HHS Public Access

Author manuscript

IEEE Nucl Sci Symp Conf Rec (1997). Author manuscript; available in PMC 2015 November 13.

Published in final edited form as:

IEEE Nucl Sci Symp Conf Rec (1997). 2011 October ; 2011: 3548–3553. doi:10.1109/NSSMIC.

2011.6153666.

A System Calibration and Fast Iterative Reconstruction Method for Next-Generation SPECT Imagers

Brian W. Miller [Member IEEE],

Center for Gamma-Ray Imaging and the College of Optical Sciences, University of Arizona, Tucson, AZ 85724 USA

Roel Van Holen,

MEDISIP, Department of Electronics and Information Systems, Ghent University, B-9000 Ghent, Belgium. He is supported by a postdoctoral fellowship of the Research Foundation (FWO)

Harrison H. Barrett [Fellow IEEE], and

Center for Gamma-Ray Imaging and the College of Optical Sciences, University of Arizona, Tucson, AZ 85724 USA

Lars R. Furenlid [Member IEEE]

Center for Gamma-Ray Imaging and the College of Optical Sciences, University of Arizona, Tucson, AZ 85724 USA

Brian W. Miller: molinero@email.arizona.edu

Abstract

Recently, high-resolution gamma cameras have been developed with detectors containing $> 10^5$ – 10^6 elements. SPECT imagers based on these detectors usually also have a large number of voxel bins and therefore face memory storage issues for the system matrix when performing fast tomographic reconstructions using iterative algorithms. To address these issues, we have developed a method that parameterizes the detector response to a point source and generates the system matrix on the fly during MLEM or OSEM on graphics hardware. The calibration method, interpolation of coefficient data, and reconstruction results are presented in the context of a recently commissioned small-animal SPECT imager, called FastSPECT III.

Index Terms

Single-photon emission computed tomography (SPECT); calibration; system matrix; image reconstruction; high-performance computing; graphics processing units; OSEM

I. Introduction

A discrete-to-discrete model of a SPECT system, described by a system matrix $\mathbf{H}$ that maps an object $\mathbf{f}$ into measured data $\mathbf{g}$ can be described as:

$$\mathbf{g} = \mathbf{H}\mathbf{f}, \quad (1)$$

where $\mathbf{f}$ is an $N \times 1$ vector of voxel elements, $\mathbf{g}$ is an $M \times 1$ vector of data bins (pixels), and $\mathbf{H}$ is an $M \times N$ matrix. When properly normalized, the individual elements of $\mathbf{H}$, h_{mn} ,

represent the probability of a photon emitted from the n^{th} voxel being detected in the m^{th} detector element [1], [2]. Detector and collimator blur, and pinhole and aperture misalignment, can be included in $\mathbf{H}$ by measuring the system response at a grid of discrete locations in the field of view (FOV) of the camera [3]–[6]. Using iterative algorithms, these system imperfections are at least partially compensated for, leading to improved quality of reconstructed images.

Traditional scintillation detectors that use photomultiplier tubes (PMT) typically have $\sim 10^4$ resolvable detector elements. In small-animal SPECT systems based on these low-resolution detectors, $\mathbf{g}$ has on the order of $\sim 10^5$ elements, depending on the number of projection views. The number of voxel elements in $\mathbf{f}$ is typically on the order of $\sim 10^6$.

In recent years, high-resolution gamma cameras have been developed with detectors having 10^5 – 10^6 pixel elements [7]–[11]. In SPECT systems based on these detectors the size of $\mathbf{g}$ ranges from 10^6 – 10^7 elements, one to two orders of magnitude more elements compared to PMT-based SPECT systems. Consequently, we are presented with new challenges with regards to system calibration procedures, storage issues of the system matrix, and methods for performing fast tomographic reconstructions using iterative algorithms.

At the Center for Gamma-Ray Imaging, we recently completed the system integration of FastSPECT III, a next generation high-resolution stationary SPECT imager [12]. Stationary SPECT imagers are composed of rings of gamma-ray detectors that provide sufficient angular sampling for tomographic reconstruction without requiring movement of the detector, imaging aperture, or imaging subject. Simultaneous acquisition of projection image data allows for dynamic 4D imaging, time-dependent activity studies, and avoidance of artifacts due to subject motion. An image of the system is shown in Figure 1. FastSPECT III has 20 CCD-based scintillation gamma cameras, called BazookaSPECT [7]. Each BazookaSPECT comprises a columnar CsI(Tl) scintillator, an image intensifier, and a 640×480 CCD sensor that operates at up to 200 frames per second. Currently, the FastSPECT III imaging aperture has 20 pinholes, one per camera. The size of $\mathbf{g}$ for the system is $\sim 6 \times 10^6$ elements. Assuming a 100^3 voxel volume, storage of the entire system matrix would require ~ 22 TB of space. Fortunately, for pinhole SPECT most elements of $\mathbf{H}$ are zero, and storing only the non-zero elements (sparse $\mathbf{H}$) significantly reduces memory storage requirements. In PMT-based systems, the entire sparse $\mathbf{H}$ can be loaded into memory for fast iterative reconstruction. This is done in the FastSPECT II system which has 16 modular PMT detectors (80×80 elements per detector). However, for FastSPECT III even the sparse $\mathbf{H}$ is very large because of fine sampling with high-resolution detectors, and memory storage remains an issue. If we assume that the point spread function (PSF) from a given voxel can be modeled as a 2D Gaussian function, then storage of the parameters that describe the Gaussian function results in a file which is significantly smaller for both types of SPECT systems. Table I shows the relative sizes of the system matrices for FastSPECT II and FastSPECT III.

To address the storage issues of the system matrix that new high-resolution gamma cameras pose for iterative reconstruction, we have developed and validated a method that uses multi-core GPUs for fast, on-the-fly computation of $\mathbf{H}$ from coefficient data. This method can also

be applied to stationary SPECT systems where $\mathbf{g}$ is relatively small (FastSPECT II) but $\mathbf{f}$ is large, such as in full-body imaging. Additionally, the method is readily extendable to other imaging geometries where the detector response to a point source is best modeled by some function other than a 2D Gaussian, e.g. a rectangular point-source projection image from a crossed-slit collimator [13].

II. Materials and Methods

For stationary SPECT imagers at the Center for Gamma-Ray Imaging, we typically measure the system-specific components of $\mathbf{H}$ experimentally using a radioactive point source [4]. Using a 3-axis positioning stage, the point source is stepped in object space in a 3D grid of measurements points. The span of the 3D grid determines the system field of view (FOV). From the point-source projection images, we build up a system matrix that incorporates imperfections or misalignment for the imaging system as well as non-uniformity in the camera response such as distortion, for example. Figure 2 shows data from a FastSPECT III calibration measurement acquired at one position during a 3D scan. The ^{99m}Tc point source is made using a ~ 500 microns ion-exchange resin bead. For a given source position, we obtain 20 projection images that correspond to one column of $\mathbf{H}$, of the size $640 \times 480 \times 20 = 6,144,000$ elements. As can be visualized in Figure 2, most elements of the projection images are zero. Only the non-zero elements of the system matrix contain information needed for reconstruction. Figure 3a shows the non-zero regions of Figure 2 that correspond to the system PSF for the given source position.

For FastSPECT III, we have adopted the method employed by Chen with FastSPECT II [15], where the PSF is estimated to be a 2D Gaussian function. For each source position, a 2D Gaussian fit is performed on projection images using a least-squares algorithm. A total of six parameters (amplitude, x-position, y-position, x-width, y-width, and a correlation coefficient) are used to parameterize the 2D Gaussian. Gaussian fits to projection images from Figure 2 are shown in Figure 3b. To avoid prohibitively long calibration measurements, we measure the 3D grid of points at a relatively large step size of ~ 1 mm and estimate intermediate voxels by interpolating neighboring Gaussian coefficients. The next step used in previous generation imagers (FastSPECT I and II) is to generate a sparse $\mathbf{H}$ from the Gaussian coefficient data and store the non-zero matrix elements to file. The system matrix is then be loaded into system memory during tomographic reconstruction. For FastSPECT II, depending on the number of voxels and detector elements, the size of a sparse $\mathbf{H}$ ranges from 340 MB – 12 GB, a size well within the standard range of system memory in current computing systems. However for FastSPECT III, as previously mentioned, because of the increased number of detector elements, loading a sparse system matrix, e.g., 80 GB, into system memory for tomographic reconstruction is not an attractive option as the reconstruction time would be prohibitively slow.

Since the entire coefficient file is relatively small (see Table I) and it contains all the information necessary to construct the system matrix, our solution to this dilemma with FastSPECT III is to use the inherently parallel nature of modern graphics cards, with the additional benefit of being low cost, to generate the system matrix on the fly. For tomographic reconstruction, both projection images and the coefficient data are copied to

GPU memory and elements of $\mathbf{H}$ are then generated on the fly in parallel using coefficient data. To date, we have implemented both maximum-likelihood expectation maximization (MLEM) and ordered-subset expectation maximization (OSEM) algorithms using the NVIDIA[®] CUDA [16] programming environment on NVIDIA[®] Fermi graphics cards. Using CUDA, within a block of parallel threads, elements of a 2D Gaussian (PSF) are computed from a set of six coefficients. The NVIDIA GPU used for MLEM/OSEM reconstruction is a Fermi GeForce GTX 580 that has 512 processor cores and 1.5 GB of memory.

III. Results

To validate our GPU-based MLEM/OSEM reconstruction algorithm, we imaged a micro-Derenzo (Jaszczak) phantom produced by VANDERWILT techniques by [14]. The phantom bores were filled with a total of 5 mCi of ^{99m}Tc . A set of $\phi 500$ micron pinholes were used to acquire projection images. A photograph of the phantom is shown in Figure 4a, and the twenty projection images are shown in Figure 5. The smallest bore is $\phi 350$ microns and the height of the bore is 8 mm. For the tomographic reconstruction, 100 iterations of MLEM were used with a reconstruction volume of 104^3 voxels (14.95 mm in each dimension). A 3D rendering of the reconstructed Jaszczak phantom is shown in Figure 4b. Currently, each iteration of MLEM for a 104^3 voxel volume takes approximately 19 seconds, during which approximately 160 GB of data are generated and processed using the stored coefficients, i.e., elements of the system matrix are generated for forward and back projection operations using a 31×31 sampling of the 2D Gaussian PSFs. To further increase the speed of the reconstruction algorithm, we have implemented OSEM on the GPU and present results using 20 iterations of OSEM with 5 subsets.

Since we are generating the PSFs of the system matrix on the fly in the GPU, we have the freedom to choose what fraction of the 2D Gaussian to generate. For example, instead of generating a 2D Gaussian that is sampled with 31×31 elements, we can sample a truncated region using 11×11 elements. The benefit of such an approach is reduced reconstruction time at the expense of potential reconstruction artifacts. Additionally, decreased reconstruction time can be achieved using a coarser voxel volume, e.g. 52^3 voxel volume instead of a 104^3 volume. Depending upon the imaging task, reduced resolution and potential artifacts may be acceptable tradeoffs for a shorter reconstruction time, especially with imaging studies that would benefit from the capability of real-time tomographic reconstruction while projection data is being acquired. Some benefits of real-time tomography include:

- The capability of quickly knowing whether or not the subject is properly aligned within the imaging field of view.
- Rapidly determining whether or not tracer has arrived in the target volume of interest.
- Knowledge of when sufficient data are obtained so that the acquisition can then be stopped. This would optimize the acquisition time and consequently increase the

throughput capability of the imaging system. Also, it would minimize the amount of time the subject would need to be placed under anesthesia.

- The ability to acquire scout scans for adaptive SPECT systems.

Regarding the last point, future adaptive SPECT systems [13], [17]–[20] are currently being built that will have the capability to dynamically change system geometry for optimal imaging performance. These systems will initially generate a low resolution, large volume reconstruction (scout scan) that is used to identify a target volume of interest. The system then dynamically reconfigures the aperture/detector configuration for an optimal imaging acquisition. The capability to quickly obtain the scout tomographic reconstruction is an integral feature of adaptive SPECT systems.

We have successfully demonstrated that the entire object volume can be reconstructed in a mere few seconds using a reduced voxel volume and/or by truncating the PSF. To examine the performance variability in terms of time and spatial resolution, we reconstructed the Derenzo phantom with 104^3 and 52^3 voxel volumes at various truncations of the PSF. Results are shown in Table II and Table III. Note that the GPU computational time of the reconstruction (20 iterations of OSEM using 5 subsets) is for the entire voxel volume. For FastSPECT III, finer sampling of the PSF (e.g., 27×27 or 31×31) qualitatively improves the reconstructed image. For comparison, in systems with lower detector resolution, such as FastSPECT II, a non-truncated sampling of the 2D Gaussian PSF could be obtained with 9×9 elements. Examining Tables II–III, a key point is that even as the PSF is truncated it is possible to reconstruct the object without significant artifacts, even with a highly truncated PSF, e.g., 7×7 or 5×5 region of the 2D Gaussian. Shown in Table III at these sampling values, it takes only ~ 3 seconds to complete 20 iterations of OSEM.

IV. Conclusions

The inherently parallel nature of GPUs, with hundreds of processing cores, is an attractive feature for performing fast tomographic reconstructions using iterative methods. An additional attractive feature of GPUs is their relatively low cost. In modern computing systems, which provide gigabytes of system and GPU memory, storing the non-zero, system matrix elements in memory for fast reconstruction is not feasible in next-generation SPECT systems because of the massive number of voxels and/or detector elements. Our solution to this problem is to represent the non-zero system matrix elements with a model, e.g., 2D Gaussians, which can then be parameterized by coefficients. Storing only the coefficients is a data reduction process that overcomes memory storage issues much in the way storing listmode data in PMT-based scintillation cameras is more efficient than binning when there are many event attributes. Implementing iterative reconstruction algorithms using coefficient data requires that the system matrix elements be generated on the fly, which is accomplished in parallel using GPU processors.

We propose that as SPECT imagers with vastly more detector elements and vastly more voxels are developed, generation of the system matrix on the fly in GPUs is the method that will have to be employed to reconstruct in a reasonable time frame. We have successfully developed and validated this method with MLEM and OSEM in the FastSPECT III, small-

animal stationary SPECT imager. We have demonstrated that this method allows for real-time tomography of the entire voxel volume in a matter of seconds. Real-time tomography is a feature that provides SPECT imagers with a number of benefits such as a method for obtaining the optimal acquisition time and rapid knowledge as to whether or not tracer has arrived in the target volume of interest. Most importantly, we propose it as a solution to a critical feature needed in future next-generation adaptive SPECT imagers that will require fast tomographic reconstruction of scout data.

Acknowledgments

This work was supported by the National Institutes of Health under NIBIB Grant P41-EB002035 and R37-EB000803.

References

1. Barrett, H.; Myers, K. Foundations of Image Science. John Wiley and Sons; 2004.
2. Wilson, D. Small-Animal SPECT Imaging. Springer; 2005. Computational algorithms in small-animal imaging; p. 139-162.
3. Rowe R, Aarsvold J, Barrett H, Chen J, Klein W, Moore B, Pang I, Patton D, White T. A stationary hemispherical SPECT imager for three-dimensional brain imaging. J Nucl Med. 1993; 34(3):474–480. [Online]. Available: <http://jnm.snmjournals.org/cgi/content/abstract/34/3/474>. [PubMed: 8441043]
4. Chen, Y.; Furenlid, L.; Wilson, D.; Barrett, H. Small-Animal SPECT Imaging. Springer; 2005. Calibration of scintillation cameras and pinhole SPECT imaging systems; p. 195-201.
5. van der Have F, Vastenhouw B, Rentmeester M, Beekman F. System calibration and statistical image reconstruction for ultra-high resolution stationary pinhole SPECT. Medical Imaging, IEEE Transactions on. 2008; 27(7):960–971.
6. Van Holen R, Miller BW, Moore JW, Vandenberghe S, Barrett HH. Object-space interpolation of SPECT system matrices from point-source measurements. Fully 3D Conference Proceedings. 2011:419–422.
7. Miller B, Barrett H, Furenlid L, Barber H, Hunter R. Recent advances in BazookaSPECT: Real-time data processing and the development of a gamma-ray microscope. Nuclear Inst and Methods in Physics Research, A. 2008; 591(1):272–275.
8. Miller, B.; Barber, H.; Barrett, H.; Wilson, D.; Chen, L. A low-cost approach to high-resolution, single-photon imaging using columnar scintillators and image intensifiers. Nuclear Science Symposium Conference Record; 29 2006-Nov. 1 2006; IEEE; p. 3540-3545.
9. de Vree, G.; van der Have, F.; Beekman, F. EMCCD-based photon-counting mini gamma camera with a spatial resolution <100 microns. Nuclear Science Symposium Conference Record, 2004 IEEE; Oct. 2004; p. 2724-2728.
10. Meng L. An intensified EMCCD camera for low energy gamma ray imaging applications. Nuclear Science, IEEE Transactions on. Aug; 2006 53(4):2376–2384.
11. Peterson T, Wilson D, Barrett H. Application of silicon strip detectors to small-animal imaging. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 2003; 505(1–2):608–611.
12. Miller, B.; Moore, S.; Barber, H.; Furenlid, L.; Barrett, H. System integration of FastSPECT III, a dedicated SPECT rodent-brain imager based on BazookaSPECT detector technology. Nuclear Science Symposium Conference Record; IEEE; 2009. p. 4004-4008.
13. Durko H, Peterson T, Barrett H, Furenlid L. High-resolution, anamorphic, adaptive small-animal SPECT imaging with silicon double-sided strip detectors. Proceedings of SPIE. 2011; 8143:81430G.
14. VANDERWILT techniques bv. <http://www.for-med.nl>

15. Chen, Y. PhD dissertation. University of Arizona, Department of Optical Sciences; Tucson, Arizona: 2006. System calibration and image reconstruction for a new small-animal SPECT system.
16. NVIDIA, CUDA. Compute Unified Device Architecture-Programming Guide Version 2.0. 2009.
17. Clarkson, E.; Kupinski, M.; Barrett, H.; Furenlid, L. A task-based approach to adaptive and multimodality imaging. *Proceedings of the IEEE*; March 2008; p. 500-511.
18. Barrett H, Furenlid L, Freed M, Hesterman J, Kupinski M, Clarkson E, Whitaker M. Adaptive SPECT. *Medical Imaging, IEEE Transactions on*. Jun; 2008 27(6):775–788.
19. Freed M, Kupinski MA, Furenlid LR, Wilson DW, Barrett HH. A prototype instrument for single pinhole small animal adaptive SPECT imaging. *Medical Physics*. 2008; 35(5):1912–1925. [Online]. Available: <http://link.aip.org/link/?MPH/35/1912/1>. [PubMed: 18561667]
20. Van Holen, R.; Moore, J.; Clarkson, E.; Furenlid, L.; Barrett, H. Design and validation of an adaptive SPECT system: AdaptiSPECT. *Nuclear Science Symposium Conference Record (NSS/MIC)*, 2010 IEEE; 30 2010–nov. 6 2010; p. 2539-2544.

Fig. 1.
The FastSPECT III stationary SPECT imaging system. The system comprises 20 BazookaSPECT detectors and acquisition hardware that processes ~ 1.23 Gpix/sec.

Fig. 2.

FastSPECT III system calibration. A radioactive point source bead, e.g. $^{99\text{m}}\text{Tc}$, is mechanically stepped in a sparse 3D grid of positions inside the FastSPECT III pinhole aperture. The system has 20 pinholes (one per detector), and for a given source position we acquire 20 projection images that correspond to one column of $\mathbf{H}$, $640 \times 480 \times 20 = 6,144,000$ elements in length. FastSPECT III projection images in the first column, second-third rows show magnified regions of the point source image.

Fig. 3.

The non-zero regions of Figure 2 that correspond to the PSF from one voxel position and 2D Gaussian fits to the associated data. (a) 10 second, raw projection data acquired using a ~ 1 mCi, ^{99m}Tc point source. (b) Least-squares Gaussian fits to the projection data.

Fig. 4.

(a) High-resolution micro-Derenzo phantom. Bore sizes range from $\phi 0.75\text{mm}$ to $\phi 0.35\text{mm}$ [14]. The length of the phantom is 8 mm. (b) On-the-fly, GPU-based MLEM reconstruction of the phantom filled with 5 mCi of $^{99m}\text{TcO}_4^-$. The voxel volume is $104 \times 104 \times 104$, ~ 144 micron voxels.

Fig. 5.
FastSPECT III projection images of a micro-Derenzo phantom.

TABLE I

H matrix storage size assuming a 31×31 region of pixels that sample the detector's response to a point source in FastSPECT III and a 9×9 region for the low-resolution detectors in FastSPECT II.

H-matrix for $104 \times 104 \times 104$ voxels		
Storage Options	FastSPECT II	Fast SPECT III
Full H	0.37 TB	25.14 TB
Sparse H	4.82 GB	80.54 GB
Gaussian Coefficients only	366 MB	515 MB

TABLE II

104³ voxel volume reconstruction time and resolution comparison of the GPU-based OSEM reconstruction where the fraction of the 2D Gaussian (PSF) is varied.

<div>FastSPECT III</div> <div>On-The-Fly OSEM Reconstruction</div> <div>104³ voxels (15 × 15 × 15 mm³)</div> <div>20 Iterations OSEM,</div> <div>5 Subsets</div>				<div>0.35 mm</div> <div>0.75 mm</div> <div>0.4 mm</div> <div>0.6 mm</div> <div>0.45 mm</div> <div>0.5 mm</div>
PSF (2D Gaussian Sampled Region)	H (Sparse)	GPU Computation Time (Total Volume)	Versus CPU (Single Core i7)	Slice (144μm)
<div> </div> <div>31 × 31</div>	80.54 GB	387 sec	60.37×
<div> </div> <div>27 × 27</div>	61 GB	287.8 sec	60.87×
<div> </div> <div>21 × 21</div>	37 GB	148.4 sec	70.57×
<div> </div> <div>15 × 15</div>	18.8 GB	83.22 sec	64×
<div> </div> <div>11 × 11</div>	10.1 GB	43.5 sec	66.09×

FastSPECT III On-The-Fly OSEM Reconstruction 104³ voxels (15 × 15 × 15 mm³) 20 Iterations OSEM, 5 Subsets
PSF (2D Gaussian Sampled Region)	H (Sparse)	GPU Computation Time (Total Volume)	Versus CPU (Single Core i7)	Slice (144μm)
9 × 9	6.7 GB	33.3 sec	58.34×

7 × 7	4.12 GB	27.09 sec	40×

5 × 5	2.09 GB	24.39 sec	25.67×

3 × 3	0.75 GB	24.10 sec	10.68×

1 × 1	0.08 GB	19.52 sec	3.12×

TABLE III

52³ voxel volume reconstruction time and resolution comparison of the GPU-based OSEM reconstruction where the fraction of the 2D Gaussian (PSF) is varied.

FastSPECT III On-The-Fly OSEM Reconstruction 52 ³ voxels (15 × 15 × 15 mm ³) 20 Iterations OSEM, 5 Subsets
PSF (2D Gaussian Sampled Region)	H (Sparse)	GPU Computation Time (Total Volume)	Versus CPU (Single Core i7)	Slice (287.5μm)
31 × 31	10.1 GB	47.27 sec	56.67×

27 × 27	7.63 GB	35.05 sec	57.7×

21 × 21	4.62 GB	18.06 sec	67.8×

15 × 15	2.35 GB	10.02 sec	61.8×

11 × 11	1.26 GB	5.41 sec	61.89×

FastSPECT III On-The-Fly OSEM Reconstruction 52³ voxels (15 × 15 × 15 mm³) 20 Iterations OSEM, 5 Subsets
PSF (2D Gaussian Sampled Region)	H (Sparse)	GPU Computation Time (Total Volume)	Versus CPU (Single Core i7)	Slice (287.5μm)
9 × 9	0.85 GB	4.22 sec	53.8×

7 × 7	0.51 GB	3.48 sec	40.06×

5 × 5	0.262 GB	3.18 sec	23.42×

3 × 3	0.094 GB	3.15 sec	9.98×

1 × 1	0.014 GB	2.62 sec	3.02×