

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

Nathan G. Johnson-Williams, *Student Member IEEE*, Robert S. Miyaoka, *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 Processing (SBP) method with depth of interaction processing for 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) solution of the detector's intrinsic spatial resolution. 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 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 [1,2]. SBP is used to improve the resolution of the monolithic sensor module and increase the effective image 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 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 [4].

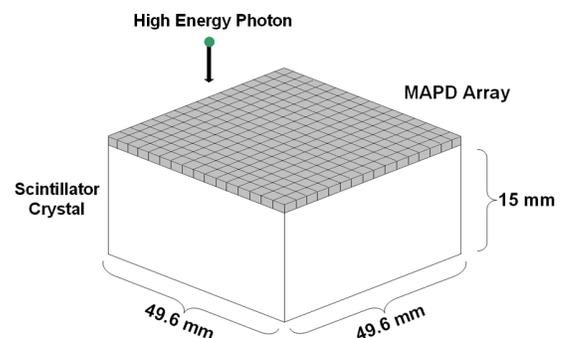
While the majority of events occur near the surface of the module crystal, the depth of interaction does vary for each event. Accurate calculation of the depth of interaction permits more accurate calculations of the coincidence line of response (LOR) and yields better intrinsic spatial resolution performance 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 three dimensional 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 [3]. The advantage of placing the sensor array on the entrance surface of the crystal 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., 16×16) MAPD elements.

Manuscript received November 13, 2009. This work was supported in part by Zecotech, Altera, DOE DE-FG02-08ER64676, and NIH grants EB001563 and EB002117.

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

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

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. To reduce the computational load of handling 256 individual sensors outputs row-column summing is used and 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 [5,6] 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 [7] 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 [8]. 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:

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

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 .3875mm apart giving a uniform distribution across the 49.6mm X 49.6mm crystal. For more detailed explanations about the SBP implementation refer to Joung and Ling [1,2].

C. Three Dimensional SBP

The same principles of the two dimensional SBP can be applied to a 3D search. Instead of two dimensions, the light response function is characterized in 3D, giving μ_i and σ_i as a function of X, Y and Z. The module is characterized for 127x127x15 positions, with 127x127 representing the X and Y dimensions with each X and Y position characterized for 15 depths. As with the 2D implementation, each XY position is on a grid with each point separated by .3875mm. The 15 depths, however, are not equally distributed over the 15mm deep crystal; depth is characterized at a higher resolution near the entrance of the crystal where events are most likely to occur [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.

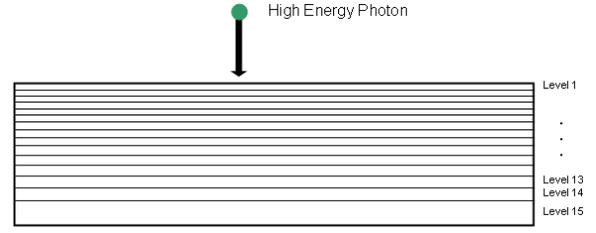


Fig. 2. Illustration of the distribution of depth levels

	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.

$$\ln(F(event, X, Y, Z)) = -\sum_{row=1}^{16} \left[\frac{[event_{row} - (\mu_{X,Y,Z})_{row}]^2}{2[(\sigma_{X,Y,Z})_{row}]^2} + \ln[(\sigma_{X,Y,Z})_{row}] \right] - \sum_{column=1}^{16} \left[\frac{[event_{column} - (\mu_{X,Y,Z})_{column}]^2}{2[(\sigma_{X,Y,Z})_{column}]^2} + \ln[(\sigma_{X,Y,Z})_{column}] \right] \quad (3)$$

D. Hierarchical Search

A hierarchical search can be used to produce a SBP solution in many fewer iterations than an exhaustive search. A hierarchical search begins with a coarse grained search, and through each stage becomes more refined, until the maximum resolution is reached in the final stage [Fig. 3]. A six stage, two dimensional, hierarchical search has been implemented and tested for previous versions of the cMiCE detector module [4]. 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 [Figure 9]. Comparing 9 different points in each search stage allows the hierarchical SBP algorithm to solve for the positioning of a 127x127 grid in six stages. In total, 54 data point computations are required, which is a drastic reduction from the 16,129 data point calculations needed for an exhaustive search.

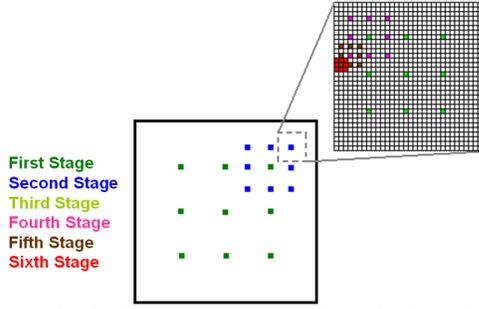


Fig. 3. Example of a six stage hierarchical solution for 2D SBP.

E. Two Dimensional SBP Implementation

A two dimensional hierarchical SBP algorithm has been designed and implemented on an FPGA development board [4]. The implementation produced similar accuracy to exhaustive methods 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, the time it takes to solve a single stage. The FPGA implementation was capable of processing data in real time, in excess of 200,000 events per second.

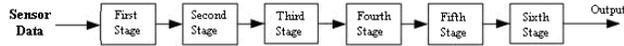


Fig. 4. Pipelined implementation of Hierarchical 2D SBP

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.

Custom hardware is currently under design for the application. The custom hardware will feature Altera's Stratix III FPGA with three large banks of external memory with sufficient capacity to store the large data tables required for 3D SBP.

With the custom hardware in progress a 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 which features the Stratix II FGPA. 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.

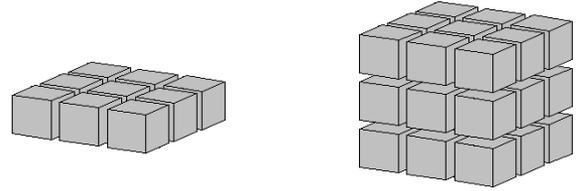
B. Algorithm Derivation

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 three dimensional characterization table in place of a two dimensional characterization table. In a 3D table, values are dependent on all three dimensions rather than in a 2D table where X and Y values are averaged according to the expected distribution of depth. 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. Since a 3D search compares the same XY points of a 2D search at multiple depths, a 3D search will consider a greater total number of points than the comparable 2D search.

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 [Fig. 6]. Since the existing pipeline has six stages, only half of the stages will be converted to 3D stages. Stages not converted to a three dimensional search will remain two dimensional to complete the XY search. With only half of the search stages required to be volumetric, the order of those stages in the pipeline becomes a variable for optimization of accuracy and memory requirements.

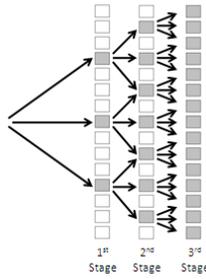


Fig. 6: Outline of hierarchical search to find depth among 15 levels.

Planar searches following any volumetric stages in the pipeline will also use three dimensional characterization tables. The depth of interaction found in the preceding stages is used as a constant with the 3D characterization table producing a 2D dimensional 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 3x3x3 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 a 2D stage with a 3D stage increases the 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 dependant 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. 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 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 tradeoffs 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 resolution 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 XY basis for calculating the depth while producing a depth

solution early enough to increase the accuracy of the finer-grained XY searches. The best combination tested places a volumetric search third, fourth, and fifth 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

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

Placing a volumetric search third, fourth, and fifth in the pipeline produces very good results; however there are several hardware limitations which prohibit its use. 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 will only permit the 27 calculations necessary in a volumetric stage to be computed in approximately 500 clock cycles, yielding a throughput of 140,000 events per second. This is 0.7x the required throughput.

To increase the throughput of the system, the volumetric search can be split into two independent stages: an XY planar search followed by a linear Z search. Using this method the algorithm will alternate between a 3x3 planar search and a 3 depth linear search. 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 XY, and then Z [Fig. 7]. This change dramatically increases the throughput of the algorithm, reducing the 27 calculation operations to 9 operations.

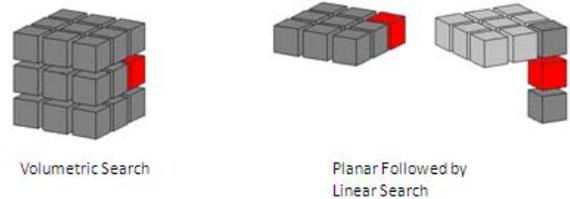


Fig. 7. Comparison between a volumetric search stage (left) and a planar/linear search stage (right).

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 XY at the given depth. The linear search will use the XY information found in the previous stage as a constant, solving for the best depth fit at the given XY. 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 act independently as separate 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 as a single stage performing two operations. In this configuration the stage first performs a planar search, followed by the 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 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 there is a significant quality gain when performing the depth searches after the first stage and little to no gain in beginning the depth searches after the second stage. The best option is placing a hybrid search stage second, third and fourth in the pipeline. Compared with the best option from table 1, the hybrid search stage performs similarly, with slightly improved depth resolution and slightly poorer XY resolution.

Location of Linear/Planar Stages	FWHM-XY (mm)	FWHM-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, performing the entire depth search at once. Using the previously outlined hierarchical depth search, the DOI can be found in a single stage in 9 calculations. Given that a hierarchical search can find the depth in 9 calculations, placing the depth search in a single stage maintains the original 2D search's critical path length to 9 calculations. This change offers several other benefits without much cost. 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 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 third stage of planar XY search. Since the

depth search characterization table size increases by a factor of 4 for each planar search preceding it, the depth search is ideally placed after the third XY search, where a high degree of accuracy is achieved without requiring a larger memory than necessary.

Location of Linear Depth Search	FWHM-XY (mm)	FWHM-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.

While not required, an XY planar search can benefit from the use of a three dimensional characterization table over a two dimensional one. A three dimensional characterization table 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 stage 4, 5 and 6 the full data table is not required and a simplified three dimensional 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, where significant improvements in accuracy correlate with larger tables up to a certain point when accuracy is not gained as readily. In the best case of high accuracy with a reduced table size, 4 depths are used in stage 4, 8 in stage 5, and 15 in stage 6. 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. The general trend is that the more depths used, the greater the accuracy. Following this 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 .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. Although an 8 depth table can be used for the sixth stage, a 4 depth table produces slightly better results.

The recommended number of depths used for the development board implementation uses 1 depth for stage 4, 2

depths for stage 5, and 4 depths for stage 6 (highlighted in Table 5). While this solution is not ideal, it is a significant improvement over using a single depth for the final stages and also represents a significant savings in memory. 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 three dimensional algorithm [Fig. 8]. The first three stages are XY planar searches, similar to ones used in the original two dimensional algorithm. The depth of interaction is found in a linear search after the first three stages and the depth is then used as part of the planar searches in stages 5 and 6. The depth remains unchanged through the last three stages of the pipeline.

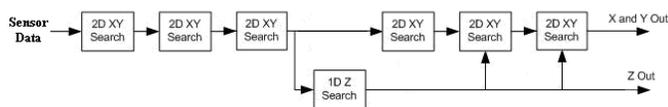


Fig. 8. Basic framework of 3D SBP hierarchical search.

C. 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.

There are many reasons behind the unbalanced gradient and local minima: reflections from the edges of the crystal, truncation of the signals due to the edges of the crystal, and noise in the system. Reflections from the edge of the crystal are most detrimental to system accuracy. In this case light can reflect from the edge of the crystal, giving the appearance of an event in a different location. Lastly, as with any system, noise can have a detrimental effect to the system's accuracy.

Most detrimental to system accuracy are local minima. 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 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 5mm of the crystal edge produce significantly poorer results than positions located closer to the center of the crystal.

A high occurrence of the same local minima leads to a tendency to produce incorrect solutions in 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) found with a 2D hierarchical search compared to the desired solution (right) found with an exhaustive search.

Additional peaks, even those close to the correct solution, have a great effect on the usefulness of the PET system. When constructing an image from positioning data it is impossible to determine the difference between a peak caused by a correct event or by a local minimum. Incorrectly identifying a peak caused by a local minimum as a true position leads to image ghosting, an object that appears in the produced image but does not actually exist. Inversely, ignoring closely grouped peaks that might be caused by local minima but are actually different data points can eliminate valuable 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.

Examination of data sets with local minima shows that local minima have a tendency to occur near the absolute minima. Since the local minima have a tendency to be located near the true minima a broader fine grain search becomes a viable correction option. A broader 5x5 search increases the range of the search by .3875mm in all directions. Expanding the final search stage from a 3x3 to a 5x5 reduces the occurrence of multiple peaking by 41% and decreases average error by .011 mm over the entire range of positions. A 7x7 search increases the range of the search by .775 mm. While a 7x7 doesn't further decrease the rate of dual peaking, expanding the final search stage to a 7x7 search reduces the average error by an additional .001mm compared to the 5x5. When expanded to a 9x9 search there is little benefit over a 7x7, with only a slight decrease in average error.

Further analysis shows that positions located in the corner of the crystal are more greatly affected than those that are near only one edge. Nearly every data point within 5mm 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. When focusing on an area 5mm from a corner a 5x5 search reduces the instances of dual peaking by 36%, a 7x7 reduces the rate by 68%, and a 9x9 reduces the rate by a comparable amount. Similar performance gains are also seen in average error: a 5x5 reduces average error by .09mm, a 7x7 reduces average error by .13mm, and a 9x9 reduces average error by .15mm.

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, 5.4 for a 7x7, and 9 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 5mm of the crystal edge are benefited by a broader search. Approximately 37% of the crystal area falls within 5mm of an edge. Given an equal distribution of events across the crystal this corresponds to 37% requiring a broader search. If only events within 5mm of the edge of the crystal invoke a 7x7, throughput is reduced to .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. 10]. 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.

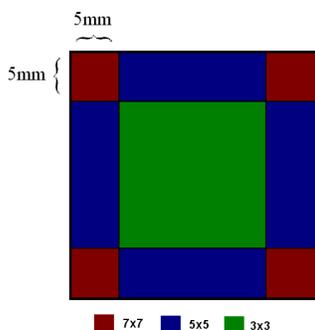


Fig. 10: Search breadth regions

A 7x7 search is computational intensive, requiring more than five times the calculations of a 3x3 search. While only a portion 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 it 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. 9 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.

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 5mm 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.

IV. RESULTS

The new three dimensional SBP algorithm produces higher quality results than the previous two dimensional SBP for our new detector module. The difference between the two algorithms is greatest at the edge of the crystal where the three dimensional method produces greatly improved results and the two dimensional 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 “dual 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. 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 a similar accuracy [Table 6].

Distance From Edge of Crystal	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)	FWHM-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.

REFERENCES

- [1] J Joung, et al., “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.
- [2] T. Ling, et al., “Performance comparisons of continuous miniature crystal elements (cMiCE) detectors,” *IEEE TNS*, vol. 53, pp. 2513-2518, 2006.
- [3] X. Li, et al., “A high resolution, monolithic crystal PET/MRI detector with DOI positioning capability *IEEE EMBS*,” pp. 2287-2290, 2008
- [4] D. DeWitt, Master’s Thesis, Department of Electrical Engineering, University of Washington, 2008.

- [5] G. Tsang, et al., "A simulation to model position encoding multicrystal PET detectors," *IEEE TNS*, vol. 42, pp. 2236-2243, 1995.
- [6] G. F. Knoll, et al., "Light collection in scintillation detector composites for neutron detection," *IEEE TNS*, vol. 35, pp. 872-875, 1988.
- [7] S. Agostinelli, et al., "Geant4 – a simulation toolkit," *NIM Section A*, vol. 506, pp. 250-303, 2003.
- [8] B. D. Rooney, et al., "Scintillator light yield nonproportionality: calculating photon response using measure electron response," *IEEE TNS*, vol. 44, pp. 509-516, 1997.