EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Real-time target detection in hyperspectral images based on spatial-spectral information extraction

Bing Zhang[1*], Wei Yang[1,3], Lianru Gao[1] and Dongmei Chen[2]

Abstract

Recently, real-time image data processing is a popular research area for hyperspectral remote sensing. In particular, target detection surveillance, which is an important military application of hyperspectral remote sensing, demands real-time or near real-time processing. The massive amount of hyperspectral image data seriously limits the processing speed. In this article, a strategy named spatial-spectral information extraction (SSIE) is presented to accelerate hyperspectral image processing. SSIE is composed of band selection and sample covariance matrix estimation. Band selection fully utilizes the high-spectral correlation in spectral image, while sample covariance matrix estimation fully utilizes the high-spatial correlation in remote sensing image. To overcome the inconsistent and irreproducible shortage of random distribution, we present an effective scalar method to select sample pixels. Meanwhile, we have implemented this target detection algorithm based on the SSIE strategy on the hardware of a digital signal processor (DSP). The implementation of a constrained energy minimization algorithm is composed of hardware and software architectures. The hardware architecture contains chips and peripheral interfaces, while software architecture contains a data transferring model. In the experiments, we compared the performance of hardware of DSP with that of Environment for Visualizing Images software. DSP speed up the data processing and also results in more effective in terms of recognition rate, which demonstrate that the SSIE implemented by DSP is sufficient to enable near real-time supervised target detection.

Keywords: Hyperspectral image, Target detection, Real-time processing, SSIE, DSP, CEM

1. Introduction

Hyperspectral remote sensing is a new multi-dimensional information acquisition technology combining image and spectral technologies for monitoring and detecting chemical substances, anomalies, and camouflaged objects, as well as their visual surveillance. Due to the high spectral resolution of hyperspectral remote sensing image, the amount of data that can be processed in hyperspectral image processing is high compared to that in a typical image processing. Some applications of hyperspectral images require real-time processing or near real-time processing. The computational complexity that comes with the enormous amounts of data involved limits hyperspectral image superiority. Digital signal processor (DSP) is quite suitable for hyperspectral image processing, and such a processor is optimized for performing multiply-accumulation operations. However, the capability of DSP is limited by the memory space, arithmetic logic units, and the clock speed of chip. When the architecture of DSP is fixed, an effective way to accelerate processing speed is to use or develop new processing methods such as parallel and distributed algorithms. In this article, an effective strategy named spatial-spectral information extraction (SSIE) to reduce redundant information from spatial and spectral dimensions is introduced. This strategy has improved the computing capability of DSP, and has acquired better detecting results than the algorithm in popular Environment for Visualizing Images (ENVI) software package. After using SSIE strategy, the hardware accelerator based on DSP has met the requirements of real-time processing system.

* Correspondence: zhangbing@ceode.ac.cn
[1] Key Laboratory of Digital Earth Sciences, Center for Earth Observation and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China
Full list of author information is available at the end of the article

A real-time system requires its hardware and software to perform the assigned work within the specified time while the system has the ability of detecting and reacting to internal and external errors in a controlled fashion [1]. However, in many applications of the hyperspectral image real-time processing, the processing time is often too long. There are three ways to improve the hyperspectral image processing speed.

The first way is to apply the high-performance processors or complex hardware architectures. A recent development in the area of image processing is the introduction of fully programmable graphics processing units (GPU). Tarabalka and Haavardshlom [2] have implemented the anomaly detection in hyperspectral image using multivariate normal mixture models on GPU. The experiment has shown that the GPU provides a significant speedup of the algorithm compared to the CPU implementation. The GPU enables real-time execution of the algorithm on a hyperspectral data stream with high spatial and spectral resolution, with acceptable detection performance and a significant margin on computing time [2]. Although real-time Principal Component Analysis (PCA) calculation [3], near real-time endmember extraction [4], Pixel Purity Index (PPI) [5], and Automatic Morphological Endmember Extraction algorithms [6] have been implemented on GPU, Field Programmable Gate Arrays (FPGA) or DSP is always chosen as main processor in the space missions and Unmanned Air Vehicle (UAV) instruments. Reconfigurable FPGA can include dual core PowerPC processors thereby providing a flexible hardware and software co-design architectures to meet the on-board processing challenges of these missions. Reconfigurable FPGA technologies also provide in-flight flexibility and the ability to update processing algorithms as needed post-launch [7]. Wang and Chang [8] studied an FPGA implementation of the causal constrained energy minimization (CEM) for hyperspectral target detection, and the experiment showed that the data's correlation matrix is calculated in a causal manner that only needs data samples up to the sample at the time it is processed. One of major difficulties with implementing the algorithm in real time is the computation of the inverse of a matrix. They use systolic arrays architecture in conjunction with the CORDIC algorithm to implement the computation of a matrix inverse [8]. In addition, SVM classifiers [7], PPI algorithm [5], and parallel independent component analysis algorithm [9] have also been implemented on FPGA. Some other hardware structures are also suitable for real-time processing, such as a massively parallel Beowulf cluster [5].

Another way is to develop real-time algorithms to improve the execution efficiency of processors. Wu et al. [10] re-designed N-finder algorithm (N-FINDR) in a real-time processing fashion, and four versions of real-time N-FINDR were developed. Experiment results demonstrated that the correlation and covariance matrices can be calculated and updated in a causal manner, so the real-time processing requirement was met [10]. Du and Ren [11] presented a real-time constrained linear discriminant analysis (CLDA) approach to implement target detection and classification. In fact, both of CLDA and target-constrained interference-minimized filter are based on a similar principle, and their only difference is the use of covariance and correlation matrices. The CLDA algorithm is difficult to implement in a real-time fashion, and then modified versions are presented to avoid the data whitening process. Experiment results demonstrate that the entire real-time simulation takes a Sun Ultra10 440-MHz workstation only about 50 s [11].

The last way is to ameliorate data processing strategy. An appropriate strategy can make real-time processing easy to implement. Acitoet al. [12] presented a complexity-aware algorithm architecture for real-time enhancement and detection of local anomalies. The algorithm is actually a data processing strategy. The key of the proposed architecture is an efficient procedure for updating the local background parameter estimates [12]. Parket al. [13] proposed real-time target detection architecture as an efficient solution for hyperspectral image processing. Data partitioning is the main strategy to reduce complexity algorithm for high-throughput applications [13].

In this article, a near-real-time data processing strategy named SSIE is introduced and implemented on the TI TMS320DM642 DSP. The remainder of this article is organized as follows: the following section introduces target detection algorithm and describes the near-real-time data processing strategy. Section 3 focuses on introducing the hardware implementation of SSIE. Section 4 assesses the performance of the data processing strategy by some experiments. Finally, the article ends with conclusion section.

2. Near-real-time processing strategy of target detection algorithm

2.1 Target detection algorithm

Target detection includes supervised target detection, unsupervised target detection, and change detection. In this article, we focus on the supervised target detection problem.

Spectral matched filters such as CEM are one of the most useful supervised target detection algorithms. CEM is suitable when the spectrum for targets of interest is known but the background is unknown. This concept is based on the extraction of some signals at the specific direction and the suppression of other signals. It designs a finite impulse response filter vector $\boldsymbol{w}$ so that the filter output energy is minimized, subject to a constraint that the output of desired target vector $\boldsymbol{d}$ is constant,where, $\boldsymbol{w} = \{w_1,\ w_2,\ \ldots,\ w_L\}^{\mathbf{T}}$, and $\boldsymbol{w}^{\mathbf{T}}\boldsymbol{d} = 1$,let input be $\boldsymbol{x}_i$, detection statistic be $\boldsymbol{y}_i$, then $\boldsymbol{y}_i = \boldsymbol{w}^{\mathbf{T}}\boldsymbol{x}_i = \boldsymbol{x}_i^{\mathbf{T}}\boldsymbol{w}$. So, the average output energy would then be

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} y^2 &= \frac{1}{n}\left[\sum \left(x_i^T w\right)^T x_i^T w\right] \\ &= w^T\left(\frac{1}{n}\left[\sum x_i x_i^T\right]\right)w = w^T R w \end{aligned} \tag{1}$$

where $\boldsymbol{R}$ is the self-correlation matrix of sample set, $\boldsymbol{x}_i$ is the pixel vector, and $\boldsymbol{n}$ is the number of pixels in the original hyperspectral images. Obviously, removing the mean from $\boldsymbol{X}$ will result in $\boldsymbol{R}$ replaced by covariance matrix $\sum$. So, filter algorithm is boiled down to a minimum problem:

$$\begin{cases} \min_w \left(\dfrac{1}{n}\left[\displaystyle\sum_{i=1}^{n} y_i^2\right]\right) = \min_w \left(w^T R w\right) \\ \qquad\qquad d^T w = 1 \end{cases} \tag{2}$$

Solve this problem by Langrage multiplying operator. The solution is CEM operator [14], namely

$$w* = \frac{R^{-1}d}{d^T R^{-1} d} \tag{3}$$

If we apply the CEM operator to every pixel in the image, the detection result will be

$$\begin{aligned} y = D_{CEM}(x) = w*^T x = \left(\frac{R^{-1}d}{d^T R^{-1} d}\right)^T x \\ = \frac{x^T R^{-1} d}{d^T R^{-1} d} \end{aligned} \tag{4}$$

2.2 Near-real-time data processing strategy

An effective method to improve the speed of hyperspectral image processing is to reduce the amount of redundant information. The key to target detection is calculating covariance matrix or correlation matrix, which take up majority of the whole computing amount in target detection. In this article, we introduce anSSIE strategy to fast near-real-time data processing by taking advantage of high spatial and spectral correlations in hyperspectral image.

2.2.1 Spatial and spectral correlations of hyperspectral image

Spatial correlation is the correlation from pixel-to-pixels at every spectral band, and spectral correlation is the correlation among bands at the same spatial location [15].

1. *Spatial correlation analysis*

Let $r\ (l, k)$ be correlation function between $f\ (x, y)$ and $f\ (x + l, y + k)$, where l represents the varying value in the direction of row and k represents the varying value in the direction of column, $f\ (x, y)$ be grayscale function, spatial correlation function is defined as follow

$$r(l,k) = \iint f(x,y) f(x+l, y+k)dxdy \tag{5}$$

After doing normalization and discretization for spatial correlation function, Equation (5) would then be

$$R(l,k) = \frac{\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N}\left[f(x,y) - u_f\right]\left[f(x+l, y+k) - u_f\right]}{\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N}\left[f(x,y) - u_f\right]^2} \tag{6}$$

where M and $N$are the image sizes, u_f is the average of image's grayscale, and u_f is given by

$$u_f = \frac{1}{M \times N}\sum_{x=1}^{M}\sum_{y=1}^{N} f(x,y) \tag{7}$$

If $k = 1$ and $l = 1$, $R\ (1, 1)$ is the correlation coefficient between two adjacent pixels.Since an image is a two-dimensional signal, correlation coefficient of adjacent rows or columns is valuable to demonstrate the spatial correlation in total image.So, the correlation coefficient of adjacent rows at every band can be represented as

$$\begin{aligned} R_x(1) &= R(1,0) \\ &= \frac{\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N}\left[f(x,y) - u_f\right]\left[f(x+1, y) - u_f\right]}{\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N}\left[f(x,y) - u_f\right]^2} \end{aligned} \tag{8}$$

Similarly, the correlation coefficient of adjacent columns at every band can be represented as

$$\begin{aligned} R_y(1) &= R(0,1) \\ &= \frac{\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N}\left[f(x,y) - u_f\right]\left[f(x, y+1) - u_f\right]}{\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N}\left[f(x,y) - u_f\right]^2} \end{aligned} \tag{9}$$

2. *Spectral correlation analysis*

Let $f\ (x, y)$ be grayscale function, $g\ (x, y)$ the standard image, and $h\ (l, k)$ cross-correlation function between $f\ (x, y)$ and $g\ (x, y)$. So,the spectral cross-correlation function is defined as follows:

$$h(l,k) = \iint f(x+l, y+k) g(x,y) dxdy \tag{10}$$

After doing normalization and discretization for spectral cross-correlation function, Equation (10) would then be

$$H(l,k) = \frac{\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N}\left[f(x+l, y+k) g(x,y)\right]}{\sqrt{\left(\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N}\left[f(x,y)\right]^2\right)\left(\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N}\left[g(x,y)\right]^2\right)}} \tag{11}$$

If $k = 0$ and $l = 0$, $h\ (0, 0)$ is cross-correlation coefficient.

After analyzing spatial and spectral correlations, we know that these high correlations will cause to redundant information.

2.2.2 Band selection

Band selection maps dimensional data into lower dimension while preserving the main features of the original data. For specific targets, bands known to carry little or no information should be removed from spectral analysis.

Some common methods of band selection are PCA, Isometric Feature Mapping [16], Diffusion map, Locally linear embedding [17], Local Tangent Space Alignment [18], and so on. However, these methods need complex calculation, which is not suitable for real-time processing. There are some straightforward methods of band reduction in [19]. One option is the rather primitive step of averaging two or more adjacent bands and forming a smaller number of lower spectral resolution bands subset. Actually, the selection of band subsets in this fashion cannot represent all of the spectral features of the specific target. The shape of an absorption feature or the presence of combinations of absorption features at the specific wavelength can represent a target's spectral feature. So, we developed averaging adjacent bands for band selection to join the bands representing absorption features into the band subset.

As shown in Figure 1, the new method of band selection is presented as follows:

1. Remove bands that carry little or no information about the specific target. For example, bands at atmospheric absorption and bands with low SNR should be removed. The key of band selection is spectrum analysis. The level of SNR is decided by the hyperspectral image sensor, so it is easy to know by parameters of sensor. At the same time, the band wavelengths at the atmospheric absorption are easy to know by prior parameter of sensor. These bands are preliminarily set by people.
2. Analyze the specific target's spectral absorption features and preserve the band at the minimum reflectance and several bands at the left and the right sides of the minimum. The number of bands at the left or right side of the minimum is decided by the type of targets, different targets have different spectral features and the number of spectral bands is also different. Some targets having obvious features need fewer bands, but some need more bands.
3. Select band at every step of two or more adjacent bands from the remaining bands to form a band subset.
4. Combine the bands at absorption feature and the band subset to form a new bands subset.

After band reduction, a new hyperspectral image is generated to represent the major information of original data. In Section 2.3, we will analyze the change of computing complexity, and in Section 4.2.1 we will assess the actual effect of band selection.

2.2.3 Sample covariance matrix estimation

2.2.3.1 Small sample covariance matrix estimation

Covariance matrix computing is the most time-consuming component in target detection, so a simple and effective computing method is very important for real-time processing. Remote sensing image usually has high spatial correlation, and a small number of pixels can still capture the major statistics [20]. So, using a portion of pixels for correlation matrix $\boldsymbol{R}$ or covariance matrix $\sum$ calculation is a commendable method to reduce computation complexity.

From the previous spatial correlation analysis, we have found that a pixel has a similar statistical distribution to the pixels in its neighborhood. Furthermore, this similarity is determined by spatial resolution and homogeneous background in hyperspectral image. So, we can select a portion of pixels as a sample to represent total pixels to compute the statistics. The spatial resolution is higher, and the number of samples can be smaller. If the number of pixels in the original data is $\boldsymbol{N}$ and the number of bands is $\boldsymbol{L}$, the minimum number of pixels which are used to estimate the covariance matrix is $\boldsymbol{L}$. If the number of pixels in the sample is less than the number of bands, $\mathbf{w}^*$ in Equation (3) can be affected by the so-called singularity problem.

2.2.3.2 The singular problem of small sample inverse covariance matrix estimation

Singularity problem is common in the problem of hyperspectral classification. Sufficient training pixels for each spectral class must be available to allow reasonable estimates to be obtained of the elements of the class conditional mean vector and covariance matrix. For an $\boldsymbol{L}$-dimensional hyperspectral space, the covariance matrix is symmetric of size $\boldsymbol{L}\times\boldsymbol{L}$, the minimum number of independent training pixels required is $(\boldsymbol{L}+\mathbf{1})$[21]. Swain and Davis [22] recommend as a practical minimum that $10\boldsymbol{L}$ training pixels per spectral class be used, with as many as $100\boldsymbol{L}$ per class if possible.

In the target detection problem, we try to reduce the computation of covariance matrix using a portion of pixels as a sample to estimate background statistics. However, this method will bring the same problem as hyperspectral classification. Fortunately, it is different with classification that inverse covariance matrix in target detection will weaken the main information of background without targets. So, it is an effective method to avoid singularity matrix that the mean component in

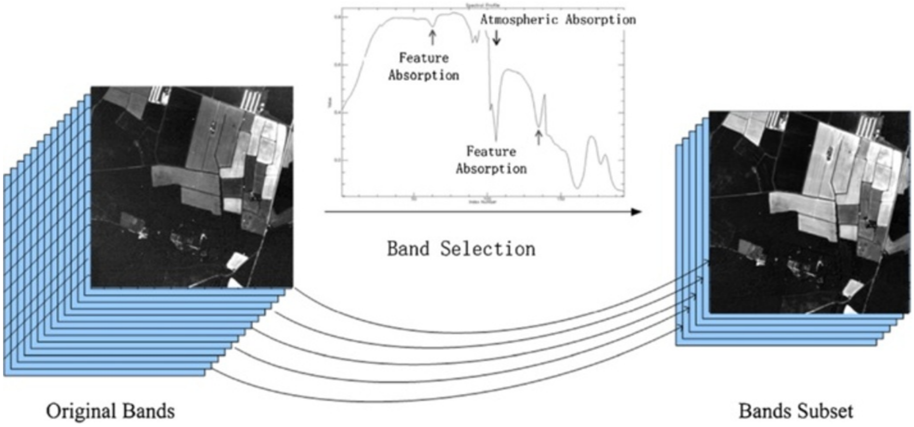


Figure 1 **Illustration of the method of band selection.**

every band is added in the covariance matrix. The estimated covariance matrix $\sum_M$ is shown in Equation (12)

$$\sum\nolimits_M = \sum + m\bar{I} \tag{12}$$

where $\boldsymbol{I}$ is a scaled identity matrix whose size is $\boldsymbol{L} \times \boldsymbol{L}$, and the whole of bands' mean vector $\boldsymbol{m} = \{m_1, m_2, m_3, \ldots, m_L\}$, especially,

$$m_j = \frac{1}{N}\sum_{i=1}^{N} x_{i,j}, (j = 1, 2, \ldots, L) \tag{13}$$

where $\boldsymbol{N}$ is the number of pixels in the original image, and $\boldsymbol{L}$ is the number of bands.

Similarly, the estimated correlation matrix $R_M = R + m\bar{I}$, so, Equation (4) would then be

$$y_{CEM}(x) = \frac{x^T R_M{}^{-1} d}{d^T R_M{}^{-1} d} = \frac{x^T (R + m\bar{I})^{-1} d}{d^T (R + m\bar{I})^{-1} d} \tag{14}$$

2.2.3.3 The method of selecting pixels to form the sample

It is difficult to make a rule for selecting pixels for the estimation of population covariance matrix. In general, these pixels should include the main information in the image. We present an effective method to extract these pixels in order to form a representative sample. The image shown in Figure 2 has 216 × 216 pixels and 193 bands. The location of pixels selected to form a sample is shown in Figure 3.

Different objects are located at different locations in an image, and the probability of difference is larger when pixels are farther apart. Pixels should try to represent the information at all locations. So, we put all the pixel vectors into a column, and select a pixel vector after several pixels apart to form a sample set. The distribution of location for every pixel is shown in Figure 3a. Although this sample set could represent all of the information in the original image, the method will become invalid when the length of adjacent pixel's interval in the sample set is several times the number of pixels at every column. Also, it is not a good choice when the sample size is small. So, we try to utilize the spectral information to increase the veracity of the sample.

Owning to a hypothesis that statistics used to compute covariance matrix or correlation matrix in target detection algorithms cannot include target information [23], we try to search for several pixels that is not from targets but form different geographic objects in the original image.

A matched filter is an L-dimensional vector $\boldsymbol{d}$ that is applied to a pixel vector $\boldsymbol{X}$ to produce a scalar $\boldsymbol{d}^T\boldsymbol{X}$ which emphasizes the signal while suppressing the background clutter [24]. Then we will make the scalar $\boldsymbol{d}^T\boldsymbol{X}$

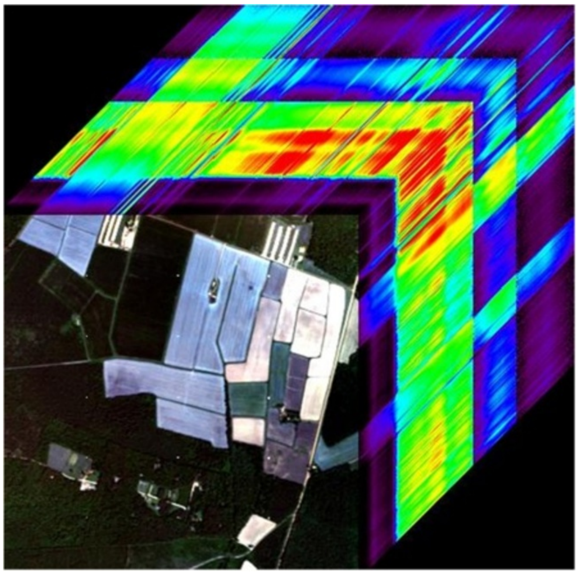


Figure 2 **The hyperspectral remote sensing image used in this study.**

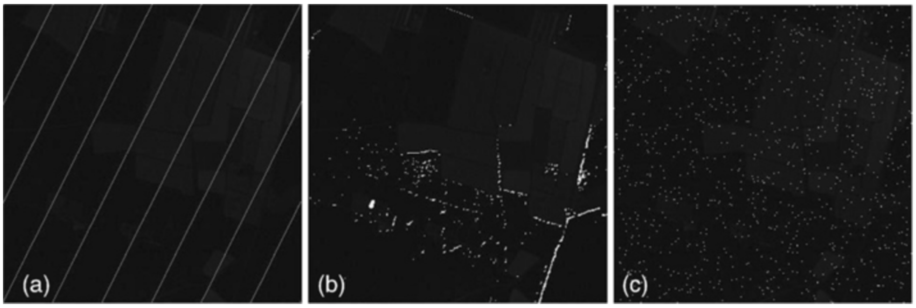


Figure 3 Location of pixels selected to compute covariance matrix.

as the standard of selecting pixels. If $\boldsymbol{d}^T X$is small, the difference between the pixel vector $\boldsymbol{X}$ and the target vector $\boldsymbol{d}$ is large. Figure 3b shows the location for $5\boldsymbol{L}$ pixels that make $\boldsymbol{d}^T\boldsymbol{X}$ larger. It is clearly shown that these pixels were mainly selected from similar objects. If the background is homogeneous, the sample will only represent one or two objects. So, it is difficult to represent the population's statistics.

After sorting the pixels by the value of $\boldsymbol{d}^T\boldsymbol{X}$ in descending order, we can select a pixel after several pixels apart to form a sample set. The distribution of location for every pixel is shown in Figure 3c. It is shown that these pixels are irregularly distributed in the whole of image. Although it is like the random distribution whose location is unpredictable, these locations are decided by the number of pixels in the sample. This rule can ensure that the sample would represent the total image's statistics.

After discussing the rule of selecting pixels, the calculating workflow of sample covariance matrix is shown in Figure 4.

The process flow of sample covariance matrix calculation is presented as follows:

1. Determine the number of pixels in the sample, which is then used to compute the population's covariance matrix by the requirement for recognition rate. Generally, higher number of pixels selected to compute the covariance matrix will result in a higher recognition rate. Ten times number of bands is an appropriate number of pixels.
2. Select pixels as the sample from the original data on the basis of pre-designed rule. After sorting the pixels by the value of $d^T X$ in descending order, we can select a pixel after several pixels apart to form a sample set. The rule can select pixels that can represent the statistics of population.
3. Compute the covariance matrixes in every pixel vector and accumulate these covariance matrixes. The result of this accumulation is the estimated covariance matrix for all pixels. Note that a covariance matrix is the symmetrical matrix, so there is only a triangle matrix to be calculated for very pixel vector.
4. Do a normalization and inverse operation for the estimated covariance matrix.

2.3 Assessment of the fast processing strategy

Figure 5 shows the whole workflow of target detection algorithm. The key accelerating components are band selection and sample covariance matrix calculation. We will assess the calculated amount after using fast processing strategy, so it will be clearly found that the strategy reduces the computing complexity.

Let $\boldsymbol{L}$ be the number of bands in original data, and $\boldsymbol{N}$ be the number of pixels, so the number of multiplication is $\boldsymbol{NL^2}$. After band selection, $\boldsymbol{\alpha L}$ is the number of bands in the band subset, where $\boldsymbol{0} \le \boldsymbol{\alpha} \le \boldsymbol{1}$, and the number of multiplication is $(\boldsymbol{\alpha L})^2\boldsymbol{N}$. The number of multiplication is decreased by $(\mathbf{1} - \boldsymbol{\alpha}^2)\boldsymbol{NL}^2$.

The number of multiplication is $(\boldsymbol{\alpha L})\boldsymbol{N}$ by the pre-designed rule for selecting pixels. And, after pixel selection, the number of pixels which are used to compute sample covariance matrix is $\boldsymbol{\beta L}$, where $\mathbf{1} \le \boldsymbol{\beta} \le (\boldsymbol{N/L})$, so the number of multiplication will be $\boldsymbol{\beta\alpha^2L^3}$. The number of multiplication reduced is $(\boldsymbol{N} - \boldsymbol{\beta\alpha^2L})\boldsymbol{L}^2 - (\boldsymbol{\alpha L})\ \boldsymbol{N}$. Let speed-up ratio be $\boldsymbol{a}$, then $\boldsymbol{a} = \boldsymbol{NL}/(\boldsymbol{\beta\alpha^2L^2} + \boldsymbol{\alpha N})$, where $\boldsymbol{N} \gg \boldsymbol{L}$, and $\mathbf{0} \le \boldsymbol{\beta\alpha^2} \le \mathbf{1}$, so $\boldsymbol{a}$ will be a very large number.

Multiplication is time-consuming for DSP, so the SSIE strategy will reach a high-speed-up ratio compared with no processing strategy.

3. DSP implementation

3.1 Hardware architecture

Our experiments are conducted on the TMS320DM642 DSP, which is a high-performance digital media processor.

The TMS320DM642 device is based on the second-generation high-performance, advanced velocity very-long-instruction-word architecture applications [25].

Figure 6 shows the workflow of hardware system. The computer sends hyperspectral image data to the DSP by Ethernet interface, and DSP runs the algorithm program

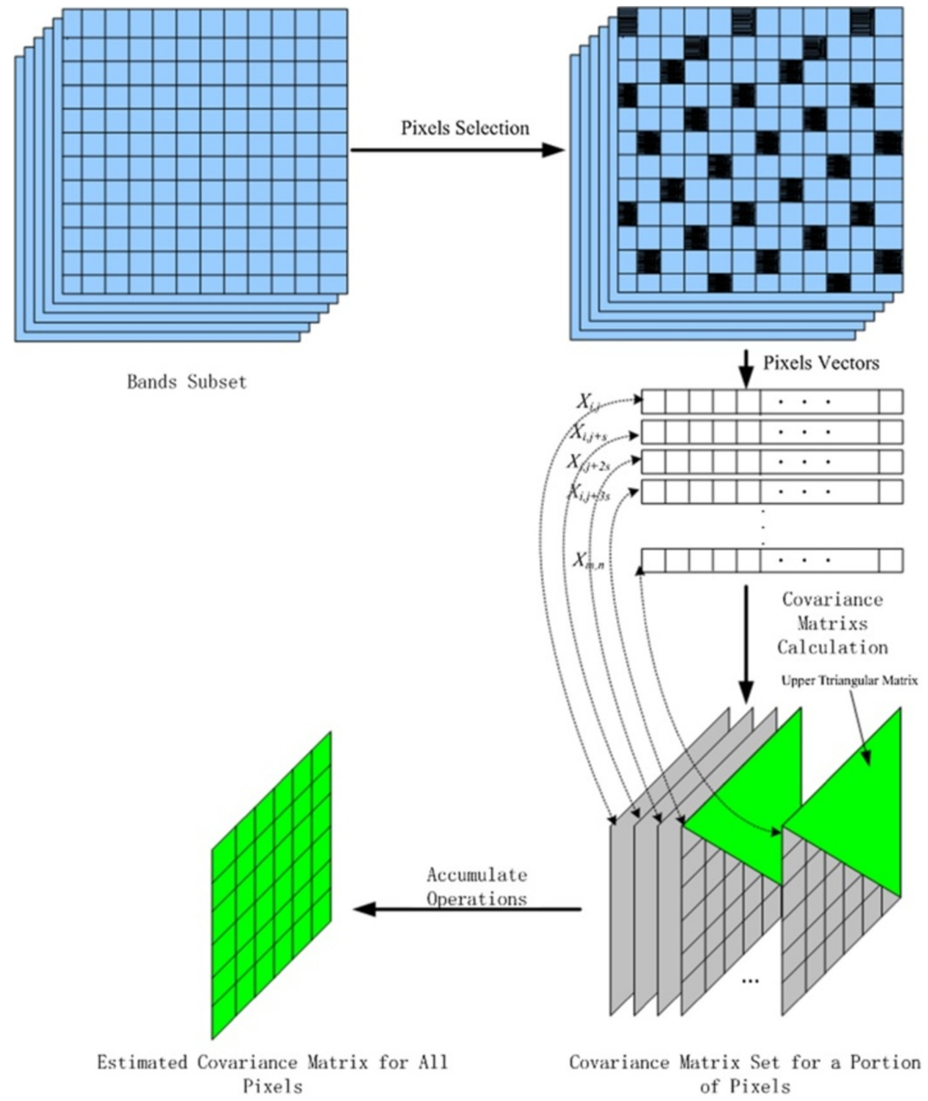


Figure 4 Illustration of the method of covariance matrix calculation.

to process image data. Then, DSP sends the processed target image to the computer by Ethernet interface. Data stream is continually exchanged by Ethernet interface whose transmitting speed can reach 100 Mbps, so the real-time processing is easy to implement on this platform.

3.2 Software architecture

To implement the communication between PC and DSP, we use of Network Developer's Kit (NDK) to establish TCP/IP stack on DSP [26]. For designing the software architecture, we aim at establishing a fast, simple channel to transmit data between PC and DSP. In this system, PC as a server performs several functions, such as initializing socket environment, reading local document and sending it by Ethernet, and receiving and saving the processed target image. Meanwhile, DSP as a client performs such functions as initializing the DM642 chip, configuring, and initializing the network environment, receiving and sending data, and executing target detection algorithm. Figure 7 shows the structure of the software system.

3.3 Implementation of fast processing strategy on DSP

As shown in Figure 5, the key components of the whole processing strategy are band selection and sample covariance matrix estimation. Band selection is implemented on PC, where a server is responsible for selecting feature bands. The server is operated by user, so it is convenient to decide which bands are preserved. Meanwhile, covariance matrix estimation is implemented on DSP which is the client of NDK. First, pixels are sampled from the image on server by the pre-designed rule, and the sample set is sent

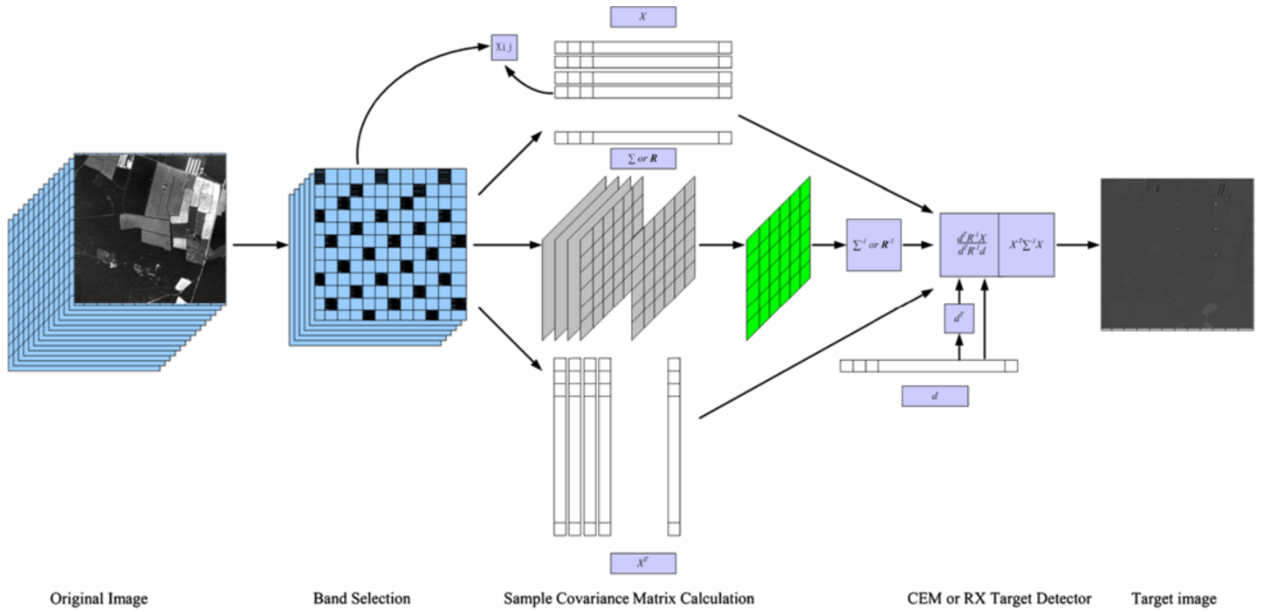


Figure 5 Illustration of the workflow of complete target detection algorithm.

to DSP to compute sample covariance matrix. Then the original data are sent to DSP to execute target algorithm. Finally, the result of target detection is sent to PC from DSP. For the hyperspectral image with BSQ format, a socket transfers one band data by the TCP/IP protocol.

In the following experiments, ENVI is run on a PC platform with an Intel Pentium 1.6 GHz processor and 256 MB of RAM. An algorithm with the strategy of SSIE is run on the platform of DSP which is a 550-MHz processor with 32 MB of SDRAM.

4. Experimental results

4.1 Hyperspectral data

In order to compare the CEM algorithm's performance on the hardware of DSP with that in the software of ENVI 4.6 version, a hyperspectral image has been selected for experiment. To evaluate the results of target detection well, we use a synthetic hyperspectral images. Figure 2 shows the hyperspectral image that is used in the subsequent experiments. The scene is a 193-band image with 320 × 320 pixels and was collecting over a farmland. As shown in Figure 8a, 15 simulated targets are embedded in the image. We also selected a region with 12 simulative targets and with 216 × 216 pixels in Figure 8b to assess the performance of the previous theory.

As shown in Figure 9a, the first column is five targets that each one was made of 1 pixel, similarly, the second column is 1 × 2 pixels, and the third column is 2 × 2 pixels. The first row is three targets made of pure pixels, but the second row is three targets made of 80% target spectral signal (labeled "S") and 20% background spectral signal (labeled "BKG"), similarly, the rest of rows are targets

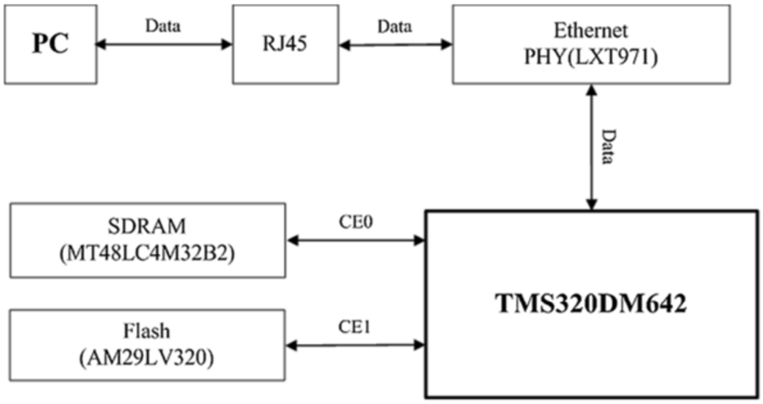


Figure 6 Illustration of the workflow of hardware system.

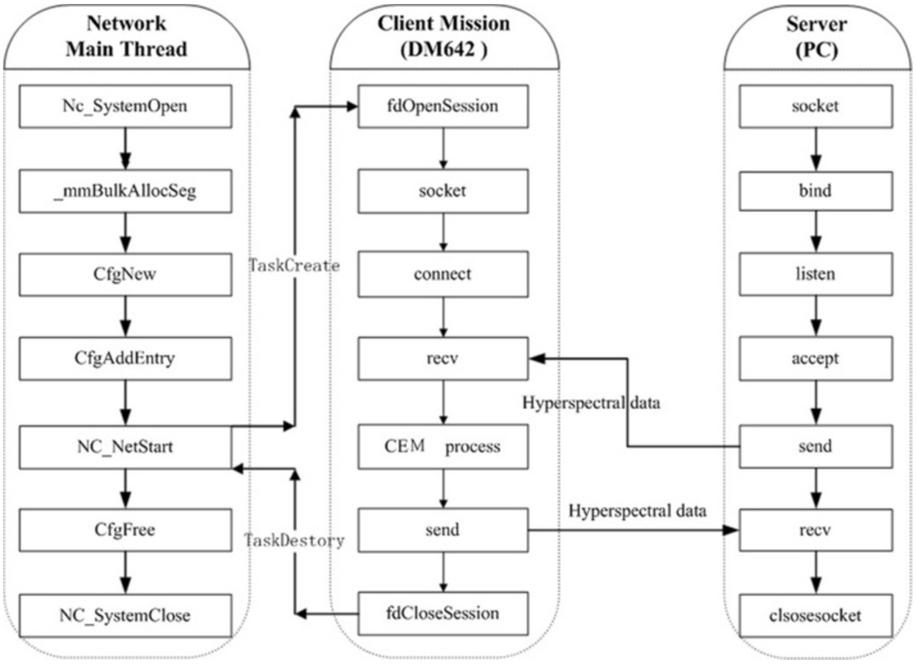


Figure 7 **Illustration of the workflow of software system.**

made of different fraction of target and background spectral signal. The section Additional file 1: Table S1 lists the location of simulative targets,where $\boldsymbol{P}_{i,j}^{k}$ is the pixel value, and $\boldsymbol{k}$ is the label of target, $\boldsymbol{i}$ is the location in the column while $\boldsymbol{j}$ is the location in the row of original image. Figure 9b is the simulative target's spectrum. The feature bands subsets are labeled with red dashed ellipse on the spectrum.

We will analyze the correlation of original data based on the previous theory in Section 2.2.1. The correlation analyzing results is showed in Figure 10 as follows.

As shown in Figure 10a, the minimum correlation coefficient of adjacent rows in every band is 0.8166 and the maximum is 0.9802, while the minimum correlation coefficient of adjacent columns in every band is 0.8208 and the maximum is 0.9779. This indicates that the pixels in the hyperspectral image are highly correlated to their neighboring pixels in both X and Y directions.

Correspondingly, as shown in Figure 10b, only 11 pairs of adjacent bands have a correlation between 0.86 and 0.98, and the correlation of the others are close to 1.00. Obviously, the hyperspectral image has

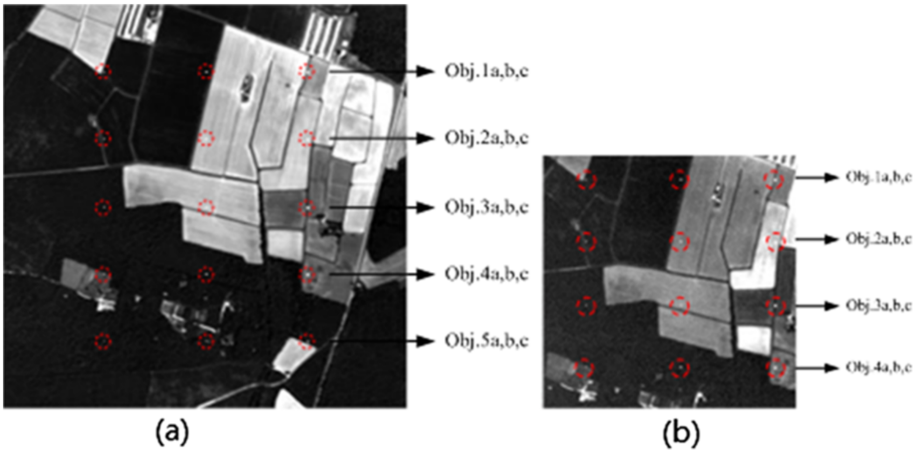


Figure 8 **Original hyperspectral image scene.** **(a)** A hyperspectral image scene containing 15 simulative targets, **(b)** target area containing 12 simulative targets.

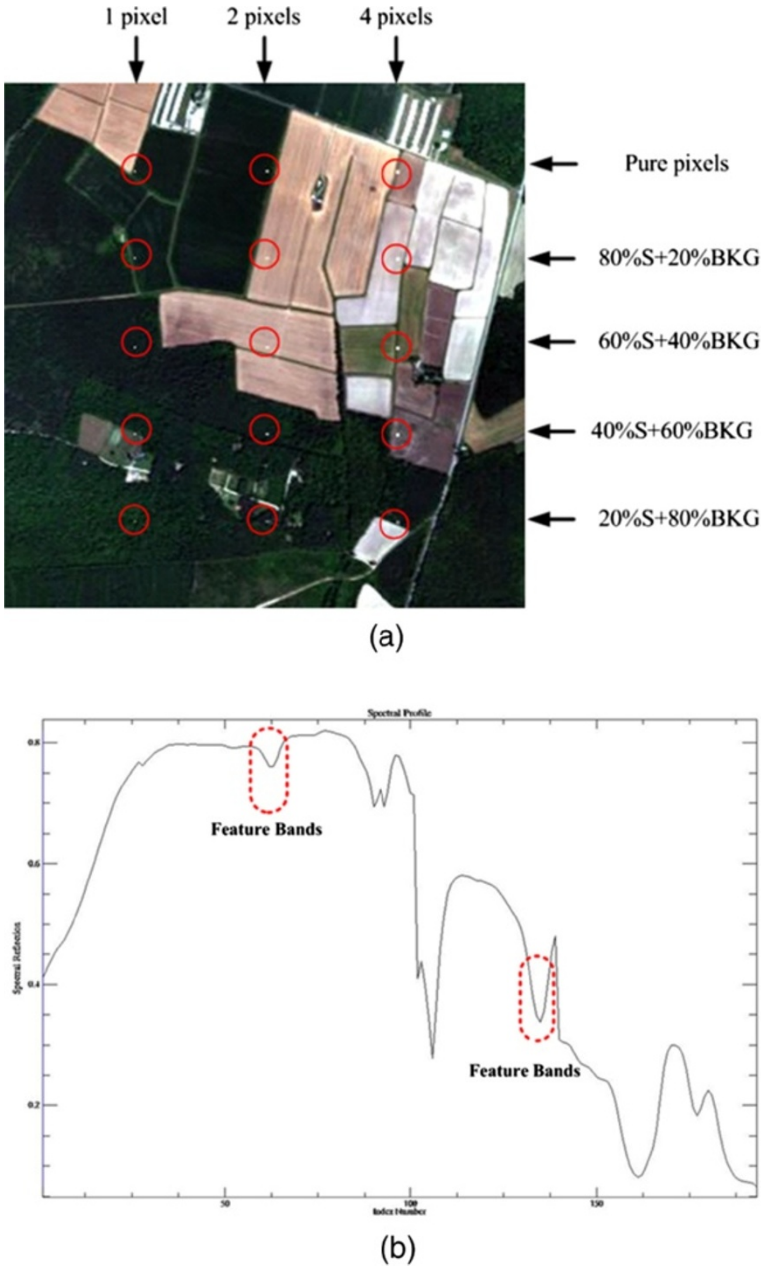


Figure 9 Hyperspectral image scene embedded simulated targets. (a) The synthetic hyperspectral image scene, **(b)** the simulated target's spectrum.

a strong spectral correlation, which indicates that the remarkable redundant information exist in the hyperspectral image.

Band selection and sample covariance matrix estimation are presented in this article to carry out spatial and spectral redundancy information reduction.

4.2 Target detection based on CEM algorithm

4.2.1 Performance of band selection

The performance of band selection method with feature absorption described in Section 2.2.2 is assessed in the following work. In Figure 8, the target region containing 12 simulative targets as shown in Figure 8b is processed with CEM algorithm on CPU-based ENVI software and DSP hardware platform. After bands selection, bands of hyperspectral image are reduced from 193 to 42, where the bands at feature absorption have been preserved and the bands at atmosphere absorption have been removed.

As shown in Figure 11, though targets are all recognized, the background is better suppressed and noise is reduced after preserving feature absorption bands on DSP. Compared with two-dimensional image, three-dimensional is

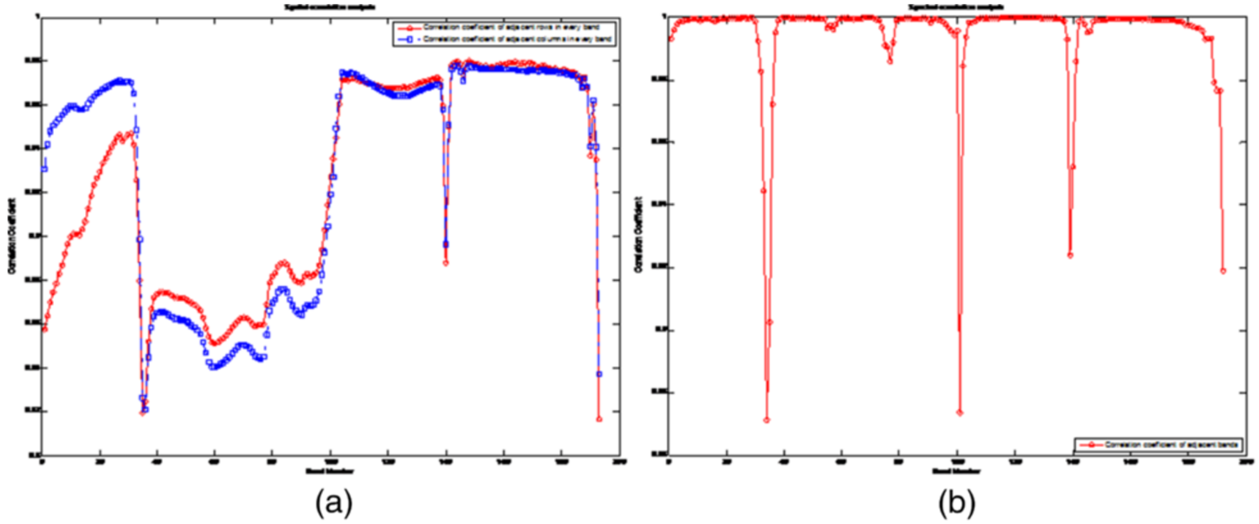


Figure 10 Spatial and spectral correlation analysis results. (a) Spatial correlation analysis, **(b)** Spectral correlation analysis.

more clearly shown that targets are easy to be distinguished from background and the background is suppressed well in the image. Calculated amount always increases with the bands' number in a forth power relation [27]. So, it has significantly improved processing speed after bands selection.

4.2.2 Performance of sample correlation matrix estimation on a DSP platform

In this section, the method proposed in Section 2.2.3 is verified. The original data are the target region containing 12 simulative targets and 42 bands, as shown in Figure 8b. We will compare the results on the condition of the different number of pixels used to estimate correlation matrix. In particular, we use correlation matrix instead of covariance matrix to whiten the data. So, the following experiments focus on sample correlation estimation.

As shown in Figure 12, the correlation matrix is sensitive when the number of sample pixels is less than 10 times of the number of bands $\boldsymbol{L}$ (Indexes III and IV). When the number is larger than 10 $\boldsymbol{L}$ (Indexes I and II in Figure 12), detection results have no obvious difference.

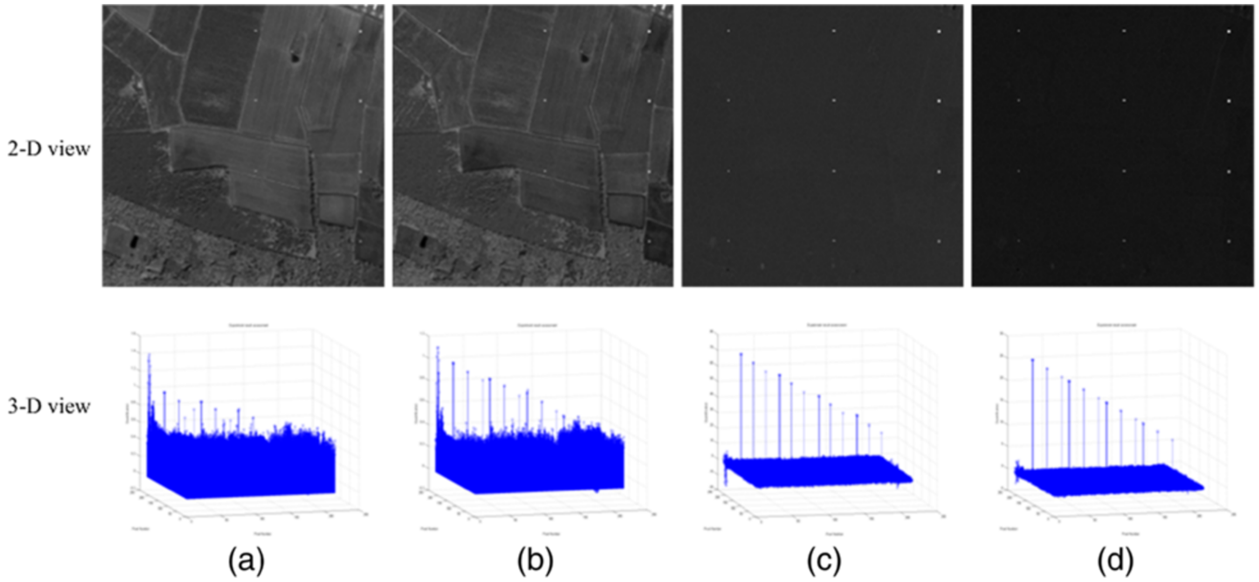


Figure 11 Analysis of band selection with feature absorption. (a) The processed result without bands selection on ENVI software, **(b)** the processed result after bands selection on ENVI software, **(c)** the processed result without bands selection on DSP platform, **(d)** the processed result after bands selection on DSP platform.

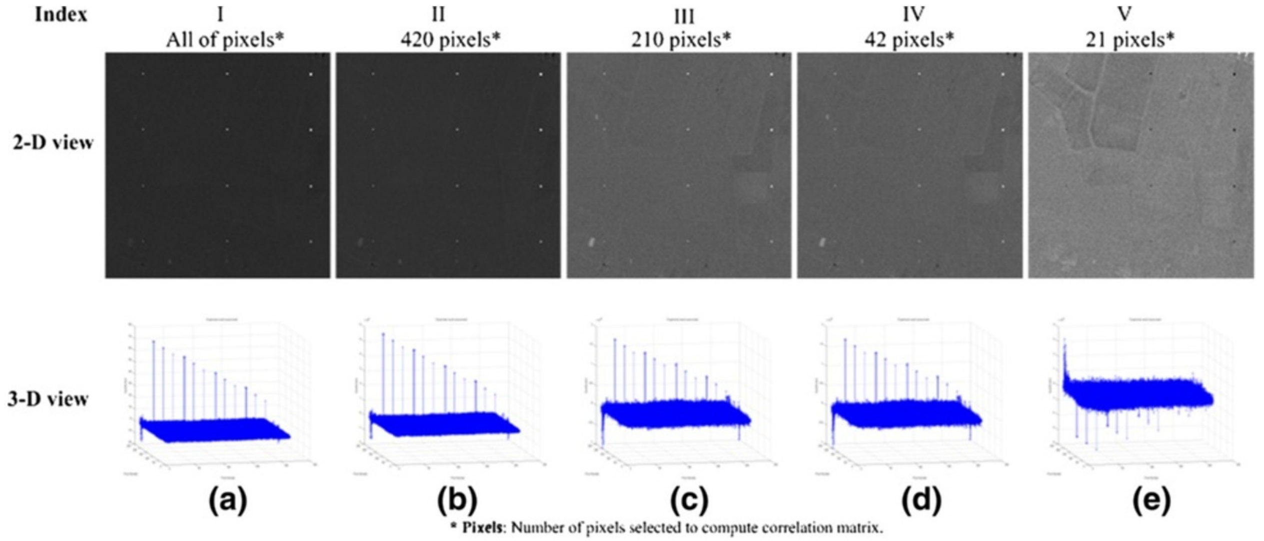


Figure 12 **Analysis of correlation matrix estimation.**

When the number of sample pixels is less than L, the background will flood targets, owing to the fact that sample correlation matrix is not a full-rank matrix (Indexes V in Figure 12). In Section 2.2.3.2, a method to solve the singular problem is provided, and the experiment result is shown in Figure 13.

As shown in Figures 12 and 13, we can see that the fewer the number of sample pixels, the more noise the background has. Correspondingly, processing time is obviously reduced after executing sample correlation matrix estimation. The results are shown in Table 1.

4.2.3 Performance comparison between ENVI processing and DSP processing

In this section, comparison of processing results between ENVI and DSP is presented. The original data

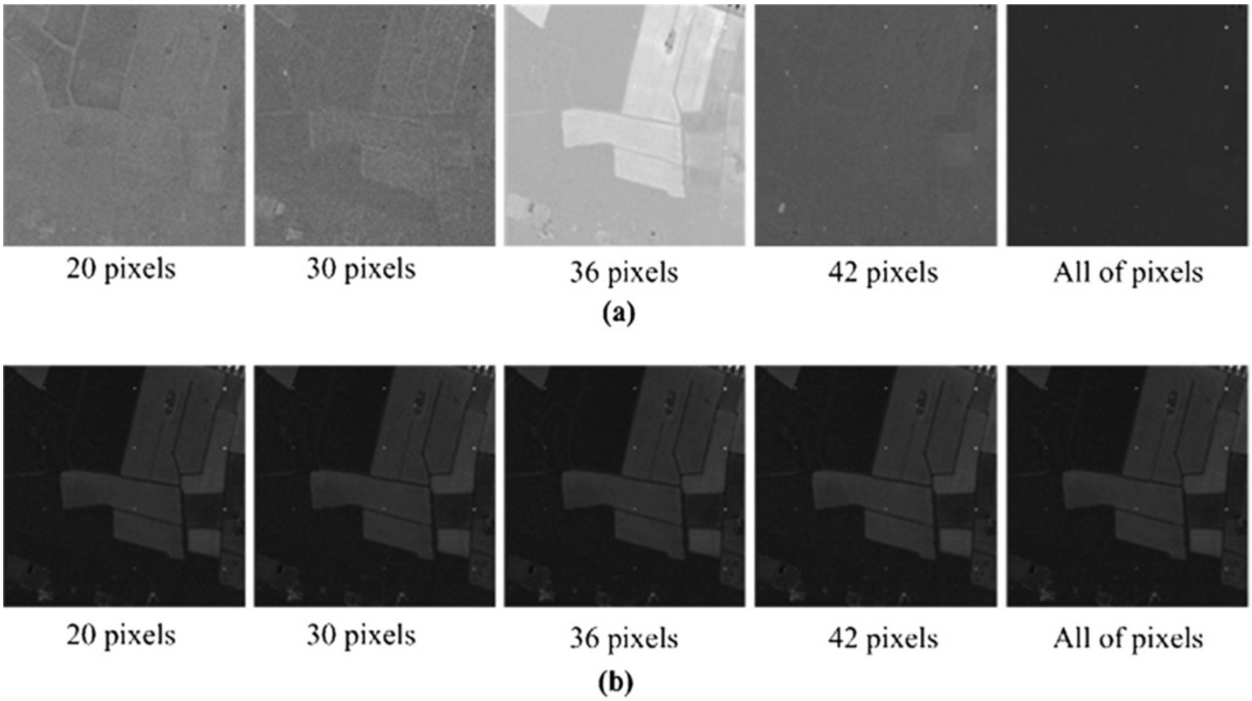


Figure 13 **The solution for singular problem of small sample inverse covariance matrix estimation.** **(a)** Covariance matrix is singular, **(b)** a method making covariance matrix nonsingular.

Table 1 Execution time on the platform of DSP

Index	I	II	III	IV	V
Pixels	46656	420	210	42	21
Processing time (s)	14.721	1.602	1.532	1.512	1.502

containing 15 simulative targets and 193 bands are shown in Figure 8a.

As shown in Figure 14, ENVI's CEM implementation could only recognize 12 targets (*obj. 5a, b, c* cannot be discriminated from background) and targets are heavily disturbed by the background. Some ground objects such as two strip objects in the top right of image are incorrectly detected.

Though details of the specific steps to implement ENVI's CEM are not available, it is known that target detection algorithm is implemented for all bands and all pixels in a hyperspectral image.

After extracting SSIE strategy on DSP, it is clearly shown that background is obviously suppressed and noise is remarkably reduced in Figure 14. However, as the number of pixels selected to calculate correlation matrix decreases, the impact from noise cannot be ignored.

For illustrative purposes, a comparison of our proposed DSP design with the algorithm on ENVI has been performed in terms of computation time and recognition rate. The execution time is shown in Table 2.

Table 2 Execution time comparison between ENVI and DSP

Platform	ENVI	DSP	DSP
Pixels	102400	102400	100
Bands	193	20	20
Processing time (s)	183.2	9.694	2.023
Recognition rate (%)	80.0	93.3	93.3

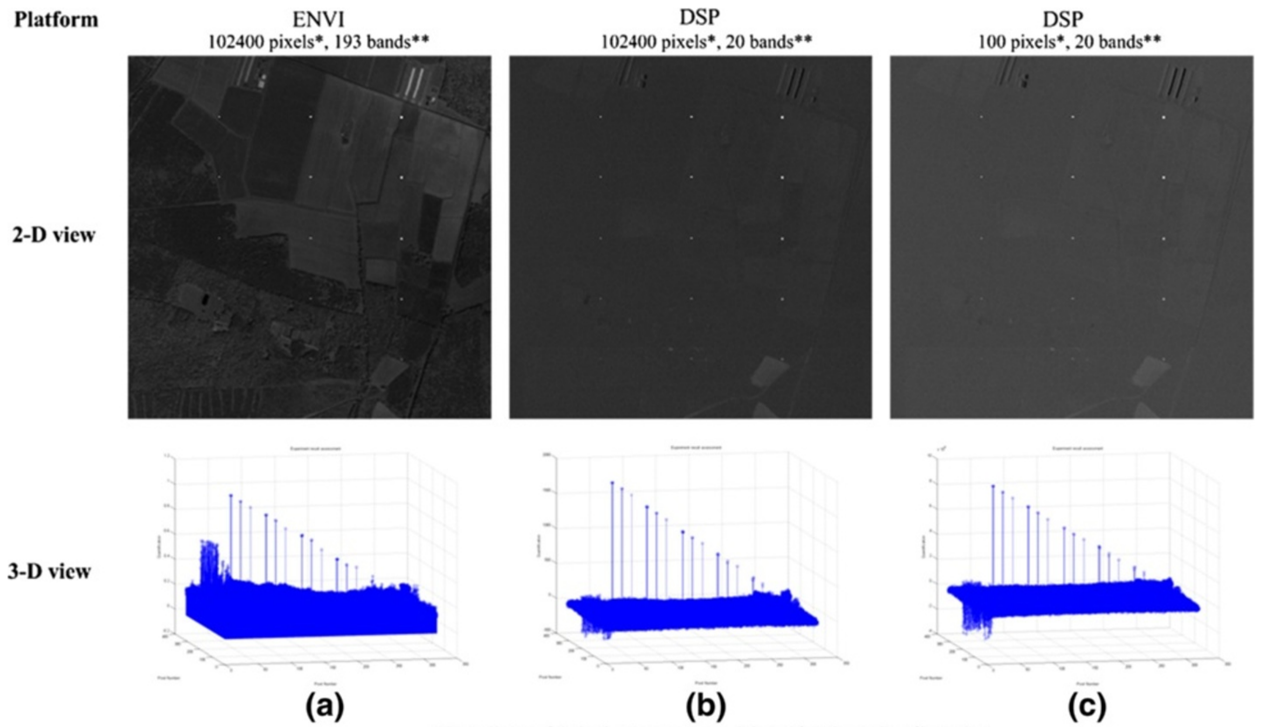
After executing the band selection on DSP, Table 2 shows that the speedup rate is 18.89 compared with processing on ENVI. Due to the removal of some bands with low SNR and the preservation of some bands at absorption features, 14 targets are recognized (*obj. 5a* cannot be discriminated from background), so the recognition rate is improved accordingly from 80.0 to 93.3%.

After executing the covariance matrix estimation, the speedup rate is 90.56 when the number of pixels calculating matrix is 100, and 14 targets are still recognized. The three-dimensional views show that *obj.5a, b, c* can be discriminated from noise in the background.

Compared with all pixels and all bands calculating the matrix in ENVI, the SSIE strategy implemented on DSP significantly improved the processing time.

4.2.4 Assessment of experimental results

This section provides an assessment of previous experimental results. The CEM algorithm in ENVI software is



* **Pixels**: Number of pixels selected to compute correlation matrix.** **Bands**: Size of band subset.

Figure 14 Analysis of SSIE strategy.

time-consuming, and the background has a significant effect on targets. After using SSIE strategy, the processing time is reduced and recognition rate is improved on DSP.

ENVI is run on the CPU, which has a higher clock speed and larger memory than the DSP. However, comparing with CEM in ENVI package, the CEM algorithm using SSIE strategy on DSP reaches a high-speed-up rate. In addition, the processing results on DSP are superior to that on ENVI, because the background is suppressed better and targets are easier to be discriminated.

5. Conclusions

In this article, we present a real-time DSP system implementation for hyperspectral target detection. This system is suitable for UAV or satellite platform, which require data automatically processed.

The implementation involves the SSIE strategy and the hardware architecture of DSP. The fundamental purpose of SSIE is to reduce the redundant information and improve the SNR of original hyperspectral data. Band selection preserving feature absorption takes advantage of the high-spectral correlation in spectral image to reduce the redundant information of spectral dimension, while sample covariance matrix estimation takes advantage of the high-spatial correlation in remote sensing image to reduce the redundant information of spatial dimension.

For sample covariance matrix estimation, we present a method that making the scalar $\boldsymbol{d}^T\boldsymbol{X}$ as the standard of selecting pixels to form a consisted and reproducible sample set. The SSIE strategy decreases the image data and significantly improves the speed of target detection algorithm.

We have used TI TMS320DM642 DSP to implement the CEM algorithm and SSIE strategy, and compared its performance to a CPU-based implementation. After using the SSIE strategy, the DSP provides a significant speedup of CEM algorithm compared to CPU implementation. The speed-up rate reaches 90.56 on DSP. In addition, CEM algorithm using SSIE strategy on DSP is more effective to recognize targets than that algorithm in ENVI running on a computer. Fourteen targets in 15 simulative targets are recognized on DSP, but only 12 targets are recognized in the same image on ENVI.

To solve the singular problem of small sample inverse covariance matrix estimation, a method of adding band mean value to covariance matrix is provided. Though it can solve the singular problem, the background information reduced the detection accuracy. This is a valuable issue in the future direction.

Finally, it can be noted that target algorithms based on SSIE strategy can be used in many platforms such as GPU, FPGA, and ASIC. The SSIE strategy reduces the computational complexity to ensure real-time processing of hyperspectral images, while the DSP platform implements data processing automatically on the unmanned device.

Additional file

Additional file 1: Table S1. Mixed panel pixels for simulative targets.

Abbreviations

CEM: constrained energy minimization; DSP: digital signal processor; ENVI: environment for visualizing images software; GPU: graphics processing units; NDK: network developer's kit; PCA: principal component analysis; SNR: signal-to-noise ratio; SSIE: spatial-spectral information extraction.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work has been supported by the National Basic Research Program of China (973) under Grant 2009CB723902, by the National High Technology Research and Development Program of China (863) under Grant 2011AA120203, and by the National Natural Science Foundation of China (40901225). The authors would like to express their gratitude to the anonymous reviewers for their detailed comments that improved both the editorial and technical quality of this article substantially.

Author details

[1]Key Laboratory of Digital Earth Sciences, Center for Earth Observation and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China. [2]Department of Geography, Queen's University, Kingston, ON K7L3N6, Canada. [3]Graduate University of Chinese Academy of Sciences, Beijing 100049, China.

Received: 23 November 2011 Accepted: 27 June 2012
Published: 13 July 2012

References

1. RB Gomz, AJ Lewis, On-board processing for spectral remote sensing, in *ISPRS Special Session Future Intelligent Earth Observing Satellites (FIEOS'02)*, (Denver, Colo, USA, 2002)
2. Y Tarabalka, TV Haavardshlom, Real-time anomaly detection in hyperspectral images using multivariate normal mixture models and GPU processing. J. Real-Time Image Process **4**, 287–230 (2009)
3. R Josth, J Antikainen, J Havel, A Herout, P Zemcik, M Hauta-Kasari, Real-time PCA calculation for spectral imaging(using SIMD and GP-GPU). J. Real-Time Image Process **7**, 95–103 (2012)
4. S Sanchez, G Martin, A Paz, A Plaza, J Plaza, Near real-time endmember extraction from remotely sensed hyperspectral data NVidia GPUs, Paper presented at the Real-time Image and Video Processing 2010, Brussels, Belgium, April 16, 2010
5. A Plaza, J Plaza, H Vegas, Improving the performance of hyperspectral image and signal processing algorithms using parallel, distributed and specialized hardware-based systems. J. Signal Process Syst. **61**(3), 293–315 (2010)
6. J Setoain, M Prieto, C Tenllado, F Tirado, Real-time onboard hyperspectral image processing using programmable graphics hardware, in *High Performance Computing in Remote sensing*, ed. by A.J. Plaza, C.-I. Chang (Chapman&Hall/CRC, New York, 2008), p. 411
7. PJ Pingree, Advancing NASA's on-board processing capabilities with reconfigurable FPGA technologies, in *Aerospace Technologies Advancements*, ed. by T.T. Arif (Intech, India, 2010), p. 69
8. J Wang, C-I Chang, FPGA design for real-time implementation of constrained energy minimization for hyperspectral target detection, in *High Performance Computing in Remote sensing*, ed. by A.J. Plaza, C.-I. Chang (Chapman&Hall/CRC, New York, 2008), p. 379
9. H Du, H Qi, A reconfigurable FPGA system for parallel independent component analysis. EURASIP J Embed Syst **2006**, 1–12 (2006)
10. C-C Wu, H-M Chen, C-I Chang, Real-time N-finder processing algorithms for hyperspectral imagery. J. Real-Time Image Process. **7**(2), 105–129 (2010)

11. Q Du, H Ren, Real-time constrained linear discriminant analysis to target detection and classification in hyperspectral imagery. Pattern Recognit. **36**(1), 1–12 (2003)
12. N Acito, S Matteoli, M Diani, G Corsin, N Acito, S Matteoli, M Diani, G Corsini, Complexity-aware algorithm architecture for real-time enhancement of local anomalies in hyperspectral images. J. Real-Time Image Process., 1–16 (2011). Article in Press
13. K-S Park, SH Cho, S Hong, W.-D. Cho, Real-time target detection architecture based on reduced complexity hyperspectral processing. EURASIP J. Adv. Signal Process **2008**(Article ID 438051), 1–14 (2008)
14. JC Harisanyi, *Detection and classification of subpixel spectral signatures in hyperspectral image sequences* (PhD Dissertation, University of Maryland, Baltimore County, 1993)
15. L Heng-shu, P Feng-hua, H Lian-qing, Feature analysis hyperspectral images. Opt Precis Eng **6**, 392–395 (2001)
16. MC Bachmann, TL Ainsworth, RA Fusina, Improved manifold coordinate representations of large-scale hyperspectral scenes. IEEE Trans Geosci Remote Sens **44**(10), 2786–2802 (2006)
17. DH Kim, LH Finkel, Hyperspectral image processing using locally linear embedding, in*Proceedings of the 1st International IEEE EMBSon Neural Engineering*. Capri Island, Italy **316–319**, 20–22 (2003)
18. Z Zhang, H Zha, Principal manifolds and nonlinear dimension reduction via local tangent space alignment. SIAM J Sci Comput **26**, 313–338 (2005)
19. JR Schott, *Remote Sensing: The Image Chain Approach*, 2nd edn. (Oxford University Press, New York, 2007), pp. 416–420
20. Q Du, R Nekovei, Fast real-time onboard processing of hyperspectral imagery for detection and classification. J. Real-Time Image Process. **4**(3), 273–286 (2009)
21. JA Richards, X Jia, *Remote Sensing Digital Image Analysis*, 4th edn. (Springer, Berlin, 2006), p. 199
22. PH Swain, SM Davis (eds.), *Remote Sensing: The Quantitative Approach* (McGraw-Hill, NY, 1978)
23. B Zhang, L Gao, *Hyperspectral Image Classification and Target Detection* (Science Press, Beijing, 2011), p. 252
24. J Theiler, BR Foy, Effect of signal contamination in matched-filter detection of signal on a cluttered background. IEEE Geosci Remote Sens Lett **3**, 98–102 (2006)
25. *TMS320DM642 Vidio/Imaging Fixed-Point Digital Signal Processor Data Manual* (2009). www.ti.com
26. *Texas Instruments TMS320C6000 TCP/IP Network Developer' Kit* (2008). www.ti.com
27. X Jia, JA Richards, Segmented principle components transformation for efficient hyperspectral remote-sensing image display and classification. IEEE Trans Geosci Remote Sens **37**(1), 538–542 (1999)

doi:10.1186/1687-6180-2012-142
Cite this article as: Zhang *et al.*: Real-time target detection in hyperspectral images based on spatial-spectral information extraction. *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:142.