Dynamic PET Reconstruction with Binning in Time

Bence Parajdi, Balázs Kovács, and László Szirmay-Kalos

Budapest University of Technology and Economics (e-mail: szirmay@iit.bme.hu)

Abstract

This paper proposes a scalable dynamic PET reconstruction method. In dynamic PET reconstruction the space-time activity function needs to be recovered from measurements, where the number of spatial basis functions is in the range of a hundred million, the number of temporal basis functions are in the range of a hundred, while the number of events can exceed billions. The complexity of a list mode reconstruction would be the product of these factors, which is far too high in high dose pharmaceutical studies. To reduce the computation time, we propose binning not only in the spatial but also in the temporal domain. We show that such binning just negligibly compromises the accuracy while it can significantly improve the speed of the reconstruction.

1. Introduction

In Positron Emission Tomography (PET) we need to find the spatial positron emission density. At a positron–electron annihilation, two oppositely directed photons are generated, which might be detected by the tomograph. A PET/CT system [15] collects the *events* of simultaneous photon incidents in detector pairs, while the material map of the examined object is obtained by a CT scan. An event is a composition of the identification of the detector pair, also called *Line Of Response* or *LOR* and its time of occurrence, i.e. a pair (L,t) where L is the index of the LOR, which is in itself also a pair of two detector crystals, and t is the time of detection.

In dynamic tomography, we focus on the dynamic nature of biological processes, like accumulation and emptying of drugs in certain organs, and track the variation of the positron density in time. Such studies are essential in pharmaceutical research, and in finding cure for Alzheimer's disease, in particular. Dynamic tomography means that the time of the events is also used and instead of a spatial radiotracer density, a space-time density $x(\vec{v},t)$ needs to be reconstructed. Using the space-time density, the expected number of positron generation in differential volume dv and in differential time dt is $x(\vec{v},t)\mathrm{d}v\mathrm{d}t$.

The unknown space-time activity function is searched in finite-element form:

$$x(\vec{v},t) = \sum_{V=1}^{N_{voxel}} \sum_{W=1}^{N_{time}} x_{V,W} b_V(\vec{v}) \tau_W(t)$$

where $x_{1,1}, x_{1,2}, \ldots, x_{2,1}, \ldots, x_{N_{voxel},N_{time}}$ are unknown coefficients, while $b_V(\vec{v})$ $(V = 1,2,\ldots,N_{voxel})$ and $\tau_W(t)$ $(W = 1,2,\ldots,N_{time})$ are predefined spatial and temporal basis functions, respectively. For example, if $b_V(\vec{v})$ were constant in voxel V, and zero otherwise, then we would obtain a piece-wise constant approximation of the unknown activity function. We choose spline basis functions because they are more accurate than the constant approximation. First order spline, i.e. linear basis functions (bilinear or trilinear in space depending on the number of dimensions) are directly supported by the graphics hardware, and higher order spline functions can be traced back to a sequence of linear interpolations [13, 8, 1].

In static tomography, the time information associated with the events is ignored and we wish to determine the total number of positrons generated in a voxel during the finite measurement time, i.e. the spatial function $x(\vec{v})$. If the time of events is discarded and we consider just the number of events in each LOR, the reconstruction is called of *binned mode*. In this case the input of the reconstruction is the number of hits y_L in each LOR L during the whole measurement time.

Unlike binned mode reconstruction, *list mode reconstruction*[4] handles events individually. Pairs (L,t) defining events can be sorted different ways. For example, we can sort events according to the time of occurrence obtaining a list (L_e,t_e) where $e = 1,2,\ldots,N_{event}$ with N_{event} being the number of events. Alternatively, we can sort events first according to the LOR, then for each LOR, according to the time, generating a list of times $t_{L,1},\ldots,t_{L,y_L}$ for each LOR. This array of variable length lists is called *timeogram*.

For a typical PET system, both the number of LORs and the number of finite elements representing the spatial distribution of the activity may be in the range of several hundred millions, while the number of events may exceed a billion, thus the reconstruction algorithm must scale up well and must be appropriate for high performance computation platforms. Among the high-performance computing possibilities, like FPGAs [3], multi-CPU systems [6], the CELL processor, and GPUs [16], the massively parallel GPU has proven to be the most cost-effective platform for such tasks [2, 5], The critical issue of the GPU programming, and parallel programming in general, is *thread mapping*, i.e. the decomposition of the algorithm to parallel threads that can run efficiently [12].

Generally, global memory access is slow on the GPU, especially when atomic writes are needed to resolve thread collisions. Particle transport needs the consideration of many sources (inputs) and many detectors (outputs). This kind of "many to many" computation can be organized in two different ways. We can take input values one-by-one, obtain the contribution of a single input value to all of the outputs, and accumulate the contributions as different input values are visited. We call this scheme *input-driven* or *scattering*. Alternatively, we can take output values (i.e. equations) one-by-one, and obtain the contribution of all input values to this particular output value. This approach is called *output-driven* or *gathering*. Generally, if possible, gathering type algorithms must be preferred since they can completely remove write collisions and may increase the coherence of memory access.

Although list mode may involve more information, like the timing of individual events, its input size is the number of events unlike in the case of conventional binned reconstruction where the input size is the number of LORs. As dynamic reconstruction is based on the timing of events, a pure binned method is not satisfactory, but either a list mode approach should be taken or the binning should be extended to the time domain as well. Binning in time means that the measurement interval is decomposed to smaller intervals and we obtain just the number of events $y_{L,T}$ in each LOR L and each time interval T ignoring the actual time of the event within the interval. Clearly, by binning in time we loose information, so the reconstruction must be less accurate. However, we can handle all events in a time bin simultaneously, which can greatly increase the performance of the solution method. The objective of this paper is to study the effect of time discretization and propose an optimal binning strategy that maintains accuracy and enables fast solutions.

2. Dynamic PET reconstruction

The correspondence between positron generation and gamma photon detection is established by *scanner sensitivity* $\mathcal{T}(\vec{v} \to L)$ that expresses the probability of generating an event in LOR L given that a positron is emitted in point $\vec{v}$ of volume $\mathcal{V}$. We assume that the scanner sensitivity is constant in time. The scanner sensitivity is a high-dimensional integral of variables unambiguously defining the path of particles from positron emission point $\vec{v}$ to the detector electronics.

The event rate $\lambda_L(t)$ in LOR L at time t is the sum of the contributions of all points in the volume at this time (note that we ignore the time elapsed between positron generation and gamma photon detection):

$$\lambda_L(t) = \int_{\mathcal{V}} x(\vec{v},t)\mathcal{T}(\vec{v} \to L)\mathrm{d}v = \sum_{V=1}^{N_{voxel}} \sum_{W=1}^{N_{time}} \mathbf{A}_{LV} x_{V,W} \tau_W(t)$$

where *system matrix*

$$\mathbf{A}_{LV} = \int_{\mathcal{V}} \mathcal{T}(\vec{v} \to L) b_V(\vec{v})\mathrm{d}v$$

defines the correspondence between voxel V and LOR L.

During iterative *Expectation Maximization* (ML-EM) reconstruction [7], unknown coefficients are found to maximize the probability of the actually measured data. The maximization is done via an iterative process. In every iteration cycle a large number of integrals need to be evaluated simultaneously, for which numerical quadrature rules can be applied [11].

Let us decompose the measurement time (t_{start}, t_{end}) by discrete time instances $t_0 = t_{start}, t_1, \ldots, t_{N_T} = t_{end}$ into finite time intervals $\Delta t_1 = t_1 - t_0, \ldots, \Delta t_{N_T} = t_{N_T} - t_{N_T-1}$ and denote the number of events in LOR L and time interval Δt_T by $y_{L,T}$. The measured number of hits in LOR L in time interval Δt_T follows a Poisson distribution of expectation $\lambda_L(t_T)\Delta t_T$:

$$P\{y_{L,T}\} = \frac{(\lambda_L(t_T)\Delta t_T)^{y_{L,T}}}{y_{L,T}!} \cdot e^{-\lambda_L(t_T)\Delta t_T}.$$

Because of the independence of different LORs and different time intervals, the combined probability considering all LORs and all time intervals is the product of elementary probabilities:

$$\mathcal{L} = \prod_{T=1}^{N_T} \prod_{L=1}^{N_{LOR}} \frac{(\lambda_L(t_T)\Delta t_T)^{y_{L,T}}}{y_{L,T}!} \cdot e^{-\lambda_L(t_T)\Delta t_T}.$$

The log-likelihood function is:

$$\log \mathcal{L} = \sum_{T=1}^{N_T} \sum_{L=1}^{N_{LOR}} \log\left(\frac{(\lambda_L(t_T)\Delta t_T)^{y_{L,T}}}{y_{L,T}!} e^{-\lambda_L(t_T)\Delta t_T}\right) =$$

$$\sum_{L=1}^{N_{LOR}} \left(\sum_{T=1}^{N_T} y_{L,T} \log \lambda_L(t_T) - \sum_{T=1}^{N_T} \lambda_L(t_T)\Delta t_T\right) + C,$$

where

$$C = \sum_{T=1}^{N_T} \sum_{L=1}^{N_{LOR}} y_{L,T} \log(\Delta t_T) - \log(y_{L,T}!)$$

is a constant that depends on the measurement intervals and hit numbers but is independent of activity function λ.

If we choose Δt_T to be infinitesimally small, then $y_{L,T}$ can be either 0 or 1 since the probability of multiple events is in the order of $o(\Delta t_T^2)$. Erasing all zero elements from the sum and replacing the Riemann sum by the integral, we obtain

$$\log \mathcal{L} = \sum_{L=1}^{N_{LOR}} \left(\sum_{e=1}^{y_L} \log \lambda_L(t_{L,e}) - \int_{t_{start}}^{t_{end}} \lambda_L(t) \mathrm{d}t \right) + C.$$

According to the concept of maximum-likelihood reconstruction, unknown coefficients are found to maximize this likelihood. To maximize the multi-variate objective function with inequality constraints of non-negativity, we can use the Kuhn-Tucker conditions, which lead to:

$$x_{V,W} \frac{\partial \log \mathcal{L}}{\partial x_{V,W}} = 0.$$

Computing the partial derivatives, we obtain

$$x_{V,W} \left(\sum_{L=1}^{N_{LOR}} \mathbf{A}_{LV} \left(\sum_{e=1}^{y_L} \frac{\tau_W(t_{L,e})}{\lambda_L(t_{L,e})} - \int_{t_{start}}^{t_{end}} \tau_W(t) \mathrm{d}t \right) \right) = 0$$

for $V = 1, 2, \ldots, N_{voxel}$ and $W = 1, \ldots, N_{time}$ since

$$\frac{\partial \lambda_L(t)}{\partial x_{V,W}} = \mathbf{A}_{LV} \tau_W(t).$$

Rearranging the terms, we obtain the following iteration sequence to find the optimum:

$$x_{V,W}^{(n+1)} = x_{V,W}^{(n)} \frac{\sum_L \mathbf{A}_{LV} \sum_{e=1}^{y_L} \frac{\tau_W(t_{L,e})}{\lambda_L(t_{L,e})}}{\sum_L \mathbf{A}_{LV} \hat{\tau}_W}$$

where

$$\hat{\tau}_W = \int_{t_{start}}^{t_{end}} \tau_W(t) \mathrm{d}t,$$

which can be pre-computed analytically.

It is worth examining the special case when the temporal basis functions are constant 1 in their respective interval and zero otherwise:

$$x_{V,W}^{(n+1)} = x_{V,W}^{(n)} \frac{\sum_L \mathbf{A}_{LV} \frac{y_{L,W}}{\tilde{y}_{L,W}}}{\sum_L \mathbf{A}_{LV}}.$$

where $\tilde{y}_{L,W}$ is the expected number of hits in LOR L in time interval W.

Note that with piece-wise constant basis functions the reconstruction process is decomposed to a sequence of independent static reconstructions in the intervals. From this point of view, non-constant basis functions correspond to a filtering in the time domain.

2.1. Forward projection

The goal of the forward projection is to compute the rate of events at the measured events' impact time based on the current approximation of voxel values at the time sampling points. The current voxel intensity estimate is $x(\vec{v},t)$, whose finite element representation is the array of lists of coefficients $x_{1,1}, x_{1,2}, \ldots, x_{2,1}, \ldots, x_{N_{voxel},N_{time}}$. We store the impact time for each event which will be used in the computation. For each event occurring in LOR L, we have to compute

$$\lambda_L(t_{L,e}) = \sum_{V'=1}^{N_{voxel}} \mathbf{A}_{LV'} \sum_{W'=1}^{N_{time}} x_{V',W'} \tau_{W'}(t_{L,e}),$$

where $t_{L,e}$ is the impact time of the event. As this event rate will divide basis function $\tau_W(t_{L,e})$ in the iteration formula, when $x_{V,W}$ is updated, only those events should be considered where the occurrence time gives non-zero result both for $\tau_{W'}$ and τ_W.

Note that an event can contribute only to a single LOR, so a possible, gathering type implementation is

```
for L = 1 to N_LOR do
    for e = 1 to y_L do
        λ_L(t_e) = 0
        for V' = 1 to N_voxel do
            for W' = 1 to N_time do λ_L(t_e) += A_LV' x_V',W' τ_W'(t_e)
        endfor
    endfor
endfor
```

In this algorithm the most time consuming step is the approximation of system matrix element $\mathbf{A}_{LV'}$, which is the evaluation of a multi-dimensional integral [9, 10, 14].

2.2. Back projection

The back projection's goal is to adjust the voxel values according to the iteration scheme. The denominator can be computed and stored for each $x_{V,W}$ before starting the first iteration because only geometric information is needed for the computation.

The back projection algorithm is:

```
for V = 1 to N_voxel do
    for W = 1 to N_time do
        n = 0   // numerator
        for e = 1 to N_event do
            L = LOR of event e
            n += A_LV τ_W(t_e)/λ_L(t_e)
        endfor
        x_V,W *= n/s_V/τ̂_W
    endfor
endfor
```

This algorithm assumes that *voxel sensitivity* $s_V = \sum_L \mathbf{A}_{LV}$ has been pre-computed.

2.3. Performance issues

The complexity of both forward projection and back projection is $O(N_{event} \cdot N_{voxel} \cdot N_{time})$. One possibility to reduce this complexity is to use temporal basis functions that do not cover the complete measurement time interval. Let us denote the maximum number of temporal basis functions that are non zero for a given time instance by N_{cover}, which can be smaller than N_{time}. If we use constant basis functions that are non-zero only in a single time interval, $N_{cover} = 1$. In case of linear basis functions $N_{cover} = 2$. Generally, B-spline basis functions of degree d result in $N_{cover} = d$. When an event time t_e is substituted into temporal basis function τ_w, we can automatically skip those basis functions that do not cover t_e. This way, the complexity can be reduced to $O(N_{event} \cdot N_{voxel} \cdot N_{cover})$.

It is tempting to select N_{cover} as low as possible and apply constant basis functions, but this is a bad idea. The problem is that an $x_{V,W}$ coefficient will then be reconstructed not from N_{event} events but from only those events that are covered by τ_W. The iteration formula scales voxel values according to the ratios of measured number of hits and their expectations, $y_{L,W}/\tilde{y}_{L,W}$, while the hits follow a Poisson distribution. The variance of the Poisson distribution is then $\tilde{y}_{L,W}$, so the standard deviation of the ratio is $1/\sqrt{\tilde{y}_{L,W}}$. Note that this standard deviation diverges when the expected number of hits goes to zero. Thus, reducing the coverage of temporal basis functions leads to very high noise levels in the reconstruction.

The other factor of the complexity formula is the number of events N_{event}, which is much larger than the number of LORs N_{LOR}, thus attacking this factor can lead to significant performance improvement. An event belongs to a LOR and has a particular time. If we use the concept of binned reconstruction, and the exact time is ignored in these intervals, then different events occurring in the same LOR during a given interval can be merged together.

3. Time binning

Let us return to the finite decomposition of the measurement time by discrete time instances $t_0, \ldots, t_{N_T}$ into finite time intervals $\Delta t_1 = t_1 - t_0, \ldots, \Delta t_{N_T} = t_{N_T} - t_{N_T-1}$. From these time instances, we select those times $t^*_0, \ldots, t^*_{N_{time}}$ which define the local domains of spline basis functions. However, $t_0, \ldots, t_{N_T}$ decomposition is finer, i.e. $N_T > N_{time}$, so a temporal basis function can expand to many intervals. The number of intervals that together provide the domain of a basis function is denoted by N_{expand}. If a new basis function started at every time interval, so time binning was as fine as the basis functions, then $N_{expand} = N_{cover}$ would hold. However, in practical cases $N_{expand} > N_{cover}$, so many neighboring intervals are covered by the same set of basis functions.

The actual time of a hit in an interval is ignored, so it is enough to store how many events occurred in LOR L in time interval Δt_{N_T}, which is denoted by $y_{L,T}$. This discretization assigns an N_T-dimensional vector to each LOR. Whenever an interval is processed during forward projection and back projection, its time is inserted into the temporal basis functions. Note that we should check at most N_{cover} basis functions, because others give zero result for this function.

The forward projection with time binning is as follows:

for $L = 1$ **to** N_{LOR} **do**
  for $T = 1$ **to** N_T **do**
    $\lambda_L[T] = 0$
    W_{start} = first basis function where $\tau_W(t_T)$ is not zero
    for $V' = 1$ **to** N_{voxel} **do**
      for $W' = W_{start}$ **to** $W_{start} + N_{cover}$ **do**
        $\lambda_L[T]$ += $\mathbf{A}_{LV'} x_{V',W'} \tau_{W'}(t_T)$
      endfor
    endfor
  endfor
endfor

The corresponding back projection algorithm is:

for $V = 1$ **to** N_{voxel} **do**
  for $W = 1$ **to** N_{time} **do**
    $n = 0$
    T_{start} = first interval where $\tau_W(t_T)$ is not zero
    for $L = 1$ **to** N_{LOR} **do**
      for $T = T_{start}$ **to** $T_{start} + N_{expand}$ **do**
        n += $\mathbf{A}_{LV} \tau_W(t_T)/\lambda_L \cdot y_{L,T}$
      endfor
    endfor
    $x_{V,W}$ *= $n/s_V/\hat{\tau}_W$
  endfor
endfor

The complexity of the proposed time binned algorithm is $O(N_{LOR} \cdot N_T \cdot N_{voxel} \cdot N_{time})$, which is better than that of the list mode reconstruction if $N_{LOR} \cdot N_T \ll N_{event}$.

4. Results

To demonstrate the results we run experiments on a simple 2D tomograph model (Fig. 1), where $N_{LOR} = 2115$ and $N_{voxel} = 1024$ [11]. We considered the *Two Squares* phantom and the *Point phantom*.

Figs. 2 and 3 show the reconstructed time dependent activity of the two hot squares executing 100 iterations and Fig. 4 is the evaluation of the complete activity map in time after 50 iterations. Note that after this number of iteration steps, the reconstruction is still a little blurry, so some activity is distributed outside the hot squares, causing that the activity in the square is less than the reference value. However, the different time discretization schemes give very similar results, so the proposed time binned method can result in similar accuracy using significantly less computation time.

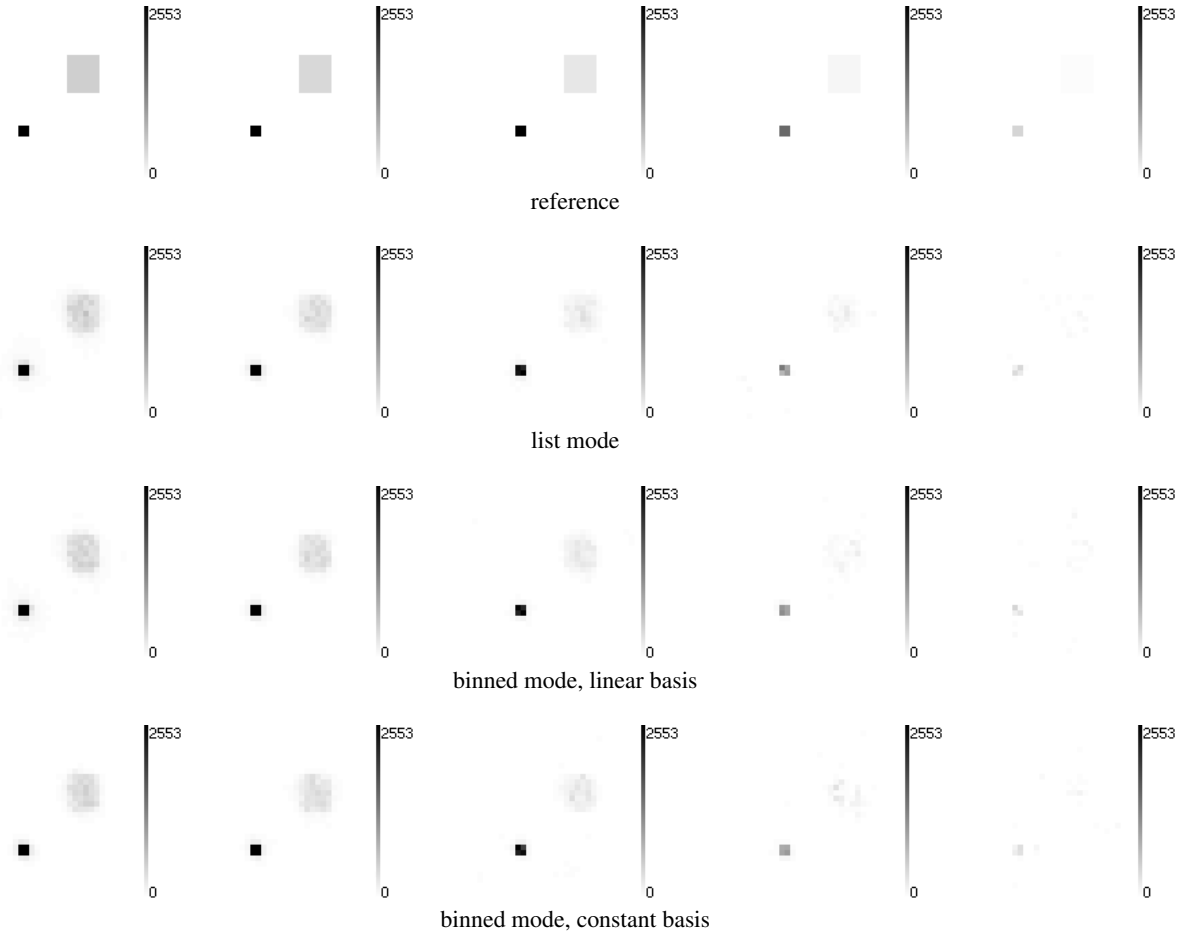

Figure 4: *Snapshots of the Two Squares phantom. The upper row shows the reference in every 3 seconds while the lower rows the corresponding reconstruction results.*

Fig. 5 shows the activity as a function of time of a Point Source phantom. Here, we can also make the conclusion that time binning does not reduce the accuracy of the reconstruction.

5. Conclusions

In this paper we investigated the problem of dynamic PET reconstruction when the total activity in a region of interest needs to be reconstructed as a function of time. We have shown that binning in time but still using non-constant basis functions is possible and can greatly reduce the computation time while maintaning the accuracy of the time consuming list mode reconstruction.

Acknowledgement

This work has been supported by OTKA K–104476.

References

1. B. Csébfalvi. *Interactive Volume-Rendering Techniques for Medical Data Visualization*. PhD thesis, Technische Universität Wien, 2001.

2. M. Magdics et al. TeraTomo project: a fully 3D GPU based reconstruction code for exploiting the imaging capability of the NanoPET/CT system. In *World Molecular Imaging Congress*, 2010.

3. M. Leeser, S. Coric, E. Miller, H. Yu, and M. Trepanier. Parallel-beam backprojection: an FPGA implementation optimized for medical imaging. In *Proc. Tenth Int. Symposium on FPGA*, pages 217–226, 2002.

4. Craig S. Levin et al. Shift-varing line projection using graphics hardware. *US patent No. 0182491 A1*, 2011.

5. M. Magdics et al. Performance Evaluation of Scatter Modeling of the GPU-based "Tera-Tomo" 3D PET Reconstruction.

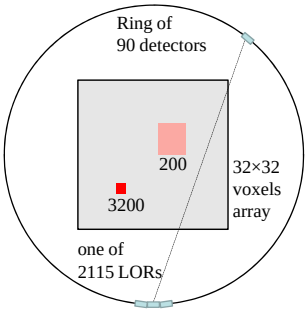

Figure 1: *A simple 2D tomograph model. The detector ring contains 90 detector crystals and each of them is of size 2.2 in voxel units and participates in 47 LORs connecting this crystal to crystals being in the opposite half circle, thus the total number of LORs is* $90 \times 47/2 = 2115$. *The voxel array to be reconstructed is in the middle of the ring and has* 32×32 *resolution, i.e. 1024 voxels.*

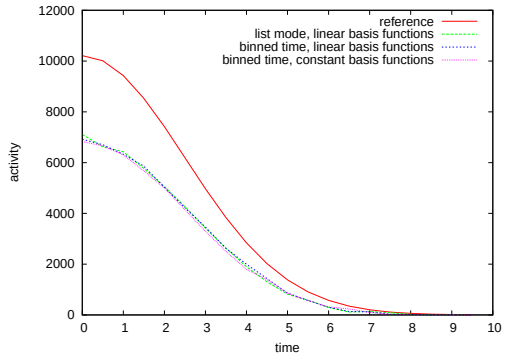

Figure 2: *Total activity as a function of time in the hot square of the higher activity in the Two Squares phantom*

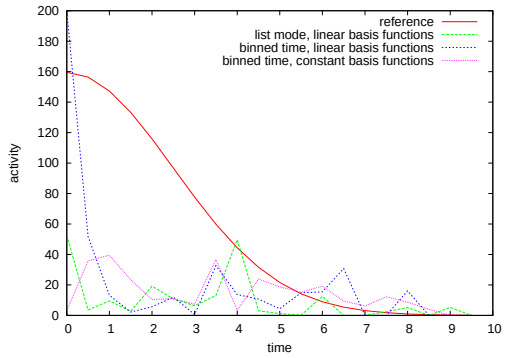

Figure 3: *Total activity as a function of time in the hot square of the lower activity in the Two Squares phantom*

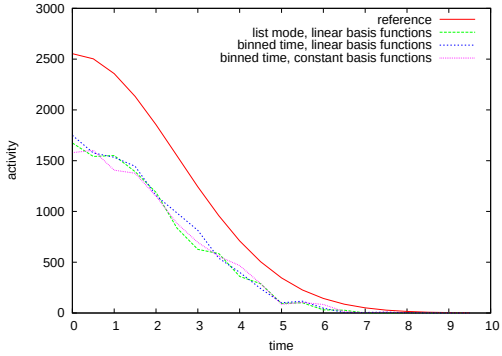

Figure 5: *Activity as a function of time in the Point Source phantom*

In *IEEE Nuclear Science Symposium and Medical Imaging*, pages 4086–4088, 2011.

6. D.W. Shattuck, J. Rapela, E. Asma, A. Chatzioannou, J. Qi, and R.M. Leahy. Internet2-based 3D PET image reconstruction using a PC cluster. *Physics in Medicine and Biology*, 47:2785–2795, 2002.

7. L. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imaging*, 1:113–122, 1982.

8. C. Sigg and M. Hadwiger. Fast third-order texture filtering. In *GPU Gems 2*, pages 313–329. Addison-Wesley, 2005.

9. L. Szirmay-Kalos. *Monte-Carlo Methods in Global Illumination — Photo-realistic Rendering with Randomization*. VDM, Verlag Dr. Müller, Saarbrücken, 2008.

10. L. Szirmay-Kalos, M. Magdics, and B. Tóth. Multiple importance sampling for PET. *IEEE Trans Med Imaging*, 33(4):970–978, 2014.

11. L. Szirmay-Kalos, M. Magdics, B. Tóth, and T. Bükki. Averaging and Metropolis iterations for positron emission tomography. *IEEE Trans Med Imaging*, 32(3):589–600, 2013.

12. L. Szirmay-Kalos and L. Szécsi. General purpose computing on graphics processing units. In A. Iványi, editor, *Algorithms of Informatics*, pages 1451–1495. MondArt Kiadó, Budapest, 2010. http://sirkan.iit.bme.hu/szirmay/gpgpu.pdf.

13. L. Szirmay-Kalos, L. Szécsi, and M. Sbert. *GPU-Based Techniques for Global Illumination Effects*. Morgan and Claypool Publishers, San Rafael, USA, 2008.

14. L. Szirmay-Kalos, B. Tóth, and M. Magdics. Free Path Sampling in High Resolution Inhomogeneous Participating Media. *Computer Graphics Forum*, 30(1):85–97, 2011.

15. D. W. Townsend and T. Beyer. A combined pet/ct scanner: the path to true image fusion. *British Journal of Radiology*, 75:24–30, 2002.

16. F. Xu and K. Mueller. Real-time 3d computed tomographic reconstruction using commodity graphics hardware. *Physics in Medicine and Biology*, pages 3405–3419, 2007.