

# RANDOM-PROJECTION-BASED DIMENSIONALITY REDUCTION AND DECISION FUSION FOR HYPERSPECTRAL TARGET DETECTION

Qian Du, James E. Fowler, Ben Ma

Department of Electrical and Computer Engineering, Geosystems Research Institute  
Mississippi State University, Mississippi State, MS 39762, USA  
{du, fowler, bm538}@ece.msstate.edu

## ABSTRACT

Random projection for dimensionality reduction of hyperspectral imagery with a goal of target detection is investigated. Random projection is attractive in this task because it is data independent and computationally more efficient than other widely-used dimensionality-reduction methods, such as principal component analysis or the maximum-noise-fraction transform. Experimental results reveal that dimensionality reduction based on random projections yields improved target detection after decision fusion across multiple instances of the projections. Parallel implementation using a graphics processing unit is also investigated.

**Index Terms** — dimensionality reduction, random projection, target detection, hyperspectral imagery, parallel computing.

## 1. INTRODUCTION

Dimensionality reduction is a frequent preprocessing step in hyperspectral image analysis. Traditional dimensionality-reduction methods, such as principal component analysis (PCA) and the maximum-noise-fraction (MNF) transform are data dependent and entail a heavy computational load. For example, for a hyperspectral image with  $N$  pixels and  $L$  bands, PCA needs  $L^2N$  multiplications for covariance matrix calculation,  $O(L^3)$  multiplications for eigen-decomposition, and  $KLN$  multiplications for the transform (with  $K$  principal components (PCs)); on the other hand, MNF requires estimation of the covariance matrix of the noise, as well as whitening, in addition to the PCA computations.

Random projection (RP) is a computationally efficient, data-independent method for dimensionality reduction [1]. We focus on the case in which the transformation matrix of RP consists of Gaussian random vectors which have been orthogonalized and normalized to unit norm. Theoretical results indicate that such a RP can preserve distances among

data points as well as the structure of data cloud [2-4]. RP is of recent interest due to the mathematical theory of compressed sensing provides that sparsely representable signals can be recovered exactly from RP [5].

RP can result in significant computational savings over techniques such as PCA and MNF. For example, if the original data is reduced from  $L$  dimensions to  $K$  dimensions, the computation time involved in the random-matrix generation is only  $O(K^2L)$ , including the expensive Gram-Schmidt orthogonalization process; it can be further reduced to  $O(KL)$  if using uniformly distributed random variables without orthogonalization. Note that, for a hyperspectral remote-sensing image,  $K < L \ll N$ .

In this paper, we show that dimensionality reduction through RP can improve target-detection accuracy. In addition, we conduct multiple RP processes for decision fusion. In each run, the random matrix is different, resulting in a slightly different data-analysis output. For instance, when the randomly projected data is applied to the target-detection task, the detection map produced for each different RP instance will be different in each run. In experimental results, we show that a detection map of greater accuracy can be produced by fusing all the individual outputs resulting from different projection matrices. Since the same detection process is repeated multiple times, the overall computing time is increased although the original data dimensionality is reduced. However, this approach is suitable to parallel computing which can significantly reduce the overall computation time.

## 2. METHODOLOGY

### A. Proposed Algorithm

The entire algorithm is illustrated in Fig. 1. A series of random matrices of size  $K \times L$  are generated as the first step; the reduction ratio for these random matrices is defined as  $P = K/L$ . Subsequently, some suitable target-detection algorithm is applied to the data in the randomly-projected,  $K$ -dimensional domain. From multiple runs with different random matrices, a higher-quality result can be generated compared to the corresponding target detection applied in

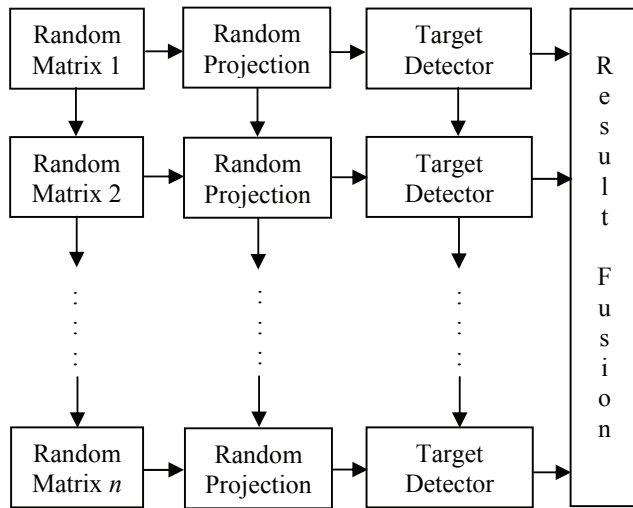
original,  $L$ -dimensional data. This is due to the phenomenon that a target pixel which is salient in the original data domain generally remains distinctive after RP; thus, the target can be detected in the RP domain (see, e.g., [9, 10]). However, with the use of multiple-detector fusion, the false-alarm rate may be reduced.

Different fusion methods can be employed. The simplest is to claim a pixel to be a target pixel if it is extracted by  $m$  times out of  $n$  runs. We will show that when  $m = n$ , the fusion output is superior than that from the original data or a single-run RP.

Here, we adopt the Target-Constrained Interference-Minimized Filter (TCIMF) [6] as the target detector. TCIMF can be expressed as

$$\mathbf{W}_{TCIMF} = \mathbf{R}^{-1}(\mathbf{S}^T \mathbf{R}^{-1} \mathbf{S})^{-1} \quad (1)$$

where  $\mathbf{R}$  is the data correlation matrix, and  $\mathbf{S} = [s_1, s_2, \dots, s_p]$  is the target-signature matrix. To detect if a pixel  $\mathbf{r}$  contains a target or not, we simply apply  $\mathbf{y} = \mathbf{W}_{TCIMF}^T \mathbf{r}$ , and then the  $i^{\text{th}}$  element in  $\mathbf{y}$  can be used to construct the  $i^{\text{th}}$  target-detection map. TCIMF is proved to have excellent capability of separating targets with similar spectral signatures [6].



**Figure 1:** The proposed approach for target detection.

### B. Parallel Implementation

The RP-based processing of Fig. 1 is well-suited to data parallelism. Thus, we implement a parallel version of Fig. 1 within a graphics processing unit (GPU) which is essentially a parallel computer with a shared-memory architecture. GPUs are of great interest to the high-performance-computing community due to the fact that GPUs yield excellent speedup performance while their portability makes them useful for onboard, real-time processing [7].

All processors of a GPU can share data within a global address space. To achieve satisfactory parallel performance,

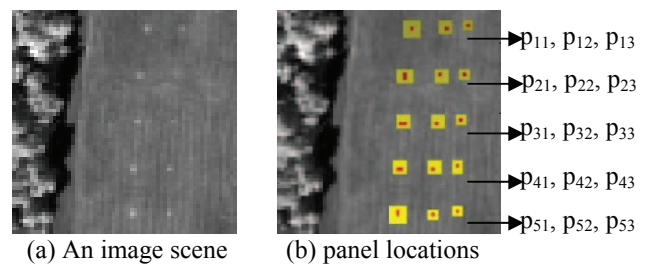
the data throughput is critical in the design of GPU-based parallel algorithms, meaning that enough data should be fed into the GPU to take advantage of the available compute power. Due to the shared-memory architecture, the major bottleneck is memory communication between the host and the device; unnecessary data transfer between host and device should be avoided. In other words, most data computation should take place in the GPU without interruption. While data sharing between GPU cores is much easier than in compute clusters, the data-throughput requirement renders current GPUs inappropriate for solving numerous small matrix-operation problems. Therefore, two key rules of GPU parallelization are: 1) to parallelize a large number of scalar/vector additions/multiplications if possible, and 2) to reduce communication between host and device as much as possible. For a hyperspectral image, the spatial size is typically much larger than the spectral size, suggesting that fulfilling computation tasks in spatial order on a GPU while leaving other tasks (such as small matrix manipulations) to the CPU is the proper approach to balance workload between the GPU and the CPU.

The proposed algorithms use matrix operations extensively. Fortunately, the CUDA CUBLAS library provides a high-performance-computing implementation of the Basic Linear Algebra Subprograms (BLAS) level-1 to level-3 operations [8]. Thus, the parallel implementation is designed to utilize this existing parallel linear-algebra library, which, in turn, requires keeping data continuity in the memory as much as possible.

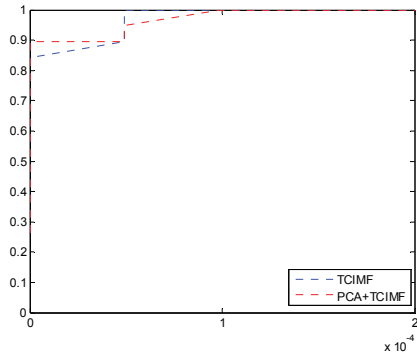
## 3. EXPERIMENT

### A. Data Used in the Experiment

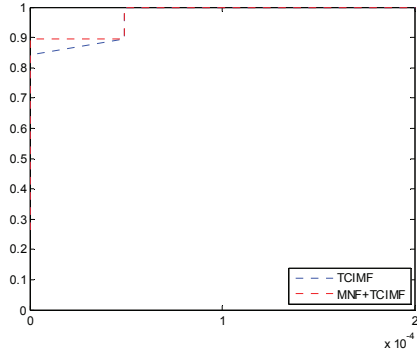
The subimage of HYDICE Forest scene of size  $64 \times 64$  shown in Fig. 2(a) was collected in Maryland in 1995 from a flight altitude of 10,000 ft with about 1.5-m spatial resolution. The spectral coverage is 0.4–2.5  $\mu\text{m}$ . The water absorption and low SNR bands were removed, reducing the spectral dimensionality from 210 to 169. This scene includes 15 panels arranged in a  $5 \times 3$  matrix. The three panels in the same row were made from the same material, and can be considered to be one class,  $P_i$  for  $1 \leq i \leq 5$ . The pixel-level ground-truth map is shown in Fig. 2(b).



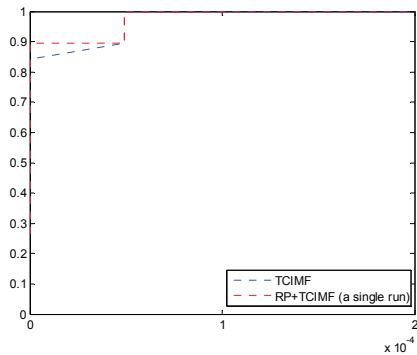
**Figure 2:** The image scene used in the experiment.



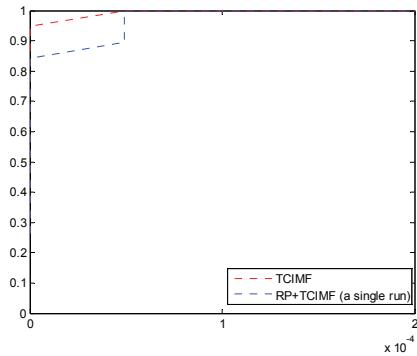
(a) PCA



(b) MNF

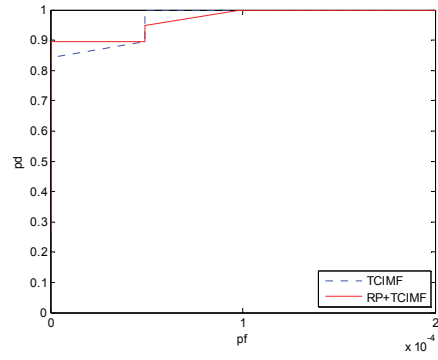


(c) a single run RP ( $P=0.5$ )

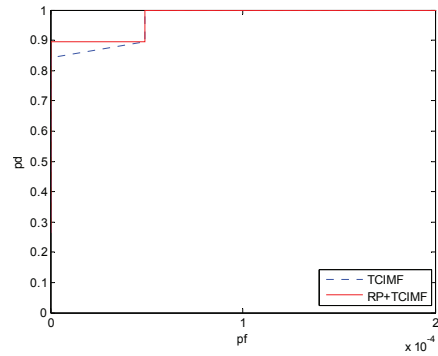


(d) another single run RP ( $P=0.5$ )

**Figure 3:** Target-detection performance using decision dimensionality reduction.

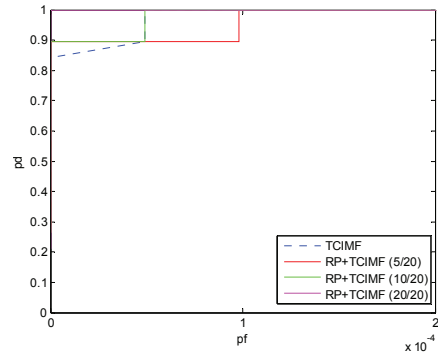


(a)  $P=0.4$  ( $m=10/n=20$ )



(b)  $P=0.5$  ( $m=10/n=20$ )

**Figure 4:** Target-detection performance using RP-based decision fusion.



**Figure 5:** Target-detection performance using decision fusion with different fusion parameters ( $P = 0.5$ ).

### B. Target Detection Performance

Based on the ground truth, a Receiver Operating Characteristic (ROC) curve can be estimated for quantitative performance assessment by gradually changing the threshold (i.e., such that  $p_f$  and  $p_d$  are varied from 0 to 1). Fig. 3 shows the ROC curves using PCA, MNF, and RP for  $P = K/L = 0.5$ . Compared to the result using the original data, PCA does not offer any improvement as is evident in Fig. 3(a); on the other hand, MNF did improve the performance as shown in Fig. 3(b). Fig. 3(c) and Fig. 3(d) concern performance using RP which is seen to also

improve the detection performance and to even outperform MNF. Note that target signatures  $\mathbf{S}$  went through the same dimensionality-reduction process.

Fig. 4 shows the fusion output when RP is used for dimensionality reduction. Here,  $n = 20$  runs were executed, and a target pixel was finally claimed if it was determined to be a target pixel more than  $m = 10$  times. We can see that, when  $P = 0.4$  as in Fig. 4(a), the fused result is not superior to that of the original data; however, when  $P = 0.5$  as in Fig. 4(b), the fused result was indeed better. Fig. 5 demonstrates that the fusion output is further enhanced if  $m$  is increased. In practice,  $m$  can be simply chosen to be equal to  $n$ .

### C. GPU Implementation

The CPU machine used in the experiment is an Intel Core i7-860 2.80GHz processor with 4GB memory. The GPU is NVidia's GeForce GTX480 which has 480 cores with 1.5GB memory. The GPU algorithm was implemented in the C++ with CUBLAS and MKL version 11.1.

Fig. 6 shows the speedup performance when  $n = 20$ . As the reduction ratio  $P$  is increased, both the serial and parallel versions of the algorithm use more compute time. However, the speedup remains relatively unchanged at about 9. Obviously, the speedup performance will be improved as the number of repetitions,  $n$ , is increased.

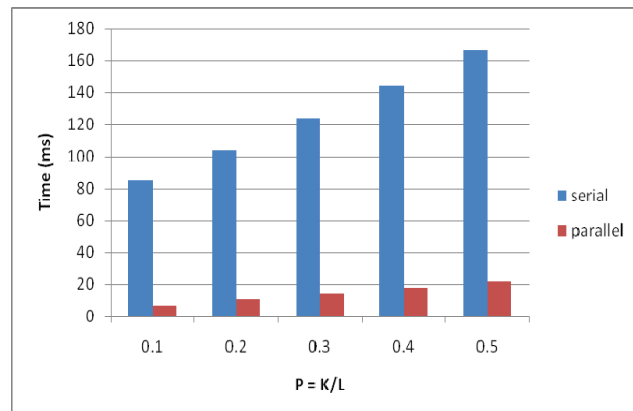


Figure 6: Speedup performance ( $n = 20$ ).

## 4. CONCLUSION

In this paper, we investigated supervised target detection using RP-based dimensionality reduction. When the reduction ratio  $P$  is not too small ( $> 0.4$  in our experiment), target detection can be robustly improved by a single-pass RP. Detection accuracy can be further improved by decision fusion of multiple runs. This approach can be easily implemented in parallel to significantly expedite computation time.

## REFERENCES

- [1] Q. Du and J. E. Fowler, "On the performance of random-projection-based dimensionality reduction for endmember extraction," *Proceedings of IEEE Geoscience and Remote Sensing Symposium*, Honolulu, HI, Jul. 2010.
- [2] S. Dasgupta, "Experiments with random projections," In *Proceedings of the Sixteen Conference on Uncertainty in Artificial Intelligence*, pp. 143-151, 2000.
- [3] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *Proceedings of ACM Conference on Knowledge Discovery and Data Mining*, pp. 245-250, 2001.
- [4] D. Achlioptas, "Database-friendly random projections," in *Proceedings of ACM Symposium on the Principles of Database Systems*, pp. 271-281, 2001.
- [5] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21-30, Mar. 2008.
- [6] H. Ren and C.-I. Chang, "A target-constrained interference-minimized approach to subpixel detection for hyperspectral images," *Optical Engineering*, vol. 39, no. 12, pp. 3138-3145, 2000.
- [7] M. D. McCool, "Signal processing and general-purpose computing on GPUs," *IEEE Signal Processing Magazine*, May 2007.
- [8] [http://developer.download.nvidia.com/compute/cuda/2\\_3/docs/CUBLAS\\_Library\\_2.3.pdf](http://developer.download.nvidia.com/compute/cuda/2_3/docs/CUBLAS_Library_2.3.pdf)
- [9] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal processing with compressive measurements," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 445-460, April 2010.
- [10] J. E. Fowler and Q. Du, "Anomaly detection and reconstruction from random projections," *IEEE Transactions on Image Processing*, submitted.