



An Efficient Hardware Implementation of Detecting Targets from Remotely Sensed Hyperspectral Images

C Sherin Shibi^{1*} and R Gayathri²

¹Institute of Artificial Intelligence and Data Science, Saveetha School of Engineering, SIMATS,,
Thandalam, Chennai 602 105, Tamil Nadu, India

²Department of Electronics and Communication Engineering, Sri Venkateswara College of Engineering,
Sriperumbudur 602 117, Tamil Nadu, India

Received 04 August 2020; revised 05 January 2022; accepted 05 January 2022

Real-time implementation of hyperspectral imagery is an emerging research area which has notable remote sensing applications. It is challenging to process a huge volume of hyperspectral data under real-time constraints. Field programmable gate arrays are considered as an efficient hardware suited for onboard processing system. ATGP is a proven target detection algorithm which can automatically detect the target without any predefined data. In the traditional method, this algorithm involves orthogonal subspace projector which makes the hardware design too complex and slow. To speed up the process, Gram-Schmidt orthogonalization operator is used. Gram-Schmidt orthogonalization technique uses inner product instead of matrix inverse which makes the hardware design easy to implement in FPGA board. A detailed comparative analysis is carried out using three different hyperspectral images to emphasize the performance of the design which is adopted in this technique. The processing speed of the proposed ATGP-GS algorithm is 3.484 s for ROSIS Pavia University dataset, 1.781 s for HYDICE Urban dataset and 1.609 s for AVIRIS Cuprite dataset. The proposed algorithm is implemented in Virtex 6 ML605 evaluation board to evaluate the real-time performance of the system.

Keywords: Automatic target generation process, Field programmable gate array, Gram-Schmidt orthogonalization, Hyperspectral imaging, Onboard processing

Introduction

Hyperspectral imaging sensors captures numerous spectral bands at varying wavelengths to characterize the objects present in the surface of Earth.^{1,2} These imaging systems generate large data volumes consist of rich information about the scene which is more detailed than the conventional imaging systems like multispectral sensors. Moreover, sub-pixel level analysis is not possible with multispectral images because it has only few spectral bands.³ The main challenge in processing of hyperspectral images is the huge dimension of data. Because of the possibility of sub-pixel analysis and availability of rich information, the hyperspectral images can be used in number of applications which includes environmental modeling, wildfire management, pollutants detection in water and atmosphere, biological threat detection, military and security applications and so on.

During two decades, a large number of target detection algorithms have been introduced by researchers.⁴⁻⁹ The main aim of target detection

algorithms is to recognize the pixels in a hyperspectral data cube when a specific target is given. Real-time identification of manmade and natural targets from hyperspectral images has become more important in the field of military and security purpose. From the literature^{10,11}, it is proved that Automatic Target Generation Process (ATGP) has good target detection accuracy. Initially, ATGP was implemented using Orthogonal Subspace Projection (OSP) operator which is more complex due its matrix inversion operation.¹ This will slow down the operation and affect the real time performance. To overcome this problem, another version of ATGP is introduced with Gram-Schmidt Orthogonalization technique.¹² This technique will reduce the complex matrix multiplications and performs only vector operations. This will reduce the computations and makes the system to perform fast. The target detection process is shown in Fig. 1.

The latest development in hyperspectral imaging instruments increases the spatial, spectral and temporal resolution of data cube. This introduces a new challenge in fast processing and analyzing hyperspectral data. To accelerate the computational

*Author for Correspondence
E-mail: shibicharles@gmail.com

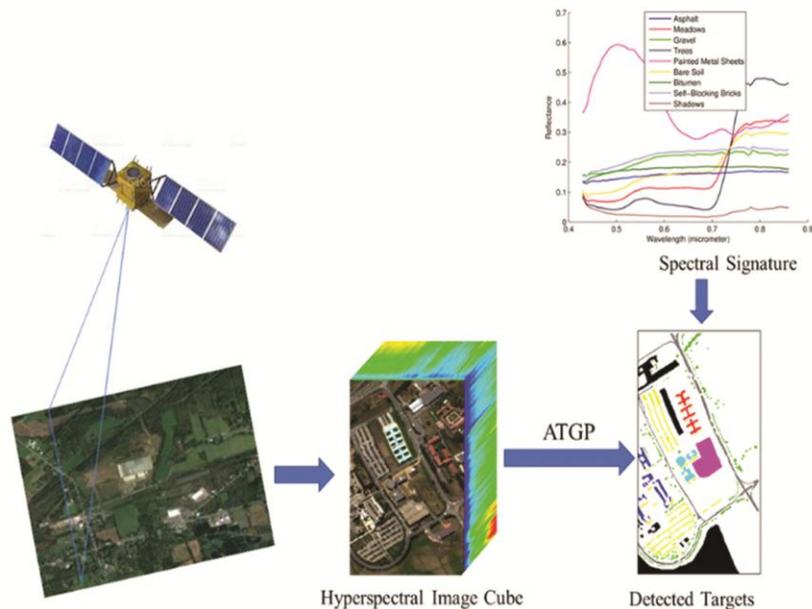


Fig. 1 — Hyperspectral target detection process

speed, it is necessary to use hardware accelerators for real-time processing of data.^{13,14} Field programmable gate arrays (FPGAs) are the preferred hardware solution compared to graphical processing units (GPUs), clusters of computers and multicore processors. This is because of the following three reasons: 1) small size, weight and low power consumption, 2) tolerance to ionizing radiation, and 3) reconfigurable property.¹⁵

The performance of the proposed algorithm is evaluated using three real hyperspectral datasets namely Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS), Hyperspectral Digital Image Collection Experiment (HYDICE) and Reflective Optics Spectrographic Imaging System (ROSIS). The proposed algorithm was implemented on the Xilinx Virtex-6 XC6VLX240T FPGA of the ML605 evaluation board.

Materials and Methods

Dataset Description

We considered three hyperspectral images for the experimental analysis. These images are chosen for analysis because they have highly reliable reference data and the results can be compared with the high spectral resolution. The dataset is available in the website <http://lesun.weebly.com/hyperspectral-dataset.html>.

AVIRIS Cuprite Mining Site Dataset

The most popular hyperspectral dataset is Cuprite data captured by AVIRIS (Airborne Visible Infrared

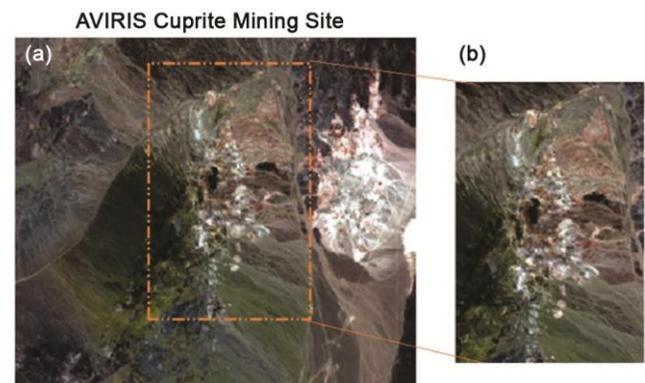


Fig. 2 — (a) AVIRIS Image Scene with RGB bands (31, 20, 12); (b) Region of interest

Imaging Spectrometer) sensor over the Cuprite mining site, Nevada, in 1997. There are 224 spectral bands in the Cuprite image ranging from 370 nm to 2480 nm. This image has 350×350 pixels with approximately 10 nm spectral resolution. There are 188 bands after removing the noisy bands (i.e. 1–2 and 221–224) and the water absorption bands (i.e. 104–113 and 148–167). This cite consist of minerals including alunite, buddingtonite, calcite, kaolinite, and muscovite. The image scene from AVIRIS considered in our experiments is shown in Fig. 2.

HYDICE Urban Dataset

The second dataset is recorded by the HYDICE (Hyperspectral Digital Image Collection Experiment) sensor in October 1995, which is an urban area at

Copperas Cove, near Fort Hood, Texas, USA. This image has 307×307 pixels with the spatial resolution of 2 m per pixel and spectral resolution of 10 nm. It has 210 spectral bands ranging from 400 nm to 2500 nm. There are only 162 bands after removing noise bands such as 1–4, 76, 87, 101–111, 136–153 and 198–210 due to dense water vapor and atmospheric effect. The urban image considered for analysis is shown in Fig. 3.

ROSIS Pavia Dataset

The third dataset is acquired using ROSIS (Reflective Optics System Imaging Spectrometer) over the Pavia University, northern Italy. This image has 610×340 pixels with a spectral coverage ranging from 0.43 to 0.86 μm . The number of bands in this image is 103 spectral bands and a spatial resolution of 1.3 m per pixel. It covers an urban environment, with various solid structures (asphalt, gravel, metal sheets, bitumen, bricks), natural objects (trees, meadows, soil), and shadows. The ROSIS image scene of Pavia University is shown in Fig. 4.

Linear Mixing Model

In the structured modelling methods, the target detection involves Linear Mixing Model (LMM) which is evaluated by considering the variability of pixels. In LMM, the mixed pixel is considered as a collection of spectra of endmembers.³ Consider L is the number of spectral bands available in the dataset and there are p target signatures, m_1, m_2, \dots, m_p present in the image. Let m_j is an $L \times 1$ column vector denoted by the j^{th} target signature and r is an $L \times 1$ column vector which denotes the spectral signature of a pixel vector. Linear Mixing Model can be represented as:

$$r = Ma + n \quad \dots (1)$$

where n is considered as an $L \times 1$ column vector denotes the additive white noise with zero mean and variance $\sigma^2 I_{L \times L}$ and $I_{L \times L}$ is the $L \times L$ identity matrix. The image pixel vector is normally a mixed pixel and the signature of the mixed pixel is the combination of target signatures.

Automatic Target Generation Procedure-Orthogonal Subspace Projection

The Automatic Target Generation Procedure (ATGP) automatically finds a set of spectrally distinct endmembers in the hyperspectral image when there is no prior knowledge of target is available.⁸ This algorithm estimates the end members by iterative

HYDICE URBAN IMAGE



Fig. 3 — HYDICE Image Scene with RGB bands (40, 30, 10)

Pavia University



Fig. 4 — ROSIS Image Scene with RGB bands (50, 30, 15)

orthogonal projections of the input and finding the largest magnitude vector of these projections.

In ATGP algorithm, the first target pixel vector is first selected and is represented as t_0 . Then the image pixels are projected onto the orthogonal space $\langle t_0 \rangle^\perp$. The first target is chosen by considering maximum length pixel vector in $\langle t_0 \rangle^\perp$ and is denoted by t_1 . This procedure is continued to obtain all the target vectors until a stopping criterion is satisfied.

Rewrite the model Eq. (1) as follows

$$r = ta_p + B\gamma + n \quad \dots (2)$$

where $t = m_p$ denotes the desired target signature and $B = [m_1, m_2, \dots, m_{p-1}]$ represents undesired spectral signature matrix. To annihilate B from image pixel vector, an orthogonal subspace projector is used and is given by

$$P_B^\perp = (I - B(B^T B)^{-1} B^T) \quad \dots (3)$$

Applying P_B^\perp to Eq. (2) results in a new model

$$P_B^\perp r = P_B^\perp t a_p + P_B^\perp n \quad \dots (4)$$

The signal-to-noise ratio (SNR) is given as

$$SNR(x) = \frac{(x^T P_B^\perp t) a_p^2 (t^T P_B^\perp x)}{x^T P_B^\perp E[nn^T] P_B^\perp x} \quad \dots (5)$$

as the standard criterion. The SNR is maximized by a matched filter represented by M_t with $x = \kappa t$ where κ is a constant and t is the desired target signature. The combination of P_B^\perp with M_t is known as an orthogonal subspace classifier P_{OSP} is denoted by

$$P_{OSP} = M_t P_B^\perp = t^T P_B^\perp \quad \dots (6)$$

The orthogonal subspace projector is given as $P_B^\perp = (I - B(B^T B)^{-1} B^T)$ which is applied to all image pixels. The steps involved in ATGP-OSP is shown in Algorithm 1.

Algorithm 1: ATGP-OSP

1. Inputs: F : hyperspectral image cube (lines x rows x bands)
- p : the number of targets to be detected
2. $c = F^T F$ % To find the largest length pixel
3. $idx = \arg \max_{[1, \dots, r]} c[:, N]$
4. $t_0 = F(:, idx)$
5. $B = [t_0 | 0, \dots, | 0]$ where t_0 be the initial endmember with maximum length
6. for $i = 1$ to $p - 1$ do
7. $P_B^\perp = (I - B(B^T B)^{-1} B^T)$;
8. $v = P_B^\perp F$;
9. $i = \arg \max_{[1, \dots, r]} v[:, i]$;
10. $t_i = B[:, i + 1] = F[:, i]$;
11. end for
12. Output: $B = [t_0, t_1, \dots, t_{p-1}]$; Endmember matrix

Automatic Target Generation Procedure-Gram Schmidt Orthogonalization

The ATGP-GS algorithm is proposed to reduce the computational complexity of the traditional OSP which adopts an enhanced version of Gram-Schmidt method.¹⁶ This method uses the previously formulated information which speeds up the entire process. The classical orthogonal projection approach involves matrix calculation which is difficult to compute in hardware. This method involves inner product space which orthogonalizes the set of vectors. The hyperspectral image F is defined in the space R^n and this process involves a linearly independent vectors

$V = \{v_1, \dots, v_k\}$ for $k \leq n$, and creates an orthogonal vectors $U = \{u_1, \dots, u_k\}$ which has the same k -dimensional subspace of R^n as V . The operation operator is expressed as

$$\text{proj}_u(v) = \frac{\langle v, u \rangle}{\langle u, u \rangle} u \quad \dots (7)$$

where $\langle v, u \rangle$ represents the inner product of vectors u and v . The initial endmember is selected by extracting the maximum magnitude vector. After the selection of first endmember, the ATGP-GS algorithm detects the new endmembers by finding the vector which has the largest orthogonal projection with the previously extracted endmember. Moreover, the ATGP-GS algorithm checks whether the new endmembers are not the same as the previously extracted endmembers and also ensures the extracted endmembers are really new endmembers. The steps involved in the ATGP-GS algorithm are explained in Algorithm 2.

Algorithm 2 ATGP-GS

1. Inputs: F : hyperspectral image cube (lines x rows x bands)
- p : the number of targets to be detected
2. $= [t_0 | 0, \dots, | 0]$; where t_0 be the initial endmember with maximum magnitude in orthogonal space.
3. $U = [0 | 0, \dots, | 0]$; for storing the orthogonal base generated by GS process
4. for $i = 1$ to $p - 1$ do
5. $U[:, i] = B[:, i]$;
6. $P_B^\perp = [1, \dots, 1]$;
7. for $j = 2$ to i do
8. $\text{proj}_{U[:, j-1]}(B[:, i]) = B[:, i]^T U[:, j-1] / U[:, j-1]^T U[:, j-1] U[:, j-1]$;
9. $U[:, i] = U[:, i] - \text{proj}_{U[:, j-1]}(B[:, i])$;
10. end for j
11. $w = [1, \dots, 1]$;
12. for $k = 1$ to i do
13. $\text{proj}_{U[:, k]}(w) = \frac{w^T U[:, k]}{U[:, k]^T U[:, k] U[:, k]}$;
14. $P_B^\perp = P_B^\perp - \text{proj}_{U[:, k]}(w)$;
15. end for k
16. $x = P_B^\perp F$;
17. $i = \arg \max_{[1, \dots, r]} x[:, i]$;
18. $v_i = B[:, i + 1] = F[:, i]$;
19. end for
20. Output: $B = [t_0, t_1, \dots, t_{p-1}]$; Endmember matrix

Hardware Implementation

The hardware architecture of the ATGP-GS algorithm to implement in FPGA board is shown in Fig. 5. The prefetching approach is adopted to store the data using off-chip memory. The data communication is managed by DDR3 SDRAM and a direct memory access (DMA). MicroBlaze is the softcore processor available in the Virtex board which controls the operations of DMA and FIFO. The proposed algorithm is implemented in the reconfigurable unit and the output is obtained using RS232 port which is controlled by RS232 controller.

In this section, an enhanced version of ATGP algorithm is implemented using high performance Virtex board. The endmembers are calculated by orthogonal projections which is formulated by Gram-Schmidt method. The modules to deploy ATGP-GS algorithm along with I/O communications using AXI Interconnect bus is shown in Fig. 6.

In this work, parallelization is adopted to minimize the execution time. The calculation of maximum projections is the time-consuming stage. This requires a huge number of dot products and should be executed in parallel manner. The modules of dot product unit are shown in Fig. 7. The hardware required to execute the dot-product is multiplier, adder and a register.

The architecture of the maximum length module to find the initial target signature and also the maximum length vector after adopting orthogonal projection is described in Fig. 8. The length of the pixel f_i is calculated using the formula $\sum_{k=1}^N f_i(k) \times f_i(k)$. To compute this step, two pixels are retrieved from DDR3 SDRAM at the same time. After calculating the length, the value is compared with the already

extracted maximum value and if it is a new maximum value, then it is saved for next iteration. This will be carried out for all number of spectral bands.

The modules to implement the projection operator are explained in Fig. 9. This module performs the operation $P_B^\perp - (B^T U / U^T U)$. This module consists of dot product followed by a divider. The result of divider is multiplied with the Matrix Memory U . Then

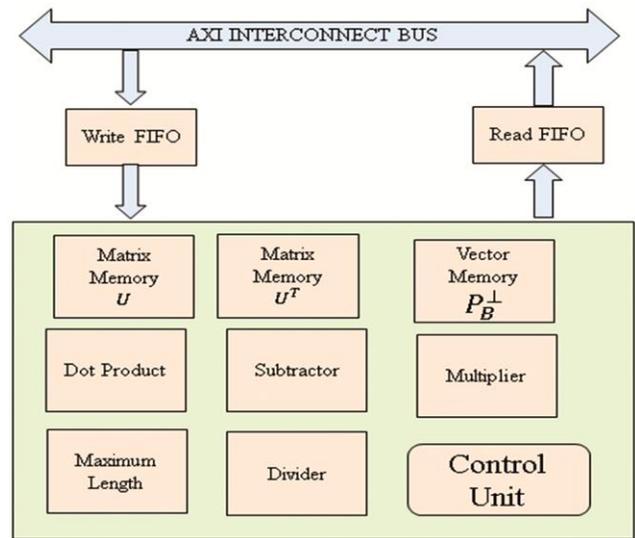


Fig. 6 — Modules to implement ATGP-GS Algorithm

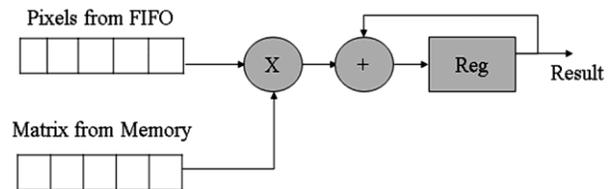


Fig. 7 — Dot Product Unit

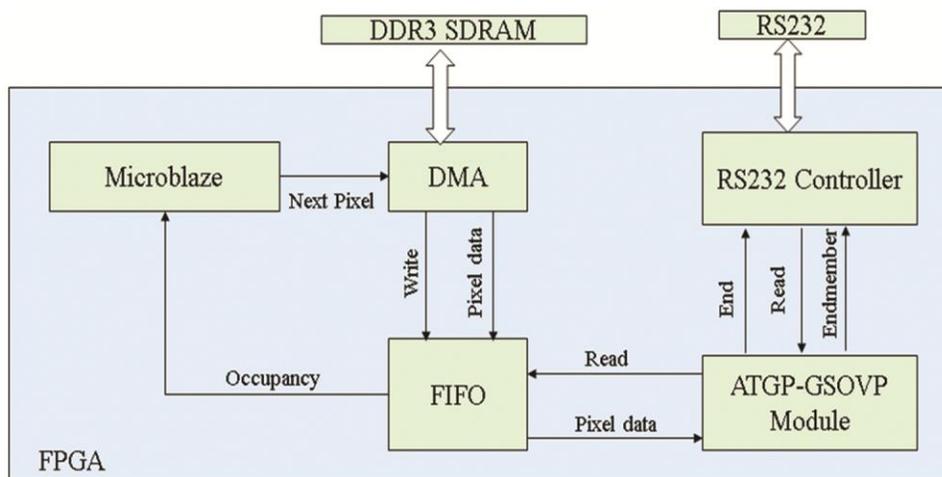


Fig. 5 — Hardware architecture of the complete system

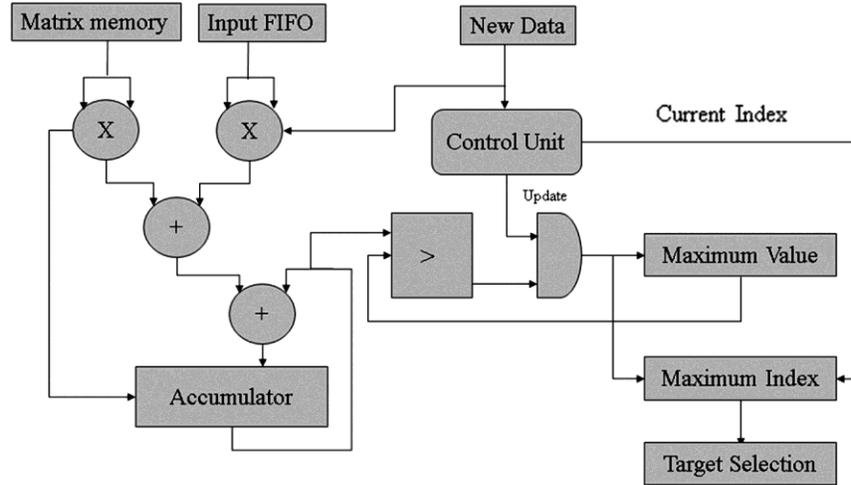


Fig. 8 — Modules to find maximum length

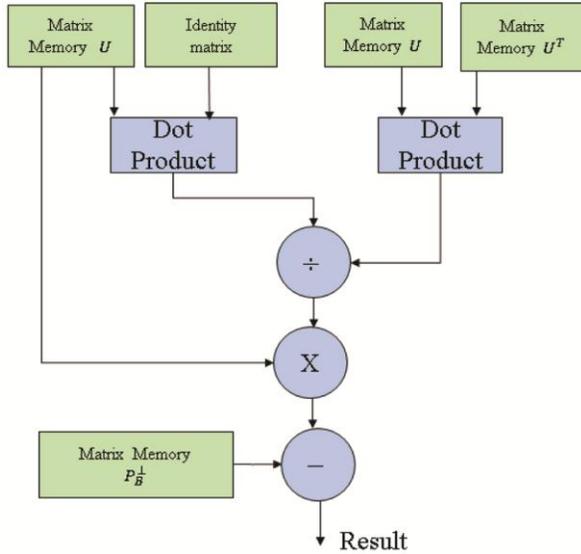


Fig. 9 — Modules to find projection operator

the result is obtained by subtracting with the vector memory P_B^1 .

The step-by-step description of proposed architecture and its workflow to extract the set of p endmembers from a hyperspectral image is given below.

- At first, Microblaze choose a set of p endmembers randomly and issue a signal to DMA for writing these initial endmembers in the FIFO.
- Then, the control unit accepts these endmembers and moves them to the registers. After storing in registers, the volume of these endmembers are formulated and stored in a register.
- Once the Microblaze store the initial set of endmembers, it sends a signal to the DMA to

transfer the data from the DDR3 SDRAM to the FIFO.

- After storing the first pixel in FIFO, the ATGP-GS module will perform the operation. A new row of pixel is sent to the dot product module for every clock cycle. The control unit monitors whether it is a first target and maximum length is calculated. It identifies this set of pixels as the first target. Then the projection operator is calculated and followed by dot product to find the next target. This will be repeated for all p number of targets.

Results and Discussion

Performance of Software Implementation

We have considered ATGP algorithm to identify the spectral signatures in the hyperspectral data cube. The optimization of this algorithm involves Gram-Schmidt Orthogonalization technique which does not have complex matrix inverse calculations. The processing time of these two algorithms is visualized in Table 1. The ATGP-GS algorithm has low computational time which indicates that the fast processing of hyperspectral data cube.

Target-Detection Accuracy Evaluation

The spectral similarity between the USGS library spectra and the corresponding endmembers extracted by the considered implementation of the ATGP algorithm is determined using spectral angle distance (SAD). It is important to emphasize that smaller SAD values indicate higher spectral similarity. The SAD between a target t_i detected by the ATGP algorithm and a reference spectral signature s_i is given by

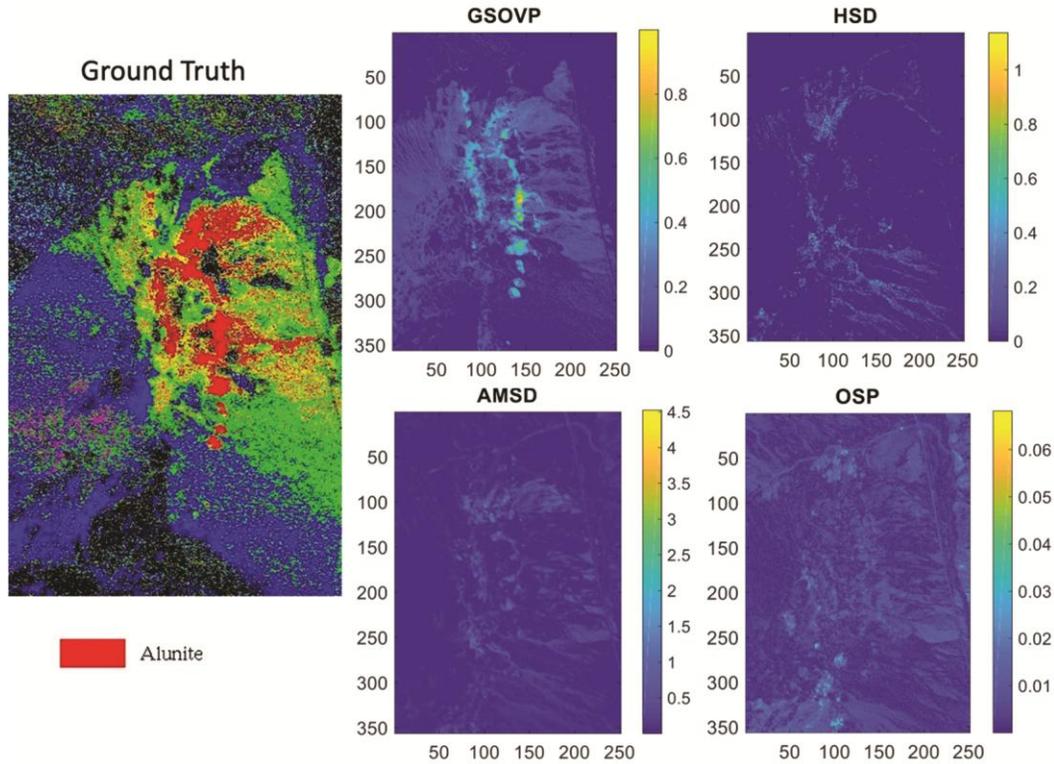


Fig. 10 — Detector results of ATGP-GS, HSD, AMSD and OSP for AVIRIS Cuprite scene with Ground truth

Table 1 — Computational time of endmember extraction step for different datasets

Hyperspectral Image Scene	ATGP	ATGP-GS
ROSIS Pavia University	23.625s	3.484s
HYDICE Urban	14.766s	1.781s
AVIRIS Cuprite, Nevada	14.406s	1.609s

$$SAD(t_i, s_i) = \cos^{-1} \left(\frac{t_i \cdot s_i}{\|t_i\| \cdot \|s_i\|} \right) \dots (8)$$

The good detection performance of the targets is determined by low SAD values. The SAD values for the endmembers of AVIRIS Cuprite, HYDICE Urban and ROSIS Pavia University is shown in Table 2. The SAD values indicate that the ATGP-GS optimization does not penalize ATGP-OSP in terms of target detection accuracy.

Performance of Target Detection Algorithm

In AVIRIS Cuprite dataset, the scene is well understood mineralogically and has several exposed minerals of interest including alunite, buddingtonite, calcite, kaolinite, and muscovite. We have considered the mineral Alunite as the target spectra. The spectral signature of the target is available in the United States Geological Survey (USGS) Digital Spectral Library. After removing water absorption and noisy bands, the bands 3–103, 114–147 and 168–220 are considered

Table 2 — SAD values for endmembers in AVIRIS Cuprite Dataset, HYDICE Urban Dataset and ROSIS Pavia University Dataset

Endmembers	ATGP-OSP	ATGP-GS
AVIRIS Cuprite Dataset		
Alunite	11.68°	11.68°
Andradite	9.52°	8.50°
Buddingtonite	7.71°	7.79°
Dumortierite	12.15°	5.89°
HYDICE Urban Dataset		
Asphalt Road	7.54°	7.54°
Grass	9.33°	9.13°
Tree	4.25°	6.82°
Roof	6.21°	5.41°
ROSIS Pavia University Dataset		
Painted Metal Sheets	7.79°	7.79°
Bare Soil	11.84°	11.91°
Bitumen	13.96°	12.98°
Self-Blocking Bricks	9.05°	8.92°

for analysis. In Fig. 10, the detector results of ATGP-GS, HSD, AMSD and OSP is shown. In HYDICE Urban dataset, we have considered only 162 bands such as 5–75, 77–86, 88–100, 112–135 and 154–197. We have considered Asphalt road as the target spectra and the detection result of ATGP-GS, HSD, AMSD and OSP is shown in Fig. 11. In ROSIS Pavia University dataset, the target to be detected is Painted

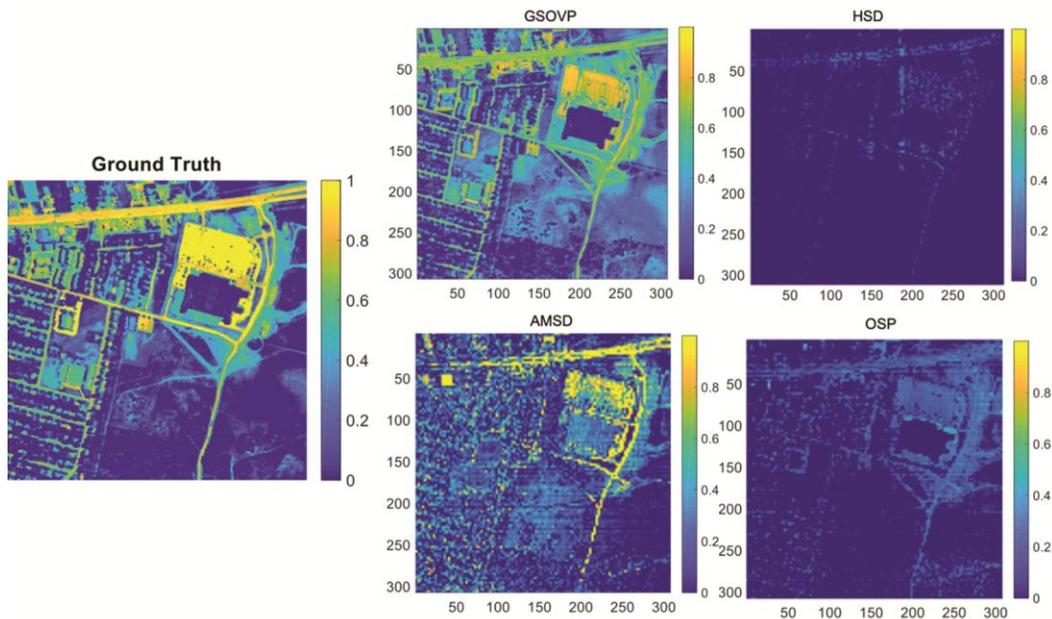


Fig. 11 — Detector results of ATGP-GS, HSD, AMSD and OSP for HYDICE Urban Image with ground truth

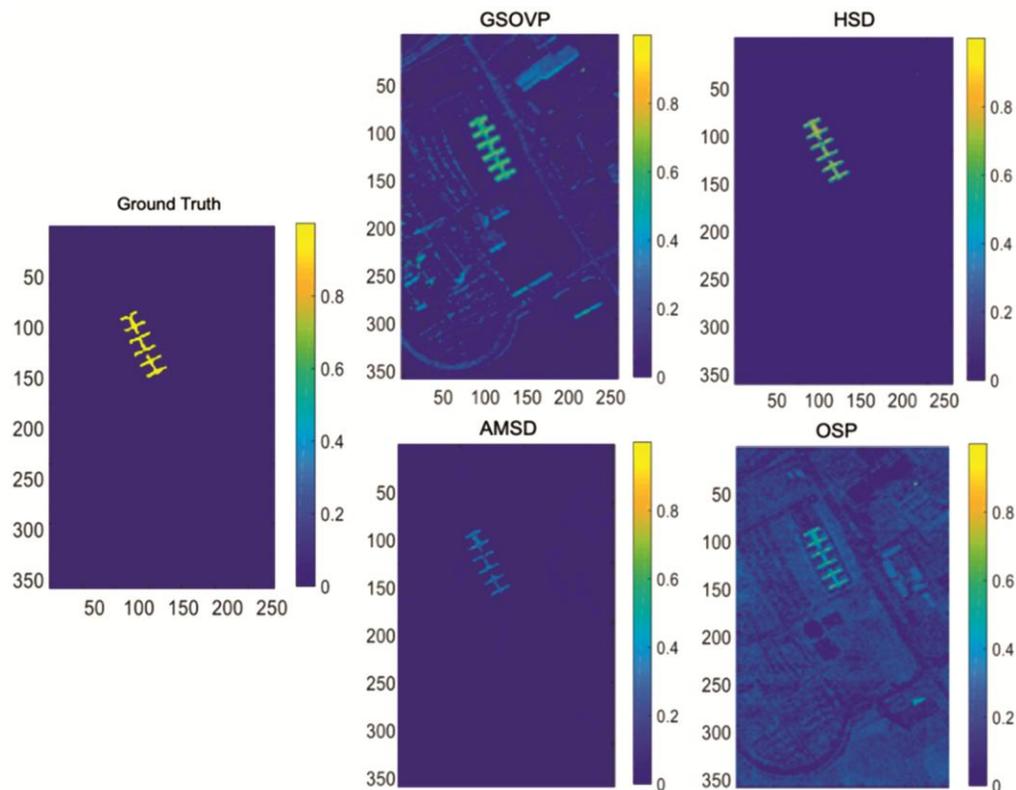


Fig. 12 — Detector results of ATGP-GS, HSD, AMSD and OSP for ROSIS Pavia University Image with ground truth

metal sheets and the detector results are shown in Fig. 12. The ATGP-GS algorithm shows reliable performance compared to HSD, AMSD and OSP algorithm with different datasets.

Performance of Hardware Implementation

In this section, an experimental evaluation of the proposed FPGA implementation was conducted. The FPGA design was implemented on the Xilinx

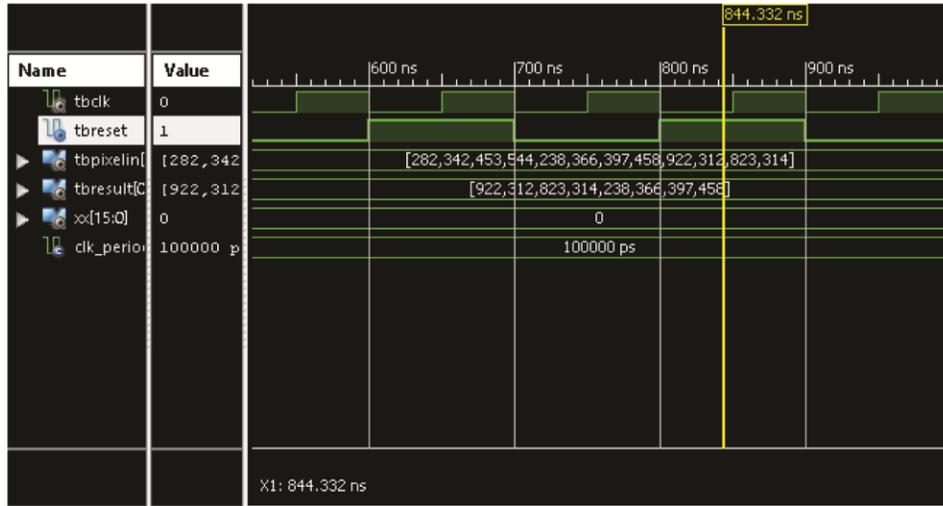


Fig. 13 — Simulation results of ATGP-GS algorithm

Virtex-6 XC6VLX240T FPGA of the ML605 board. This FPGA has a total of 3,01,440 slice registers, 1,50,720 slice lookup tables (LUTs), and 11,398 LUT Flip Flop pairs. In addition, the FPGA includes some heterogeneous resources such as 768 DSP48E1s and 416 distributed block RAMs.

Simulation Results

The simulation results of ATGP-GS algorithm are shown in Fig. 13. A sample input is given as [282, 342, 453, 544, 238, 366, 397, 458, 922, 312, 823, 314] and is represented as ‘tbpixelin’. The number of bands is 4 and the number of targets is 2. The result is represented as ‘tbresult’. From the results, it is visible that the band having maximum pixel values are taken as the first endmember.

The resource utilization of ATGP-GS algorithm is given in Table 3. FIFOs and the memories are implemented using Block RAMs. A large number of slices and DSP48E1s are used for the implementation of the ATGP-GS algorithm. The main challenge of the parallel system is the frequent communications between the modules. To reduce this prefetching approach is utilized. DMA is used to reduce the I/O overhead. When ATGP-GS modules process a set of data, the DMA will fetch the next data set and storing it in the write FIFO.

According to the experimental results, the target detection accuracy of the proposed ATGP-GS algorithm is better compared to ATGP-OSP. Regarding the processing speed performance, ATGP-GS shows 7 times speedup than that of the ATGP-OSP. It is worth noting that the proposed architecture of ATGP-GS can also gain benefits in terms of

Table 3 — Summary of Resource Utilization of the ATGP-GS Algorithm

	Available Resources	Utilized Resources	Percentage of Utilization
Number of DSP48E1s	768	637	82%
Number of Slice Registers	3,01,440	1,80,232	60%
Number of Slices LUTs	1,50,720	79,070	52%
Number of LUT Flip Flop pairs	11,398	3,048	27%

scalability, portability, and flexibility. This is particularly suitable for the real-time hyperspectral target detection applications on satellite.

Conclusions

This work focused on the FPGA version of an ATGP algorithm for remotely sensed hyperspectral image. In our proposed method, we have investigated the impact of adopting the Gram-Schmidt method for calculating the orthogonal projections instead of OSP method. The proposed algorithm is implemented on a Virtex-6 XC6VLX240T FPGA which proves that our implementation makes full use of FPGA architecture and also enhances the performance of the system. The proposed ATGP-GS algorithm can detect the targets 7 times faster than the ATGP-OSP algorithm. As a future work, we will focus to utilize the hardware resources in a better way and improve the detection speed. The spatial as well as spectral information can be used to obtain a high performance. And also, we will focus on unsupervised deep learning algorithms for remote sensing applications to enhance the detection accuracy.

Acknowledgments

This work is funded by Indian Space Research Organisation (ISRO) under RESPOND Scheme. (Sanction Letter No. ISRO/RES/3/756).

References

- 1 Manolakis D, Marden D & Shaw G A, Hyperspectral image processing for automatic target detection applications, *Lincoln Lab J*, **14** (2003) 79–116.
- 2 Nasrabadi N M, Hyperspectral target detection : An overview of current and future challenges, *IEEE Signal Process Mag*, **31** (2014) 34–44.
- 3 Manolakis D, Siracusa C & Shaw G, Hyperspectral subpixel target detection using the linear mixing model, *IEEE Trans Geosci Remote Sens*, **39** (2001) 1392–1409.
- 4 Truslow E, Manolakis D, Pieper M, Cooley T & Brueggeman M, Hyperspectral performance prediction of the adaptive cosine estimator, *Proc IEEE Int Conf Acoust Speech Signal Process* (Vancouver, BC, Canada) 2013, 6264–6268.
- 5 Manolakis D, Lockwood R, Cooley T & Jacobson J, Robust matched filters for target detection in hyperspectral imaging data, *Proc IEEE Int Conf Acoust Speech Signal Process* (Honolulu, HI, USA) 2007, 529–532.
- 6 Harsanyi J C & Chang C I, Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach, *IEEE Trans Geosci Remote Sens*, **43** (2005) 502–518.
- 7 Zhao L, Lin W, Wang Y & Li X, Recursive local summation of RX detection for hyperspectral image using sliding windows, *Remote Sens*, **10** (2018) 1–17.
- 8 Ren H & Chang C I, Automatic spectral target recognition in hyperspectral imagery, *IEEE Trans Aerosp Electron Syst*, **39** (2003) 1232–1249.
- 9 Chen S Y, Ouyang Y C, Lin C, Chen H M, Gao C & Chang C I, Progressive endmember finding by fully constrained least squares method, in *7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing* (Tokyo, Japan) 5–8 June 2015.
- 10 Chang C I, Chen SY, Li H C, Chen H M & Wen C H, Comparative study and analysis among ATGP, VCA, and SGA for finding endmembers in hyperspectral imagery, *IEEE J Sel Top Appl Earth Obs Remote Sens*, **9** (2016) 4280–4306.
- 11 Chang C I & Li Y, Recursive band processing of automatic target generation process for finding unsupervised targets in hyperspectral imagery, *IEEE Trans Geosci Remote Sens*, **54** (2016) 5081–5094.
- 12 González C, Bernabe S, Mozos D & Plaza A, FPGA Implementation of an algorithm for automatically detecting targets in remotely sensed hyperspectral images, *IEEE J Sel Top Appl Earth Obs Remote Sens*, **9** (2016) 4334–4343.
- 13 Robertson I, Irvine J, Lysaght P & Robinson D, Enhanced architectures, design methodologies and cad tools for dynamic reconfiguration of xilinx FPGAs, in *Int Conf Field Program Logic Appl* (Madrid, Spain) 28–30 August 2006.
- 14 Lee C A, Gasster S D, Plaza A, Chang C I & Huang B, Recent developments in high performance computing for remote sensing: A review, *IEEE J Sel Top Appl Earth Obs Remote Sens*, **4** (2011) 508–527.
- 15 Lopez S, Vladimirova T, Gonzalez C, Resano J, Mozos D & Plaza A, The promise of reconfigurable computing for hyperspectral imaging onboard systems: A review and trends, *Proc IEEE*, **101** (2013) 698–722.
- 16 Guerra R, Santos L, Lopez S & Sarmiento R, A new fast algorithm for linearly unmixing hyperspectral images, *IEEE Trans Geosci Remote Sens*, **53** (2015) 6752–6765.