31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

SORTEO: Monte Carlo-based simulator with List-Mode capabilities

Andrew McLennan, Anthonin Reilhac and Sir Michael Brady

Abstract— **Monte Carlo-based PET simulators are powerful tools in the evaluation and validation of new PET algorithms. Accurate generation of projection data from spatio-temporal tracer distributions enable, for a given scanner specification and attenuating media distribution, quantitative analysis based on known ground truth. High activity-related phenomena, such as the contribution of randoms, as well as block and system deadtimes, corrupt actual PET scan data and therefore must be integrated within the simulation model, along with photon interactions within tissue and scanner materials.**

The PET-SORTEO Monte Carlo simulator, dedicated to full ring tomographs, is able to generate scattered, unscattered, and randoms event distributions from voxelized phantoms, accounting for data losses due to system deadtime. We show the results of extending the simulator to include accurate generation of list-mode data. Our implementation avoids incorrect event distribution and event timing inaccuracies cause by local and propagating temporal rounding errors. List-mode events produced by the PET-SORTEO simulator, when rebinned, are now consistent with sinograms produced by the simulator.

I. INTRODUCTION

Positron Emission Tomography (PET) is a functional medical imaging modality with the potential to record accurate, quantitative pharmacokinetic information in-vivo. When a radiotracer is administered to a subject, activity concentrates into regions determined by the particular tracer administered. By recording the temporal activity curves at each voxel, it is possible to assess the metabolic activity at that point inside the subject's body. Realistic, though simulated, PET data for which ground truth is available, forms an integral part of the development and validation process for new medical image analysis algorithms, prior to clinical deployment. The ground truth information enables quantitative analysis of algorithms, which is not possible when only clinical data is available.

There are two main methods available to researchers when validating an algorithm: bio-mechanical phantoms and digital phantoms. Bio-mechanical phantoms are physical phantoms which are designed to simulate the structures and flows of the patient. Typical applications include gated cardiac imaging, validation of ejection fraction calculations and myocardial wall uniformity assessment [1]. As each such phantom is a physical object, a new phantom needs to be developed for each new process that is to be examined. Digital phantoms, or computer generated simulations, such as the four dimensional NURBS-based cardiac-torso (NCAT) phantom [2] provide realistic human models and are capable of accurately reproducing cardiac and respiratory motions. As Time Activity Curves (TACs) can be assigned to individual regions as required, this approach is considerably more flexible than using physical phantoms.

Given both a digital phantom and TACs, Monte-Carlo based simulators are able to generate highly realistic dynamic PET data. A number of simulators have been developed, such as SimSET [3], PeneloPET [4], PETSIM [5], Eidolon [6], and GATE, based on GEANT4 [7]. PET-SORTEO is a simulator developed originally at the Montreal Neurological Institute. Its main advantage is that it is orders of magnitude faster than most competitive simulators and has been explicitly designed to run using multiple processors on a cluster; yet is still able to account for attenuation, scatter of the photons, high count rate related phenomena such as random events and the system deadtime, and all the other major sources of noise in the image acquisition process [8], [9].

The version of Sorteo that is currently available has the major limitation of only being able to simulate temporally-framed projection sinogram data. However, many reconstruction algorithms currently under development require list-mode data [10], [11], [12], [13], in which each detection event is recorded. For this reason, we have extended Sorteo to generate list-mode data. In this paper, we present results for list-mode Sorteo.

II. PET-SORTEO

A. Operation

Sorteo is dedicated to full ring tomographs and is able to produce Emission, Transmission and Blank scan data, given an appropriate scanner description. Sorteo generates PET data whose count rate performance matches closely that of experimental measurements obtained from the Siemens ECAT Exact HR+ scanner operating in both 2D and 3D modes for a wide range of activity levels and distributions. There are three main steps to generate simulated PET data using Sorteo. First, the user defines a text file which contains a description of the simulation. This includes: the scan duration and temporal frame distribution; the Emission and/or Attenuation maps together with their corresponding TAC values and tissue types; the outputted data format (Sinogram and/or List-mode); and any other required descriptors of the scanner's geometry or physical properties. Second, the text file is complied into a machine readable binary protocol format. Finally, the main Sorteo simulation generates the PET data.

This work was supported by Siemens Molecular Imaging and the Department of Trade and Industry

A. McLennan is with the Department of Engineering Science, University of Oxford, UK andrew.mclennan@new.ox.ac.uk

A. Reilhac is with Biospective in Montreal, Canada anthonin@biospective.com

M. Brady is with the Department of Engineering Science, University of Oxford, UK jmb@robots.ox.ac.uk

The TACs are specified as activity values at user specified time points throughout the scan duration. Linear interpolation is assumed for the activity values between the specified time points. Transmission scans do not require an emission map, since standard rotating rod sources are assumed. Also, Blank scans do not require an emission map or an attenuation map. Specifying the use of multiple processors within Sorteo is also possible, enabling it to be run concurrently on multiple machines, including clusters, reducing run time.

B. Simulator Overview

The Sorteo code comprises four main parts: a pre-simulation stage, and the Emission, Transmission and Blank stages. The pre-scan simulation is always required and its results are dependent on which type of simulation is being performed. The role of the pre-scan is to generate statistics for each Region Of Interest (ROI) for use in the later simulation. The PET simulation uses these statistics when calculating the emission rates and the system's deadtime. The simulation procedure itself has three main parts: the generation of True and Scattered events; the generation of prompt Randoms; and the generation of delayed Randoms. The True and Scattered events follow exactly the same simulation methods and are generated simultaneously by tracking the photon paths through the phantom. The Randoms are simulated from a probability distribution that is obtained from the statistics generated in a pre-scan.

Positron emissions in Sorteo are modelled as Poisson processes, whose spatial range is radionuclide specific. The inter-event disintegration times follow an exponential distribution given by $\delta t = -\ln(\xi)/a(t)$, where ξ is a random number that is uniformly distributed in the interval $[0, 1]$ and $a(t)$ is the activity of the emitting structure. The annihilation photons incorporate non-colinearity effects; also, Photoelectric, Compton and Rayleigh interactions in the tissue and detection system are modelled. Detector efficiencies with respect to energy and spatial resolutions are modelled with a blurring function; and energy thresholding of the detected photons is carried out automatically. Losses due to deadtime are modelled at four stages: within the block and bucket controllers, and at the processing and storage stage.

Experiments on cylinders located inside and outside the FOV show that simulation results are in close agreement with real ECAT Exact HR$^+$ PET data operating in both 2D and 3D modes. In 2D mode, the simulated results deviate slightly from the reference cylinder's values for activities above 30 kBq/cm and result in an overall error of about 3% at measurement of 50 kBq/cm. These may be due to differences between the physical phantoms and the numerical volumes, by the modelling of the septa in 2D, or by imperfections of the detection model where the external shields are assumed to block all photons. Another possible cause is the simplicity of the block deadtime modelling, which currently is only a function of the photons whose energy is larger than the threshold. In 3-D, the model correctly predicts the rates with less than 10% error.

A key aspect of Sorteo is that the simulated events are not all produced sequentially. Multiple instances of Sorteo allow the generation of simulated events to be parallelised; but require a final process of combining the event lists. Additionally, Randoms events (both prompt and delayed) are simulated after the Scattered and Unscattered events and need to be merged back within the overall list-mode file. Therefore, the main requirements of the overall list-mode data set which needs to be adhered to are: 1) if temporal gaps have been specified to exist between the sinogram frames, then these gaps need to be incorporated into the list-mode data files; 2) when merging data from multiple processors, the list mode events need to be ordered back into the true order they would have originally come from; 3) local timing errors at each event must not to propagate through the final list-mode file; and 4) the temporal distribution of the list-mode events need to match that of the sinograms when rebinned.

C. List-Mode Data Format

The List-Mode File (LMF) format, which is also used by the GATE simulator, is chosen as the Sorteo list-mode file format. It consists of two files: one is an ASCII file (with a .cch extension) containing all the general information about the scan and scanner such as: the scan type, isotope half-life, and the scanner's physical geometry[14]; and the other is a Binary file (with a .ccs extension) containing headers that fix the topology of the scanner, followed by the actual fixed size event records.

The Binary file contains a list of binary encoded headers, usually 16-bits long, followed by binary records of predefined length; such as event record, the count rate record, the simulation record (or digi record), or any other auxiliary information record outsourced from an external device (such as a motion-tracking camera). Each record type header is encoded over 16-bits whose first 4-bits are used to assign a tag to the record type which is described. The headers contain the encoding rules and number of bits allowed to store the detector IDs, Scanner description, and the encoding pattern for each event record. An example of the encoding rule for detector ID corresponding to ***e601** = 1110 0110 0000 0001* $\rightarrow$ *sssM Mmmc cccc cccl* means that there are 3-bits reserved for rings and rsectors, 2 for modules, 2 for submodules, 8 for crystals and 1 for layers [15], [16].

Once the headers have been constructed, Sorteo is ready to start recording the actual events in LMF format. LMF is a very flexible file format which lets users select how much information they want to record in the *Event Records*. The header files hold the information about what fields each event needs to store. Basic information that should always be recorded includes the time of the event, and the detectors that record it. Other information which may be stored includes: Time Of Flight (TOF) information, photon detection energies, and the current gantry/bed position. To date, we store only: the detection time, detector IDs, and photon energies. Sorteo stores coincidence event times in the form of the relative time to the previous event in millisecond units on 23-bits.

Simulated data also has the ability to store the extra information produced during the simulation, in the form of a *Digi Record*. This information would not normally be available from a clinical scanner but can be used to test and validate research stage reconstruction algorithms before they are put into clinical use. Digi records can be considered to be extensions to the event record, enabling the storage of more specific simulation information. One of the reserved bits from the Event Header is used to indicate whether a Digi record should accompany each event record. Information that we choose to store includes: the emission XYZ locations, detection XYZ locations, the absolute disintegration time, number of times each photon is scattered, and the event type. When testing reconstruction algorithms it is important to know the performance difference between noise-free and noisy data, and hence being able to initially only reconstruct from true events will enable a better understanding of the strengths and weakness of the reconstruction algorithm before noise is included.

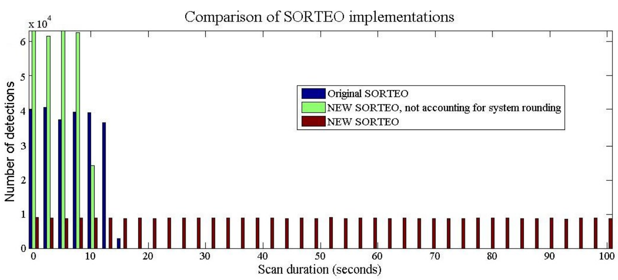

Fig. 1. Blue: When the times of each event on each processor are recorded as the difference between the current time and the previous event time, without taking into consideration the events from other processors or the actual exact time. Green: When the exact time is taken into consideration so that when the events from multiple processors are recombined into a single file the distribution is the same as that of when they were generated. Red: When the rounding errors of each event are also taken into consideration, so that rounding errors don't propagate beyond the timing window duration.

III. Implementation and Results

A. Scattered and Unscattered Events

When events are calculated on different processors, and the time is set to be the difference between the event and the previous event generated on that processor, rounding errors due to the 1ms timing windows can be introduced. This may produce two main problems. First, the event produced on any one processor may not match the true temporal distribution. Second, when the events from multiple processors are combined, the true temporal order in which the events should appear will also be incorrect. If we instead keep a temporal record of the true exact time at which each event is generated, then when events from multiple processors are combined the event order remains as it should be. Unfortunately, this method produces a larger grouping around time zero due to the exacerbated temporal rounding error problem. If we explicitly keep records of the true time and the time which the event produces, after rounding error, it is possible to localise this error so that certain events are given underestimates of time and some overestimates of time.

Results of this are shown in Fig. 1, with the distribution of events given for the various Sorteo list-mode file format implementations. Fig. 2 shows the errors between the list-mode event times and the actual detection times. No error between events now exceeds that of the 0.001 second timing resolution. The benefit of this method is that it automatically fixes the problem of gaps between frames and the temporal accuracy of multiple frames, due to the absolute detections times being used throughout.

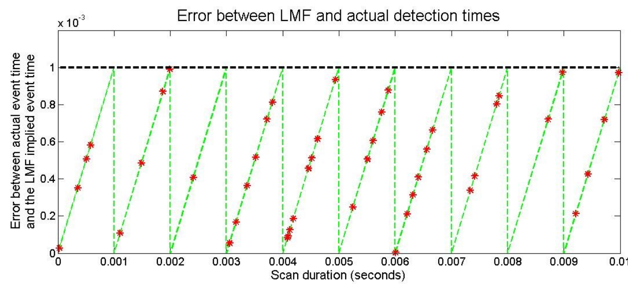

Fig. 2. Temporal distribution of list-mode events together with the error between the apparent LMF time and the absolute detection time. Stars represent the distribution in time along the horizontal axis, and their vertical values measure the error between the absolute detection time and the equivalent LMF time. Green dashes show that error increases but remains localised to the 0.001 second temporal windows.

B. Prompt and Delayed Randoms Events

As Randoms are not tracked through the phantom in the same way that scattered and unscattered events are, the original locations for the positrons that generated the events are not known. This is not important, however, as all we need to record for the list mode events are the detection times and the index of the crystals that detect them. For list-mode data, we need a method of specifying the temporal distribution of the Randoms events within the timing window, however, as the location of the Randoms is based on statistical measures calculated during the pre-scan phase. The three possible methods we consider are:

- Uniformly distribute Randoms events within window,
- Randomly distribute Randoms events within window,
- Calculate the time difference directly from the rate at which Randoms are produced.

Uniform spacing simply takes the number of Randoms events that occur within the 1 second temporal windows and distribute them with uniform spacing across the time interval. The random distribution of events takes a random number uniformly distributed in the interval $[0, 2]$ and multiplies it by the temporal spacing difference found from the uniform distribution of Randoms mentioned previously. A sufficient number of events will result in the average inter-event time being that of the uniform distribution; but with a wider variation of inter-event times. The third method calculates the inter-event times from the Randoms event rate, by taking its reciprocal. Fig. 3(a) shows the absolute inter-event times for a whole scan which includes Scattered, Unscattered and prompt Randoms events. The distribution is very similar for all three methods.

Fig. 3(b) now shows the three inter-event times distributions for Randoms events only.[1] It can be seen that uniformly distributed Randoms events are unrealistic but they are quick and easy to calculate. Randomly spaced events will be more realistic as disintegrations are inherently random processes, but require slightly more computation time. The Randoms rate derived inter-event times actually are equivalent to the uniform distribution, as the Randoms rate is only recomputed with the same periodicity as the number of events needing to be simulated is calculated and hence of no added benefit.

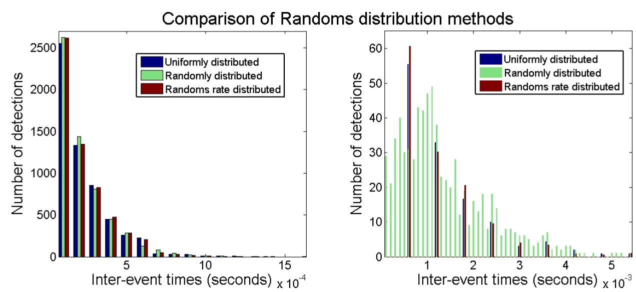

Fig. 3. (a) Histogram of absolute inter-event times for a whole scan which includes Scattered, Unscattered and prompt Randoms events. (b) Histogram of absolute inter-event times for a whole scan which includes prompt Randoms events only. Histogram values have been scaled to account for their effective duration.

Finally, when testing reconstruction algorithms, it is advantageous to have access to extra information about the events that are detected. Knowing the exact disintegration location of each event, for example, enables comparisons between emission and reconstructed distribution. Using the Digi flag in Sorteo results in the Digi Records described above being saved for all events. Care now needs to be taken when reading the LMF file, to account for whether or not the Digi records are included. If the Digi record flag is chosen, the delayed random events are also stored along with the other event types in the LMF file. The Digi record allows distinction between event types and hence all information can be kept in a single LMF file. Fig. 4 gives the histogrammed results from a list-mode file which was generated for a phantom with a single region of uniform activity. As can be seen, the number of Scattered, Unscattered and prompt Randoms are all approximately uniform for the whole duration of the scan, as they should be. This demonstrates the successful implementation of the list-mode format in Sorteo.

IV. DISCUSSION

We show results for the PET-SORTEO Monte-Carlo simulator which has been extended to enable accurate list-mode data generation. We have shown that the implementation avoids incorrect event distribution error and event timing inaccuracies cause by local and propagating temporal

[1]The number of events are scaled by their duration.

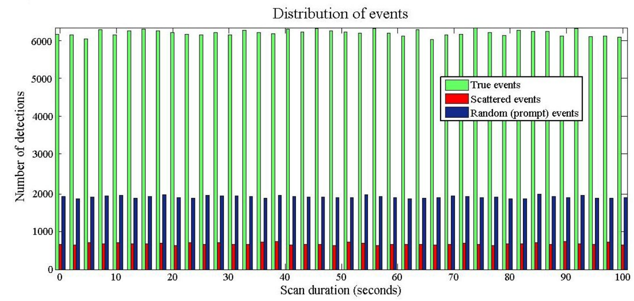

Fig. 4. Histogram of the True, Scattered and prompt Randoms events for a simulation of uniform activity showing that Sorteo is now fully capable of returning list-mode events.

rounding errors. List-mode events produced by the PET-SORTEO simulator, when rebinned, are now consistent with sinograms produced by the simulator. The addition of list-mode capabilities also does not significantly affect simulation run-times. Further extension of the simulator to include motion, such as that caused by respiration, is also currently under development.

REFERENCES

[1] T. R. Simon, B. S. Walker, *et al.*, "A Realistic Dynamic Cardiac Phantom for Evaluating Radionuclide Ventriculography: Description and Initial Studies with the Left Ventricular Chamber," *Journal of Nuclear Medicine*, vol. 30, no. 4, pp. 542–547, 1989.

[2] W. P. Segars, "Development and Application of the new Dynamic NURBS-based Cardiac-Torso (NCAT) Phantom," Ph.D. dissertation, University of North Carolina, 2001.

[3] R. L. Harrison *et al.*, "Acceleration of SimSET Photon History Generator Module of a Public Domain Simulation System for Emission Tomography," in *IEEE NSS MIC Conf. Rec.*, 2003, pp. 1835–1838.

[4] S. España, J. L. Herraiz, E. Vicente, *et al.*, "PeneloPET, a Monte Carlo PET Simulation Tool Based on PENELOPE: Features and Validation," *Phys. Med. Biol.*, vol. 54, pp. 1723–1742, 2009.

[5] C. J. Thompson, J. M. Cantu, and Y. Picard, "PETSIM: Monte Carlo Program Simulation of all Sensitivity and Resolution Parameters of Cylindrical Positron Imaging Systems," *Physics in Medicine and Biology*, vol. 37, pp. 731–749, 1992.

[6] H. Zaidi, C. Labbe, and C. Morel, "EIDOLON: Implementation of an Environment for Monte Carlo Simulation of Fully 3D Positron Tomography on a High-Performance Parallel Platform," *Parallel Computing*, vol. 24, pp. 1523–1536, 1998.

[7] S. Jan, G. Santin, *et al.*, "GATE: A Simulation Toolkit for PET and SPECT," *Phys. Med. Bio.*, vol. 49, pp. 4543–4561, 2004.

[8] A. Reilhac, G. Batan, C. Michel, C. Grova, *et al.*, "PET-SORTEO: Validation and Development of Database of Simulated PET Volumes," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 5, pp. 1321–1328, 2005.

[9] A. Reilhac, C. Lartizien, *et al.*, "PET-SORTEO: A Monte Carlo-based Simulator with High Count Rate Capabilities," *IEEE Trans. Nuc. Sci.*, vol. 51, no. 1, pp. 46–52, 2004.

[10] T. Nichols, J. Qi, *et al.*, "Spatiotemporal Reconstruction of List Mode PET Data," *Trans. Med. Imag.*, vol. 23, no. 4, pp. 396–404, 2002.

[11] B. Gundlich, P. Musmann, and S. Weber, "Dynamic List-Mode Reconstruction of PET Data based on the ML-EM Algorithm," *IEEE Nuclear Science Symposium Conference Record*, vol. 5, pp. 2791–2795, 2006.

[12] A. Reader, S. Ally, *et al.*, "One-Pass List-Mode EM Algorithm for High-Resolution 3-D PET Image Reconstruction Into Large Arrays," *IEEE Trans. Nuc. Sci.*, vol. 49, no. 3, pp. 693–699, 2002.

[13] A. J. Reader, "List-Mode EM Algorithm for Limited Precision High-Resolution PET Image Reconstruction," *International Journal of Imaging Systems and Technology*, vol. 14, no. 3, pp. 139–145, 2004.

[14] ClearPET Project, "List Mode Format Implementation: Scanner Geometry Version 4.1," Open Gate Collaboration, Tech. Rep., 2002.

[15] ——, "LMF Specification," Open Gate Collaboration, Tech. Rep., 2005.

[16] ——, "List Mode Format Implementation, Version 1.3.2," 2002.