the bottom window is a real-time progressive version of the 3D plots of the frequency for a particular pixel. Unfortunately, this nice feature can be only demonstrated in real time process.



Recursive Band Processing of Fast Iterative Pixel Purity Index (RBP-FIPPI)

RBP-FIPPI

Current Band Being Processed: 169

Figure 5.18. GUI design for PBP-FIPPI

5.6 Conclusion

Fast Iterative Pixel Purity Index (FIPPI) was previously developed to address two major issues arising in PPI which are the use of skewers whose number must be determined by *a priori* and inconsistent final results which cannot be re-produced. Recently, a new concept has been developed for hyperspectral data communication according to Band SeQuential (BSQ) acquisition format in such a way that bands can be collected band by band. By virtue of BSQ users are able to develop Progressive Band Processing (PBP) for hyperspectral imaging algorithms so that data analysts can observe progressive profiles of inter-band changes among bands. Its advantages have been justified in several applications, anomaly detection, constrained energy minimization, automatic target generation process, orthogonal subspace projection, PPI, etc. This chapter further extends RBP to FIPPI. The idea to implement RBP-FIPPI is to use two loops specified by skewers and bands to process FIPPI. Depending upon which one is implemented in the outer loop two different versions of PBP-FIPPI can be designed. When the outer loop is iterated band by band, it is called to be called Progressive Band Processing of FIPPI (PBP-FIPPI). When the outer loop is iterated by growing skewers, it is called Progressive Skewer Processing of FIPPI (PSP-FIPPI). Interestingly, both versions provide different insights into the design of FIPPI but produce close results.

Chapter 6: Recursive Band Processing of Simplex Growing Algorithm (RBP-SGA)

6.1 Introduction

Endmember finding is a fundamental task in hyperspectral data exploitation because endmembers can be used to specify spectral classes (Chang, 2003). Due to the fact that the number of endmembers, denoted by n_{E}, is relatively smaller than data dimensionality, dimensionality reduction (DR) is generally required. Idealistically, if an endmember can be accommodated by a specific spectral band, then it only needs n_{E} bands for finding all the n_{E} endmembers. This fact can be illustrates by simplex volume-based endmember finding algorithms (EFAs) such as N-finder algorithm (N-FINDR) developed by Winter (Chang, 2003) which finds a simplex of maximal volume with its vertices specified by all endmembers. As a result, an n_{E}-vertex simplex has only n_{E}-1 dimensions. This implies that there are no more than n_{E} dimensions required to accommodate an n_{E}-vertex simplex. The main issue is how to find n_{E} appropriate bands that can accommodate such an n_{E}-vertex simplex. Band selection (BS) can be used for this purpose. However, BS requires solving band optimization problems for all possible n_{E}-combinations among all the L spectral bands repeatedly, which is practically unrealistic. To mitigate this issue band prioritization is used to rank all bands according to priorities assigned to each of bands. Since a band with a higher priority its adjacent bands may also have higher priorities. To avoid selecting adjacent bands with too much overlapped spectral information band de-correlation (BD) is also needed. But a challenging issue is to select an adequate threshold. With all these issues this paper investigates a new concept of progressive band processing (PBP) to replace BS. Unlike BS PBP does not require the prior knowledge of n_{E} or select particular bands. Instead, it processes bands progressively band by band so that progressive changes in inter-band spectral variations of endmembers of interest can be dictated. Such progressive profiles cannot be offered by BS or any endmember finding algorithm (EFA) using full bands. With this interpretation a perfect candidate to be used for finding endmembers is simplex growing algorithm (SGA) developed by Chang et al. (Chang, 2013) which can be also considered a progressive EFA in (Chang, 2003) generating one endmember after another progressively by growing simplexes vertex by vertex where each vertex is specified by one endmember. By virtue of SGA we can develop a progressive band processing version of SGA, called PBP-SGA which implements SGA not only band by band but also endmember by endmember both progressively.

6.2 Derivations of RBP-SGA

Define

$$\tilde{\mathbf{U}}_{l\times p}=\mathbf{U}_{l\times p}-\mathbf{e}_{1,l\times 1} \tag{6.1}$$

$$\left[\tilde{\mathbf{U}}_{l\times p}^{T}\tilde{\mathbf{U}}_{l\times p}\right]^{-1}=\left[\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}\right]^{-1}-\frac{\left[\left(\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}\right)^{-1}\mathbf{e}(l)\right]\left[\mathbf{e}^{T}(l)\left(\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}\right)^{-1}\right]}{1+\mathbf{e}^{T}(l)\left(\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}\right)^{-1}\mathbf{e}(l)} \tag{6.2}$$

$$\begin{aligned}
\tilde{\mathbf{U}}_{l\times p}^{\#}&=\left[\tilde{\mathbf{U}}_{l\times p}^{T}\tilde{\mathbf{U}}_{l\times p}\right]^{-1}\tilde{\mathbf{U}}_{l\times p}^{T}\\
&=\left[\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}\right]^{-1}\left[\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\quad \mathbf{e}(l)\right]-\left\{\frac{\left[\left(\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}\right)^{-1}\mathbf{e}(l)\right]\left[\mathbf{e}^{T}(l)\left(\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}\right)^{-1}\right]}{1+\mathbf{e}^{T}(l)\left(\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}\right)^{-1}\mathbf{e}(l)}\right\}\left[\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\quad \mathbf{e}(l)\right]\\
&=\left[\left(\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}\right)^{-1}\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\quad \boldsymbol{\rho}_{l|(l-1)}\right]-\frac{1}{1+\mathbf{e}^{T}(l)\boldsymbol{\rho}_{l|(l-1)}}\left[\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\quad \boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\mathbf{e}(l)\right]
\end{aligned} \tag{6.3}$$

where $\boldsymbol{\rho}_{l|(l-1)}=\left[\left(\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}\right)^{-1}\mathbf{e}(l)\right]$ is a p-dimensional vector.

$$\begin{aligned}
P_{\tilde{\mathbf{U}}_{l\times p}}^{\perp}&=\mathbf{I}_{l\times l}-\tilde{\mathbf{U}}_{l\times p}\tilde{\mathbf{U}}_{l\times p}^{\#}\\
&=\mathbf{I}_{l\times l}-\begin{bmatrix}\tilde{\mathbf{U}}_{(l-1)\times p}\\ \mathbf{e}^{T}(l)\end{bmatrix}\left(\left[\left(\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}\right)^{-1}\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\quad \boldsymbol{\rho}_{l|(l-1)}\right]-\frac{1}{1+\mathbf{e}^{T}(l)\boldsymbol{\rho}_{l|(l-1)}}\left[\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\quad \boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\mathbf{e}(l)\right]\right)\\
&=\begin{bmatrix}\mathbf{I}_{(l-1)\times(l-1)} & 0\\ 0 & 1\end{bmatrix}-\begin{bmatrix}\tilde{\mathbf{U}}_{(l-1)\times p}\left(\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}\right)^{-1}\tilde{\mathbf{U}}_{(l-1)\times p}^{T} & \tilde{\mathbf{U}}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\\ \boldsymbol{\rho}_{l\|(l-1)}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}^{T} & \mathbf{e}^{T}(l)\boldsymbol{\rho}_{l|(l-1)}\end{bmatrix}\\
&\quad+\frac{1}{1+\mathbf{e}^{T}(l)\boldsymbol{\rho}_{l|(l-1)}}\begin{bmatrix}\tilde{\mathbf{U}}_{(l-1)\times p}\\ \mathbf{e}^{T}(l)\end{bmatrix}\left[\boldsymbol{\rho}_{l\backslash(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\quad \boldsymbol{\rho}_{l\backslash(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\mathbf{e}(l)\right]\\
&=\begin{bmatrix}\mathbf{I}_{(l-1)\times(l-1)} & 0\\ 0 & 1\end{bmatrix}-\begin{bmatrix}\tilde{\mathbf{U}}_{(l-1)\times p}\tilde{\mathbf{U}}_{(l-1)\times p}^{\#} & \tilde{\mathbf{U}}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\\ \boldsymbol{\rho}_{l\|(l-1)}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}^{T} & \mathbf{e}^{T}(l)\boldsymbol{\rho}_{l|(l-1)}\end{bmatrix}\\
&\quad+\frac{1}{1+\mathbf{e}^{T}(l)\boldsymbol{\rho}_{l|(l-1)}}\begin{bmatrix}\tilde{\mathbf{U}}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}^{T} & \tilde{\mathbf{U}}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\mathbf{e}(l)\\ \mathbf{e}^{T}(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}^{T} & \mathbf{e}^{T}(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\mathbf{e}(l)\end{bmatrix}\\
&=\begin{bmatrix}P_{\tilde{\mathbf{U}}_{(l-1)\times p}}^{\perp} & -\tilde{\mathbf{U}}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\\ -\left[\tilde{\mathbf{U}}_{(l-1)\times p}\left(\tilde{\mathbf{U}}_{(l-1)\times p}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}\right)^{-1}\mathbf{e}(l)\right]^{T} & 1-\mathbf{e}^{T}(l)\boldsymbol{\rho}_{l|(l-1)}\end{bmatrix}\\
&\quad+\frac{1}{1+\mathbf{e}^{T}(l)\boldsymbol{\rho}_{l|(l-1)}}\begin{bmatrix}\tilde{\mathbf{U}}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}^{T} & \tilde{\mathbf{U}}_{(l-1)\times p}\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\mathbf{e}(l)\\ \mathbf{e}^{T}(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\tilde{\mathbf{U}}_{(l-1)\times p}^{T} & \mathbf{e}^{T}(l)\boldsymbol{\rho}_{l|(l-1)}\boldsymbol{\rho}_{l|(l-1)}^{T}\mathbf{e}(l)\end{bmatrix}
\end{aligned} \tag{6.4}$$

Let $\mathbf{r}_{l}=\left(r_{1},r_{2},\cdots,r_{l-1},r_{l}\right)^{T}=\left(\mathbf{r}_{l-1}^{T},r_{l}\right)$. Then

$$
\begin{aligned}
&\left\| P^{\perp}_{\tilde{\mathbf{U}}_{l\times p}} \mathbf{r}_l \right\|^2 \\
&= \mathbf{r}_l^T P^{\perp}_{\tilde{\mathbf{U}}_{l\times p}} \mathbf{r}_l = \begin{bmatrix} \mathbf{r}_{l-1}^T & r_l \end{bmatrix} P^{\perp}_{\tilde{\mathbf{U}}_{l\times p}} \begin{bmatrix} \mathbf{r}_{l-1} \\ r_l \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{r}_{l-1}^T & r_l \end{bmatrix} \begin{bmatrix} P^{\perp}_{\tilde{\mathbf{U}}_{(l-1)\times p}} & -\tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)} \\ -\left[\tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)}\right]^T & 1-\mathbf{e}^T(l)\boldsymbol{\rho}_{l|(l-1)} \end{bmatrix} \begin{bmatrix} \mathbf{r}_{l-1} \\ r_l \end{bmatrix} \\
&+ \frac{1}{1+\mathbf{e}^T(l)\boldsymbol{\rho}_{l|(l-1)}} \begin{bmatrix} \mathbf{r}_{l-1}^T & r_l \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)} \boldsymbol{\rho}_{l|(l-1)}^T \tilde{\mathbf{U}}_{(l-1)\times p}^T & \tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)} \boldsymbol{\rho}_{l|(l-1)}^T \mathbf{e}(l) \\ \mathbf{e}^T(l) \boldsymbol{\rho}_{l|(l-1)} \boldsymbol{\rho}_{l|(l-1)}^T \tilde{\mathbf{U}}_{(l-1)\times p}^T & \mathbf{e}^T(l) \boldsymbol{\rho}_{l|(l-1)} \boldsymbol{\rho}_{l|(l-1)}^T \mathbf{e}(l) \end{bmatrix} \begin{bmatrix} \mathbf{r}_{l-1} \\ r_l \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{r}_{l-1}^T P^{\perp}_{\tilde{\mathbf{U}}_{(l-1)\times p}} - r_l \left[\tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)}\right]^T & -\mathbf{r}_{l-1}^T \tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)} + r_l \left(1-\mathbf{e}^T(l)\boldsymbol{\rho}_{l|(l-1)}\right) \end{bmatrix} \begin{bmatrix} \mathbf{r}_{l-1} \\ r_l \end{bmatrix} \\
&+ \frac{1}{1+\mathbf{e}^T(l)\boldsymbol{\rho}_{l|(l-1)}} \begin{bmatrix} \mathbf{r}_{l-1}^T \tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)} \boldsymbol{\rho}_{l|(l-1)}^T \tilde{\mathbf{U}}_{(l-1)\times p}^T + r_l \mathbf{e}^T(l) \boldsymbol{\rho}_{l|(l-1)} \boldsymbol{\rho}_{l|(l-1)}^T \tilde{\mathbf{U}}_{(l-1)\times p}^T & \mathbf{r}_{l-1}^T \tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)} \boldsymbol{\rho}_{l|(l-1)}^T \mathbf{e}(l) + r_l \mathbf{e}^T(l) \boldsymbol{\rho}_{l|(l-1)} \boldsymbol{\rho}_{l|(l-1)}^T \mathbf{e}(l) \end{bmatrix} \begin{bmatrix} \mathbf{r}_{l-1} \\ r_l \end{bmatrix} \\
&= \mathbf{r}_{l-1}^T P^{\perp}_{\tilde{\mathbf{U}}_{(l-1)\times p}} \mathbf{r}_{l-1} - r_l \left[\tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)}\right]^T \mathbf{r}_{l-1} - \mathbf{r}_{l-1}^T \tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)} r_l + r_l \left(1-\mathbf{e}^T(l)\boldsymbol{\rho}_{l|(l-1)}\right) r_l \\
&+ \frac{1}{1+\mathbf{e}^T(l)\boldsymbol{\rho}_{l|(l-1)}} \left\{ \mathbf{r}_{l-1}^T \tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)} \boldsymbol{\rho}_{l|(l-1)}^T \tilde{\mathbf{U}}_{(l-1)\times p}^T \mathbf{r}_{l-1} + r_l \mathbf{e}^T(l) \boldsymbol{\rho}_{l|(l-1)} \boldsymbol{\rho}_{l|(l-1)}^T \tilde{\mathbf{U}}_{(l-1)\times p}^T \mathbf{r}_{l-1} \right\} \\
&+ \frac{1}{1+\mathbf{e}^T(l)\boldsymbol{\rho}_{l|(l-1)}} \left\{ \mathbf{r}_{l-1}^T \tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)} \boldsymbol{\rho}_{l|(l-1)}^T \mathbf{e}(l) r_l + r_l \mathbf{e}^T(l) \boldsymbol{\rho}_{l|(l-1)} \boldsymbol{\rho}_{l|(l-1)}^T \mathbf{e}(l) r_l \right\} \\
&= \mathbf{r}_{l-1}^T P^{\perp}_{\tilde{\mathbf{U}}_{(l-1)\times p}} \mathbf{r}_{l-1} - 2r_l \left[\tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)}\right]^T \mathbf{r}_{l-1} + r_l \left(1-\mathbf{e}^T(l)\boldsymbol{\rho}_{l|(l-1)}\right) r_l \\
&+ \frac{1}{1+\mathbf{e}^T(l)\boldsymbol{\rho}_{l|(l-1)}} \left(\mathbf{r}_{l-1}^T \tilde{\mathbf{U}}_{(l-1)\times p} \boldsymbol{\rho}_{l|(l-1)} + \boldsymbol{\rho}_{l|(l-1)}^T \mathbf{e}(l) r_l \right) \left(\boldsymbol{\rho}_{l|(l-1)}^T \tilde{\mathbf{U}}_{(l-1)\times p}^T \mathbf{r}_{l-1} + \boldsymbol{\rho}_{l|(l-1)}^T \mathbf{e}(l) r_l \right)
\end{aligned}
\tag{6.5}
$$

6.3 Various Versions of RBP-SGA Algorithm

The concept of PBP of SGA (PBP-SGA) was first explored and investigated in (Chang, 2003) where finding SV was carried out by finding matrix determinants via singular value decomposition (SVD). There are two issues needed to be resolved. One is excessive computational cost of calculating SV by SVD. Another is the SV calculated by the matrix determinant is not true SV as noted in (Chang, 2013). As a result, PBP-SGA remains a conceptual approach. This section looks into PBP-SGA from a rather different perspective.

6.3.1 Recursive-Band-Processing of OPSGA (RBP-OPSGA) Algorithm

Recursive-Band-Processing of OPSGA (RBP-OPSGA) Algorithm

Initial condition: p is the number of endmembers to be found by PBP-OPSGA algorithm, determined by VD.

Outer Loop ($p \le l \le L$)

Inner Loop ($1 \le j \le p$)

1. Initial condition:

Find an initial target pixel vector $\mathbf{e}_1^l = \arg\left\{\max_{\mathbf{r}_l} \mathbf{r}_l^T \mathbf{r}_l\right\}$. $\mathbf{U}_{l\times 1} = \left[\mathbf{e}_1^l\right]$, $\tilde{\mathbf{U}}_{l\times 1} = \mathbf{U}_{l\times 1} - \mathbf{e}_1^l$ and calculate $P^{\perp}_{\tilde{\mathbf{U}}_{l\times 1}}$. Set $j = 2$.

2. At the j^{th} iteration, find $\mathbf{e}_j^l$ by finding the maximum height of a simplex over all data sample vectors, $\left\{\mathbf{r}_l\ (i)\right\}_{i=1}^{N}$. This can be done by the following
    a. Find $\mathbf{e}_j^l = \arg\left\{\max_{\tilde{\mathbf{r}}} \left\| P^{\perp}_{\tilde{\mathbf{U}}_{l\times(j-1)}} \tilde{\mathbf{r}} \right\|\right\}$ via (5).
    b. Calculate $\left\| \mathbf{e}_j^{l\ \mathrm{OP}} \right\| = \left\| P^{\perp}_{\tilde{\mathbf{U}}_{l\times(j-1)}} \mathbf{e}_j^l \right\|$
    c. Use $\mathbf{e}_j^{l\ \mathrm{OP}}$ obtained in step 2(b) to compute $V\left(\mathbf{e}_1^l, \mathbf{e}_2^l \cdots, \mathbf{e}_j^l\right)$.
    d. Update $P^{\perp}_{\tilde{\mathbf{U}}_{l\times j}}$ via $P^{\perp}_{\tilde{\mathbf{U}}_{l\times(j-1)}}$ via (4).
3. If $j < p$, then $j \leftarrow j+1$ and go step 1. Otherwise, output $\left\{\mathbf{e}_1^l, \mathbf{e}_2^l \cdots, \mathbf{e}_p^l\right\}$ and go to outer loop.

End (Inner Loop)

if $l < L$, then $l \leftarrow l+1$. Otherwise, output $\bigcup_{l=1}^{L} \left\{\mathbf{e}_1^l, \mathbf{e}_2^l \cdots, \mathbf{e}_p^l\right\}$.

End (Outer Loop).

6.3.2 Recursive-Band-Processing of GSGA (RBP-GSGA) Algorithm

Recursive-Band-Processing of GSGA (RBP-GSGA) Algorithm

Initial condition: p is the number of endmembers to be found by RBP-GSGA algorithm, determined by VD.

Outer Loop ($p \le l \le L$)

Inner Loop ($1 \le j \le p$)

1. Initial condition:

   Find two endmembers, $\mathbf{e}_1^l$ and $\mathbf{e}_2^l$ that yield a line segment with the maximal length. In other words, find $\mathbf{e}_1^l$ and $\mathbf{e}_2^l$ with the maximal Euclidean distance, i.e.,

$\left(\mathbf{e}_1^l,\mathbf{e}_2^l\right)=\arg\left\{\max_{(\mathbf{r},\mathbf{s})} d(\mathbf{r},\mathbf{s})\right\}$ where $d(\cdot,\cdot)$ is the Euclidean distance measure. $\mathbf{U}_{l\times 2}=\left[\mathbf{e}_1^l,\mathbf{e}_2^l\right]$, $\tilde{\mathbf{U}}_{l\times 2}=\mathbf{U}_{l\times 2}-\mathbf{e}_1^l$ and calculate $P^{\perp}_{\tilde{\mathbf{U}}_{l\times 2}}$. Set j = 3.

2. At the jth iteration, find $\mathbf{e}_j^l$ by finding the maximum height of a simplex over all data sample vectors, $\left\{\mathbf{r}_l(i)\right\}_{i=1}^N$. This can be done by the following
   a. Find $\mathbf{e}_j^l=\arg\left\{\max_{\tilde{\mathbf{r}}}\left\|P^{\perp}_{\tilde{\mathbf{U}}_{l\times(j-1)}}\tilde{\mathbf{r}}\right\|\right\}$ via (5).
   b. Calculate $\left\|\mathbf{e}_j^{l\ \mathrm{OP}}\right\|=\left\|P^{\perp}_{\tilde{\mathbf{U}}_{l\times(j-1)}}\mathbf{e}_j^l\right\|$
   c. Use $\mathbf{e}_j^{l\ \mathrm{OP}}$ obtained in step 2(b) to compute $V\left(\mathbf{e}_1^l,\mathbf{e}_2^l\cdots,\mathbf{e}_j^l\right)$.
   d. Update $P^{\perp}_{\tilde{\mathbf{U}}_{l\times j}}$ via $P^{\perp}_{\tilde{\mathbf{U}}_{l\times(j-1)}}$ via (4).
3. If $j < p$, then $j \leftarrow j+1$ and go step 1. Otherwise, output $\left\{\mathbf{e}_1^l,\mathbf{e}_2^l\cdots,\mathbf{e}_p^l\right\}$ and go to outer loop.

End (Inner Loop)

if $l < L$, then $l \leftarrow l+1$. Otherwise, output $\bigcup_{l=1}^{L}\left\{\mathbf{e}_1^l,\mathbf{e}_2^l\cdots,\mathbf{e}_p^l\right\}$.

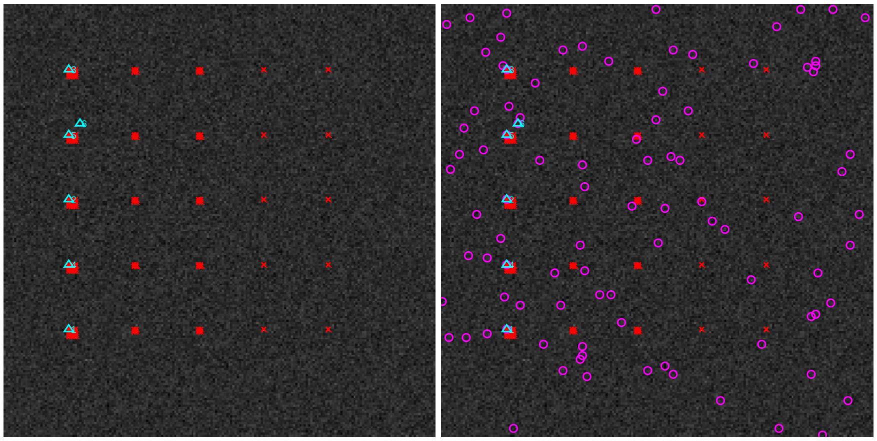
End (Outer Loop).

6.4 Detailed Experiments

6.4.1 Synthetic Data

A. Synthetic TI Experiments

Fig. 6.1 conducts a comparative study between OPSGA and PBP-OPSGA. Fig. 6.1(a) shows the results of original OPSGA applying on TI scene with VD = 6, Fig. 6.1(b) displays the spatial location of PBP-OPSGA results with the same value of VD. As shown in Fig. 6.1, the red cross indicates the spatial location of the ground truth pixels in TI scene, the magenta circle denotes the endmembers found in the previous bands, the cyan upper triangle highlights the endmembers identified in the current band, and the number next to the triangle points out the order of the endmembers been found. PBP-OPSGA acquires the same results as SGA in the last iteration, which verified the correction of the algorithm. OPSGA is a one-shot process, while PBP-OPSGA is not, it is a slow motion of OPSGA and some moving targets can be discovered during the process, the pixels circled by magenta color presents the additional endmembers picked up by PBP-OPSGA.



(a) (b)

Figure 6.1. Comparative study between OPSGA and PBP-OPSGA applying on TI Scene. (a) OPSGA experimental results with VD = 6 (b) PBP-OPSGA experimental results with VD = 6

There are 5 ground truth pixels are identified as endmembers by PBP-OPSGA. Fig. 6.2 displays the order of the ground truth pixels been discovered versus the number of bands processed. It is clear shown in the graph that, material C and material K are the first two materials been discovered.

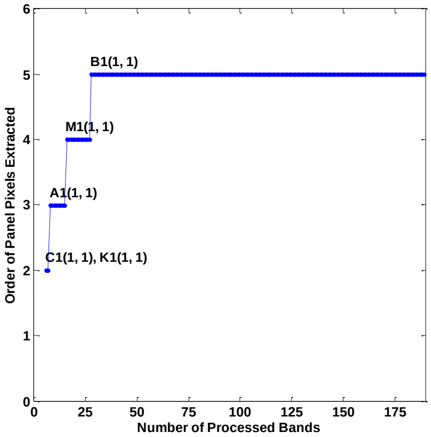
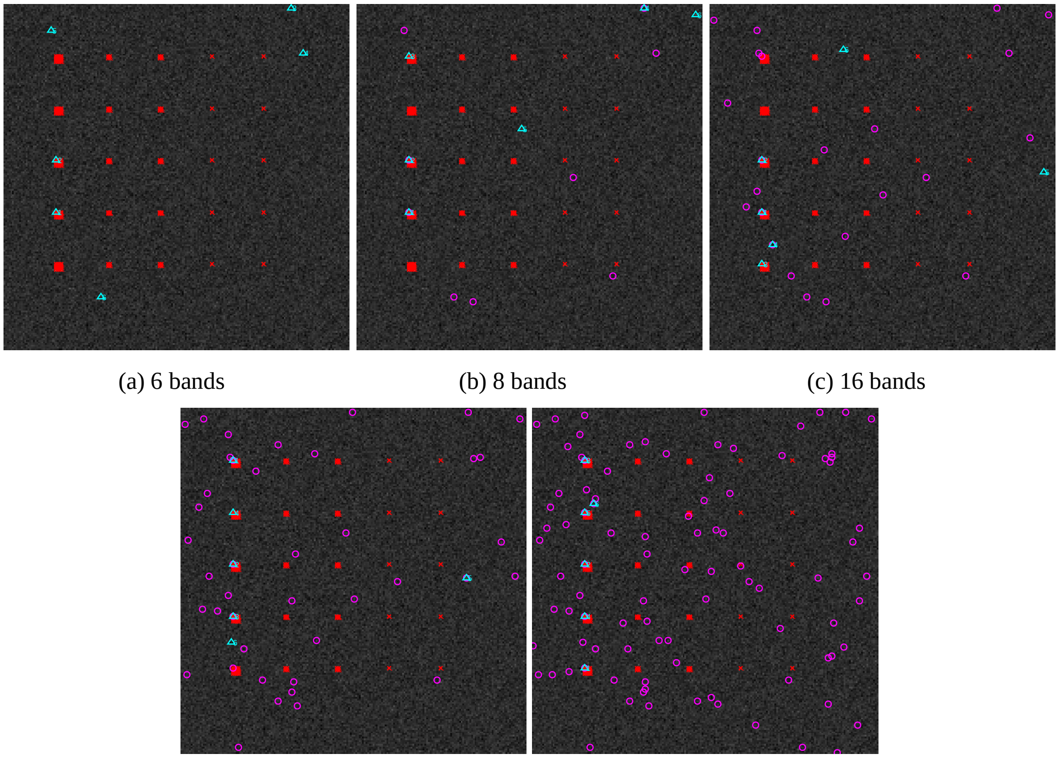


Figure 6.2. Order of the ground truth pixels been discovered versus the number of bands processed

Figure 6.3 further shows the spatial location of the endmember candidates identified by PBP-OPSGA with different number of bands processed starting from 18 bands. In these figures, the red cross indicates the spatial location of the R-panel pixels in HYDICE scene, the magenta circle denotes the endmembers found in the previous bands, the cyan upper triangle highlights the endmembers identified in the current band, and the number next to the triangle points out the order of the endmembers been found.



(a) 6 bands (b) 8 bands (c) 16 bands

(d) 28 bands (e) 189 bands (full bands)

Figure 6.3. Endmembers identified by PBP-OPSGA with different number of bands processed (a) 6 bands (b) 8 bands (c) 16 bands (d) 28 bands (e) 189 bands (full bands)

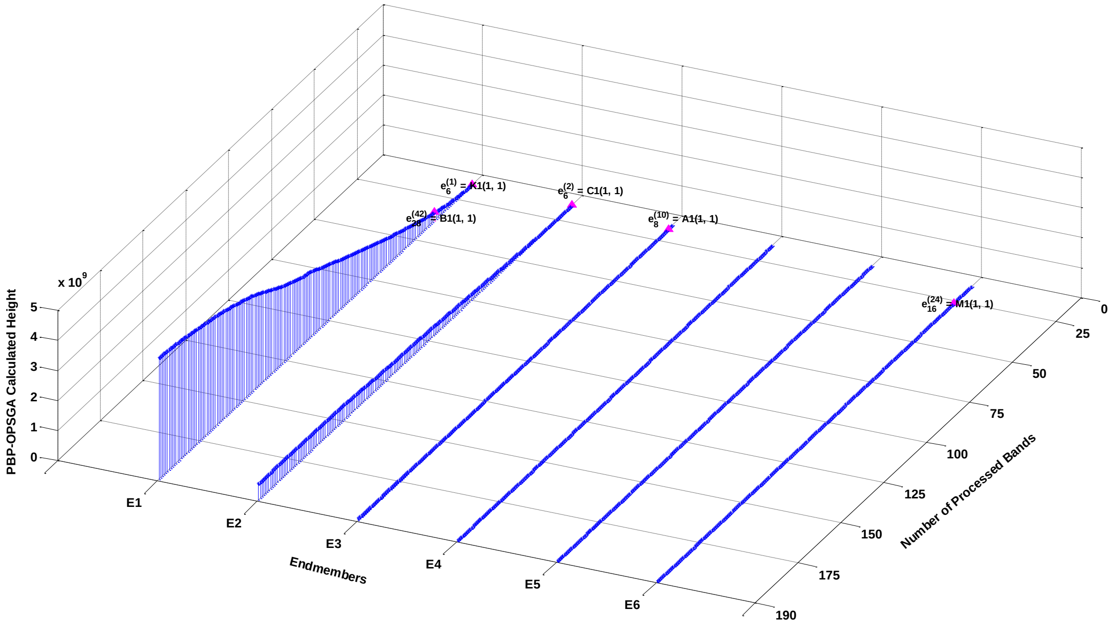


Figure 6.4. 3D plot of the progressive change of PBP-OPSGA calculated height applying on TI scene

Fig. 6.4 shows a progressive changes of PBP-OPSGA calculated height of the 18 endmembers identified by PBP-OPSGA, where *x*, *y* and *z* axes denote found members, the number of the first bands being used for data processing, and the calculated height by PBP-

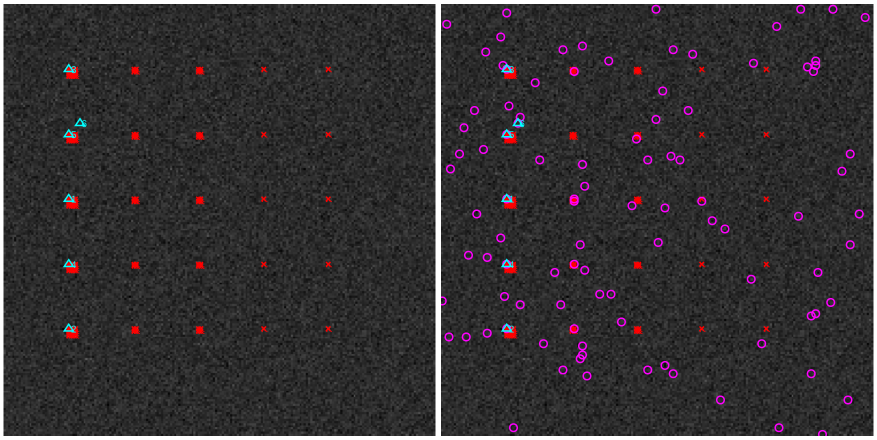
OPSGA. The magenta arrow indicates a particular panel pixel is first identified as an endmember. For example, C1(1, 1) can be discovered with 6 bands collected, it is the 2 endmember found by PBP-OPSGA algorithm.

Table 6.1 summarize the minimal number of bands required identifying signatures, order of signatures to be found and the order of signatures found in the particular band for all the 5 signatures been identified with PBP-OPSGA in HYDICE scene.

Table 6.1. Summary of minimal number of bands processed to identify signatures and order of signatures to be found in particular bands

Signatures found by RBP-GSGA	Minimal n_l identifying signatures	Order of signatures to be found	Order of signatures found in the particular band
$\mathrm{A}1(1,1)=\mathbf{e}_{8}^{(10)}$	8	10	3
$\mathrm{B}1(1,1)=\mathbf{e}_{28}^{(42)}$	28	42	1
$\mathrm{C}1(1,1)=\mathbf{e}_{6}^{(2)}$	6	2	2
$\mathrm{K}1(1,1)=\mathbf{e}_{6}^{(1)}$	6	1	1
$\mathrm{M}1(1,1)=\mathbf{e}_{16}^{(24)}$	16	24	6

Fig. 6.5 conducts a comparative study between GSGA and RBP-GSGA. Fig. 6.5(a) shows the results of original GSGA applying on synthetic TI scene with VD = 6, Fig. 6.5(b) displays the spatial location of RBP-GSGA results with the same value of VD. As shown in Fig. 6.5, the red cross indicates the spatial location of the ground truth pixels, the magenta circle denotes the endmembers found in the previous bands, the cyan upper triangle highlights the endmembers identified in the current band, and the number next to the triangle points out the order of the endmembers been found. RBP-GSGA acquires the same results as GSGA in the last iteration, which verified the correction of the algorithm. GSGA is a one-shot process, while RBP-GSGA is not, it is a slow motion of GSGA and some moving targets can be discovered during the process, the pixels circled by magenta color presents the additional endmembers picked up by RBP-GSGA. Furthermore, a total of 10 ground truth pixels can be discovered by RBP-GSGA comparing with only 5 ground truth pixels picked up by the original GSGA.



(a) (b)

Figure 6.5. Comparative study between GSGA and RBP-GSGA applying on synthetic TI scene. (a) GSGA experimental results with VD = 6 (b) RBP-GSGA experimental results with VD = 6

As mentioned in the previous sections, there are 10 ground truth pixels are identified as endmembers by RBP-GSGA. Fig. 6.6 displays the order of the ground truth pixels been discovered versus the number of bands processed. It is clear shown in the graph that, both material C and K can be discovered with 6 bands collected and processed.

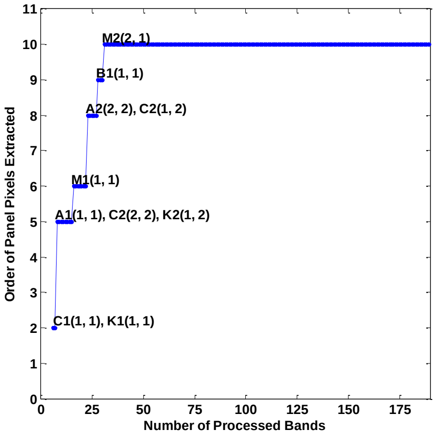
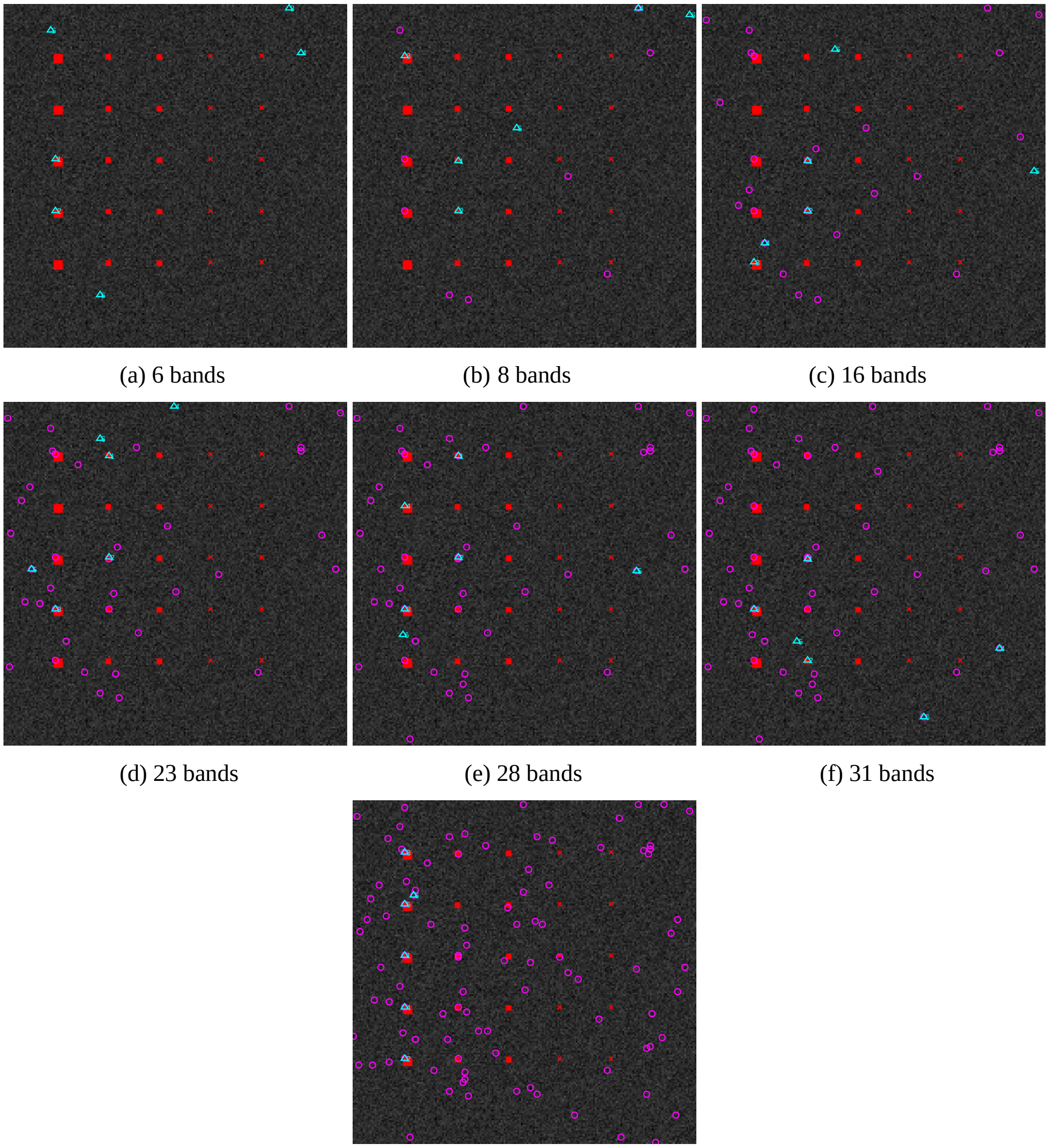


Figure 6.6. Order of the ground truth pixels been discovered versus the number of bands processed

Figure 6.7 further shows the spatial location of the endmember candidates identified by RBP-GSGA with different number of bands processed starting from 6 bands. In these figures, the red cross indicates the spatial location of the ground truth pixels, the magenta circle denotes the endmembers found in the previous bands, the cyan upper triangle highlights the endmembers identified in the current band, and the number next to the triangle points out the order of the endmembers been found. Material C and K are found with 6 bands collected and processed, A can be discovered with 8 bands collected, M and B are identified with 16 bands and 28 bands accordingly. 4 ground truth pixels sitting on the second column can be discovered as well.



(a) 6 bands

(b) 8 bands

(c) 16 bands

(d) 23 bands

(e) 28 bands

(f) 31 bands

(g) 189 bands (full bands)

Figure 6.7. Endmembers identified by RBP-GSGA with different number of bands processed (a) 6 bands (b) 8 bands (c) 16 bands (d) 23 bands (e) 28 bands (f) 31 bands (g) 189 bands (full bands)

Fig. 6.8 shows a progressive changes of RBP-GSGA calculated height of the 6 endmembers identified by RBP-GSGA, where *x*, *y* and *z* axes denote found members, the number of the first bands being used for data processing, and the calculated height by RBP-GSGA. The magenta arrow indicates a particular panel pixel is first identified as an endmember. For example, A1(1, 1) is identified with 8 bands received and processed, it is the 12th endmembers found by RBP-GSGA algorithm.

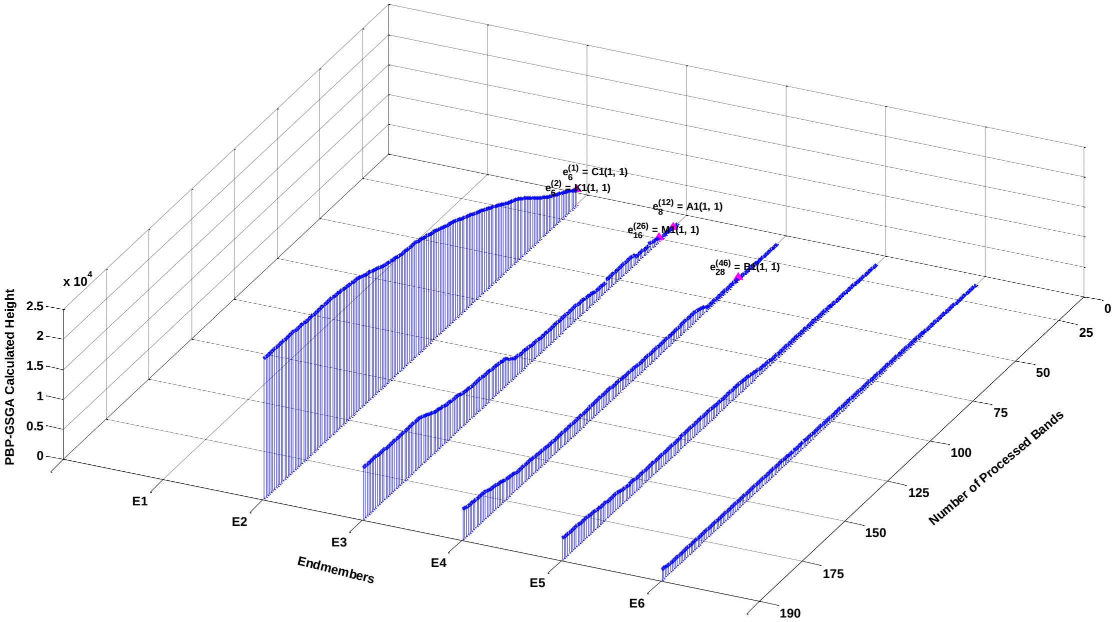


Figure 6.8. 3D plot of the progressive change of RBP-GSGA calculated height

Table 6.2 summarize the minimal number of bands required identifying signatures, order of signatures to be found and the order of signatures found in the particular band for all the 10 ground truth pixels been identified with RBP-GSGA in TI scene.

Table 6.2. Summary of minimal number of bands processed to identify signatures and order of signatures to be found in particular bands

Signatures found by RBP-GSGA	Minimal n_l identifying signatures	Order of signatures to be found	Order of signatures found in the particular band
$\text{A}1(1,1) = \mathbf{e}_8^{(12)}$	8	12	3
$\text{B}1(1,1) = \mathbf{e}_{28}^{(46)}$	28	46	4
$\text{C}1(1,1) = \mathbf{e}_6^{(1)}$	6	1	1
$\text{K}1(1,1) = \mathbf{e}_6^{(2)}$	6	2	2
$\text{M}1(1,1) = \mathbf{e}_{16}^{(26)}$	16	26	3
$\text{A}2(2,2) = \mathbf{e}_{23}^{(38)}$	23	38	1
$\text{C}2(1,2) = \mathbf{e}_{23}^{(39)}$	23	39	2
$\text{C}2(2,2) = \mathbf{e}_8^{(10)}$	8	10	1
$\text{K}2(1,2) = \mathbf{e}_8^{(11)}$	8	11	2
$\text{M}2(2,2) = \mathbf{e}_{31}^{(52)}$	31	52	2

B. Synthetic TE Experiments

Fig. 6.9 conducts a comparative study between OPSGA and PBP-OPSGA. Fig. 6.9(a) shows the results of original OPSGA applying on TE scene with VD = 6, Fig. 6.9(b) displays the spatial location of PBP-OPSGA results with the same value of VD. As shown in Fig. 6.9, the red cross indicates the spatial location of the ground truth pixels in TE scene, the magenta circle denotes the endmembers found in the previous bands, the cyan upper triangle highlights the endmembers identified in the current band, and the number next to the triangle points out the order of the endmembers been found. PBP-OPSGA acquires the same results as SGA in the last iteration, which verified the correction of the algorithm. OPSGA is a one-shot process, while PBP-OPSGA is not, it is a slow motion of OPSGA and some moving targets can be discovered during the process, the pixels circled by magenta color presents the additional endmembers picked up by PBP-OPSGA.

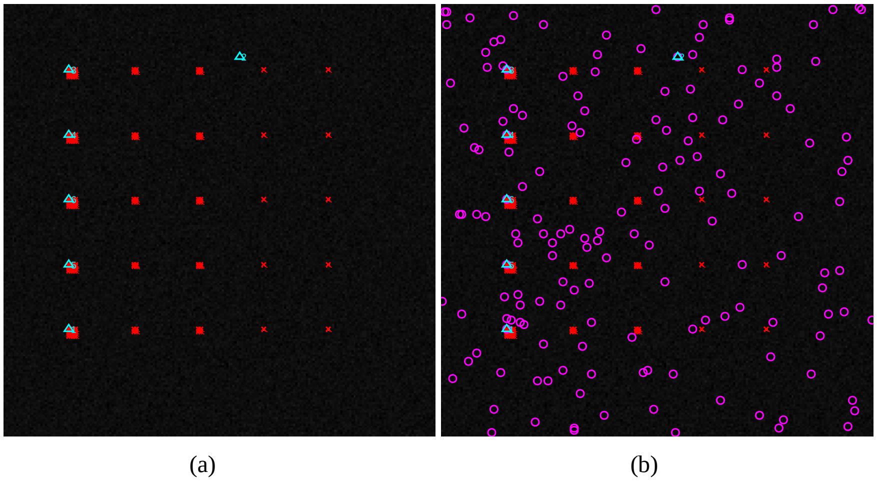


Figure 6.9. Comparative study between OPSGA and PBP-OPSGA applying on TE Scene. (a) OPSGA experimental results with VD = 6 (b) PBP-OPSGA experimental results with VD = 6

There are 5 ground truth pixels are identified as endmembers by PBP-OPSGA. Fig. 6.10 displays the order of the ground truth pixels been discovered versus the number of bands processed. It is clear shown in the graph that, material A and material K are the first two materials been discovered.

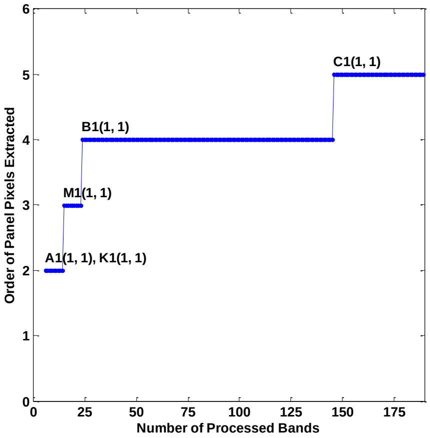
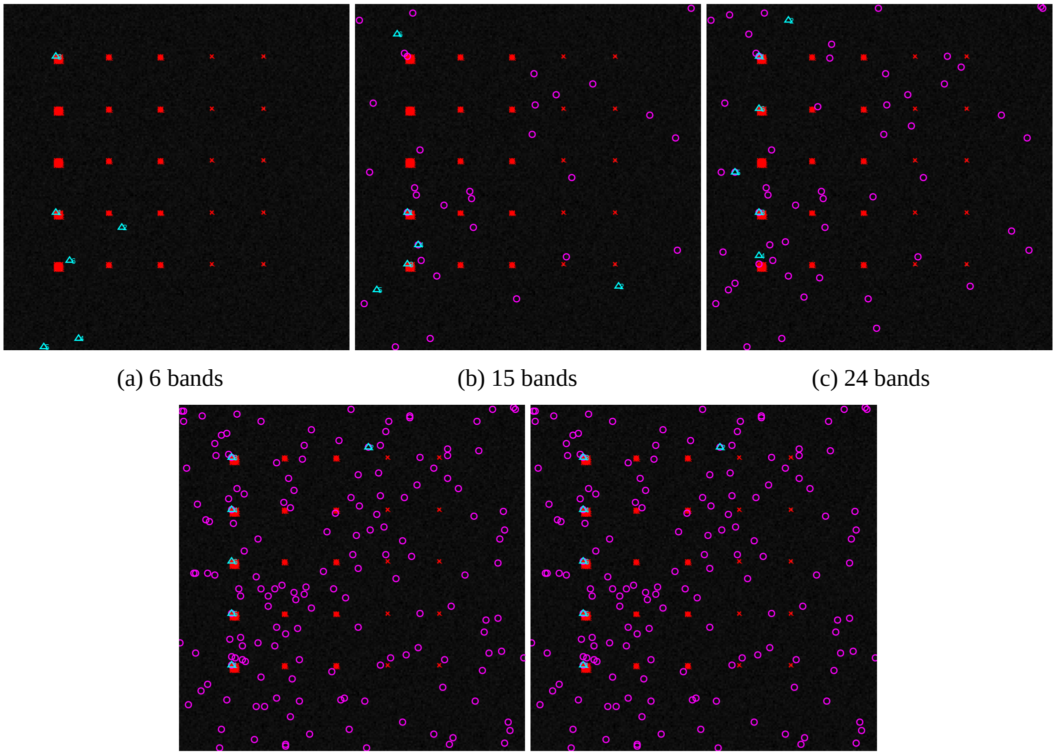


Figure 6.10. Order of the ground truth pixels been discovered versus the number of bands processed

Figure 6.11 further shows the spatial location of the endmember candidates identified by PBP-OPSGA with different number of bands processed starting from 6 bands. In these figures, the red cross indicates the spatial location of the R-panel pixels in HYDICE scene, the magenta circle denotes the endmembers found in the previous bands, the cyan upper triangle highlights the endmembers identified in the current band, and the number next to the triangle points out the order of the endmembers been found.



(a) 6 bands (b) 15 bands (c) 24 bands

(d) 146 bands (e) 189 bands (full bands)

Figure 6.11. Endmembers identified by PBP-OPSGA with different number of bands processed (a) 6 bands (b) 15 bands (c) 24 bands (d) 146 bands (e) 189 bands (full bands)

Fig. 6.12 shows a progressive changes of PBP-OPSGA calculated height of the 18 endmembers identified by PBP-OPSGA, where *x*, *y* and *z* axes denote found members, the number of the first bands being used for data processing, and the calculated height by PBP-OPSGA. The magenta arrow indicates a particular panel pixel is first identified as an endmember.

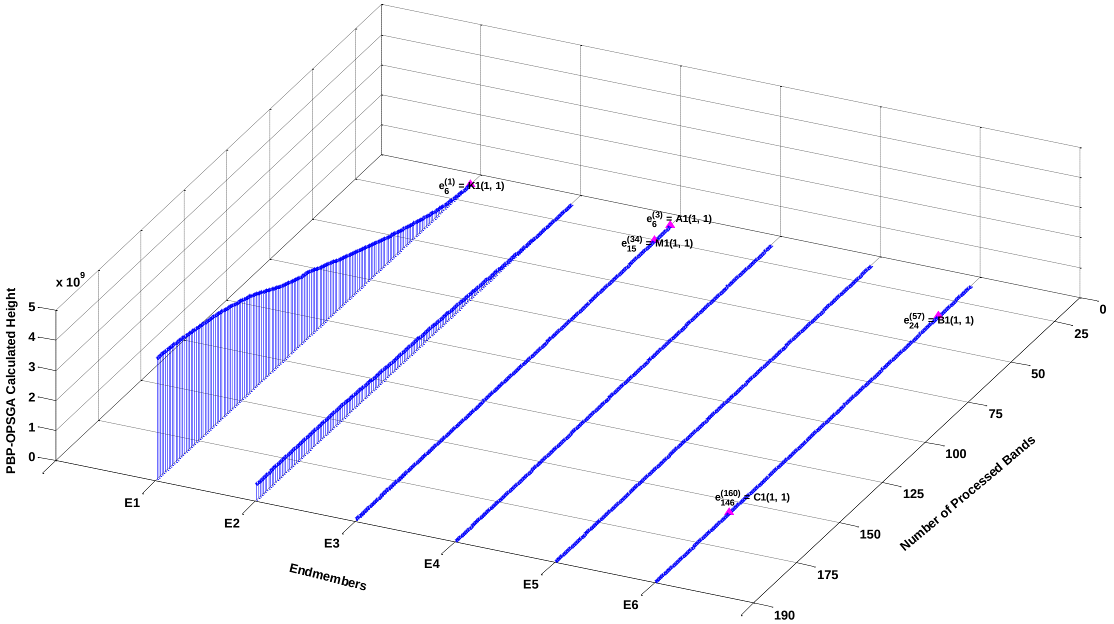


Figure 6.12. 3D plot of the progressive change of PBP-OPSGA calculated height applying on TI scene

Table 6.3 summarize the minimal number of bands required identifying signatures, order of signatures to be found and the order of signatures found in the particular band for all the 5 signatures been identified with PBP-OPSGA in TE scene.

Table 6.3. Summary of minimal number of bands processed to identify signatures and order of signatures to be found in particular bands

Signatures found by RBP-GSGA	Minimal n_l identifying signatures	Order of signatures to be found	Order of signatures found in the particular band
$\mathrm{A}1(1,1) = \mathbf{e}_6^{(3)}$	6	3	3
$\mathrm{B}1(1,1) = \mathbf{e}_{24}^{(57)}$	24	57	6
$\mathrm{C}1(1,1) = \mathbf{e}_{146}^{(160)}$	146	160	6
$\mathrm{K}1(1,1) = \mathbf{e}_6^{(1)}$	6	1	1
$\mathrm{M}1(1,1) = \mathbf{e}_{15}^{(34)}$	15	34	3

Fig. 6.13 conducts a comparative study between GSGA and RBP-GSGA. Fig. 6.13(a) shows the results of original GSGA applying on synthetic TE scene with VD = 6, Fig. 6.13(b) displays the spatial location of RBP-GSGA results with the same value of VD. As shown in Fig. 6.13, the red cross indicates the spatial location of the ground truth pixels, the magenta circle denotes the endmembers found in the previous bands, the cyan upper triangle highlights the endmembers identified in the current band, and the number next to the triangle points out the order of the endmembers been found. RBP-GSGA acquires the same results as GSGA in the last iteration, which verified the correction of the algorithm. GSGA is a one-

shot process, while RBP-GSGA is not, it is a slow motion of GSGA and some moving targets can be discovered during the process, the pixels circled by magenta color presents the additional endmembers picked up by RBP-GSGA. Furthermore, a total of 11 ground truth pixels can be discovered by RBP-GSGA comparing with only 5 ground truth pixels picked up by the original GSGA.

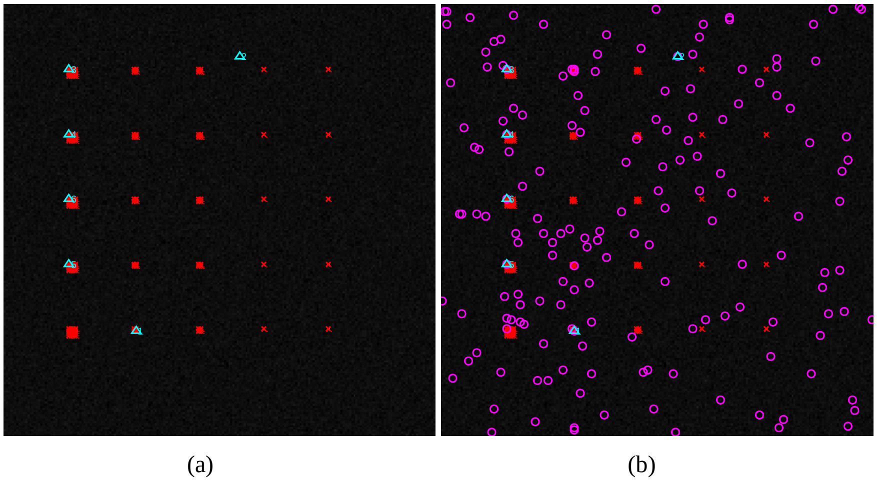


Figure 6.13. Comparative study between GSGA and RBP-GSGA applying on synthetic TE scene. (a) GSGA experimental results with VD = 6 (b) RBP-GSGA experimental results with VD = 6

As mentioned in the previous sections, there are 11 ground truth pixels are identified as endmembers by RBP-GSGA. Fig. 6.14 displays the order of the ground truth pixels been discovered versus the number of bands processed. It is clear shown in the graph that, both material A and K can be discovered with 6 bands collected and processed.

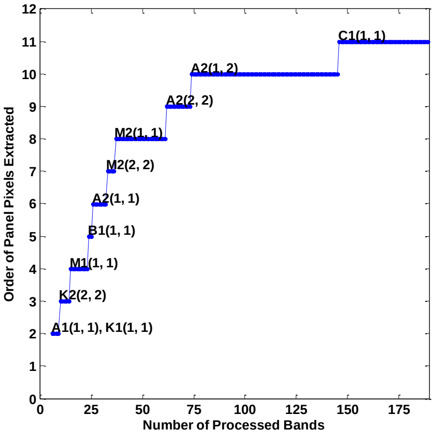
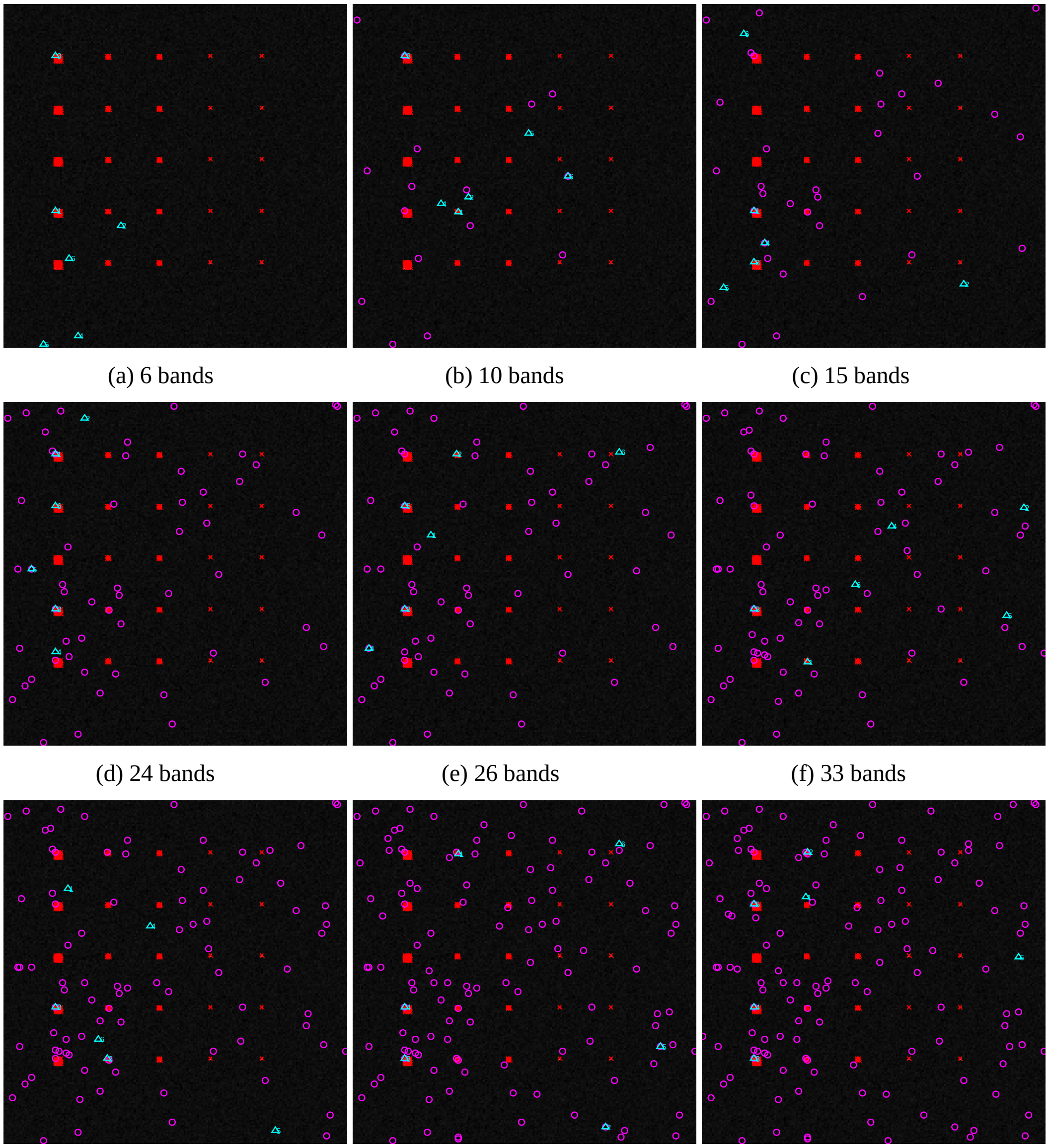


Figure 6.14. Order of the ground truth pixels been discovered versus the number of bands processed

Figure 6.15 further shows the spatial location of the endmember candidates identified by RBP-GSGA with different number of bands processed starting from 6 bands. In these figures, the red cross indicates the spatial location of the ground truth pixels, the magenta circle denotes the endmembers found in the previous bands, the cyan upper triangle highlights the endmembers identified in the current band, and the number next to the triangle points out the order of the endmembers been found. Material A and K are found with 6 bands collected and processed, M can be discovered with 15 bands collected, B and C are identified with 24 bands and 146 bands accordingly. 5 ground truth pixels sitting on the second column can be discovered as well.



(a) 6 bands

(b) 10 bands

(c) 15 bands

(d) 24 bands

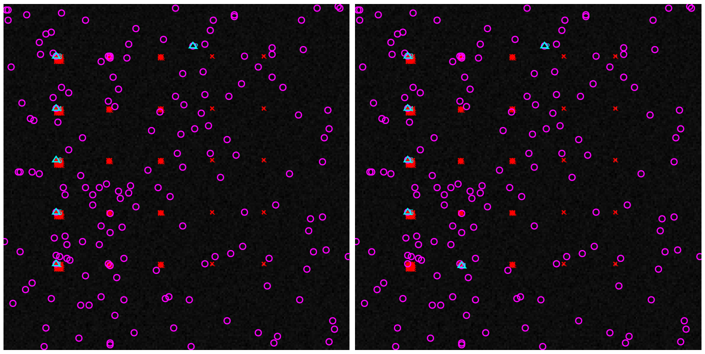
(e) 26 bands

(f) 33 bands

(g) 37 bands

(h) 62 bands

(i) 74 bands



(j) 146 bands (k) 189 bands (full bands)

Figure 6.15. Endmembers identified by RBP-GSGA with different number of bands processed (a) 6 bands (b) 10 bands (c) 15 bands (d) 24 bands (e) 26 bands (f) 33 bands (g) 37 bands (h) 62 bands (i) 74 bands (j) 146 bands (k) 189 bands (full bands)

Fig. 6.16 shows a progressive changes of RBP-GSGA calculated height of the 6 endmembers identified by RBP-GSGA, where *x*, *y* and *z* axes denote found members, the number of the first bands being used for data processing, and the calculated height by RBP-GSGA. The magenta arrow indicates a particular panel pixel is first identified as an endmember. For example, A1(1, 1) is identified with 6 bands received and processed, it is the 8th endmembers found by RBP-GSGA algorithm.

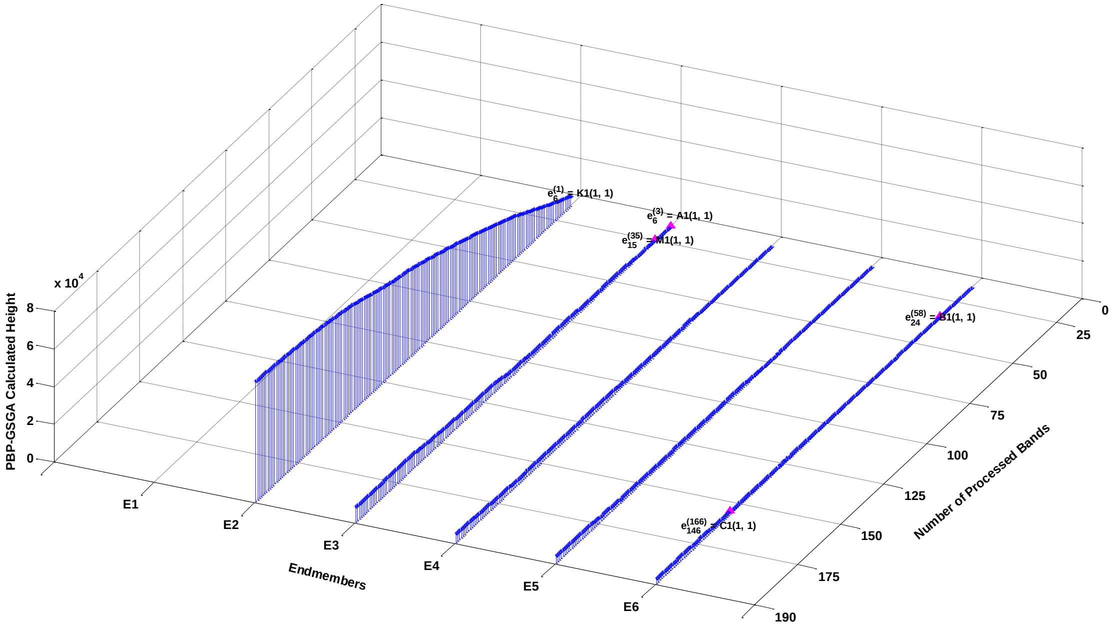


Figure 6.16. 3D plot of the progressive change of RBP-GSGA calculated height

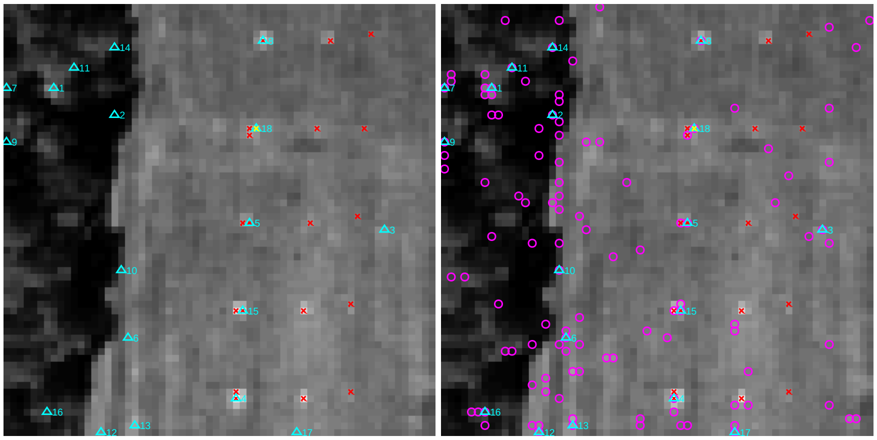
Table 6.4 summarize the minimal number of bands required identifying signatures, order of signatures to be found and the order of signatures found in the particular band for all the 11 ground truth pixels been identified with RBP-GSGA in TE scene.

Table 6.4. Summary of minimal number of bands processed to identify signatures and order of signatures to be found in particular bands

Signatures found by RBP-GSGA	Minimal n_l identifying signatures	Order of signatures to be found	Order of signatures found in the particular band
$\mathrm{A}1(1,1) = \mathbf{e}_{6}^{(3)}$	6	3	3
$\mathrm{B}1(1,1) = \mathbf{e}_{24}^{(58)}$	24	58	6
$\mathrm{C}1(1,1) = \mathbf{e}_{146}^{(166)}$	146	166	6
$\mathrm{K}1(1,1) = \mathbf{e}_{6}^{(1)}$	6	1	1
$\mathrm{M}1(1,1) = \mathbf{e}_{15}^{(35)}$	15	35	3
$\mathrm{A}2(1,1) = \mathbf{e}_{26}^{(62)}$	26	62	2
$\mathrm{A}2(1,2) = \mathbf{e}_{74}^{(131)}$	74	131	2
$\mathrm{A}2(2,2) = \mathbf{e}_{62}^{(120)}$	62	120	1
$\mathrm{K}2(2,2) = \mathbf{e}_{10}^{(17)}$	10	17	1
$\mathrm{M}2(1,1) = \mathbf{e}_{37}^{(89)}$	37	89	2
$\mathrm{M}2(2,2) = \mathbf{e}_{33}^{(77)}$	33	77	1

6.4.2 Real Data (HYDICE Scene)

Fig. 6.17 conducts a comparative study between OPSGA and PBP-OPSGA. Fig. 6.17(a) shows the results of original OPSGA applying on HYDICE scene with VD = 18, Fig. 6.17(b) displays the spatial location of PBP-OPSGA results with the same value of VD. As shown in Fig. 6.17, the red cross indicates the spatial location of the R-panel pixels in HYDICE scene, the yellow cross displays the spatial location of the Y-panel pixels in the scene, the magenta circle denotes the endmembers found in the previous bands, the cyan upper triangle highlights the endmembers identified in the current band, and the number next to the triangle points out the order of the endmembers been found. PBP-OPSGA acquires the same results as SGA in the last iteration, which verified the correction of the algorithm. OPSGA is a one-shot process, while PBP-OPSGA is not, it is a slow motion of OPSGA and some moving targets can be discovered during the process, the pixels circled by magenta color presents the additional endmembers picked up by PBP-OPSGA. Furthermore, a total of 8 ground truth pixels (7 R-panel pixels and 1 Y-panel pixel) can be discovered by PBP-OPSGA comparing with only 5 ground truth pixels (4 R-panel pixels and 1 Y-panel pixel) picked up by the original OPSGA.



(a) (b)

Figure 6.17. Comparative study between OPSGA and PBP-OPSGA applying on HYDICE Scene. (a) OPSGA experimental results with VD = 18 (b) PBP-OPSGA experimental results with VD = 18

As mentioned in the previous sections, there are 8 ground truth pixels (7 R-panel pixels and 1 Y-panel pixel) are identified as endmembers by PBP-OPSGA. Fig. 6.18 displays the order of the ground truth pixels been discovered versus the number of bands processed. It is clear shown in the graph that, p_{212} and p_{521} are the first two panel pixels been found with 18 band collected and received.

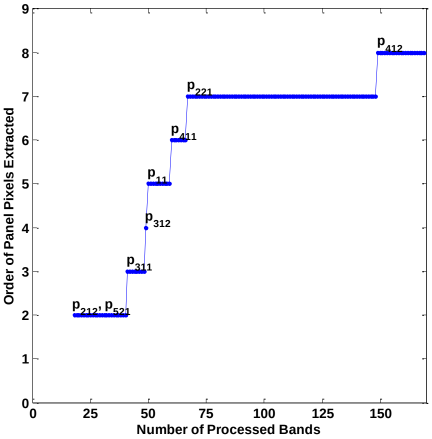
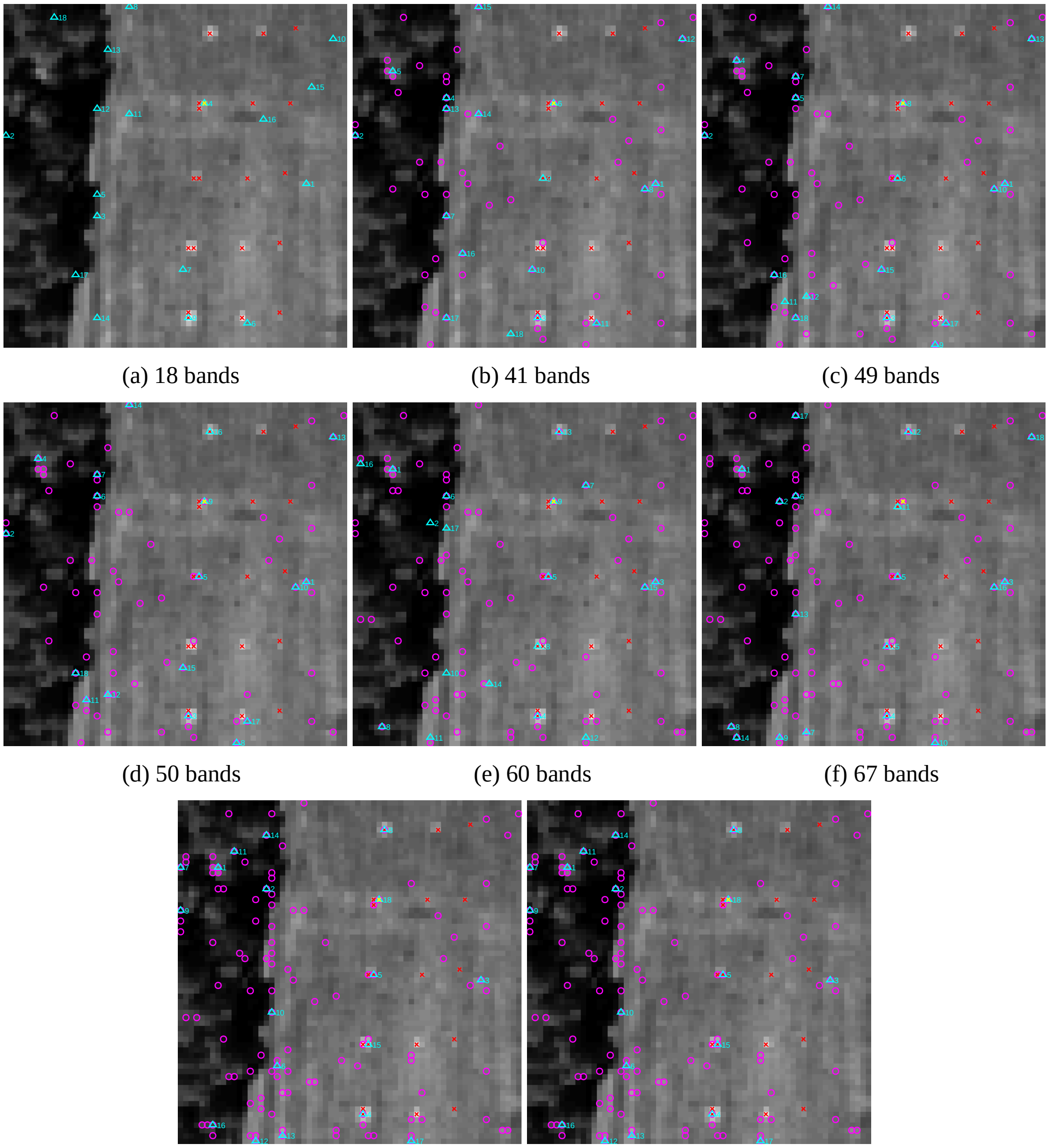


Figure 6.18. Order of the ground truth pixels been discovered versus the number of bands processed

Figure 6.19 further shows the spatial location of the endmember candidates identified by PBP-OPSGA with different number of bands processed starting from 18 bands. In these figures, the red cross indicates the spatial location of the R-panel pixels in HYDICE scene, the yellow cross displays the spatial location of the Y-panel pixels in the scene, the magenta circle denotes the endmembers found in the previous bands, the cyan upper triangle highlights the endmembers identified in the current band, and the number next to the triangle points out the order of the endmembers been found. p_{212} and p_{521} are the first two panel pixels been found with 18 band collected and received. Followed by p_{311} with 41 bands received,

p_{312} with 49 band received, p_{11} with 50 bands collected, p_{411} with 60 bands collected, p_{221} with 67 bands collected, and p_{412} is the last panel pixels been identified with 149 band processed. There is no more panel pixels been found after band 149.



(a) 18 bands

(b) 41 bands

(c) 49 bands

(d) 50 bands

(e) 60 bands

(f) 67 bands

(g) 149 bands

(h) 169 bands (full bands)

Figure 6.19. Endmembers identified by PBP-OPSGA with different number of bands processed (a) 18 bands (b) 41 bands (c) 49 bands (d) 50 bands (e) 60 bands (f) 67 bands (g) 149 bands (h) 169 bands (full bands)

Fig. 6.20 shows a progressive changes of PBP-OPSGA calculated height of the 18 endmembers identified by PBP-OPSGA, where *x*, *y* and *z* axes denote found members, the number of the first bands being used for data processing, and the calculated height by PBP-OPSGA. The magenta arrow indicates a particular panel pixel is first identified as an endmember. For example, p_{212} is first identified as endmember with 18 bands collected and processed, it is the 4th endmember identified by PBP-OPSGA algorithm as well.

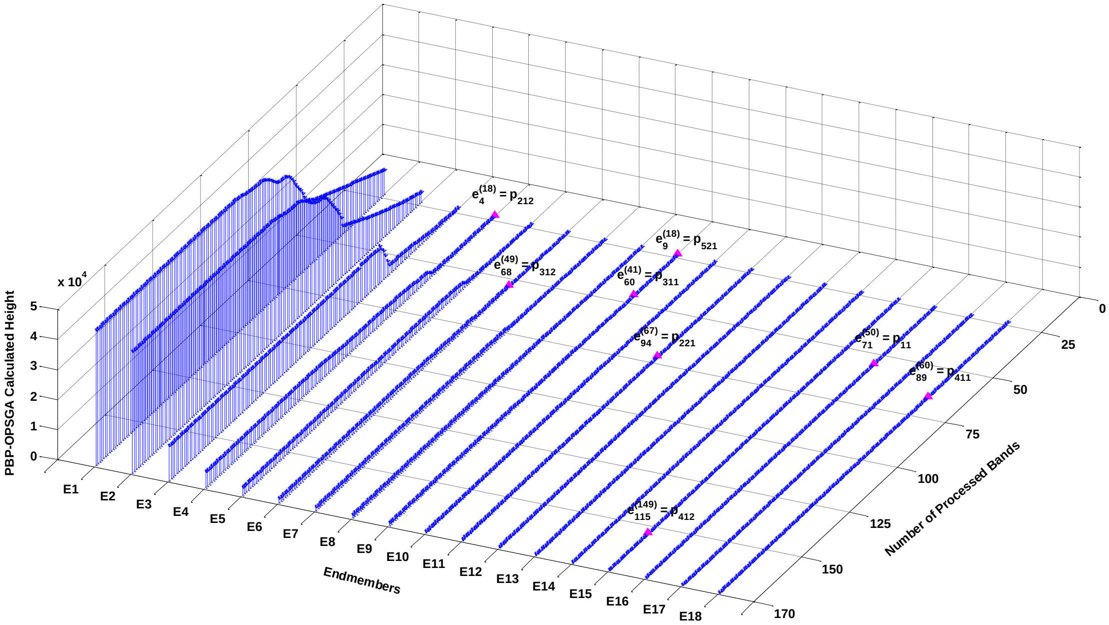


Figure 6.20. 3D plot of the progressive change of PBP-OPSGA calculated height

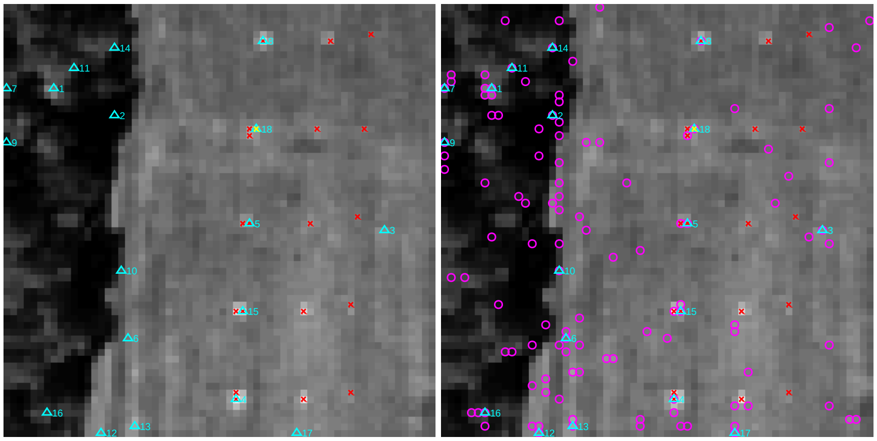
Table 6.5 summarize the minimal number of bands required identifying signatures, order of signatures to be found and the order of signatures found in the particular band for all the 8 panel pixels been identified with PBP-OPSGA in HYDICE scene.

Table 6.5. Summary of minimal number of bands processed to identify signatures and order of signatures to be found in particular bands

Signatures found by RBP-ATGP	Minimal n_l identifying signatures	Order of signatures to be found	Order of signatures found in the particular band
$p_{11} = \mathbf{t}_{71}^{(50)}$	50	71	16
$p_{212} = \mathbf{t}_{4}^{(18)}$	18	4	4
$p_{221} = \mathbf{t}_{94}^{(67)}$	67	94	11
$p_{311} = \mathbf{t}_{60}^{(41)}$	41	60	9
$p_{312} = \mathbf{t}_{68}^{(49)}$	49	68	6
$p_{411} = \mathbf{t}_{89}^{(60)}$	60	89	18
$p_{412} = \mathbf{t}_{115}^{(149)}$	149	115	15
$p_{521} = \mathbf{t}_{9}^{(18)}$	18	9	9

Fig. 6.21 conducts a comparative study between GSGA and RBP-GSGA. Fig. 6.21(a) shows the results of original GSGA applying on HYDICE scene with VD = 18, Fig. 6.21(b)

displays the spatial location of RBP-GSGA results with the same value of VD. As shown in Fig. 6.21, the red cross indicates the spatial location of the R-panel pixels in HYDICE scene, the yellow cross displays the spatial location of the Y-panel pixels in the scene, the magenta circle denotes the endmembers found in the previous bands, the cyan upper triangle highlights the endmembers identified in the current band, and the number next to the triangle points out the order of the endmembers been found. RBP-GSGA acquires the same results as GSGA in the last iteration, which verified the correction of the algorithm. GSGA is a one-shot process, while RBP-GSGA is not, it is a slow motion of GSGA and some moving targets can be discovered during the process, the pixels circled by magenta color presents the additional endmembers picked up by RBP-GSGA. Furthermore, a total of 8 ground truth pixels (7 R-panel pixels and 1 Y-panel pixel) can be discovered by RBP-GSGA comparing with only 5 ground truth pixels (4 R-panel pixels and 1 Y-panel pixel) picked up by the original GSGA.



(a) (b)

Figure 6.21. Comparative study between GSGA and RBP-GSGA applying on HYDICE Scene. (a) GSGA experimental results with VD = 18 (b) RBP-GSGA experimental results with VD = 18

As mentioned in the previous sections, there are 8 ground truth pixels (7 R-panel pixels and 1 Y-panel pixel) are identified as endmembers by RBP-GSGA. Fig. 6.22 displays the order of the ground truth pixels been discovered versus the number of bands processed. It is clear shown in the graph that, p_{212} and p_{521} are the first two panel pixels been found with 18 band collected and received.

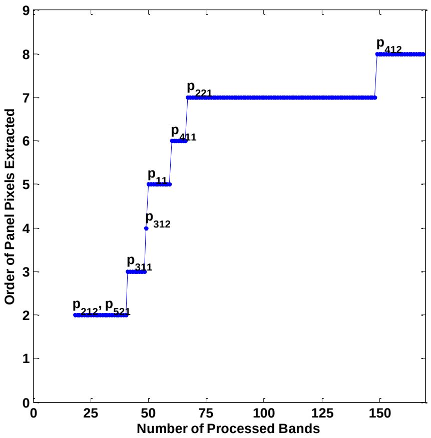
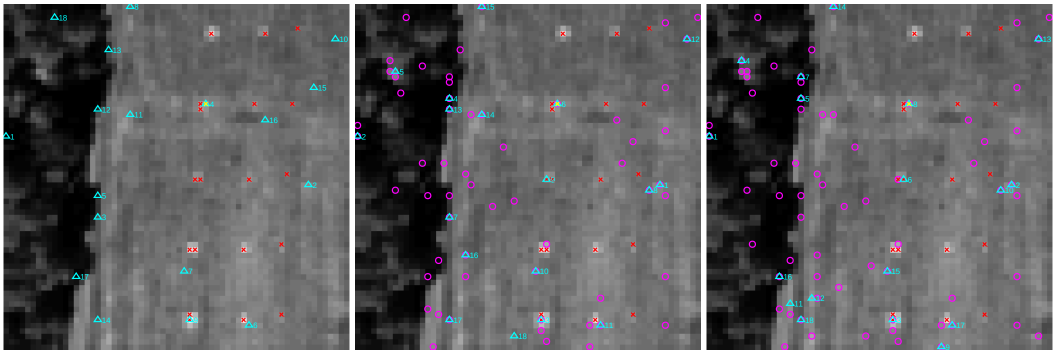
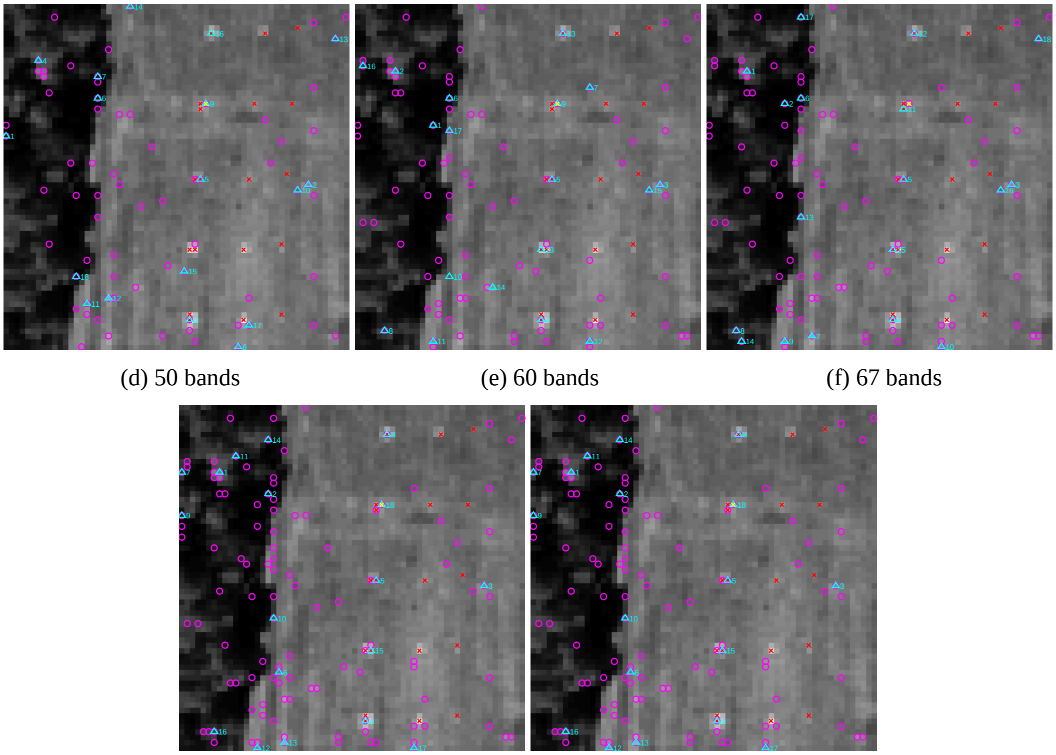


Figure 6.22. Order of the ground truth pixels been discovered versus the number of bands processed

Figure 6.23 further shows the spatial location of the endmember candidates identified by RBP-GSGA with different number of bands processed starting from 18 bands. In these figures, the red cross indicates the spatial location of the R-panel pixels in HYDICE scene, the yellow cross displays the spatial location of the Y-panel pixels in the scene, the magenta circle denotes the endmembers found in the previous bands, the cyan upper triangle highlights the endmembers identified in the current band, and the number next to the triangle points out the order of the endmembers been found. p_{212} and p_{521} are the first two panel pixels been found with 18 band collected and received. Followed by p_{311} with 41 bands received, p_{312} with 49 band received, p_{11} with 50 bands collected, p_{411} with 60 bands collected, p_{221} with 67 bands collected, and p_{412} is the last panel pixels been identified with 149 band processed. There is no more panel pixels been found after band 149.



(a) 18 bands (b) 41 bands (c) 49 bands



(d) 50 bands (e) 60 bands (f) 67 bands

(g) 149 bands (h) 169 bands (full bands)

Figure 6.23. Endmembers identified by RBP-GSGA with different number of bands processed (a) 18 bands (b) 41 bands (c) 49 bands (d) 50 bands (e) 60 bands (f) 67 bands (g) 149 bands (h) 169 bands (full bands)

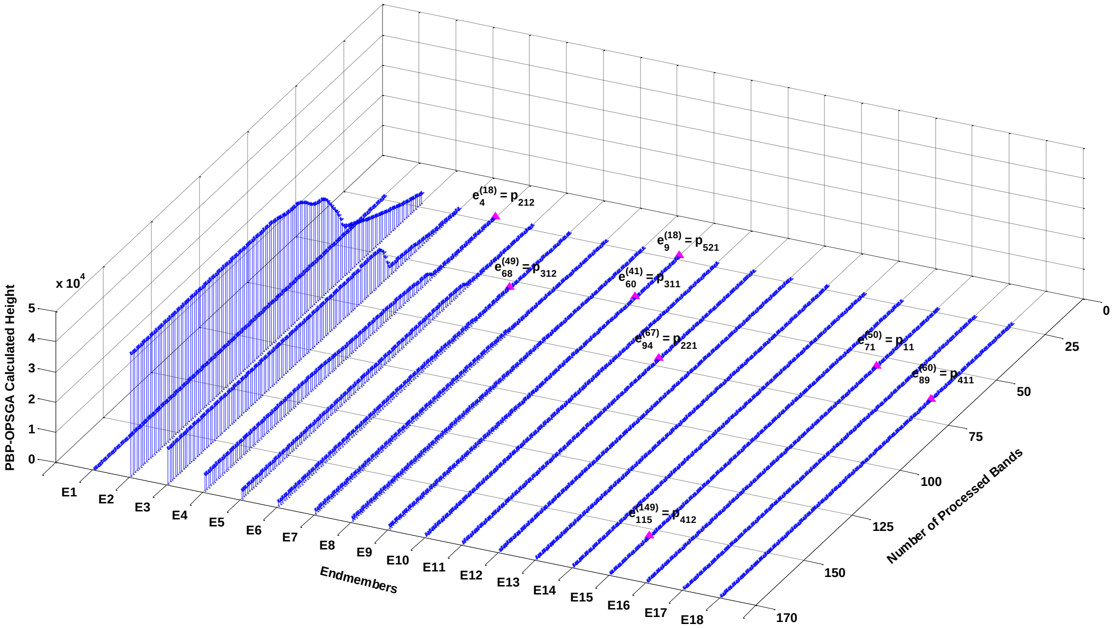


Figure 6.24. 3D plot of the progressive change of RBP-GSGA calculated height

Fig. 6.24 shows a progressive changes of RBP-GSGA calculated height of the 18 endmembers identified by RBP-GSGA, where *x*, *y* and *z* axes denote found members, the number of the first bands being used for data processing, and the calculated height by RBP-GSGA. The magenta arrow indicates a particular panel pixel is first identified as an endmember. For example, p_{212} is first identified as endmember with 18 bands collected and processed, it is the 4th endmember identified by RBP-GSGA algorithm as well.

Table 6.6 summarize the minimal number of bands required identifying signatures, order of signatures to be found and the order of signatures found in the particular band for all the 8 panel pixels been identified with RBP-GSGA in HYDICE scene.

Table 6.6. Summary of minimal number of bands processed to identify signatures and order of signatures to be found in particular bands

Signatures found by RBP-GSGA	Minimal n_l identifying signatures	Order of signatures to be found	Order of signatures found in the particular band
$p_{11} = \mathbf{t}_{71}^{(50)}$	50	71	16
$p_{221} = \mathbf{t}_{4}^{(18)}$	18	4	4
$p_{212} = \mathbf{t}_{94}^{(67)}$	67	94	11
$p_{311} = \mathbf{t}_{60}^{(41)}$	41	60	9
$p_{312} = \mathbf{t}_{68}^{(49)}$	49	68	6
$p_{411} = \mathbf{t}_{89}^{(60)}$	60	89	18
$p_{412} = \mathbf{t}_{115}^{(149)}$	149	115	15
$p_{521} = \mathbf{t}_{9}^{(18)}$	18	9	9

6.5 GUI Design

A graphical user interface (GUI) with a screenshot shown in Fig. 6.25 was developed using Matlab's GUIDE to aid in algorithm performance analysis. The three images displayed on top of the window show a color image of the scene, a gray scale image of the current band being processed, and the result after the current band is processed. At the bottom a window shows the RBP-OPSGA calculated height for each discovered endmember in a 3D view. Once the data is loaded, the user can start the process by clicking the start button. At each iteration each band is received and being processed. Upon completion of processing, the current band and the resulting image are updated to allow the user to observe the results. A final note on Fig. 6.25 is worthwhile. The plots shown at the bottom window is a real-time progressive version of the 3D plots of the RBP-OPSGA calculated height for each found endmember. Unfortunately, this nice feature can be only demonstrated in real time process.

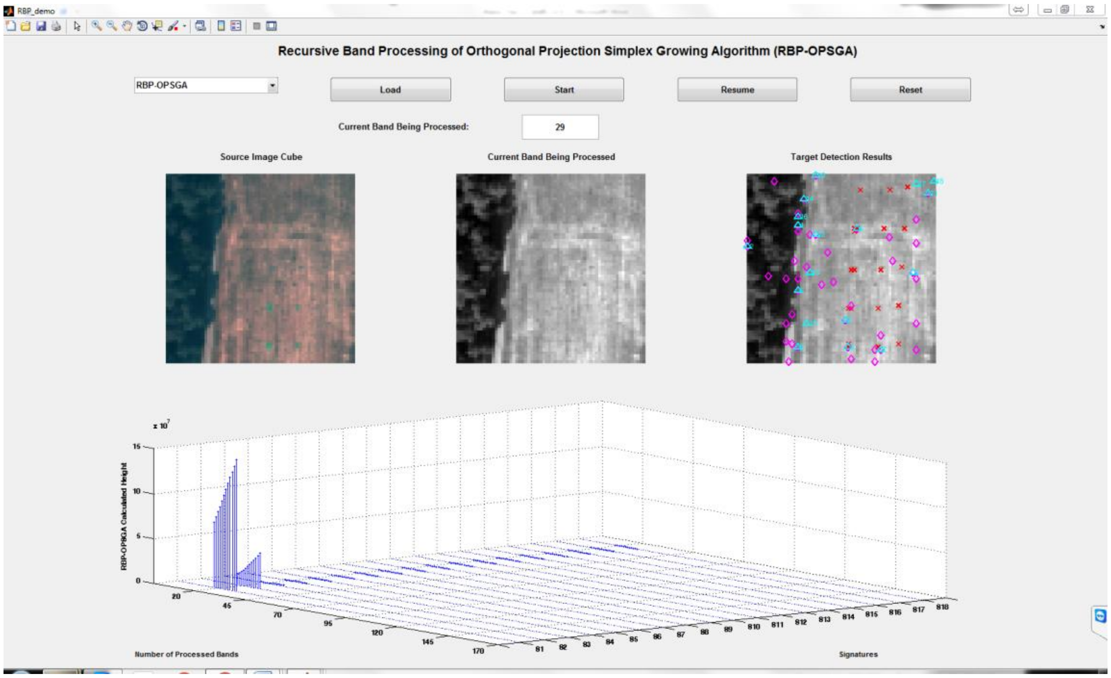

RBP_demo
Recursive Band Processing of Orthogonal Projection Simplex Growing Algorithm (RBP-OPSGA)
RBP-OPSGA
Load
Start
Resume
Reset
Current Band Being Processed:
29
Source Image Cube
Current Band Being Processed
Target Detection Results
x 10^7
15
10
5
0
RBP-OPSGA Calculated Height
20
45
70
95
120
145
170
Number of Processed Bands
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
S17
S18
Signatures

Figure 6.25. GUI Design for RBP-SGA

6.6 Conclusions

This chapter presents a new approach to finding endmembers which allows users to find endmembers band by band progressively and recursively. It is referred to as recursive band processing (RBP) and extends a well-known SGA to RBP-SGA. Since endmembers are generally considered as insignificant targets because of their rare appearance in the data. Accordingly, they may be very likely missed in many cases. The proposed RBP provides advantages of finding endmembers during its inter-band processing. In fact, as shown in experiments, RBP-SGA can find more endmembers than SGA does during its progressive endmember finding process.

Chapter 7: Conclusions

7.1 Conclusions

This dissertation develops a theory of Recursive Band Processing (RBP) for finding unsupervised targets in hyperspectral imagery. The proposed RBP allows users to process algorithms one band at a time so as to achieve real time processing. To the author's best knowledge, the idea of RBP has not been previously explored before. It is particularly suitable for applications that have limited bandwidth for data transmission and communication where progressive results provide early assessment as well as timely decision on whether or not the current process needs to be continued. In this dissertation, several potential algorithms such as Automatic Target Generation Process (ATGP) for active target detection, Pixel Purity Index (PPI) and Fast Iterative Pixel Purity Index (FIPPI) find endmembers for passive target detection, are studied and extended to their RBP versions. Most importantly, in order to demonstrate its real time processing capability, this dissertation also develops Graphical User Interfaces (GUIs) to show all the algorithms processing in real time. These GUIs pave the way for future hardware implementation.

Chapter 3 presents an active target detection approach RBP-ATGP which is shown not only possible but also have comparable performance to the traditional ATGP for target detection with less data transmission required. Moreover, RBP-ATGP algorithm provides a spectral profile of the detected abundance fractions, this will provide clues for band selection, prioritization. Furthermore, comparing to the original ATGP RBP-ATGP can find more endmembers and offer more valuable information for data analyst.

Chapter 4 develops RBP-PPI which conducts target detection in a passive manner with no assumed prior knowledge. The theory behind PPI is OP, but is it is not as easy as it looks. This is because there are 4 parameters involves in the process, the number of skewers K, the number of processed bands l, the number of processed pixels n and the proper threshold t for PPI Count value. RBP-P-PPI and RBP-C-PPI are developed to address the issues that how to choose a proper number of skewers K and how to process data if only partial pixel data is available correspondingly.

Chapter 5 shows RBP-FIPPI, which is a special case of RBP-PPI, with no need of determining the number of skewers K and the value of threshold t. In the mealtime, since

RBP-FIPPI using a small amount of skewers it reduces computational complexity. In addition, RBP-FIPPI has also the ability to reduce the amount of data necessary to transmit. This supports the concept that RBP algorithms can be used to speed the decision making process while the data acquisition is ongoing.

Chapter 6 introduces and presents a recursive-band-processing version of a well-known endmember finding algorithm, Simplex Growing Algorithm (SGA) referred to as PBP-SGA. SGA was previously developed by Chang et al. (2006) to ease the computational complexity of N-finder algorithm (N-FINDR) developed by Winter. PBP-SGA allows users to find endmembers by growing simplexes band by band without waiting for completion of all bands.

7.2 Future Work

There are a number of extensions for future research.. One potential method that hasn't been fully explored is the concept of progressive band tuning (PBT). Conceptually, band tuning can add and remove bands in a forward and backward manner to identify bands of interest and improve results.

Another area is to determine an optimal number of bands needed to be processed. Presently, the algorithms are intended to have users decide if any further bands should be processed according to their interpreted results. Without operator interaction, they will continue to process data band by band until all data is received. In the future, automatic band processing should be designed in such a fashion that the algorithm can be terminated according to custom designed stopping rules.

While the algorithms presented have been successfully implemented in software, the datasets used were relatively small. One can expect a significant increase as the spatial dimensions of the hyperspectral image increase. Additionally, computing time required for newer sensors that may contain thousands of bands of spectral information may exceed the ability of software implementation. RBP paves the way for designing hardware devices such as FPGAs or ASICs for future hyperspectral data processing.

List of Reference

Berman, M. (2004) “ICE: A statistical approach to identifying endmembers in hyperspectral images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 10, pp. 2085–2095, Oct. 2004.

Bernabe, S, Lopez, S., Plaza, A., Sarmiento, R. and Rodriguze, P.G. (2011) “FPGA design of an automatic target generation process for hyperspectral image analysis,” *2011 IEEE 17th International Conference on Parallel and Distributed Systems* (ICPADS), pp. 1010-1015, 7-9 Dec., Tainan, Taiwan.

Bernabe, S., Lopez, S., Plaza, A. and Sarmiento, R. (2013) “GPU Implementation of an automatic target detection and classification algorithm for hyperspectral image analysis,” *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 2, pp. 221-225, March.

Bioucas-Dias, J. M. (2012) “Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.

Boardman, J. W. (1994) “Geometric mixture analysis of imaging spectrometry data,” in *Proc. Int. Geosci. Remote Sens. Symp.*, vol. 4, pp. 2369 - 2371.

Brown, A. J. (2006) “Spectral curve fitting for automatic hyperspectral data analysis,” *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1601–1608, Jun.

Chang, C.-I. (2003). *Hyperspectral Imaging Techniques for Spectral Detection and Classification.* New York: Kluwer Academic

Chang, C.-I, Wu, C., Liu, W., and Ouyang, Y. C. (2006) “A growing method for simplex-based endmember extraction algorithms,” *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 10, pp. 2804–2819, Oct.

Chang C. – I and Plaza, A., (2006) "A fast iterative algorithm for implementation of pixel purity index," in *Geoscience and Remote Sensing Letters, IEEE,* vol.3, no.1, pp.63-67, Jan.

Chang, C.-I, Wu, C. C., Lo, C.-S., and Chang, M.-L. (2010) “Real-time simplex growing algorithms for hyperspecral endmember extraction,” *IEEE Trans. Geosci. Remote Sens.,* vol. 40, no. 4, pp. 1834–1850, Apr. 2010.

Chang, C.-I. (2012). Progressive hyperspectral imaging. *Proceedings SPIE 8539, High-Performance Computing in Remote Sensing II.* SPIE.

Chang, C.-I. (2013) *Hyperspectral Data Processing: Signal Processing Algorithm Design and Analysis*, Wiley, N.J.

Chang, C.-I, Wen, C. H., and Wu, C. C. (2013) "Relationship exploration among PPI, ATGP and VCA via theoretical analysis," *Int. J. Comput. Sci. Eng.*, vol. 8, no. 4, pp. 361–367, 2013.

Chang, C.-I, Wen, C.H., and Wu, C.C. (2013) "Relationship exploration among PPI, ATGP and VCA via theoretical analysis," *Int. J. of Computational Science and Engineering*, vol. 8, no. 4, pp. 361-367.

Chaudhry, F. (2005) Pixel Purity Index-Based Endmember Extraction for Hyperspectral Data Exploitation, MS Thesis, Department of Computer Science and Electrical Engineering, August.

Chen, H.M., Lin, C., Chen, S.Y., Wen, C.H., Chen, C.C.C., Ouyang, Y.C. and Chang, C.-I. (2013) "PPI-iterative FLDA approach to unsupervised multispectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing*, vol. 6, no. 4, pp. 1834-1842, August.

Chen, S.Y., Li, H.C., Gao, C, and Chang, C.-I. (2014) "Algorithm analysis for ATGP, VCA and SGA," *Satellite Data Compression, Communication and Processing X* (ST146), *SPIE International Symposium on* SPIE Sensing Technology + Applications, Baltimore, MD, 5-9 May.

Greg, I. (2010) 'An evaluation of three endmember extraction algorithms, ATGP, ICA-EEA and VCA," *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing* (WHISPERS), pp. 1-4.

Heylen, R., Parente, M., and Gader, P. (2014) "A review of nonlinear hyperspectral unmixing methods," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 1844–1868, Jun. 2014.

HYDICE Data Caveats Version 1.0. (1996, March 15).

HYMSMO Program Office. (1995). *Forest Radiance Collection and Exploitation Operations Plan.* HYMSMO Program Office

Nascimento, J. M. P. and Bioucas-Dias, J. M. (2005) "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, Apr.

Paola, J. D. and R. A. Schowengerdt (1997). "The effect of neural network structure on a multispectral land-use/land-cover classification. Photogrammetric Engineering and Remote Sensing.

Poor, H. V. (1994) *An Introduction to Signal Detection and Estimation*. New York: Springer.

Reed, I. S., & Yu, X. (1990, October). Adaptive Multiple-Band CFAR Detection of an Optical Pattern with Unknown Spectral Distribution. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 38*(10), 1760-1770.

Ren, H., Chang, C.-I. (2003) "Automatic spectral target recognition in hyperspectral imagery," in *Aerospace and Electronic Systems, IEEE Transactions on*, vol.39, no.4, pp.1232-1249, Oct.

Resmini, Ron, HYMSMO. (1997, March 12). Memo: Using the target pixels. HYMSMO.

Schultz R. (2014) Progressive Band Processing for Hyperspectral Imaging (Doctoral dissertation)

Seber, G. A. (2008). *A Matrix Handbook for Statisticians.* Hoboken: John Wiley & Sons, Inc.

Topographic Technology Laboratory Topographic Engineering Center. (1995). *Field Data Collection at Aberdeen Proving Ground, MD Final Report.* Aberdeen Proving Ground: US Army Corp of Engineers.

Tompkins, S., Mustarrd, J. F., Pieters, C. M., and Forsyth, D. W. (1997) "Optimization of targets for spectral mixture analysis," *Remote Sens. Environ.*, vol. 59, pp. 472–489, 1997.

Wang, X. R., Brown, A. J., and Upcroft, B. (2005) "Applying incremental EM to Bayesian classifiers in the learning of hyperspectral remote sensing data," in *Proc. 7th Int. Conf. Inf. Fusion*, pp. 606–613.

Winter, M. E., "N-finder: An algorithm for fast autonomous spectral endmember determination in hyperspectral data," in *Proc. SPIE Image Spectrom. V*, 1999, vol. 3753, pp. 266–277.

Xiong, W., Wu, C.-C., Chang, C.-I, Kapalkis, K. and Chen, H.M. (2011) "Fast algorithms to implement N-FINDR for hyperspectral endmember extraction," *IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing*, vol. 4, no. 3, pp. 545-564, September.