

Graphical Processing Unit based Implementations of Crystal Identification Algorithms

M. Sayed

Radiation Engineering Department,
Atomic Energy Authority,
EAEA, 3 Ahmed El-Zomar Str., 8th
District, Nasr City, Cairo Egypt

A. Arafa

Radiation Engineering Department,
Atomic Energy Authority,
EAEA, 3 Ahmed El-Zomar Str., 8th
District, Nasr City, Cairo Egypt

H.I. Saleh

Radiation Engineering Department,
Atomic Energy Authority,
EAEA, 3 Ahmed El-Zomar Str., 8th
District, Nasr City, Cairo Egypt

ABSTRACT

Crystal identification (CI) provides a solution for parallax error that occurs within the Positron Emission Tomography (PET) scanners. The CI rate is one of the main challenges to reconstruct the image in PET system. This paper proposed a high rate CI algorithm based on fractional Fourier transform (FRFT) and a powerful classifier that is Support Vector Machine (SVM). The computation of the proposed algorithm is significantly reduced to a single weighted sum of pulse's samples. In addition, the computations were accelerated using two different approaches; Compute Unified Device Architecture (CUDA) or multi-threading high-level parallelism model (openMP) in order to satisfy a high rate for processing the scintillation pulses of PET systems. A huge number of scintillation pulses (100 000 pulses from LSO-LuYAP crystals) were processed to take full advantage of the hardware speeding up provided by a parallel implementation on a graphics processing unit (GPU). The event rates of openMP are 13 or 76 M events/s on a serial single core or parallel 8 cores processor respectively. On the other hand, the pulses were processed using Tesla K20 GPU at 942 M events/sec. The proposed implementations provide a high-speed rate of scintillation pulses that enables the designers of PET systems to increase the number of detectors for high-resolution PET images.

Keywords

Graphical processing unit - Crystal identification - OpenMp - CUDA

1. INTRODUCTION

The spatial resolution of PET images is mainly affected by the detector width, positron range, a co-linearity, the detector ring, and sampling time [1]. These factors can be eliminated or reduced through a careful camera design. Furthermore, the spatial resolution can be enhanced by increasing number of detectors. Due to increase the amount of channels, a high-performance sorting circuits could be used to process the read out channel accurately. However, the ring diameter of the PET system must be designed carefully to reduce the parallax error. This error occurs because of gamma ray that penetrates the crystal of the detector at non-perpendicular angles. One of good solutions to this error is the phosphor sandwich (phoswich) detector. This detector is constructed from multilayer scintillation materials (two or more layers). Where, the response to the radiation of each layer differs from the other. Therefore, the depth of interaction into the crystal can defined accurately based on the decay time of the scintillation pulse [2].

The crystal identification can determined based on the pulse shape discrimination (PSD) method as in [3]. In this work,

PSD was performed based on FFT and SVM using FPGA. The processing rate of this work was 6 M events/s. However, a high rate PSD method increases the PET system accuracy. Therefore, in the proposed work, the CI rate based on parallel computing platform was enhanced. However, a suitable sorting method should be used to record the scintillation pulses accurately. An implementation of sorting algorithm based on FPGA for the PET system events was presented in [4]. Based on this work, all events are detected from the PET channels, recorded. Thus, the scintillation event can be sent to host computer. Therefore, the proposed CI algorithm using GPU can be applied on the recorded events. Moreover, the coincidence identification processing and image reconstruction can be performed.

The proposed work could be used to process a huge number of events in parallel approach. Consequently, the computation time of CI algorithm is reduced. Here, we aim to implement a SVM based CI algorithm based on the computational capability of GPUs to enhance computational event rate. GPU is a specialized device designed to accelerate the image processing which is displayed on computer screens in real time. Recently, the massively parallel clusters can be replaced by GPU due to maintenance, energy costs, lower purchase, and GPU can be programmed by simple instruction in programming frameworks such as Compute Unified Device Architecture (CUDA). Moreover, GPU are more effective than CPU in scientific computation speed due to their massively parallel architecture, where GPU process a huge number of data in parallel [5].

In the proposed algorithm, the feature of parallel architecture of GPU based on CUDA model was used to process the scintillation pulses. The proposed algorithm is tested and compared against multicore architecture which code is written by openMP. The openMP is a multi-threading high-level parallelism model. OpenMP gives shared-memory parallel programmers an interface to develop parallel applications on multicore CPU.

This paper is structured as follows: the mathematical background of the proposed algorithm is presented in section 2. Section 3 presents the proposed CI algorithm. GPU Realization of CI algorithm is discussed in section 4. Section 5 presents Result and discussion. Finally, the conclusion of this work is provided in section 6.

2. MATHEMATICAL BACKGROUND

2.1 Fractional Fourier transforms (FRFT)

The FRFT is a generalization of the Fourier transform (FT) with an order parameter α . Equation (1) represents the value of (α), and (R) which is the fractional factor that ranges from

0 up to 1.

$$\alpha = R \frac{\pi}{2} \quad (1)$$

The value of the parameter α ranges from 0 up to $\pi/2$. Each value of α can be defined as a rotated time–frequency representation of the original signal. When, the value of α equals to $\pi/2$ the FRFT becomes the FT. Thus, the FT is a special case of the FRFT. Equations (2) and (3) show the mathematical representation of the FRFT, and the kernel function respectively [6].

$$X_\alpha = F_\alpha(x(t)) = \int x(t)K_\alpha(t, u)dt \quad (2)$$

$$K_\alpha(t, u) = \begin{cases} \sqrt{\frac{1-\cot(\alpha)}{2\pi}} & \text{if } \alpha \neq n(2\pi) \\ e^{j(u^2/2)\cot(\alpha)} e^{j(t^2/2)\cot(\alpha) - jut\csc(\alpha)} & \\ \delta(t - u) & \text{if } \alpha = n(2\pi) \\ \delta(t + u) & \text{if } (\alpha + \pi) = n(2\pi) \end{cases} \quad (3)$$

Where $n=1, 2, 3$

The proposed algorithm requires a discrete form of the FRFT transformation where the scintillation pulses were digitized as will be explained in the preprocessing section. So, the scintillation signals are transformed using DFRFT. There are different types to implement the DFRFT such as Eigenvector decomposition, Linear combination, and Sampling-FrFT [6]. In this work, the Linear combination type was applied. Where, this type of DFRFT approach meets the periodicity and can be computed using a transformation matrix that can be merged in the proposed algorithm. if the input digital signal is $x[n]$ where n is the number of samples, the output DFRFT coefficients for this signal is $X[k]$ as in equation (4). The transformation matrix of the DFRFT (F^R) is computed as follows in equation (5-7) [7]:

$$X[k] = F^R x[n] \quad (4)$$

$$F^R = \sum_{i=0}^3 \varphi_i(\alpha) F^i \quad (5)$$

$$\varphi_i(\alpha) = \frac{1}{4} \sum_{k=1}^4 e^{j(\alpha - i\frac{\pi}{2})k} \quad (6)$$

$$X[k] = F^R x[n] \quad (7)$$

2.2 Support Vector Machine (SVM)

SVM is a supervised machine-learning algorithm. This algorithm is used to classify data. The classification is performed by determine the hyperplane that differentiates the two classes precisely. The margin between hyperplane and each type of data should be maximized to get an accurate classification. Moreover, the input data to the SVM could be converted from a low dimension space to a higher another dimension using a kernel function. This function is used to make the classified data to be separable. There are different types of the kernel function such as linear, quadratic, radial basis, and Gaussian. The selection of the kernel function depends on the form of the classified data and the implementation method. The optimal hyperplane for a training input data x_i ($i=1, 2, 3... M$) is determined. Where, x_i is classified into to two classes; class one and class two. Moreover, the label for each class is y_i where $y_i=1$ or -1 for class one or class two respectively. The decision function for these variables is represented in equation (8):

$$D(x) = W^T x + b \quad (8)$$

Where W is m -dimensional vector, b is the bias. Consequently, the optimal hyperplane can be determined by solving the following problem for w and b :

$$\text{Mimimize } Q(W, b) = \frac{1}{2} \|W\|^2 \quad (9)$$

$$\text{Where } y_i(W^T x_i + b) \geq 1 \quad \text{for } i = 1, 2, 3, \dots M \quad (10)$$

Equations 9 and 10 can be solved by the quadratic programming technique for the small number of input data. However, input data could be classified efficiently when it converted to a high dimensions space. Hence, equations 9 and 10 are converted into dual problem where the number of variables is equal to the number of training data by optimizing the following Lagrange equation as in equation (11).

$$Q(w, b, \alpha) = \frac{1}{2} \|W\|^2 - \sum_{i=1}^M g_i \{y_i(W^T x_i + b) - 1\} \quad (11)$$

g_i is the nonnegative Lagrange Multipliers

The unknown vector (u) can be classified by predicting the decision function $C(u)$ as in equation (12):

$$C(u) = \text{sign}(\sum_i H(u, S_i) y_i g_i + b) \quad (12)$$

The value of equation (12) is positive for class one and negative for class two. Where, S_i is called the support vectors.

For the linear kernel which is used in this research work.

$$H(u, S_i) = u \sigma_i S_i \quad (13)$$

σ_i is a scale factor for i th feature. The main purpose of this work is reducing the processing time of the scintillation pulses. Hence, the linear kernel function of the SVM is used in this work where it can be merged with the feature extractor (DFRFT) in one function with a small number of multiplication and additions. Where, the direct implementation of Equation (12) (with L support vectors and M features) requires $(LN + 1)$ additions and $(N + 1)$ multiplications. Thus, the implementation time for this kernel function is the lowest.

3. CRYSTAL IDENTIFICATION ALGORITHM

Crystal identification algorithm is implemented based on merging the feature extractor (DFRFT) and the classifier (SVM) to create the classification vector (Z). The proposed merged FRFT-SVM algorithm is illustrated in Figure 1. As shown, this figure the proposed algorithm is implemented in two phases; in the first phase the vector (Z) is computed based on training the DFRFT coefficients of the training set of the scintillation pulses. In the second phase, the classification vector is multiplied with the testing set of the scintillation pulses and it added to the bias of the SVM to identify the pulse type. Thus, the acceleration of the proposed algorithm depends on the multiplication of the huge number of pulses by the vector (Z). This vector is obtained in the flowing section.

The first phase of the SVM is performed to get the support vectors $S_{N \times L}$ as in equation (12). Then, the type can be identified based on equation (14). While, $P_{1 \times N}$ is the scintillation pulse with N samples and index 1, and $F_{N \times N}^R$ in

the FRFT matrix for N number of samples.

$$d(p) = \text{sign}[(P_{1 \times N} \cdot F_{N \times N}^R) \cdot \sigma_{1 \times N}] \cdot (S_{N \times L} \cdot (yg)_{L \times 1}) + b \quad (14)$$

Equation (14) can be reduced to equation (15). Where, the discrimination vector (Z) is computed as in equation (16).

$$d(p) = \text{sign}[p_{1 \times N} \cdot Z_{N \times 1} + b] \quad (15)$$

$$Z_n = (f_{n,0} \dots \dots f_{n,N-1}) \begin{pmatrix} \sigma_0 \sum_{l=0}^{L-1} s_{0,l} \gamma_l^{\alpha_l} \\ \vdots \\ \sigma_{N-1} \sum_{l=0}^{L-1} s_{N-1,l} \gamma_l^{\alpha_l} \end{pmatrix} \quad (16)$$

For n=0, 1, ...N-1

Based on (16), discrimination step are significantly reduced where the feature extraction and classification steps are merged in only single step. Hence, the unknown scintillation pulse is multiplied with the vector (Z) and the bias (b) is added to discriminate the pulse. This algorithm includes N multiplications and N+1 additions. Thus, the proposed algorithm gives low complexity. On the other hand, the processing time of the proposed algorithm is independent on the feature extractor where the vector z is computed offline.

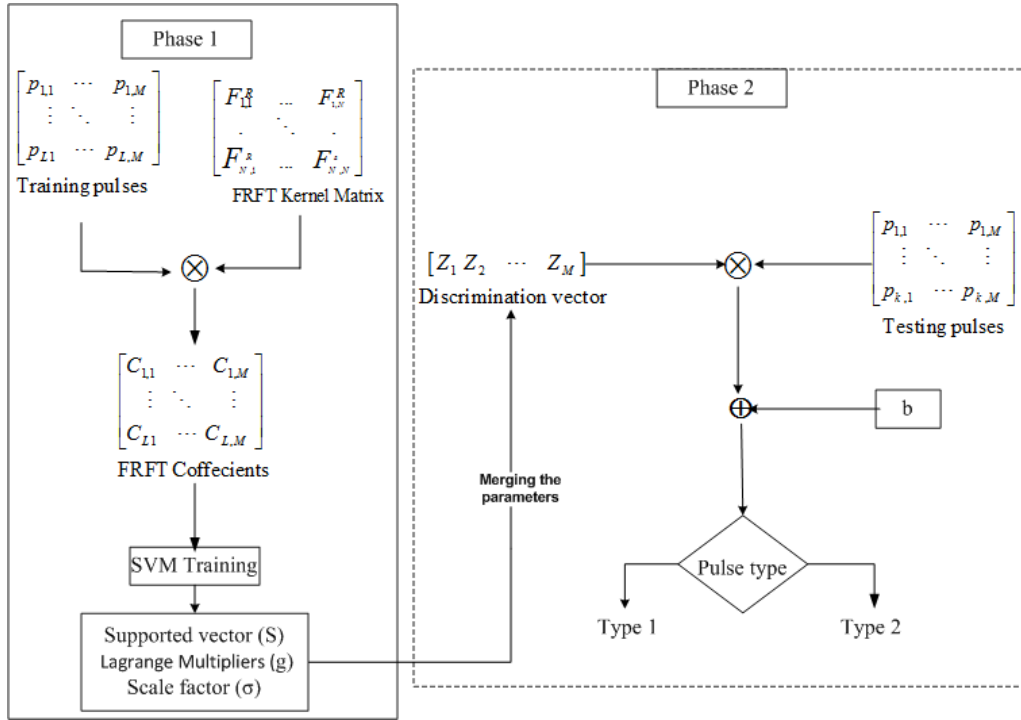


Figure 1 the proposed DFRFT-SVM PSD algorithm

4. GPU REALIZATION OF CI ALGORITHM

The proposed work is implemented to be executed on single core, multicore (CPU) or many core (GPU) based on C++, OpenMP, or CUDA respectively. Firstly, the scintillation pulses are divided into two data sets; training set and testing set. The number of testing set is 10% from the scintillation pulses. Phase one is applied on training set to get the discrimination vector (Z). Where the scintillation pulses is transformed from time domain to time-frequency domain using DFRFT. Then the coefficients of the DFRFT are trained using SVM to get the supported vectors (S) and the other parameter that are used to determine the vector (Z). This phase can be executed using single core (C++) due to the small arithmetic operations in this phase. However, phase two takes large time where huge scintillation pulses of the testing phase are tested. Therefore, this phase was executed using multicore or many cores to reduce the processing time. The speedup factors of the two different implementation algorithms are discussed in the next section. In phase two, the proposed algorithm identifies huge of scintillation pulse based matrix multiplication. The method is implemented in an approach that makes full use of the function hierarchy present within the CUDA framework. Two functions are used to implement the algorithm on the GPU; host function and

kernel function. A host function is executed and called by the CPU. However, the kernel function is executed in parallel by the GPU. The proposed algorithm can be integrated with a real time sorting algorithm that was implemented based on timestamp [4]. In this paper, pulses of the scintillation detector of the pet system were detected and recorded. Thus, the recorded pulses can be sent to a personal computer and the proposed algorithm can be applied on the recorded scintillation pulses. The CPU used in this work is Intel E5507, with 32GB of RAM and a clock speed of 2.27GHz. Two different GPUs have been used Tesla K20C and Quadro 6000. The Tesla K20 card is a dedicated scientific computation device used in professional workstations, whereas the Quadro is a graphics card that is used in a PC for games. The program was written with the minimal interaction between the GPU and the CPU without increasing the latency.

5. RESULT AND DISCUSSION

The proposed algorithm was applied on 100000 scintillation pulses of LuYAP and LSO crystals. These pulses were obtained as in [12]. The size of the LSO and LuYAP crystals is 2mm X 2mm X10 mm. ⁶⁸Ge radiation source was coupled to PMT to get the scintillation pulses. This PMT was coupled to a low pass filter with a cutoff of 3.6 MHz. Then, the output of this filter was digitized using 16 samples ADC [8, 9]. The mean and standard deviation of the two types of scintillation

pulses is shown in figure 2. The standard deviations of the tail of two types of pulses are different where the difference in decay time of the LSO and LuYAP crystal are about 20 n sec.

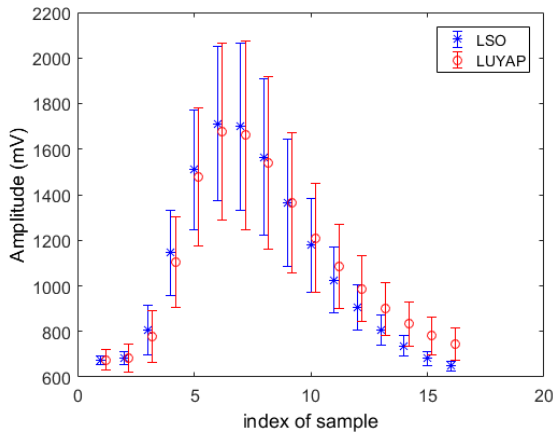


Figure 2 mean and standard deviation of the original scintillation pulses of the LSO and LuYAP crystals

To obtain the best factor of the DFRFT, the proposed algorithm was applied on the scintillation pulses then the efficiency of identification was computed. The factor is varied from 0 up to 1 with step .05 to get the best one. Figure 3 shows the efficiencies of identification for the predefined factors. The best factor is 0.2 where it separates the scintillation pulses with efficiency of 92.08%. Thus, this factor was used to get the transformation matrix of DFRFT of phase one from the proposed algorithm then we find the discrimination vector. After that, the testing phase is executed on the proposed platforms (multicore or many cores) to study the processing time, which is the aim of this work.

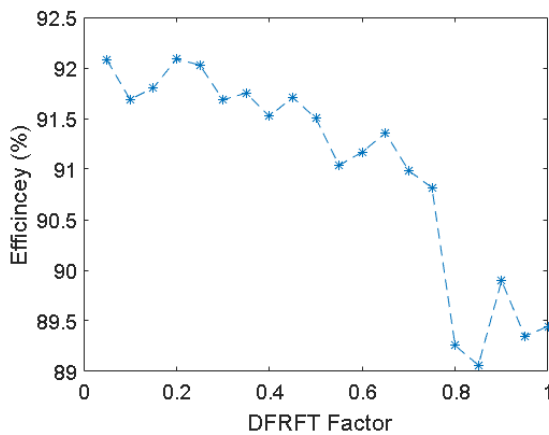


Figure 3 discrimination efficiencies of different FRFT factors

Finally, the proposed algorithm was executed on the multicore or many cores to find the speed up factor of each tool. As shown in figure 4 Tesla k20 and Quadro 6000 give a discrimination event rate of 942 M events/s and 186 M events/s respectively. On the other hand, eight cores gives 76 M events/s. Due to the ability of the GPU of parallelization, the speed up over 8 cores and 1 core is about 12.4 and 69 times respectively.

As mentioned in the introduction, the event rate of our previous work was 6.2 M events/s [3]. However, the PET systems require a high-speed rate of the CI. Table 1 represents a comparison between the previous algorithms and the

proposed one. The discrimination efficiency of the proposed algorithm are very close to the pervious algorithms. However, the discrimination rate of the proposed algorithms enhance the previous event rates in [3, 7] by factor 152 for the Tesla GPU.

Algorithm	Rate (M event/sec)	Discrimination Efficiency (%)
One Dimension ZM-BASED CI (FPGA)[10]	6.2	90.6
DFT-SVM CI (FPGA) [3]	6.2	92
Proposed CI (8 cores CPU)	76	92.08
Proposed CI (Tesla GPU)	942	92.08

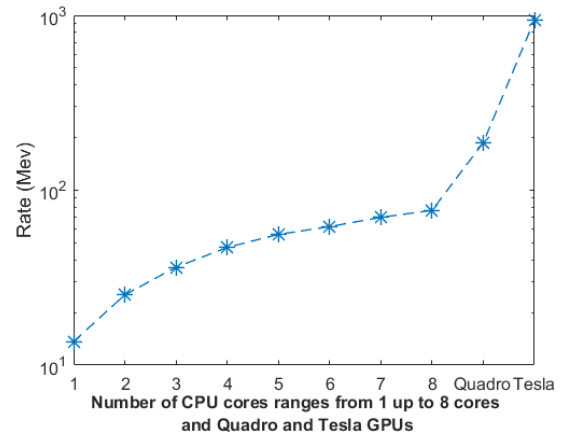


Figure 4 comparison between the discrimination rates of the multicores processor (from 1 up to 8 cores) and many cores (using Quadro and Tesla GPU)

6. CONCLUSION

A merged FRFT-SVM algorithm was proposed to perform the CI in a high rate up to 942 M event/sec using the GPU. The proposed algorithm was executed based the parallel computing in the CI algorithm. Parallel computing achieves high classification efficiency besides a high discrimination rate, which was 92.08%. Thus, high-speed applications can be implemented based on the proposed algorithm. The proposed algorithm can be integrated with a real time sorting algorithm that can be used to send the scintillation pulses to PC. On the other hand, timing information and the image reconstruction can be performed after the proposed algorithm based on GPU

7. REFERENCES

- [1] W. W. Moses, "Fundamental limits of spatial resolution in PET," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 648, pp. S236-S240, 2011.
- [2] H. Peng and C. S. Levin, "Recent Developments in PET Instrumentation," *Current pharmaceutical biotechnology*, vol. 11, pp. 555-571, 2010.
- [3] H. Saleh and A. Arafa, "Real time depth of interaction determination based on Fourier Transform and Support Vector Machine," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 872, pp. 74-79, 2017.
- [4] L. Ke, J. Yan, J. Chen, C. Wang, X. Zhang, C. Du, *et al.*, "A real-time sorting algorithm for In-Beam PET of Heavy-Ion cancer Therapy Device," *Nuclear Engineering and Technology*, 2021.

- [5] H. Singh, R. S. Venkat, S. Swagatika, and S. Saxena, "GPU and CUDA in hard computing approaches: analytical review," *Proceedings of ICRIC 2019*, pp. 177-196, 2020.
- [6] E. Sejdić, I. Djurović, and L. Stanković, "Fractional Fourier transform as a signal processing tool: An overview of recent developments," *Signal Processing*, vol. 91, pp. 1351-1369, 2011.
- [7] A. Gómez-Echavarría, J. P. Ugarte, and C. Tobón, "The fractional Fourier transform as a biomedical signal and image processing tool: A review," *Biocybernetics and Biomedical Engineering*, vol. 40, pp. 1081-1093, 2020.
- [8] M. Streun, G. Brandenburg, H. Larue, E. Zimmermann, K. Ziemons, and H. Halling, "Pulse recording by free-running sampling," *IEEE transactions on nuclear science*, vol. 48, pp. 524-526, 2001.
- [9] M. Streun, G. Brandenburg, H. Larue, H. Saleh, E. Zimmermann, K. Ziemons, *et al.*, "Pulse shape discrimination of LSO and LuYAP scintillators for depth of interaction detection in PET," *IEEE Transactions on Nuclear Science*, vol. 50, pp. 344-347, 2003.
- [10] A. Arafa and H. Saleh, "A real-time scintillation crystal identification method and ITS FPGA implementation," *IEEE Transactions on Nuclear Science*, vol. 61, pp. 2439-2445, 2014.