

Design of a Real Time FPGA-based Three Dimensional Positioning Algorithm

Nathan G. Johnson-Williams, *Student Member IEEE*, Robert S. Miyaoka, *Senior Member IEEE*, Xiaoli Li, *Student Member IEEE*, Tom K. Lewellen, *Fellow IEEE*, and Scott Hauck, *Senior Member IEEE*

Abstract—We report on the implementation and hardware platform of a real time Statistics-Based Positioning (SBP) method with depth of interaction processing for a positron emission tomography detector. The processing method works in conjunction with continuous miniature crystal element (cMiCE) detectors using a sensor on the entrance surface design. Our group previously reported on a Field Programmable Gate Array (FPGA) SBP implementation that provided a two dimensional (2D) detector positioning solution [1]. This new implementation extends that work to take advantage of three dimensional (3D) look up tables to provide a 3D positioning solution that improves intrinsic spatial resolution. Resolution is most improved along the edges of the crystal, an area where the 2D algorithm's performance suffers. The algorithm allows an intrinsic spatial resolution of ~0.90 mm FWHM in X and Y and a resolution of ~1.90 mm FWHM in Z (i.e., the depth of the crystal) based upon DETECT2000 simulation results that include the effects of Compton scatter in the crystal. A pipelined FPGA implementation is able to process events in excess of 220K events per second, which is greater than the maximum expected coincidence rate for an individual detector. In contrast to all detectors being processed at a centralized host (as in the current system) a separate FPGA is available at each detector, thus dividing the computational load. A prototype design has been implemented and tested on an Altera Stratix II FPGA using a reduced word size due to memory limitations of our commercial prototyping board.

I. INTRODUCTION

A continuous miniature crystal element (cMiCE) detector is a low cost alternative to discrete crystal detector modules that have traditionally been used to achieve high spatial resolution for small animal positron emission tomography (PET) scanners. A key to the imaging performance of the cMiCE detector is the use of a statistics based positioning (SBP) algorithm [2,3]. SBP is used to improve the resolution of the monolithic sensor module and increase the effective imaging area of the sensor compared to Anger positioning. The basis of SBP involves the calculation of the most likely location of an event given the sensor outputs. We previously

Manuscript received April 2, 2010. This work was supported in part by Altera, DOE DE-FG02-08ER64676, and NIH grants EB001563, EB002117, and CA136569.

N. Johnson-Williams, and S. Hauck are with the University of Washington Department of Electrical Engineering, Seattle, WA USA (e-mail: njohns@Sandia.Gov, hauck@ee.washington.edu).

R. Miyaoka, and T. Lewellen are with University of Washington Department of Radiology, Seattle, WA USA.

X. Li is with the University of Washington Department of Physics, Seattle, WA USA.

reported on a field programmable gate array (FPGA) implementation of our SBP method for a two dimensional (2D) solution (i.e., X and Y positioning) for a cMiCE detector [1].

While a majority of events occur near the entrance surface of a detector, depth of interaction varies from event to event. Accurate calculation of the depth of interaction permits more accurate calculations of the coincidence line of response (LOR) and yields better intrinsic spatial resolution performance for our cMiCE detectors.

A three dimensional (3D) SBP algorithm provides a solution that includes the X and Y dimensions and the depth of interaction. While the primary goal of finding the depth of interaction is for use with the LOR, the 3D SBP algorithm also provides better X and Y resolution. The use of depth of interaction in calculations allows a closer match between sensor outputs and characterization tables. The advantage of properly matched characterization tables is most evident near the edges and corners of the crystal where the resolution of 2D SBP usually suffers.

II. DETECTOR DESIGN AND METHODS

A. cMiCE Detector Module

The cMiCE detector module is modeled as being composed of a 49.6 mm by 49.6 mm by 15 mm LYSO crystal coupled to a 2D array of sensors [Fig. 1]. The module uses a novel sensor on the entrance surface (SES) design which places the sensors on the entrance of the crystal [4]. The advantage of the SES design is that it places the sensors closer to where the majority of events occur.

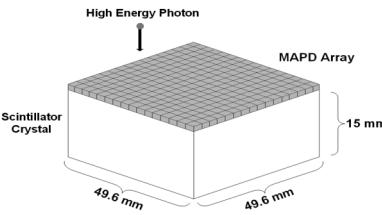


Fig. 1. Illustration of detector module with sensor on the entrance surface design. The sensor array contains 256 (i.e., 16x16) MAPD elements.

The sensor array consists of a 16 by 16 array of micro-pixel avalanche photodiodes (MAPD). MAPDs are one vendor's version of the Geiger mode APD [5,6]. To reduce the computational load of handling 256 individual sensor outputs

row-column summing is used, which reduces the 16 by 16 array to 32 outputs.

Simulation tools were used to determine the light response function of the described cMiCE detector. To summarize, DETECT2000 [7,8] is used to determine the probability that a light photon generated at a specific (X, Y, Z) position in the crystal is detected by a specific photosensor. GEANT [9] is used to track the gamma interactions (both Compton and photoelectric) within the crystal. For each interaction, the number of light photons produced by the scintillator crystal is determined. That number is adjusted for the non-proportionality of LSO according to the tables reported by Rooney [10]. While the non-proportionality tables reported by Rooney were for LSO, Rooney, Balcerzyk [11] and Chewpraditkul [12] show that LYSO and LSO have similar non-proportionality characteristics. Poisson noise is then added to the number of light photons produced.

B. Statistics-based Positioning Method (SBP)

Suppose the distributions of the 32 row-column sum outputs, $M = m_1, m_2, \dots, m_{32}$, for scintillation position X,Y are independent normal distributions with mean $\{\mu_i(x)\}$ and standard deviation $\{\sigma_i(x)\}$. The likelihood function for making any single observation m_i given x is:

$$L[m_i | x] = \prod_{i=1}^{32} \frac{1}{\sigma_i(x)\sqrt{2\pi}} \exp\left(-\frac{(m_i - \mu_i(x))^2}{2\sigma_i^2(x)}\right) \quad (1)$$

The maximum likelihood estimator of the event position x is given by:

$$\arg \min_x \left[\sum_i \frac{(m_i - \mu_i(x))^2}{2\sigma_i^2(x)} + \ln(\sigma_i(x)) \right] \quad (2)$$

The mean and variance of the light probability density function (PDF) are created by characterizing the detector module. For the two dimensional method the module is characterized at a resolution of 127x127 (X,Y). Each point is spaced 0.3875 mm apart, giving a uniform distribution across the 49.6 mm X 49.6 mm crystal. For more detailed explanations about the SBP implementation refer to Joung and Ling [1,2]. Methods to extend the 2D characterization for SBP to 3D are fully described in references [4, 13]. For the 3D SBP implementation 3D mean and standard deviation look up tables are created. A maximum likelihood clustering method is used to extract depth of interaction (DOI) information from the detector during the characterization process.

C. Three Dimensional SBP

The same principles of the two dimensional SBP algorithm can be applied to a 3D algorithm. Instead of two dimensions, the light response function is characterized in 3D as described in [4, 13], giving μ_i and σ_i as a function of the X, Y and Z position in the crystal. The module is characterized for 127x127x15 positions, with 127x127 representing the X and Y dimension, and each X and Y position characterized for 15 depths. As with the 2D implementation, each X, Y position is on a grid with each point separated by 0.3875 mm. The 15 depths, however, are not equally distributed over the 15 mm deep crystal; depth is characterized at a higher resolution near the entrance of the crystal where events are more likely to

interact [Table 1, Fig. 2]. In total the module is characterized for 241,935 different positions, 15 times more than the 2D implementation. The likelihood function of 3D SBP is seen in Equation 3; it is identical to the 2D likelihood function, but values are characterized for all three dimensions rather than two.

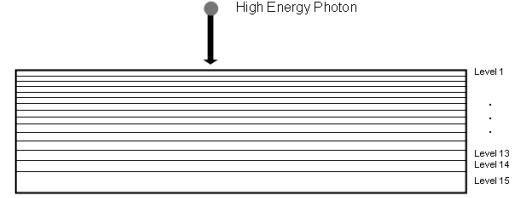


Fig. 2. Illustration of the distribution of depth levels.

	Start Depth (mm)	End Depth (mm)
Level 1	0.0000	0.8275
Level 2	0.8275	1.4131
Level 3	1.4131	2.0290
Level 4	2.0290	2.6784
Level 5	2.6784	3.3653
Level 6	3.3653	4.0943
Level 7	4.0943	4.8709
Level 8	4.8709	5.7016
Level 9	5.7016	6.5948
Level 10	6.5948	7.5604
Level 11	7.5604	8.6113
Level 12	8.6113	9.7640
Level 13	9.7640	11.0404
Level 14	11.0404	12.4703
Level 15	12.4703	15.0000

Table 1. Location of depth levels in the 15 mm thick scintillator crystal.

$$\begin{aligned} \ln(P(X, Y, Z)) = & \\ - \sum_{row=1}^{16} & \left[\frac{[event_{row} - \mu_{row}(X, Y, Z)]^2}{2[\sigma_{row}(X, Y, Z)]^2} + \ln(\sigma_{row}(X, Y, Z)) \right] \\ - \sum_{column=1}^{16} & \left[\frac{[event_{column} - \mu_{column}(X, Y, Z)]^2}{2[\sigma_{column}(X, Y, Z)]^2} + \ln(\sigma_{column}(X, Y, Z)) \right] \end{aligned} \quad (3)$$

D. Hierarchical Search

A hierarchical search can be used to produce a SBP solution in many fewer operations than an exhaustive search. A hierarchical search begins with a coarse grained search covering the entire field. The most likely point becomes the center point of a more refined search carried out in the next stage. This process continues with each stage becoming more refined until the maximum resolution is reached in the final stage [Fig. 3]. Each search stage compares nine different points equally distributed across the search area. The likelihood of each point is calculated, and the most likely of the nine points is selected as the center point for the next more refined search. Comparing 9 different points in each search stage allows the hierarchical SBP algorithm to solve for the positioning using a 127x127 grid in six stages. In total, 49 data point computations are required, which is a drastic reduction from the 16,129 data point calculations needed for an exhaustive search of the X, Y dimensions.

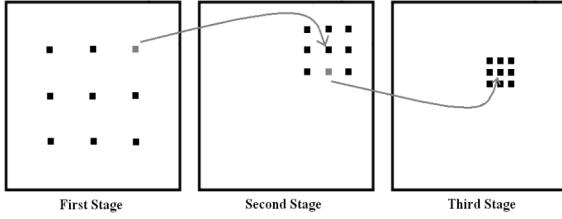


Fig. 3. Example of the first three stages of a hierarchical solution for 2D SBP.

E.Two Dimensional SBP Implementation

A 2D hierarchical SBP algorithm has been designed and implemented on an FPGA development board [1]. The implementation produced similar accuracy to an exhaustive search implementation with a drastically reduced number of calculations.

The six stages of the hierarchical search were pipelined in the FPGA implementation [Fig. 4]. In this arrangement the algorithm could produce a solution with a much higher throughput (i.e., the time it takes to solve a single stage). The FPGA implementation was capable of processing data in real time, in excess of 350,000 events per second.

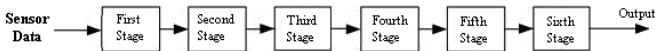


Fig. 4. Pipelined implementation of Hierarchical 2D SBP.

III.ALGORITHM DESIGN

A.Hardware Limitations

The limiting factor in implementing the SBP algorithm is the large data tables required for the fine grained search stages toward the end of the pipeline. In the final stage of the 2D implementation the sixth search stage required a data table over 2.75MB. Since 3D SBP characterizes the light response function for 15 depths, it is expected that the required tables will be significantly larger.

An interesting limitation in the hardware design is a ceiling for the clock frequency. The PET system is being designed to be able to operate within the imaging field of view of an MRI scanner. To avoid harmful interference caused by the PET insert, the hardware must be designed to avoid operating at the MRI's resonant frequency of ~120 MHz. Since operating well above this frequency is difficult with many FPGAs, a clock frequency below 120 MHz was selected. 70 MHz was established as a frequency which would allow high performance yet is significantly different from the MRI's resonant frequency.

With a custom board in progress a commercial development board was used to aid in the design and implementation of the 3D algorithm. The development board used is the Stratix II DSP Development Board Professional Edition. Of particular interest to this application is the available ~8Mb of on-chip memory, 32Mb of off-chip SRAM memory, and 128Mb of off-chip SDRAM memory.

B.Algorithm Derivation

Many of the same ideas and approaches from the 2D implementation were carried over to the 3D implementation. As a basic approach, the hierarchical search method was used to reduce the number of calculations required to produce a solution. Since the 3D implementation adds an extra dimension and the detector module is slightly different some changes were needed.

The simplest way to extend the 2D hierarchical search to a 3D hierarchical search is to expand some of the stages in the previous 2D algorithm to 3D stages. The 3D search treats the depth of interaction as an independent variable (just like the X and Y dimensions), and adds the depth variable to the search algorithm. A 3D stage uses a volumetric search, performing the same basic comparison operation as a 2D stage but comparing a distribution of points across the X, Y, and Z dimensions in place of just the X and Y dimensions [Fig. 5]. The 3D stage finds the best fit combination of X, Y, and, Z within the stage's distribution of points.

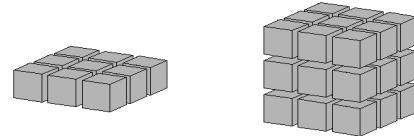


Fig. 5. Grouping of voxels in a 2D search (left) and grouping of voxels in a 3D search (right).

Functionally, a 3D stage works similar to a 2D stage with one major difference; a 3D stage uses a 3D characterization table in place of a 2D characterization table. In a 3D table, values are based on all three dimensions, rather than in a 2D table where X and Y values are averaged according to the expected distribution over all depths. The 3D table allows a 3D stage to examine independent points across all three dimensions and select the best fit point in X, Y, and Z. The computation process in a 3D stage is the same as in a 2D stage: the likelihood of each point is calculated with the best fit becoming the center point in the more refined search carried out in the next stage.

For the given system, the depth of interaction is defined by 15 depths. As with the XY planar search, the depth search can be divided into a hierarchical search, reducing the number of computations required to produce a solution. Dividing the 15 levels equally into a hierarchical search would allow the depth solution to be found in only 3 stages, with each stage comparing three depths. The most likely depth is selected and becomes the center point of a more refined search in the next stage [Fig. 6]. A solution for depth is reached after three stages.

Since the existing pipeline has six stages, only half of the stages will be converted to 3D stages. Stages not converted to a 3D search will remain two dimensional to complete the XY search. Since only half of the search stages are required to be volumetric, the order of those stages in the pipeline becomes a variable for optimization of accuracy and memory requirements.

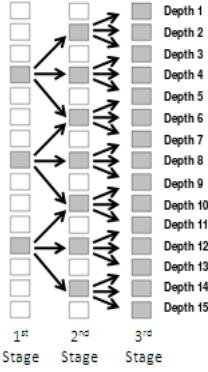


Fig. 6: Outline of hierarchical search to find depth among 15 levels.

Planar searches following any volumetric stage in the pipeline will also use three dimensional characterization tables. The depth of interaction found in the preceding stages is used as a per-event constant, with the 3D characterization table producing a 2D characterization table matched to the calculated depth of interaction. The matched table offers the advantage that characterization values from the table are based on a specific depth, which is more accurate than a 2D table where the values are based on an average depth.

The principle downside to the volumetric search stage approach is the increased number of computations required. As outlined above, each volumetric search in the pipeline performs a $3 \times 3 \times 3$ search, 3 times as many calculations as the original 2D search. With memory bandwidth limitations restricting each stage to operating on one point at a time, the 3D stage has a critical path of 27 calculations. Replacing any 2D stage with a 3D stage increases the overall critical path length in the pipeline to that of the 3D stage, reducing throughput by a factor of three.

The memory requirement for a 3D search is also drastically increased. Each volumetric search and each 2D stage following a volumetric search requires a 3D table. The size increase of a 3D table over a 2D table is dependent on the number of levels included. For example, a 3D table with all 15 depths requires 15 times the memory of the corresponding 2D table. However, since the depth search is organized as a hierarchical search not every volumetric search requires a full 3D table. The table for the first of the three volumetric stages is increased by a factor of three, the second by a factor of seven, and the third by a factor of fifteen. Any 2D stage following a 3D stage is required to have at least as many depths as the preceding 3D stage. With the 2D sixth stage table already requiring close to 3MB, converting to 3D tables is a significant increase in capacity requirements.

To evaluate trade offs the algorithm was tested with the volumetric search stages placed in different locations in the pipeline to study the effects of volumetric search placement compared to system accuracy [Table 2]. The results show that placing the depth search too early or too late in the pipeline is detrimental to system accuracy across all dimensions. When the volumetric stages were placed early, the X and Y positioning proved to be too coarse/inaccurate to provide a good basis for a depth search. When placed late, results indicated that many of the calculations early in the pipeline were inaccurate due to a lack of information on the actual depth of interaction. Placing the volumetric searches toward

the middle of the pipeline allowed a sufficiently accurate X, Y basis for calculating the depth while producing a depth solution early enough to increase the accuracy of the finer-grained X, Y searches. The best combination tested places the volumetric searches third, fourth, and fifth in the pipeline.

Table 2. Accuracy of SBP Algorithm with volumetric search stages placed at different locations in the pipeline.

Location of Volumetric Searches	FWHM-XY (mm)	FWHM-Z (mm)
Stages 1,2,3	1.03	3.49
Stages 2,3,4	0.95	1.95
Stages 3,4,5	0.91	1.87
Stages 4,5,6	0.92	1.88

Placing the volumetric searches third, fourth, and fifth in the pipeline produces very good results; however there are several hardware limitations which prohibit this arrangement. The added memory requirement of the volumetric searches would exceed the capacity of the on-chip memory currently available, which would move the later volumetric stage's table off-chip where memory bandwidth is much more limited. Total memory capacity is also a problem since the sixth stage table is required to be fifteen times larger than the original 2D table. The number of computations required for each volumetric search is also prohibitive. Given the current hardware design, there is insufficient memory bandwidth to perform the 27 calculations required for a volumetric search and meet the throughput requirements of the system. The existing hardware requires at least 600 clock cycles for the 27 calculations associated with a volumetric search stage. This yields a throughput of 117,000 events per second, 0.7X the required throughput.

To increase the throughput of the system, the volumetric search can be split into two separate independent stages: an XY planar search followed by a linear Z search. Using this method the algorithm will alternate between a 3×3 planar search and a hierarchical depth search stage. The two stages combined do not perform the same operation as the volumetric search; in place of comparing all possible points (X,Y,Z) the algorithm first solves for X, Y (9 calculations), and then Z (3 calculations) [Fig. 7]. This change dramatically increases the throughput of the algorithm, replacing the 27 calculation stage with a 9 calculation stage and a 3 calculation stage.

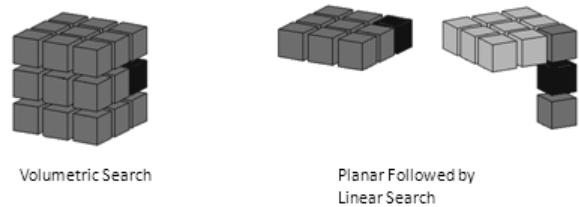


Fig. 7. Comparison between a volumetric search stage (left) and a planar/linear search stage (right).

Both the planar search and the linear search will use 3D characterization tables. The planar search will use the depth information found in the preceding stages as a constant, solving for the best fit X, Y at the given depth. The linear search will use the X, Y information found in the previous

stage as a constant, solving for the best depth fit at the given X, Y. However, the additional 3 independent depth search stages significantly increases hardware requirements over the single volumetric search stage. By having both the planar and linear searches as independent stages, the memory bandwidth and capacity requirements are significantly increased compared to the volumetric search, as each stage would require its own characterization tables and perform their calculation independently.

At the cost of some throughput, the increase in required memory bandwidth and capacity requirements can be addressed. The independent planar search and linear search can be grouped into a single stage performing two operations. In this configuration the stage first performs a planar search, followed by a linear search. This allows the planar search and linear search to share one memory table. The cost of this is an increase in the critical path caused by the addition of the 3 linear search stage calculations to the planar stage. Compared to the original volumetric search approach first described, this variation allows an algorithm with no additional memory bandwidth or memory capacity requirements, but with a greater than 2X increase in throughput.

Using the same test setup as the volumetric search algorithm, several orderings of the planar/linear stages were compared [Table 3]. As was seen in Table 1 the algorithm has its greatest accuracy when depth related search stages are placed towards the middle of the pipeline. The best option is placing a hybrid search for stages 3, 4 and 5 in the pipeline. Compared with the best option from table 1, the hybrid search stage method has similar accuracy.

Location of Linear/Planar Stages	FWHM-XY (mm)	FHWM-Z (mm)
Stages 1,2,3	1.20	3.52
Stages 2,3,4	0.95	1.95
Stages 3,4,5	0.90	1.87
Stages 4,5,6	1.02	1.90

Table 3. Accuracy of SBP Algorithm with linear/planar search stages placed at different locations in the pipeline.

Another option available to increase throughput is placing the depth search by itself in a single stage, i.e., performing the entire depth search at once. Using the previously outlined hierarchical depth search (figure 6), the DOI can be found in a single stage in 7 calculations. Given that a hierarchical search can find the depth in 7 calculations, placing the depth search in a single stage maintains the original 2D search's critical path length of 9 calculations. Also, by compressing all of the depth searches into a single stage the memory tables needed for the coarser grained depth searches can be eliminated, reducing the capacity requirement of on-chip memory.

Table 4 shows the simulated results of the algorithm with the linear depth search placed at different locations in the pipeline. Compared with the results from table 2, it can be seen that performing the entire depth search in a single stage has little effect on the accuracy of the system. Based on the results shown in Table 3 there is no benefit in performing the depth search after the fourth stage of planar X, Y search. Since the depth search characterization table size increases by a

factor of 4 for each planar search preceding it, we elected to place the depth search after the third X, Y search, where a high degree of accuracy is achieved without exceeding the capacity of the development board.

Location of Linear Depth Search	FWHM-XY (mm)	FHWM-Z (mm)
After Stage 1	1.25	3.82
After Stage 2	1.00	2.36
After Stage 3	0.90	2.02
After Stage 4	0.91	1.89
After Stage 5	1.14	1.92
After Stage 6	1.09	1.91

Table 4. Accuracy of SBP Algorithm with single hierarchical search stages placed at different locations in the pipeline.

A common problem to all the previously described methods is the large memory capacity required. The full data tables for the later stages of the algorithm are quite large, with the sixth stage 15-depth table requiring 45MB of storage space. The total memory requirement for the system exceeds 60MB, an amount greater than the development board capacity.

Once the pipeline determines the depth, using a 3D characterization table for subsequent stages allows a better match between sensor outputs and table values based on the known depth rather than an average or assumed depth. However, not all 15 depths are required to take advantage of this. In the case of stages 4, 5 and 6 the full data table is not required, and a simplified 3D table can be used which has a reduced number of levels. This reduction saves memory at the potential cost of lower accuracy.

In general, the greater number of levels that can be used, the greater the accuracy. However, the results indicate that the number of depths used is subject to diminishing returns. Overall, the best compromise in our sweep of the space was 4 depths are used in stage 4, 8 in stage 5, and 15 in stage 6 [14]. This corresponds to the following memory requirements: 768KB for the 4th stage, 6MB for the 5th stage, and 45MB for the 6th stage.

However, the development board cannot support such a large memory requirement. Following the general trend, the best combination will be the most detailed characterization tables that can fit on the development board. After allocating some of the on-chip memory for the first three stages and the depth stage, roughly 0.2MB remains for the fourth stage. This is only sufficient space for a single depth 4th stage table. The off chip SRAM has 32KB of addressing, allowing at most a 2 depth 5th stage table. The 500KB of addressing for the SDRAM memory is sufficient to allow an 8 depth 6th stage table.

The number of depths used to squeeze into the development board was 1 depth for stage 4, 2 depths for stage 5, and 8 depths for stage 6. While this solution is not ideal, it is a significant improvement over using a single depth for the final stages, and represents a significant savings in memory from the full version. Reducing stage 4 from a 15 depth table to a 1 depth table saves 2.625MB, while the reduction in stages 5 and 6 save 9.75MB and 24MB respectively. The reductions amount to a total 58% lower memory capacity requirement for the system.

Combining observations from the previous tests gives the basic framework of the 3D algorithm [Fig. 8]. The first three stages are XY planar searches, similar to ones used in the original 2D algorithm. The depth of interaction is found in a linear hierarchical search after the third stage and the depth is then used as part of the planar searches in stages 5 and 6.

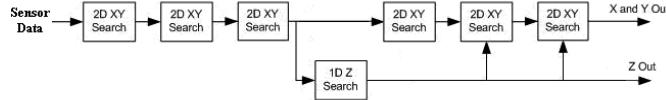


Fig. 8. Basic framework of 3D SBP hierarchical search.

A. Dynamic Breadth Search

In the ideal case the solution set forms a hyperbolic paraboloid, with a smooth and consistent gradient to the most likely solution. In this ideal case a hierarchical search will consistently produce the same result as an exhaustive search, as any point closer to the absolute minimum has a greater likelihood, allowing the algorithm to always converge to the correct solution. However, in practice the solution set is rarely ideal and often has local minima or an uneven gradient.

Local minima are detrimental to positioning accuracy and can lead to artifacts in an image. A local minimum is a portion of the solution set which breaks from the general gradient of the set and, when viewed in a limited scope, appears to be the absolute minimum of the system. As a result of a local minimum, a hierarchical search can incorrectly select a point farther from the absolute minima in coarse grained stages and eventually converge to the local minima as a solution in place of the true minimum. Generally, the local minima that can trap the hierarchical search are located near the true minimum. Local minima and noise that have a great effect on system accuracy have the highest occurrence on the outer edges of the crystal where the signals are truncated and reflections are most likely to occur. Examining the results of data sets across a range of positions on the crystal show that events occurring within 5 mm of the crystal edge produce significantly poorer results than positions located closer to the center of the crystal [14].

A high occurrence of the same local minima leads to a tendency to produce incorrect solutions at the same location with a higher than normal frequency. When viewing a large number of samples for a data point with local minima, the distribution of the results shows multiple peaks: often the correct solution and a peak occurring at each location of a local minimum [Fig. 9].

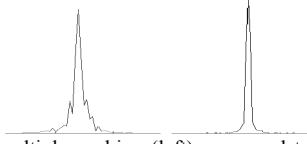


Fig. 9. Example of multiple peaking (left) compared to the desired solution (right) found with an exhaustive search.

Additional peaks, even those close to the correct solution, have a negative effect on the usefulness of the PET system. When reconstructing an image from the positioned data it is impossible to determine the difference between a peak caused by a correct event or by a local minimum. Incorrectly positioning events caused by a local minimum can lead to

artifacts in an image. Inversely, ignoring closely grouped peaks that might be caused by local minima but are actually different data points can eliminate valuable spatial information. Since image reconstruction cannot differentiate between the two, it is important to eliminate the problem at the source and ensure that the hierarchical search is not affected by local minima in the solution set.

Positions located in the corner of the crystal are more affected than those that are near only one edge. Nearly every data point within 5 mm of the corner had instances of multiple peaking. If the data point is located along a single edge the occurrence is generally much lower and less severe. Data points in the center of crystal do not have a problem with multiple peaking as illustrated in Fig. 9.

Since the local minima have a tendency to be located near the true minima, a broader fine grain search becomes a viable correction option for additional peaks in the distribution (Fig. 10). A broader 5x5 search increases the range of the search by 0.3875 mm in all directions and a noticeable improvement is seen over a 3x3 search. A 7x7 search increases the range of the search by 0.775 mm in each direction and even further improvements can be seen. When expanded to a 9x9 search there was little benefit over a 7x7 search.

The cost of the expanded search is increased computational requirements. Compared to a 3x3 search, a 5x5 search requires 2.7 times as many calculations. A 7x7 search requires 5.4 times as many calculations as a 3x3, and a 9x9 search requires 9 times as many calculations. A broader final search stage, while addressing the multiple peak issue, has a great effect on system throughput. Replacing the final stage with a broader search reduces throughput by 2.7X for a 5x5, 5.4X for a 7x7, and 9X for a 9x9. Since 9x9 search requires a significantly greater number of calculations, with an only slight benefit over the 7x7, it is not considered.

The majority of crystal positions receive little to no benefit from using an expanded search. Examining system accuracy and occurrences of multiple peaks shows that only positions within 5 mm of the crystal edge benefit by a broader search. Approximately 37% of the crystal area falls within 5 mm of an edge. Given an equal distribution of events across the crystal this corresponds to 37% requiring a broader search. If only events within 5 mm of the edge of the crystal invoke a 7x7 search, throughput is reduced to 0.38X compared to a 3x3 final stage. Further improvements can be made using a 5x5 search for data located near only one edge of the crystal and a 7x7 search for data located in the corner of the crystal [Fig. 11]. This modification results in an increase of throughput of 1.5X over only using a 7x7 search, reducing throughput to .57X when compared to the 3x3 search.

A 7x7 search is computational intensive, requiring more than five times the calculations of a 3x3 search. While only a fraction of data points are expected to invoke a 7x7 search, the computational intensity of those select data points have a profound impact. While multiple final stage data tables allowing parallel calculations would be ideal, the size of a full table makes this solution impractical. Alternatively, in the event of a 7x7 search the pipeline is stalled while the 7x7 search performs the extra calculations; in this situation the characterization tables of the previous stage are unused for

much of the time. In an effort to increase the throughput of a 7x7 search, the fifth stage table can be used to perform some of the calculations. Nine of the points in a 7x7 search are available in the fifth stage table; performing those calculations in parallel increases the throughput of a 7x7 search by 22%. Overall this change results in an expected 1.05X speed up averaged across all data points, compared to using the sixth stage table by itself.

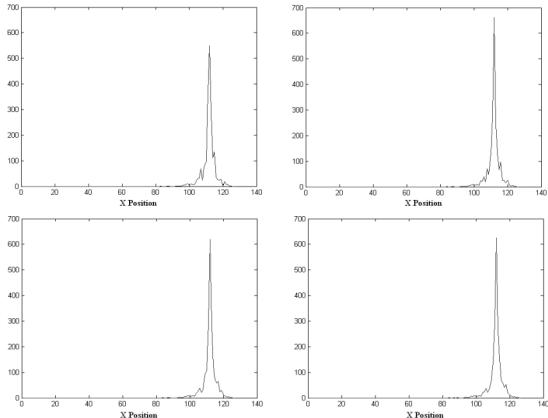


Fig. 10: Example of X distribution for 2000 samples of a point near the corner of the crystal. Different final stage search breadths are used for each distribution: 3x3 (top left), 5x5 (top right), 7x7 (bottom left), 9x9 (bottom right).

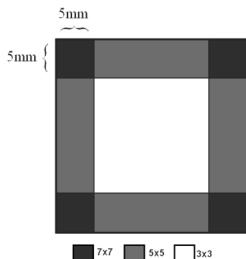


Fig. 11: Search breadth regions.

The decision of search breadth is based on the position found in the preceding stage. When the preceding stage indicates that the location of the event is within 5 mm of the edge of the crystal the algorithm is adjusted accordingly. If the previous stage indicates that the position is near the edge the algorithm enters the 5x5 search state, and if the previous stage indicates a corner position the algorithm enters the 7x7 search state. Note that the state of the final stage has no effect on the operations of the preceding stages.

II.FPGA IMPLEMENTATION

The first step in implementing the design in hardware is modifying the calculations to be better suited for hardware implementation. All floating point numbers used in the likelihood calculation are converted to fixed point numbers with a precision high enough that accuracy is not lost due to this conversion.

Since many values used in the calculation are stored in a lookup table, there is the possibility of pre-calculating some values before storing them. This increases throughput by reducing the number of operations that must be performed on

the FPGA. There are several cases in the SBP equation [Equation 2] where this is possible. The standard deviation value used in the equation is involved in two computationally intensive operations: a multiplication and a natural log. Since these operations are performed on only the standard deviation value and not any external independent variable they can be pre-computed. In place of being stored as simply the standard deviation they are stored as $\frac{1}{\sqrt{2 \times \delta}}$ and $\ln(\sigma)$. With pre-

computed values the equation is simplified to four operations: A subtraction, two multiplications, and an addition [Equation 4]. In each stage the mathematics were pipelined to take full advantage of the available memory bandwidth.

$$\left((E - i) \cdot \frac{1}{\sqrt{2 \cdot \delta}} \right)^2 + \ln(\sigma) \quad (4)$$

In addition, for many events some sensor row/columns do not measure any light. In most cases a reduced number of rows/columns can capture the majority of the light response. Testing showed that using the highest reading row and the nearest ten rows and the highest reading column and nearest ten columns produces results with no significant difference compared to using all 32 row/columns. Throughput is improved by using this reduced number of row/columns for the likelihood calculations. The reduction from using all 32 row/columns to 22 gives a significant improvement in throughput, increasing it by 44%.

III.RESULTS

The new 3D SBP algorithm produces higher quality results than the previous 2D SBP for our new detector module. The difference between the two algorithms is greatest at the edge of the crystal where the 3D method produces greatly improved results and the 2D method typically performs poorly (Table 5). Another advantage that is not captured by the FWHM results is that when using a 2D look up table there is a significant problem with “multiple peaking” of the positioning estimate. This has been corrected in part by using 3D SBP and further improvement was gained by the use of the dynamic breadth search.

The 3D SPB algorithm was pipelined and has been implemented on a commercially available FPGA development board using Verilog. With a 70 MHz clock the implementation can process events in excess of 220k events per second, which is greater than the maximum expected coincidence rate for an individual detector. In comparison to an exhaustive search, which compares the likelihood of every point, the implemented algorithm produces a solution in much fewer calculations and with similar accuracy [Table 6].

The algorithm design was limited by both the memory capacity and bandwidth of the FPGA development board. A custom board with additional memory banks and capacity is currently under design for this specific application. While the development board implementation provides acceptable levels of accuracy, additional memory provides room for improvement. The new board under development features three very large banks of off chip memory that will add a greater degree of flexibility to the design.

Most notably the addition of the third bank of off chip memory will allow the depth search to be placed in the optimal location between X, Y stage 4 and X, Y stage 5. Memory limitations of the development board required the depth search to be placed after the third X, Y stage, a location that provides very good although not ideal results. Additionally, the larger off chip memory banks allow more detailed characterization tables to be stored for stage 5 and stage 6. With the larger banks, the full 15 depth tables can be stored in both locations, while the development board only allowed a 2 depth stage 5 and a 4 depth stage 6.

Distance From Edge of Crystal (mm)	FWHM-XY (mm)	
	2D Algorithm	3D Algorithm
10.46	0.8	0.94
8.91	0.81	0.67
7.36	0.9	0.41
5.81	1.4	0.94
4.26	1.58	0.74

Table 5. Comparison between two and three dimensional SBP methods

Method	FWHM-XY (mm)	FHWM-Z (mm)
Exhaustive SBP	0.9	1.88
3D Hierarchical SBP	1.02	2.07

Table 6. Comparison between exhaustive and three dimensional SBP methods.

The largest difference in performance between the 2D and 3D method is seen in the results from near the edge of the crystal. While the FWHM of the 2D method worsens as it approaches the edge of the crystal, the 3D method produces results more consistent with those seen at points closer to the center of the crystal. For points closest to the center of the crystal the two methods produce very similar results on average. It is important to note that the 3D method does not universally produce better results. The two methods use different enough processing methods that the random noise can, on occasion, cause the 2D method to outperform the 3D method. The occurrence of these instances does not appear to follow a pattern or occur frequently enough to be significant.

VI. REFERENCES

- [1] D. DeWitt, N.G. Johnson-Wiliams, R.S. Miyaoka, X. Li, C. Lockhart, T.K. Lewellen, S. Hauck, "Design of an FPGA based algorithm for real-time solutions of statistics-based positioning," *IEEE TNS*, vol. 57(1): pp. 2771-2777, February 2010.
- [2] J. Joung, R.S. Miyaoka, T.K. Lewellen, "cMiCE: a high resolution animal PET using continuous LSO with a statistics based positioning scheme," *NIM Phys. Res. A*, vol. 489, no. 1-3, pp. 584-589, Aug. 2002.
- [3] T. Ling, K. Lee, R.S. Miyaoka, "Performance comparisons of continuous miniature crystal elements (cMiCE) detectors," *IEEE TNS*, vol. 53, pp. 2513-2518, 2006.
- [4] X. Li, C. Lockhart, T.K. Lewellen, R.S. Miyaoka, "A high resolution, monolithic crystal PET/MRI detector with DOI positioning capability," *IEEE EMBS*, pp. 2287-2290, 2008
- [5] Z. Sadygov, A. Olshevski, I. Chirikov, I. Zheleznykh, A. Novikov, "Three advanced designs of micro-pixel avalanche photodiodes: Their present status, maximum possibilities and limitations," *NIM Phys. Res. A*, vol. 567, pp. 70-73, 2006.
- [6] V. Saveliev and V. Golovin, "Silicon avalanche photodiodes on the base of metal-resistor-semiconductor (MRS) structures," *NIM Phys. Res. A*, vol. 442, pp. 223-229, 2000.
- [7] G. Tsang, C. Moisan, and J. G. Rogers, "A simulation to model position encoding multicrystal PET detectors," *IEEE Transactions on Nuclear Science*, vol. 42, pp. 2236-2243, 1995.
- [8] G. F. Knoll, T. F. Knoll, and T. M. Henderson, "Light Collection in Scintillation Detector Composites for Neutron Detection," *IEEE Transactions on Nuclear Science*, vol. 35, pp. 872-875, 1988.
- [9] S. Agostinelli, et al., "Geant4 – a simulation toolkit," *NIM Section A*, vol. 506, pp. 250-303, 2003.
- [10] B. D. Rooney and J. D. Valentine, "Scintillator light yield nonproportionality: Calculating photon response using measured electron response," *IEEE Transactions on Nuclear Science*, vol. 44, pp. 509-516, 1997.
- [11] M. Balcerzyk, M. Moszynski, M. Kapusta, D. Wolski, J. Pawelke, C.L. Mechler, "YSO, LSO, GSO and LGSO. A study of energy resolution and nonproportionality," *IEEE Transactions on Nuclear Science*, vol. 47(4), pp. 1319-1323, 2000.
- [12] W. Chewpraditkul, L. Swiderski, M. Moszynski, T. Szczesniak, A. Syntfeld-Kazuch, C. Wanarak, P. Limsuwan, "Scintillation properties of LuAG:Ce, YAG:Ce and LYSO:Ce crystals for gamma-ray detection," *IEEE Transactions on Nuclear Science*, vol. 56(6), pp. 3800-3805, 2009.
- [13] T. Ling, T.K. Lewellen, R.S. Miyaoka, "Depth of interaction decoding of a continuous crystal detector module," *Phys. Med. Biol.*, vol. 52, pp. 2213-2228, 2007.
- [14] N. Johnson-Williams, "Design of a Real Time FPGA-based Three Dimensional Positioning Algorithm," Master's thesis, Department of Electrical Engineering, University of Washington, 2009