Enabling Real-Time Reconstruction for High Intrinsic Resolution SPECT systems

Mélanie Bernard[1], Guillaume Montémont[1], Sylvain Stanchina[1], Stéphane Mancini[2], and Loïck Verger[1]

***Abstract*—Single Photon Emission Tomography (SPECT) is mainly limited by the trade-off between spatial resolution and sensitivity determined by the collimation. In this context, CdZnTe detectors enable higher intrinsic spatial resolution compared to previously used scintillators, even improved by using techniques for 3D positioning. Moreover, the compactness of these detectors enables new system architectures. To improve the sensitivity of SPECT systems without degrading their spatial resolution, a possible way is to adapt the field of view depending on the kind of acquisition and the morphology of the patient, using flexible detection heads. That means computing reconstruction fast enough to determine interesting area where to focus detectors. This task is problematic because of the complexity of data to be processed, as the high resolution of detectors makes measurements sparse, and because of the dependency of reconstruction to head position, that must be taken into account.**

The aim of the present study is to propose new approaches to deal with this massive amount of complex information in real time to dynamically adapt the field of view. Implementation techniques from MLEM algorithm are proposed in order to fasten the reconstruction and adapt the geometrical configuration during the examination, and thus improve the sensitivity of the system without degrading its spatial resolution.

I. Introduction

PERFORMANCES of SPECT systems are mainly limited by the trade-off between their sensitivity and their spatial resolution. Significant improvements have been made on CZT detectors, and collimation is optimized but performances can still be enhanced by adapting the field of view of the system to the emitting object during the examination. Real-time computing is necessary to adapt the configuration, in order to determine the most informative areas from the results. A nowadays practicable configuration for cardiac imaging to empower such adaptations has been used as an example to lead our studies. The system is composed of 10 independent detection heads placed on a 120 degree arc around the patient, at 220 mm from the center of the object to be imaged (Fig. 1), in the spirit of the architecture proposed by Spectrum Dynamics. Each head can rotate on itself independently from others with 20 determined orientations that can cover up to 45 degrees. Heads are collimated with 20 mm high parallel holes collimators placed at 15 mm from the detector to enhance sensitivity. Septa are 1.5 mm wide, and holes size is 1 mm.

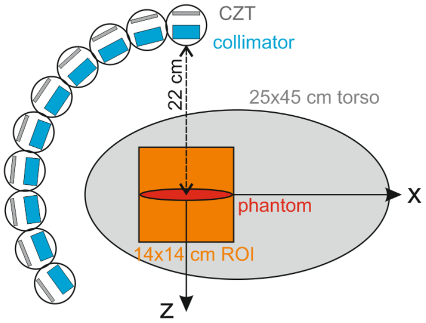


Fig. 1. Flexible configuration for cardiac imaging used in this study.

Heads uses 4 detection modules (HiSPECT [1]) developed in CEA-LETI to read and process data from CZT detectors. Modules are made of $40 \times 40 \times 5$ mm detector with 256 pixels (2.5×2.5 mm) and 8 ASICs (IDeFX [2]) right next to the detector for signal readout. The FPGA which reads out data from ASICs embeds signal processing techniques that enables 3D-positioning inside the detector and resolution recovery [3]. In that way, 8 levels of depth of interaction inside the detector and 8×8 sub-pixels per anode can be achieved. We consider that the size of the region o f interest (RoI) to be imaged is $140 \times 140 \times 140$ mm. The numerical size necessary to correctly cover this space with the spatial resolution empowered by described head is $64 \times 64 \times 64$ voxels.

In the followings, $\mathcal{M}$ stands for the detecting space and $\mathcal{O}$ is the imaging space. Classical capital letters (as D) are used for functions, italic capital letters (as M) for distributions, lowercase letters for voxels, and bold characters (as $\mathbf{R}$) for operators.

MLEM is commonly used to estimate emission distribution in the object. This algorithm originally works on binned representations of the object (O) and the measurements (M), linked by a huge matrix ($\mathbf{R}$) describing the model:

$$\forall m \in \mathcal{M}, \forall o \in \mathcal{O}, M_m = \mathbf{R}_{m,o}.O_o \tag{1}$$

where $\mathbf{R}_{m,o}$ stands for the probability to get a measurement in the detecting voxel m when having an event in the imaging voxel o. In the configuration we described, our detecting space dimension is about 10^8 bins considering every possible orientation. The size of the model would thus be about 10^{13} bins. Usually, the model is determined by calibration. This method is not practicable for this size because the storage and the reading would be problematic. Moreover, real-time

Manuscript received May 30, 2016.

[1] Mélanie Bernard, Guillaume Montémont, Sylvain Stanchina and Loïck Verger are with Univ. Grenoble Alpes, F-38000, France
CEA LETI, MINATEC Campus, 17 rue des Martyrs, F-38054 Grenoble, France. (email : melanie.bernard@cea.fr).

[2] Stéphane Mancini is with TIMA Laboratory, 46 avenue Félix Viallet, Grenoble Alpes, 38000 Grenoble, France.

computing is necessary to focus rapidly detecting heads on the most informative areas. Complete estimation of the emitting object have to be computed before changing the configuration to know what additional measurement could improve the best reconstruction quality. Knowing the amount of data to be processed, adaptation on classical MLEM algorithm have to be made to empower adapativity.

II. Implementation techniques for MLEM algorithm

A. List-Mode

Classical MLEM algorithm works on the following update on estimation at the end of each iteration from binned representation (M) of measurements :

$$O_o^{(n+1)} = \frac{O_o^{(n)}}{N(o)} \times \sum_{m \in \mathcal{M}} \frac{\mathbf{R}_{m,o}.M_m}{\sum_{o' \in \mathcal{O}} \mathbf{R}_{m,o'}.O_{o'}^{(n)}} \quad (2)$$

where $N(o)$ is a normalization factor to take into account the visibility of each voxel o. Histogram events is questionable since detecting matrix (M) is sparse. Consequently, two measurements are unlikely to occur on the same voxel, and most of the detecting bins would stay null. MLEM is computable in list mode without any change on the result [4]. Instead of browsing bins m of the detecting space, only voxels corresponding to a measurement m_k are considered. The update on estimation becomes :

$$O_o^{(n+1)} = \frac{O_o^{(n)}}{N(o)} \times \sum_{k \in \mathcal{K}} \frac{\mathbf{R}_{m_k,o}}{\sum_{o' \in \mathcal{O}} \mathbf{R}_{m_k,o'}.O_{o'}^{(n)}} \quad (3)$$

where $\mathcal{K}$ is the list of detected events k. MLEM is faster in list-mode since there is far less events to process than voxels in the detecting space. There is thus less loops to compute.
Another interesting point with the list-mode for real-time processing is that it is not worth waiting for the end of the acquisition to compute back projections of the stored events. Thus, list-mode offers some paralellization possibilities for online reconstruction pipeline.

B. On the Fly Processing

The complexity of the model $\mathbf{R}$ introduced in section I is a challenge to face to empower adaptations. To simplify it, we propose to split it into three sub-models: the detector $(\mathbf{D})$, the collimator $(\mathbf{C})$, and the geometry $(\mathbf{G})$. The model expressed in (1) becomes:

$$\forall m \in \mathcal{M}, \forall o \in \mathcal{O}, M_m = \mathbf{D}_{m,m'}.\mathbf{C}_{m',l}.\mathbf{G}_{l,o}^{m}.O_o \quad (4)$$

where

- $\mathbf{G}_{l,o}^{m}$ embodies the geometrical transformation applied on head which reported detection m at its detection time;
- $\mathbf{C}_{m,l}$ is equal to 1 if imaging voxel l is visible from detecting voxel m through the collimator;
- $\mathbf{D}$ is a diagonal matrix composed of detection homogeneity for every detecting voxel obtained by calibration.

It is not efficient to compute and store all the model in a matrix form. Therefore, distribution of the activity is obtained by successive ray traced projections and back projections.

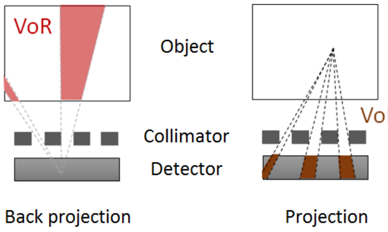


Fig. 2. Ray-tracing using front face and back face of the collimator.

Only the model of the detector is calibrated and stored, because it represents the non-homogeneity of detectors and signal readout, that can not be calculated. Sensitivity of each sub-pixel is determined using a homogeneous source before the examination. Every voxel m of the detecting space is thus normalized depending on its sensitivity $\mathsf{D}(m)$.
The model of the collimator $\mathbf{C}_{m,l}$ is determined using straight line equations. Projection of an object from the imaging space is made by calculating equations of lines delimiting the area of detecting voxels where from elements of the object are visible. Indexes m of voxels included into these lines are memorized and embodies the detecting volume of the imaging voxel l. Back projection of the detecting space are likewise obtained by calculating equations of lines delimiting the area of visible imaging voxels l from a measurement m. VoR are calculated using front face and back face of the collimator, as depict on Fig. 2.

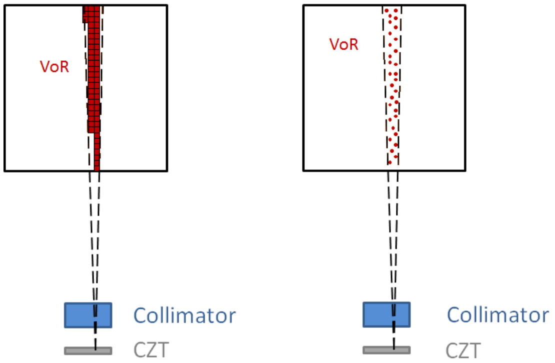
Events l obtained in the previous step are rotated and translated depending on the position G_m of the head which detected m at the detection time to get the list of final events o composing the estimation.
Finally, we can write the model in the following form:

$$\forall m \in \mathcal{M}, \forall o \in \mathcal{O}, M_m = \mathsf{D}(m).\mathbf{C}_{m,\mathsf{G}_m(o)}.O_o \quad (5)$$

This partition reduces the complexity of the model, because the detector model $\mathsf{D}(m)$ and the geometry model $\mathsf{G}_m(o)$ are now bijective functions: projections and back projections through these models lead to only one result per origin event. Only the model of the collimator is still complex, but working spaces are reduced: the size of the detecting space is divided by 200, since detecting heads position is no more taken into account. Moreover, using more possible orientations per head would no more be problematic considering the size of the model. Finally, computing projections and back projections on-the-fly is faster because there is no need to store and access a sparse matrix. Indeed, only voxels included in response volumes are actually considered. Calculating and browsing null coefficients is thus avoided.

C. Monte-Carlo Back Projections

Although models of geometry and detector have been simplified, the model of the collimator is still complex. Additional adjustments are made on it. In a systematic approach, back-project through the collimator model requires calculating and memorizing the complete list of imaging voxel included in the Volume of Response (VoR) of the collimator for each



(a) Systematic voxel browsing

(b) Monte-Carlo drawings

Fig. 3. Back projections methods through the collimator.

detecting voxel. We propose a more flexible approach using a Monte-Carlo method, that consists in randomly picking some events within the response volume of measurements through the collimator, as illustrated on Fig. 3. The number of drawings must be adjusted depending on the size of the volume of response, to take into account the sensitivity of detecting voxels trough the collimator.

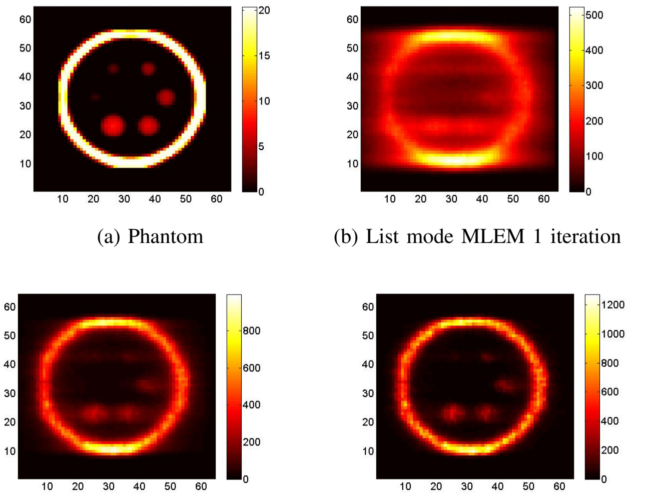
The first advantage of this method is to avoid the problem of interpolation of voxels partially included in the volume response, as picked events are represented by floating points in the imaging space. Spatial resolution is thus closer to the optimum using this approach. Then, simulation results exposed in the next section show that it is not necessary to pick as many drawings as there is voxels in the volume to get a precise result. We can thus save time if we are able to determine in advance how many drawings are necessary to correctly compute the reconstruction. Moreover, the number of drawings per detection can be seen as an adjustable parameter depending on the trade-off between computation time and precision needed. Finally, back projection of each Monte-Carlo drawn event can be run in parallel.

D. Partial Update

More could still be done on real-time processing by computing partial preliminary updates during the reconstruction. Events can be divided by groups, and one partial update is got from each group. Updating made with (3) is decomposed into the sum of contributions of each group :

$$O_o^{(n+1)} = \frac{O_o^{(n)}}{N(o)} \times \sum_g \left[\sum_{k\in\mathcal{K}_g} \frac{\mathbf{R}_{m_k,o}}{\sum_{o'\in\mathcal{O}} \mathbf{R}_{m_k,o'}.O_{o'}^{(n)}} \right] \tag{6}$$

where $\mathcal{K}_g$ is the g^{th} group of events. In our approach, we distribute events depending on their date. This approach is very similar to Ordered Subset Expectation Maximization (OSEM)[5], except that OSEM algorithm consists in sorting events depending on their location. The problem of OSEM algorithm is that it is not convergent, conversely to MLEM. Nevertheless, results are still reliable. As it is showed with simulation results in the next section, estimation is more reliable when using a large amount of events in each groups. Nevertheless, more time will be necessary to get enough measurements to compute several updates.



(a) Phantom

(b) List mode MLEM 1 iteration

(c) List mode MLEM 3 iterations

(d) List mode MLEM 5iterations

Fig. 4. Simulation results of List mode MLEM with 50k events for each iteration.

Indeed, as events are sorted depending on their date, these method enables interesting results if contributions of each group is used for next updates :

$$O_o^{(g+1)} = O_o^{(g)} + \frac{O_o^{(g)}}{N(o)} \times \sum_{k\in\mathcal{K}_g} \frac{\mathbf{R}_{m_k,o}}{\sum_{o'\in\mathcal{O}} \mathbf{R}_{m_k,o'}.O_{o'}^{(g)}} \tag{7}$$

Contributions can thus be computed separately to allow parallel computing, or one by one to use contributions between each updates and avoid to compute several iterations on the same events. It also avoid to store and replay events.

III. Simulations

A. List-mode Processing

A simulation tool has been developed to implement algorithms proposed above on the configuration described in introduction. Measurements were simulated from the phantom given if Fig. 4a. Sizes of spheres of the phantom from the smallest to the biggest are 4 mm, 6.5 mm, 9 mm, 11 mm, 15 mm and 17.5 mm. As the reconstructions is not manageable in an acceptable time using classical MLEM, we directly simulate the reconstruction using list-mode and on-the-fly computing. Projections of resulting reconstructions are showed on Fig. 4b, 4c and 4d for 1, 3 and 5 iterations on 50k measurements. Spatial resolutions we get from 3 iterations is thus between 6.5mm and 9 mm, that is consistent with the intrinsic spatial resolution of the collimator (8 mm).

5 iterations of classical list-mode MLEM are necessary to get a correct result. In the followings, this algorithm will be used as baseline to assess method performances.

B. Monte-Carlo Back projection

To assess the method consisting in picking points in the response volume of the collimator as it is developed

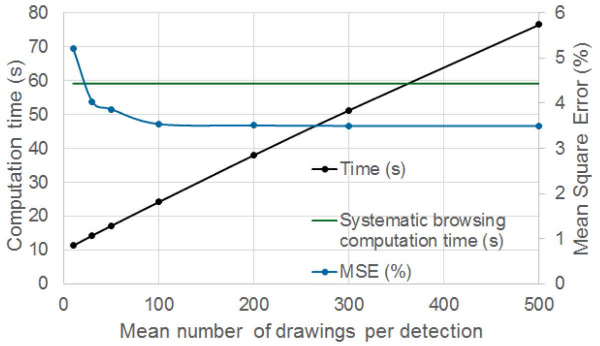
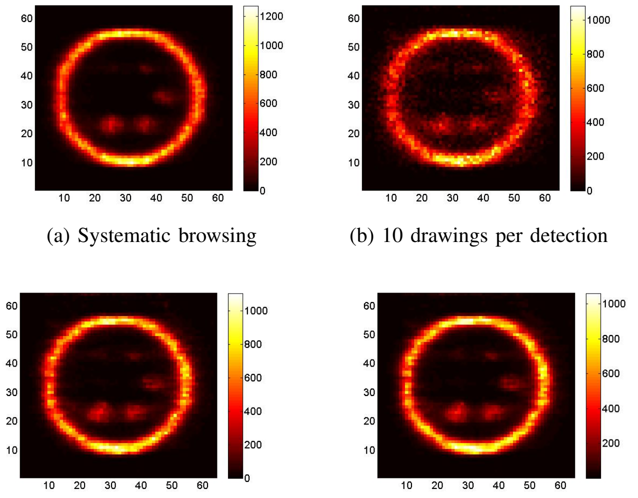


Fig. 5. Computation Time and quadratic Error for Monte-Carlo back projections.



(a) Systematic browsing

(b) 10 drawings per detection

(c) 100 drawings per detection

(d) 300 drawings per detection

Fig. 6. Simulation results using Monte-Carlo back projections.

in the section II-C instead of memorize every voxel, we compared computation times and estimations to those we get using a systematic method, for 5 iterations of list-mode MLEM. Estimations are compared using the Mean Square Error (MSE) normalized to the maximum between resulting estimation and result of systematic list-mode MLEM. Fig. 5 depicts the evolution of computation time (red curve) and MSE (blue curve) with the number of drawings per detection. Computing time for systematic browsing is 59.2 seconds (green line on Fig. 5). The evolution of the MSE points out that picking more than 100 events does not improve the quality of the result. Computing time increases linearly with the number of drawings.

Visually, Monte-Carlo back projections does not significantly affect spatial resolution, but reconstruction is grainy when the number of drawings is too low. Figure 6 confirms that image quality gets better when more events are picked, up to about 100 events. Then, no significant difference between estimations is visible.
The number of necessary drawings depends on the aperture angle of the collimator. In our configuration, the response volume for our collimator is on average 212 voxels per measurement.

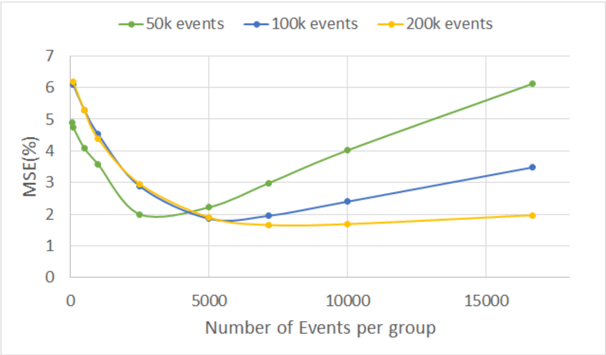


Fig. 7. Evolution of Mean Square Error with the number of Partial Updates.

C. Partial update

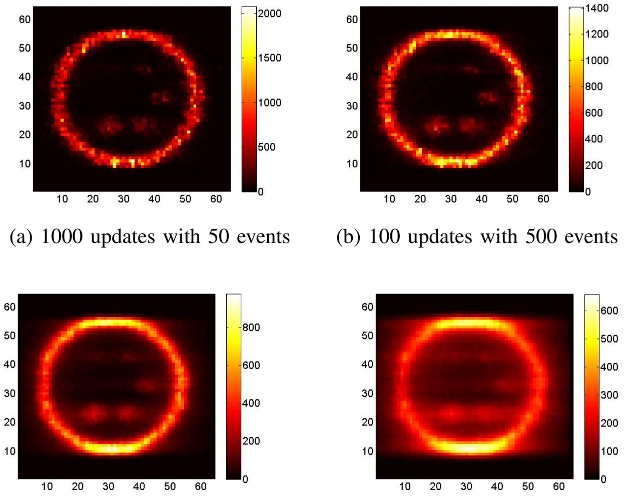
Quality of results given by reconstruction with partial updates as proposed in section II-D is very dependent on the size of groups used to compute each update. To determine this impact, we propose to compare results obtained from groups of 50, 500, 5k and 16k events for partial updates to those we get with 5 iterations of list-mode MLEM.

Evolution of MSE to the result from list-mode MLEM with the size of groups is plotted on Fig. 7, for different total numbers of events. It points out that there is an optimal size of groups to get a similar result to the reference. Size of groups is critical when the total number of events is low.
Fig. 8 illustrates the problems we get with a list of 50k events. Projections of estimations obtained with 50 events for different sizes of groups from 50 to 16k events are compared. Indeed, results from reconstructions with big groups as in Fig. 8d are similar to back projection, because there is not enough updates. Conversely, for small groups as in Fig. 8a, estimation is grainy: the algorithm overrates the activity on most likely voxels. The granularity of reconstructions is worsen when there is a lot of updates on small groups (green curve on Fig. 7).

These results highlight the necessity to have a sufficient amount of information on each group to get a correct result. Even if the reconstruction is no that faster in this case, we benefit from intermediate estimations. Processing times for different sizes of groups are given on table I. It smoothly increases when using more groups because updating step is more frequent. However, the difference is tinny and algorithm is still between 3 and 4 times faster than 5 iterations of MLEM algorithm.
As the quality of updates is linked to the amount of information contained in each group, problem is getting harder when the emitting object is noisy. The size of groups is thus to be adapted depending on the noise and the compactness of the emitting object.

D. Comparison of algorithms

Among enhancements proposed in this study, list-mode and on-the-fly processing does not change the result and require



(a) 1000 updates with 50 events

(b) 100 updates with 500 events

(c) 10 updates with 5k events

(d) 3 updates with 16k events

Fig. 8. Simulation results using Monte-Carlo back projections.

TABLE I
COMPUTATION TIME USING PARTIAL UPDATE COMPARED TO LIST-MODE MLEM RECONSTRUCTION WITH 5 ITERATIONS.

Total number of events	Algorithm	Computation time (s)
50k	List-mode MLEM	59
	Partial Update	20 - 22
100k	List-mode MLEM	109
	Partial Update	29 - 32
200k	List-mode MLEM	207
	Partial Update	49 - 55

less time because it is more adapted to our configuration. Monte-Carlo back projection and partial update can also help to quicken computing without any significant impact on the quality of the reconstruction if parameters are adapted to the situation. Previous sections helped to determine optimal parameters for our configuration.

Table II presents computational times of different combinations of these enhancements and their impact on faithfulness of results using the MSE to the distribution of detected simulated decays.

Selected numbers of drawing on Monte-Carlo back projection, and sizes of groups for partial updates, allows the same quality of result than classical list-mode MLEM algorithm. Using both together, computation is 4 times faster, and a new update is available every 0.8 seconds. Updates can be computed separately, that enables some parallel computing. This fast availability is precious since it enables making a lot of adaptations on configuration without making the examination time longer, compared to list-mode MLEM algorithm that can not allow changes on configuration faster than every minute.

TABLE II
PERFORMANCES OF PROPOSED ALGORITHMS.

Algorithm	Computation time (s)	MSE(%)
LIST-MODE MLEM 3 iterations	39.3	6.0
List-mode MLEM 5 iterations	59.2	4.3
List-mode MLEM 10 iterations	108.2	4.2
List-mode MLEM 5 iterations 100 MC Back projections	24.2	4.4
List-mode MLEM 5 iterations 200 MC Back projections	40.1	4.3
Partial Update every 2.5k events	19.6	4.4
Partial Update every 2.5k events 100 MC Back projections	13.1	4.4
Partial Update every 2.5k events 200 MC Back projections	15.3	4.3

CONCLUSION

In order to improve trade-off between their spatial resolution and their sensitivity, we worked on adaptation of the field-of-view. We studied a configuration usable for cardiac examinations in which heads can be focused on precise areas of the object to be imaged. Adapting head positions during the acquisition requires knowing fast where to direct detecting heads to get relevant information. Real-time computing of the reconstruction is thus necessary to get preliminary estimations of the object to process a lot of adaptations on the configuration. Adjustments of MLEM algorithm have thus be made to allow this real-time computing. List-mode and on-the-fly processing are a reordering of calculations and do not change the final result. Conversely, Monte-Carlo drawings for back projection trough the collimator or partial updates are approximations of MLEM algorithm, but enables the same faithfulness. Algorithms mixing these enhancements allow faster computation. The major advantage of partial update is the availability of preliminary estimations during the reconstruction to provide necessary information to adapt the configuration. Subsequently, parallel computing enabled by Monte-Carlo drawings and partial updates is to be exploited to propose efficient reconstruction pipeline.

REFERENCES

[1] G. Montémont, S. Lux, O. Monnet, S. Stanchina, and L. Verger, “Evaluation of a CZT gamma-ray detection module concept for SPECT,” presented at the IEEE Nuclear Science Symposium Conference Record, 2012, pp. 4091–4097.

[2] O. Gevin, O. Lemaire, F. Lugiez, A. Michalowska, P. Baron, O. Limousin, and E. Delagnes, “Imaging X-ray detector front-end with high dynamic range: IDeF-X HD,” Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 695, pp. 415–419, Dec. 2012.

[3] G. Montemont, S. Lux, O. Monnet, S. Stanchina, and L. Verger, “Studying Spatial Resolution of CZT Detectors Using Sub-Pixel Positioning for SPECT,” IEEE Transactions on Nuclear Science, vol. 61, no. 5, pp. 2559–2566, Oct. 2014.

[4] H. H. Barrett, T. White, and L. C. Parra, “List-mode likelihood,” J Opt Soc Am A Opt Image Sci Vis, vol. 14, no. 11, pp. 2914–2923, Nov. 1997.

[5] H. M. Hudson and R. S. Larkin, “Accelerated image reconstruction using ordered subsets of projection data,” IEEE Transactions on Medical Imaging, vol. 13, no. 4, pp. 601–609, Dec. 1994.