Lossless Compression of List-Mode 3D PET Data [1]

Evren Asma, David W. Shattuck and Richard M. Leahy
Signal and Image Processing Institute, University of Southern California, Los Angeles, CA 90089-2564

Abstract

We describe lossless methods for compression of 3D PET data. Static list-mode data is re-sorted into sinograms and compressed using a combination of run-length and entropic coding. Dynamic list-mode data is compressed using a sinogram/timogram format in which the arrival times of each photon pair are stored in spatial order, indexed by the sinogram. Compression of the timogram is performed using entropic coding of the differential arrival times in each sinogram bin. We describe these compression methods and present results for static and dynamic data. In the dynamic case, we compare compression of multiple frame sinogram data with sinogram/timogram formats. Both formats produce substantial reductions in data size compared to the raw data, with higher compression factors achieved using the sinogram/timogram format when high temporal resolution is required.

I. Introduction

The potential number of lines of response (LORs) in a PET system increases as the square of the number of detectors. Consequently, the large numbers of detectors in modern high resolution human and animal scanners can produce huge data sets when stored as sinograms. The growth in recent years in sinogram size for a number of different scanners is illustrated in Figure 1. Large sinogram sizes present problems both in terms of reconstruction times and data archiving. Here we address the latter issue.

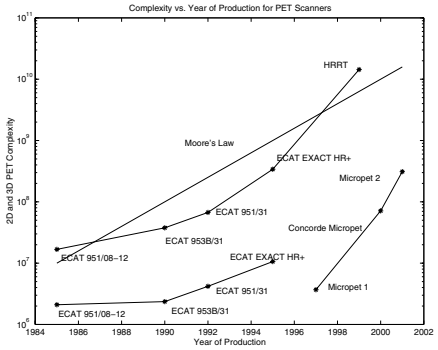


Figure 1: Illustration of the approximate sinogram sizes for 2D and 3D clinical and small animal scanners shown in comparison to "Moore's law", the observation that single-processor computing power doubles roughly every 18 months. The lower curve for the ECAT systems represents 2D complexity, the upper curve represents 3D complexity.

To reduce data sizes and reconstruction times, it is common to rebin the data by adding adjacent rows of a single sinogram and combining sinograms with small angular differences. While effective, this reduces the potential for resolution recovery during reconstruction. An alternative approach is to store the data in list-mode in which the LOR index is stored for each detected event. This becomes efficient as the maximum number of LORs begins to exceed the number of detected photon pairs. Recently, reconstruction methods have been described that make direct use of list-mode data [1], [2].

Storing list-mode data in the chronological order in which events are acquired is inherently inefficient since backprojection then requires random rather than sequential access to the image voxels. It is straightforward to rearrange list-mode data in spatial-order which effectively results in a sparse sinogram. These can then be stored either with the LOR index for each event or, as we describe here, directly using a sinogram format in which lossless compression, based on run-length and entropic coding, is used to efficiently store and retrieve the data.

There are very few previous reports of lossless compression schemes specifically tailored to PET data. Macq *et al* [3] present a scheme in which they use an adaptive pulse code modulator followed by a universal variable length coder which operates on short data blocks. Baker *et al* [4] are concerned with the hardware implementation of Lempel-Ziv coding which achieves high throughput for PET data. These methods exclude entropic compression techniques such as Huffman coding because their objective is to perform on-line compression without knowledge of the prior probabilities. Since we can generate histograms of occurrence frequency while collecting and sorting the data, here we do make use of entropy-based approaches.

Since many PET studies involve the collection of dynamic data, it is also important to consider how multiple sinograms can be efficiently stored. Traditionally, dynamic PET data is stored as a sequence of contiguous frames with one set of sinograms in each frame. Clearly, we can reorder the list-mode data into frames and apply compression as described above to each frame in turn. In this case, the total data size increases roughly linearly with the number of frames.

Here we consider an alternative which can preserve high temporal resolution in the data. We avoid sorting of events into separate sinograms by using a "sinogram/timogram" format [5], [6]. In this format, all events are collected into a standard sinogram that represents a single frame of data corresponding to the entire dynamic study. This is then augmented by a "timogram" that contains the arrival times of each event stored in spatial order so that they are indexed using the values in the associated sinogram. In list-mode, the arrival times are typically quantized at intervals on the order of 1msec. In our work we have used a temporal resolution of 256msec which

[1]This work was supported by the National Cancer Institute under Grant No. R01 CA59794.

0-7803-7324-3/02/$17.00 © 2002 IEEE

allows arrival times for a 100min study to be encoded using 16 bits. Here we describe our approach to efficient storage of data in the sinogram/timogram format and compare the results with lossless compression of multiple frame data. We note that the "timogram" format, and by extension the compression methods investigated here, can also be used in place of raw list-mode to store other attributes of events that can be appended to the sinogram. For example: depth-of-interaction measurements, energy collected in SPECT or PET photon detection, and attributes of photons detected in electronically collimated SPECT systems.

II. Methods and Results

A. Sinogram Compression

The standard data format in many commercial PET systems represents each sinogram element as 2 or 4 byte integers. There is a great deal of redundancy in this format since most sinogram elements contain few counts. In order to explore this redundancy, we applied Lempel-Ziv [7], Huffman [8] and run-length [9] coding followed by Huffman coding of both sinograms and differential sinograms. Differential sinograms store the first sinogram plane of each segment and the differences between consecutive sinogram planes. This can be seen as an attempt to estimate each sinogram plane from the previous plane in the segment. We explain why such a differencing scheme works below.

The number of counts at detector pair i are approximately Poisson(λ_i). Since corresponding entries in consecutive sinograms have almost equal means, the difference sinogram will contain entries that are approximately distributed as the difference of two independent identically distributed (iid) Poisson variables, i.e. Poisson(λ_i) - Poisson(λ_i). In Figure 2 we show the entropies of a Poisson random variable and a differenced Poisson random variable (i.e. Poisson(λ) - Poisson(λ)) as a function of mean λ.

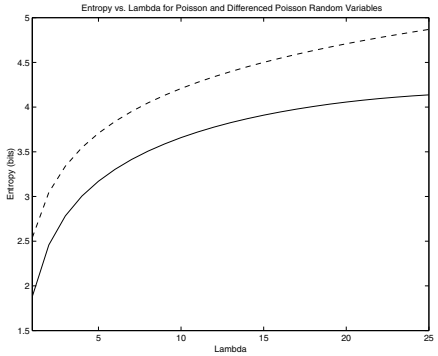
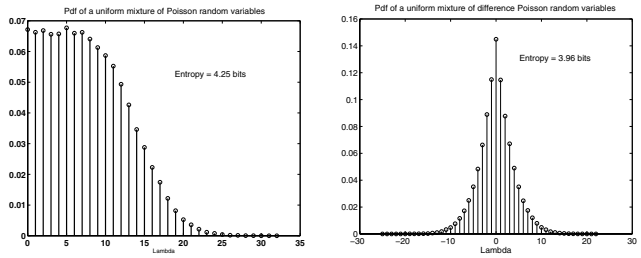


Figure 2: Entropy versus λ for Poisson (solid) and differenced Poisson (dashed) random variables. λ is the mean and variance of the Poisson random variable and is half the variance of the differenced Poisson random variable which is zero mean. We see that for all values of λ, the differenced random variable has higher entropy.

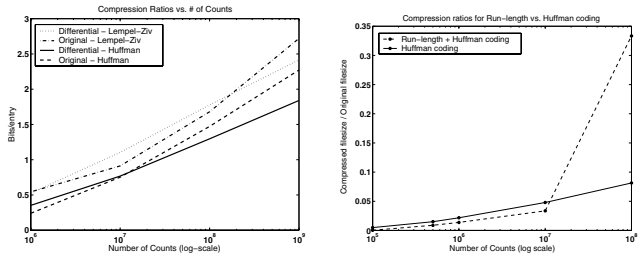
It is clear that for all values of λ, the differenced Poisson random variables have higher entropy. This means that if we had constant mean sinograms, the differential sinograms would have higher entropy, and hence, when compressed would require more, rather than fewer, bits than the original sinograms. However, in realistic cases, sinograms have spatially varying means. Therefore the probability distribution of an ordinary sinogram is a mixture of Poisson random variables with different means. Similarly, the distribution of a differenced sinogram is a mixture of difference Poisson variables with different means. Figure 3 shows the probability density functions for mixtures of Poisson and differenced Poisson random variables. The mixtures were formed by sampling from an underlying mean value uniformly distributed on $[0, 15]$.



(a) (b)

Figure 3: Probability density functions for uniform mixtures of Poisson (left) and difference Poisson (right) random variables with λ uniformly distributed on $[0, 15]$. This demonstrates that the uniform mixtures of difference random variables have lower entropy than uniform mixtures of ordinary Poisson random variables

Entropy calculations using the pdf's in Figure 3 show that the mixture of differenced Poisson random variables has lower entropy than the mixture of ordinary Poisson random variables. Thus entropy based coding of differential sinograms will, in general, achieve better compression ratios. Empirical compression ratios shown in Section III confirm this observation.



(a) (b)

Figure 4: Bits per sinogram entry for four compression methods versus count rates (left) and a direct comparison of Huffman coding (solid) of differential sinograms versus run-length+Huffman (dashed) coding of original sinograms (right)

The entropy of a process is a measure of its information content and accounts for the impact of correlations in the data. It is impossible to reduce entropy by applying a one-to-one transform (sinograms to differential sinograms in our case) [11]. However, Huffman coding is based on separate coding

of each variable and therefore cannot take advantage of correlations in the data. The Huffman code-book is constructed from the univariate mixture density formed by the data and approaches the "empirical entropy" limit of this density. Thus while Huffman coding is optimal for independent random variables, performance can be improved when coding strings of correlated data by first applying a decorrelating transformation. Thus, since the mean (noiseless) values are highly correlated between adjacent sinograms, computation of the differential sinograms reduces the "empirical entropy" resulting in improved compression rates.

Figure 4 (a) shows the average code-length of the compressed data (bits/entry) for Huffman and Lempel-Ziv coding applied to original and differential sinograms with varying count rates simulated to represent data from the CTI ECAT HR+. It is clear that for counts above 1M, Huffman coding applied to differential sinograms achieves the lowest average number of bits/entry. At between 1 and 3 bits/entry, this represents a substantial saving over the typical 16 bits/entry used in uncompressed formats. Figure 4 (b) shows a comparison between pure Huffman coding of differential sinograms and run-length coding followed by Huffman coding of the original sinograms. Run-length coding becomes increasingly advantageous as the number of counts decreases, but performs best for unrealistically low count rates. For moderate count rates, run-length coding provides some extra compression but it becomes increasingly inefficient as the number of counts increases.

B. Timogram Compression

We store temporal information associated with a sinogram as a list of arrival times indexed by sinogram entries. We call this list of arrival times a "timogram" with entries stored as 2-byte integers. To exploit the redundancies in timograms we applied Lempel-Ziv and Huffman coding to timograms and differential timograms.

Differential timograms store inter-arrival times as opposed to actual arrival times. The advantage in using differences is more obvious here compared to the differential sinogram case because we know that the n^{th} arrival time W_n is greater than or equal to the $(n-1)^{st}$ arrival time, W_{n-1}. It is redundant to store W_n if W_{n-1} is known, all we need to store is $T_n \equiv W_n - W_{n-1}$.

We model the positron emissions in each voxel in the volume as an inhomogeneous Poisson process as in our reconstruction work [5]. We denote the rate function at voxel j by $\eta_j(t)$. If processes at all voxels were homogeneous with rate function $\eta_j(t) = \eta_j$, the detection process at detector pair i would also be a homogeneous Poisson process with rate function $\lambda_i(t) = \lambda_i = \sum_j p_{ij}\eta_j$ where p_{ij} is the detection probability.

Inter-arrival times in a homogeneous Poisson process with rate function λ_i are independent and exponentially distributed with mean λ_i [10]. Therefore, the portion of the differential timogram corresponding to a particular detector pair contains independent samples from an exponential distribution. In this case, the histogram of the timogram would be a mixture of exponential random variables with different means. As the number of counts increases, the timogram's empirical histogram approaches the actual mixture density and Huffman coding would then encode independent random variables sampled from a mixture of exponentials.

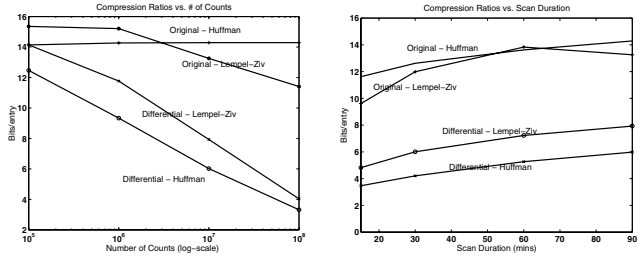
In realistic cases, individual Poisson processes at voxels are inhomogeneous and therefore the detection process at detector pair i is also an inhomogeneous Poisson process with rate function $\lambda_i(t) = \sum_j p_{ij}\eta_j(t)$. In this case, inter-arrival times are no longer independent or identically distributed. Furthermore, the arrival times form a first order Markov process from which the conditional distribution of the n^{th} inter-arrival time T_n can be easily derived as [10]

$$p(T_n|W_{n-1}) = \lambda(W_{n-1}) \exp\left(-\int_{W_{n-1}}^{W_{n-1}+T_n} \lambda(\sigma)d\sigma\right) \quad (1)$$

where $W_{n-1} = \sum_{i=1}^{n-1} T_i$ is the $(n-1)^{st}$ arrival time.

One possible solution to this complication is to fit a set of basis functions (such as decaying exponentials or B-splines) to $\lambda(t)$ using either least squares or maximum likelihood estimation and then to scale the arrival times according to the integral of $\lambda(t)$ (i.e. time warping) and finally to generate the inter-arrival times from the scaled arrival times. This would make the process homogeneous up to the error in the estimation of $\lambda(t)$. However, this technique is both computationally intensive (requires as many estimations as number of sinogram elements) and requires storing as many sinograms as the number of basis functions used in the estimation.

For these reasons, in most cases it is more practical and efficient to apply Huffman coding to loosely correlated inter-arrival times and to pay the price due to the correlation rather than attempting to fully decorrelate arrival times at the expense of increased computation and storage requirements. We can say that the inter-arrival times are loosely correlated because each inter-arrival time depends on the *sum* of previous inter-arrival times and is independent of their actual values as long as their sum remains constant.



(a) (b)

Figure 5: Bits per timogram entry for four compression methods versus count rates (left) and scan duration (right)

Figure 5 (a) shows average bits/entry for Huffman and Lempel-Ziv coding applied to original and differential timograms with varying count rates. The advantage in storing

inter-arrival times is very clear. We also see that Huffman coding outperforms Lempel-Ziv over all realistic count rates. Run-length coding of timograms is not efficient and therefore was not investigated for this application. Fig. 5b compares the compression ratios of the same techniques against scan duration. As scan duration grows, the possible number of inter-arrival times increases which implies a larger codebook for Huffman coding and fewer repetitive patterns for Lempel-Ziv coding. These result in lower compression rates for all four compression techniques.

III. Results

We applied the compression techniques described above to both differential and ordinary sinogram and timogram data representing dynamic studies. We compared the compressed data to standard uncompressed 2-byte representations in a multiframe sinogram format. We simulated brain scan sinograms and timograms for the ECAT HR+ with varying numbers of counts and varying scan durations. We also applied the same compression techniques to two real datasets: a 90 min human C-11 raclopride study (ECAT HR++) and a 90 min C-11 and F-18 dual tracer phantom study (ECAT HR+).

To make the simulations reasonably realistic we added 20% scatter and 10% randoms to the simulated sinograms. In the sinogram-timogram format, we retain only the trues timogram together with trues and randoms sinograms, i.e. the arrival times of the delayed randoms are not stored for reasons described in our previous work on dynamic list-mode image reconstruction [6].

Table 1 shows the compressed data sizes using Huffman coding of differential sinograms for multiframe dynamic studies with varying numbers of counts, with and without rebinning. For a typical dynamic study with 1000M counts, reductions of at least a factor of 10 were achieved in all cases. In Table 2 we show the data sizes for compression of the same dynamic data using the sinogram/timogram format. We show results for 256msec and 1sec temporal resolution with full spatial resolution. From this table we see that at 1sec temporal resolution and full spatial resolution the compressed sinogram/timogram data is approximately one third the size of the compressed standard-rebinned data at 30sec resolution and twice the size at 5min resolution. As we reduce the temporal and/or spatial resolution, the compressed timogram/sinogram data sizes are reduced accordingly.

Using the same methods, we also compressed a C-11 raclopride data set, collected from the ECAT HR++, with 248M trues and 19M randoms. The original file size was 1.1 GB in list mode format, and compressed to 242 MB using compressed multiple sinograms (30 sec resolution) and to 303.56 MB using compressed trues and randoms sinograms augmented with a compressed trues timogram (256 msec resolution). We achieved similar compression ratios for a C-11 F-18 dual tracer study on the ECAT HR+. The study had 405M trues and 67M randoms and the original datasize was 2 GB in list-mode format. Compression reduced filesizes to 384.54 MB as compressed multiple sinograms and 389.3 MB as compressed sinograms plus timograms.

IV. Conclusions

We have seen that by applying Huffman coding to differential sinograms and timograms we can achieve compression ratios above 90% for sinograms and 50% for timograms on both simulated and real data. Compression ratios are slightly lower in real data than in our simulations, probably because of greater complexity in the randoms and scatter profiles. However, the trends that we observe are the same. Entropy based compression, whether applied to our sinogram/timogram format or to standard multiframe sinogram data, is very effective at lossless reduction in data size. These methods are not computationally demanding, and could be routinely used to reduce requirements for both short-term storage and archiving of raw PET data. If high spatial and temporal resolution in dynamic data is called for, then the sinogram/timogram format appears to offer additional savings compared to sequential sinogram compression.

Table 1
Data sizes for compressed 3D ECAT HR+ data as a function of total number of counts in a 90min dynamic study with 30sec and 5min temporal resolution with full spatial resolution (non-rebinned) and standard rebinning.

	Standard rebinning		Non-rebinned sinograms	
Total Counts	30 sec	5 min	30 sec	5min
10M	9.20 MB	14.75 MB	39.41 MB	63.20 MB
100M	93.41 MB	36.26 MB	400.22 MB	155.35 MB
1000M	378.33 MB	63.10 MB	1.62 GB	270.35 MB
10000M	780.68 MB	107.11 MB	3.34 GB	458.91 MB
No compr.	7.16 GB	716 MB	30.6 GB	3.06 GB

Table 2
File sizes for data in Table 1 compressed in sinogram/timogram format with 256sec and 1sec temporal resolution and full spatial resolution.

Total Counts	Sinogram (prompts)+(delayed)	Timogram 256 msec	Timogram 1 sec
10M	7.70+4.74=12.44 MB	8.44 MB	5.64 MB
100M	13.30+13.50=26.80 MB	46.47 MB	23.35 MB
1000M	19.06+22.40=41.46 MB	255.58 MB	96.66 MB
10000M	24.15+31.10=55.25 MB	1.28 GB	400.1 MB

V. References

[1] H. Barrett, T. White and L. Parra : List-mode Likelihood, *Journal of the Optical Society of America* Vol 14, No 11, pp. 2914–2923, 1997

[2] L. Parra and H. Barrett: List-mode Likelihood: EM Algorithm and Image Quality Estimation Demonstrated on 2D PET, *IEEE Trans. on Medical Imaging* , Vol 17, pp. 228–235, 1998

[3] B. Macq, M. Sibomana, A. Coppens, A. Bol and C. Michel, K.Baker and B. Jones : Lossless Compression for

3D PET, *IEEE Trans. on Nucl. Sci.* , Vol 41, No 6, pp. 2765–2770, 1994

[4] K. Baker, W.F. Jones, L.Byars, M.Casey, J. Reed : Lossless Data Compression for Fast Dynamic Frames in Positron Emission Tomography, *Proc. IEEE Nucl. Sci. Symposium and Medical Imaging Conf.* , San Francisco, pp. 1831–1834, 1993

[5] T.E. Nichols, J. Qi and R. M. Leahy : Continuous Time Dynamic PET Imaging Using List Mode Data, *Information Processing in Medical Imaging* , Springer-Verlag, ed. A.Cuba et. al., pp. 98–111, 1999

[6] E. Asma, T.E. Nichols, J.Qi and R.M. Leahy : 4D PET Image Reconstruction from List Mode Data, *Proc. IEEE Nucl. Sci. Symposium and Medical Imaging Conf.* , Lyon, France, 2000

[7] J. Ziv and A.Lempel : A Universal Algorithm for Sequential Data Compression, *IEEE Trans. on Inform. Theory* , Vol 23, No 3, pp. 337–343, 1977

[8] D.A. Huffman : A Method for the Construction of Minimum-Redundancy Codes, *Proc. of the Institute of Radio Engineers* , Vol 40, No 9, pp. 1098–1101, 1952

[9] S.W. Golomb : Run-Length Encodings, *IEEE Trans. on Inform. Theory* , Vol 12, No 4, pp. 399–401, 1966

[10] D.L. Snyder, M.I. Miller : *Random Point Processes in Time and Space* , 2nd Ed., Springer-Verlag, 1991

[11] T. Cover, J. Thomas : *Elements of Information Theory* , Wiley Series in Telecommunications, 1991