

Lossless Compression of Dynamic PET Data

Evren Asma, *Student Member, IEEE*, David W. Shattuck, *Member, IEEE*, and Richard M. Leahy, *Senior Member, IEEE*

Abstract—We describe two approaches to lossless compression of dynamic PET data. In the first, a sequence of sinogram frames are compressed using differential encoding followed by lossless entropy-based compression. The second approach applies lossless compression to data stored in 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 from simulated and real data comparing multiframe and 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.

Index Terms—Entropy, list-mode data, lossless compression.

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 Fig. 1. Large sinogram sizes present problems both in terms of reconstruction times and data archiving. Here we address the latter issue.

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. The format becomes increasingly 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 this list-mode format [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 is used to efficiently store and retrieve the data.

Manuscript received December 21, 2001; revised September 24, 2002. This work was supported in part by the National Institute for Biomedical Imaging and Bioengineering under Grant R01 EB000363.

The authors are with the Signal and Image Processing Institute, University of Southern California, Los Angeles, CA 90089-8564 USA (e-mail: asma@sipi.usc.edu; shattuck@sipi.usc.edu; leahy@sipi.usc.edu).

Digital Object Identifier 10.1109/TNS.2002.807920

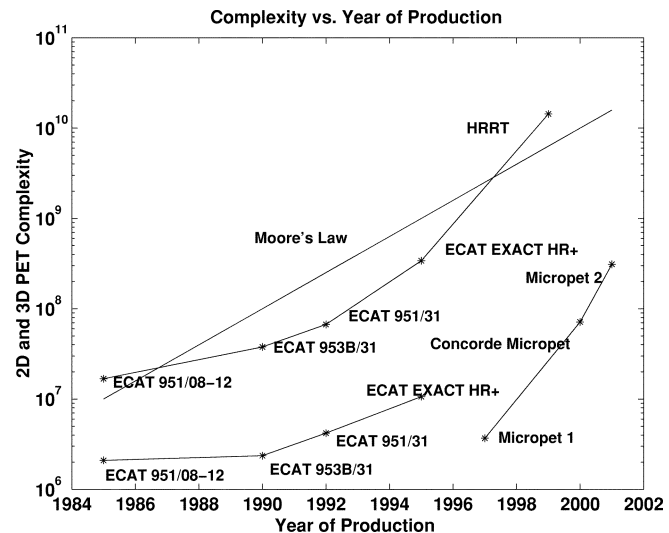


Fig. 1. Illustration of the approximate sinogram sizes for 2-D and 3-D clinical and small animal scanners shown in comparison to “Moore’s law,” the observation that single-processor computing power doubles roughly every 18 months. The lower curve for the ECAT systems represents 2-D complexity, the upper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

There are 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 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

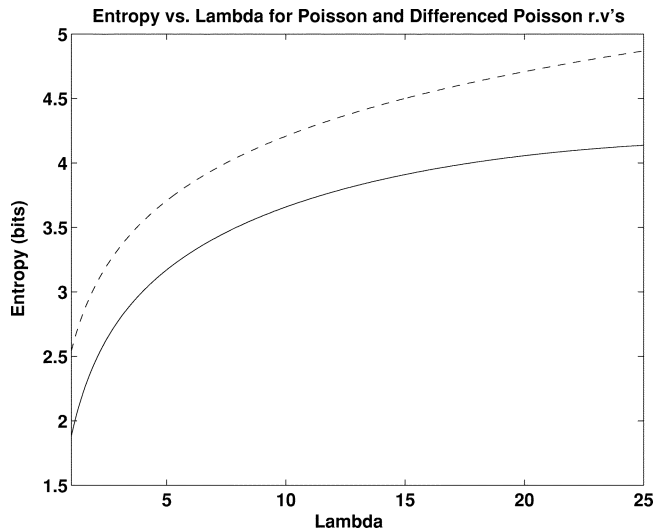


Fig. 2. Entropy versus λ for Poisson (solid) and differenced Poisson (dashed) random variables; λ is the mean and variance of the Poisson random variable. The differenced Poisson has variance 2λ and zero mean. Consequently, for all values of λ , the differenced random variable has higher entropy.

sinogram. In list-mode the arrival times are typically quantized at intervals on the order of 1 ms. In our work, we have used a temporal resolution of 256 ms which allows arrival times for a 100 min 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, such as 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 a combination of run-length [9] and Huffman coding. In each case, we considered 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 $\text{Poisson}(\lambda_i)$. 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 (i.i.d.) Poisson variables, i.e., $\text{Poisson}(\lambda_i) - \text{Poisson}(\lambda_i)$. In Fig. 2 we show the entropies of a Poisson random variable and a differenced Poisson random variable (i.e., $\text{Poisson}(\lambda) - \text{Poisson}(\lambda)$) as a

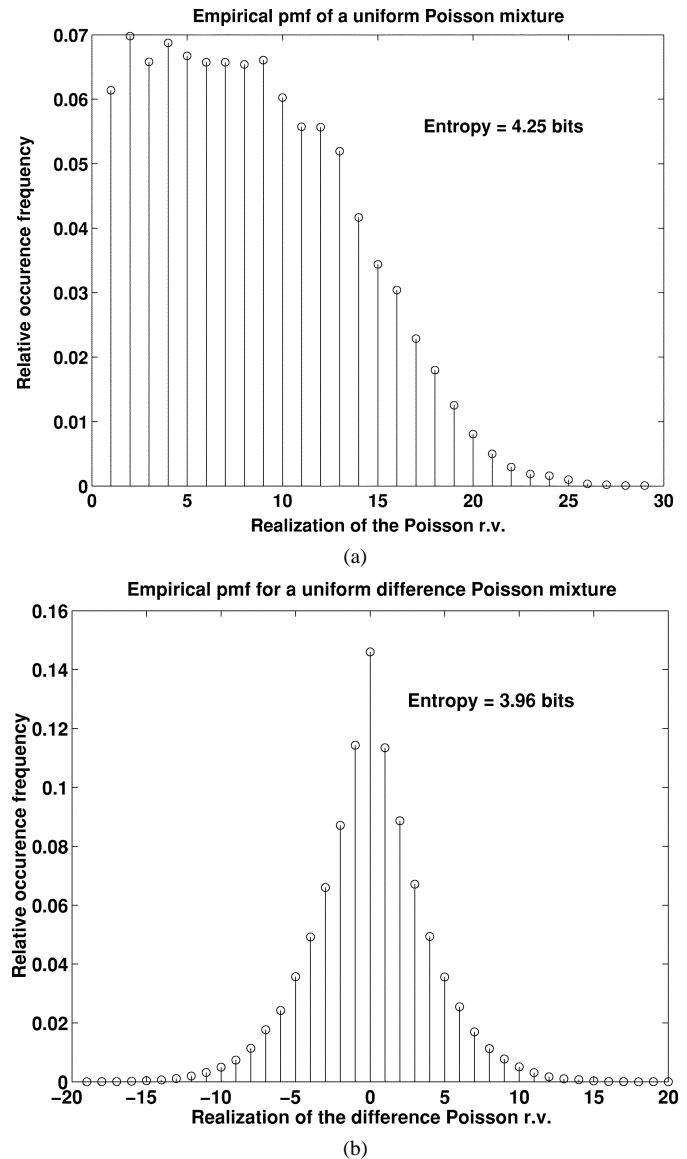


Fig. 3. Probability mass functions for uniform mixtures of (a) Poisson and (b) differenced Poisson random variables with λ uniformly distributed on $[0,15]$. This demonstrates that the uniform mixture of differenced Poisson random variables has lower entropy than the uniform mixture of ordinary Poisson random variables

function of mean λ . 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. In realistic cases, however, 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. Fig. 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 $[0,15]$.

Entropy calculations using the pdfs in Fig. 3 show that the mixture of differenced Poisson random variables has lower en-

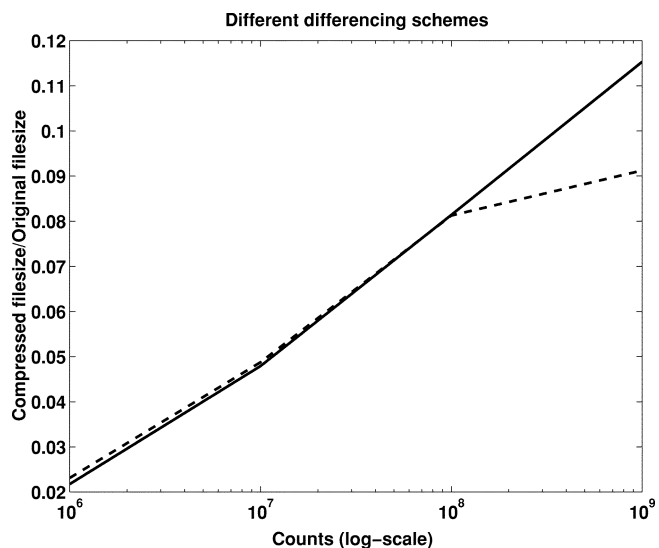


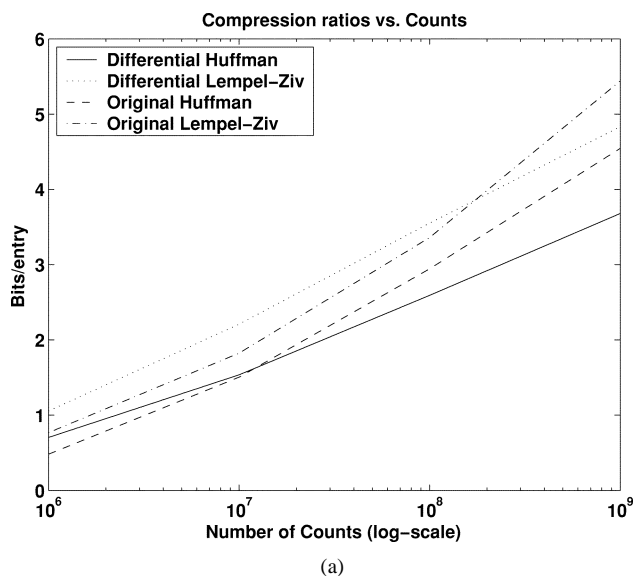
Fig. 4. Comparison of two differencing schemes. The solid line corresponds to the scheme in which each sinogram plane is estimated from the adjacent plane in the same frame. The dashed line represents the scheme in which each sinogram plane is estimated from the same plane in the previous frame. They give very similar compression ratios over a wide range of counts.

trophy 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.

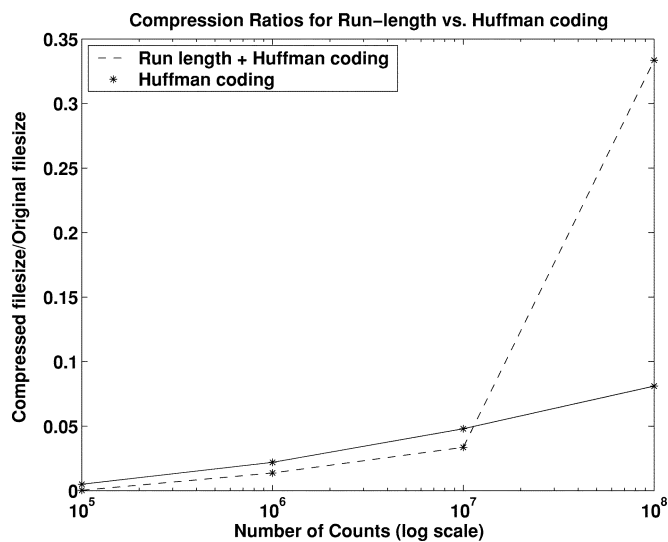
We applied the same differencing idea to dynamic studies by subtracting sinograms corresponding to consecutive frames. In this case we store the first sinogram of a dynamic study and store the differences between consecutive frames. This technique corresponds to estimating each sinogram plane from the same sinogram plane in the previous frame as opposed to estimating it from the adjacent sinogram plane in the same frame. The resulting compression ratios were very similar to the original differencing scheme as shown in Fig. 4.

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.

Fig. 5(a) shows the average code-length of the compressed data (in bits per entry) for Huffman and Lempel–Ziv coding applied to original and differential sinograms with varying count rates simulated to represent 3-D data from the CTI ECAT HR+ (span 9, maximum ring difference 22). For counts above 1M, Huffman coding applied to differential sinograms achieves



(a)



(b)

Fig. 5. (a) Bits per sinogram entry for four compression methods versus count rates applied to data simulated to represent a single frame of 3-D data from the ECAT HR+ scanner. (b) Direct comparison of Huffman coding of differential sinograms (solid) versus run-length+Huffman coding of original sinograms (dashed).

the lowest average number of bits per entry (bpe). At between 1 and 6 bpe, this represents a substantial saving over the typical 16 bpe used in uncompressed formats. Fig. 5(b) shows a comparison between pure Huffman coding of differential sinograms and run-length coding of the original sinograms followed by Huffman coding. 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

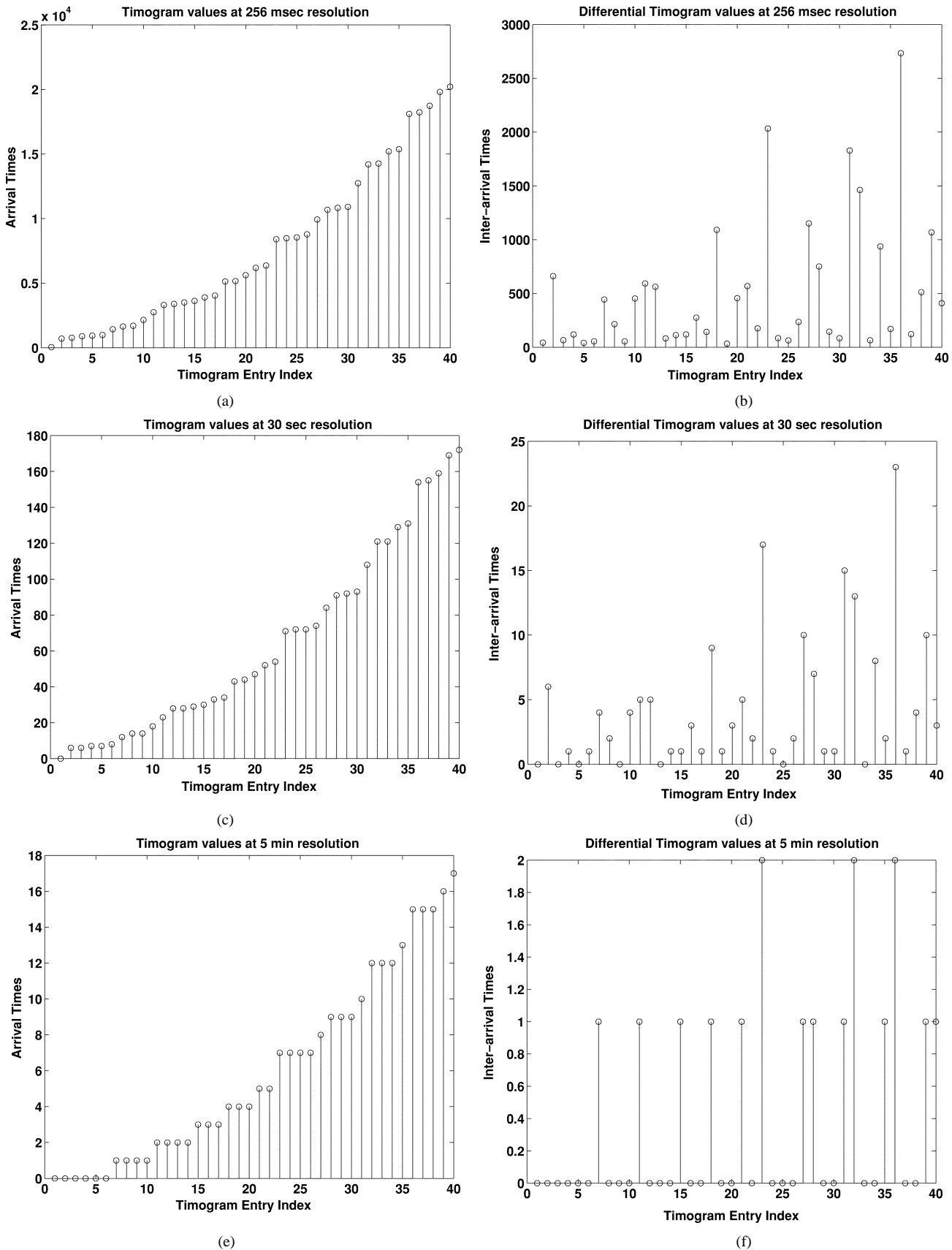


Fig. 6. Illustration of original (a) and differential (b) timogram segments at temporal resolutions of 256 ms; 30 s (c) and (d) and 5 min (e) and (f). The segment corresponds to sinogram entry (45,50) in the central plane (plane 32) of a dual C-11, F-18 tracer study. As the temporal resolution decreases to the order of minutes, timogram entries become increasingly similar because an increasing number of events fall into a single time bin and this produces a high number of zeros in the differential timogram, which can then be compressed by a large factor.

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], [6]. 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.

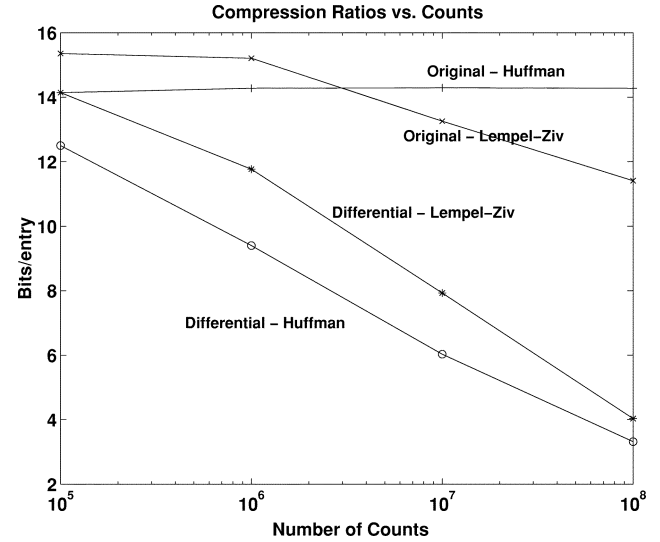
In realistic cases, individual Poisson processes at voxels are inhomogeneous and therefore the detection process at detector pair i is also an inhomogeneous Poisson process with rate function $\lambda_i(t) = \sum_j p_{ij}\eta_j(t)$. In this case, interarrival times are no longer independent or identically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

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

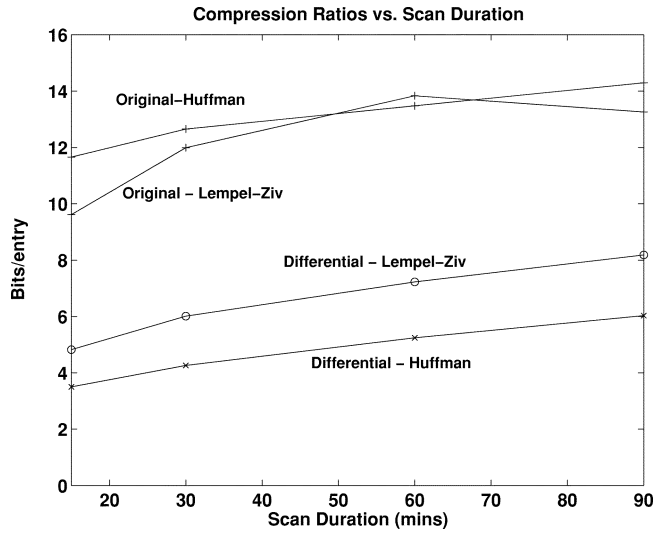
where $W_{n-1} = \sum_{i=1}^{n-1} T_i$ is the $(n - 1)$ st arrival time.

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 interarrival 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 interarrival 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 interarrival times are loosely correlated because each interarrival time depends on the *sum* of previous interarrival times



(a)



(b)

Fig. 7. Bits per timogram entry for four compression methods versus (a) count rate and (b) scan duration.

and is independent of their actual values as long as their sum remains constant.

Fig. 6 illustrates the idea of coding interarrival times and helps explain how high compression ratios can be achieved at low temporal resolutions such as 30 s and 5 min. As the temporal resolution is reduced to a few tens of seconds or minutes (as is typical for most dynamic datasets), Huffman coding of differential timograms becomes increasingly efficient. For the timogram entry shown in Fig. 6, the empirical entropies in bpe, for original and differential timograms respectively are 5.32 and 5.22 bpe at 256 ms resolution; 5.03 and 3.39 bpe at 30 s resolution and 3.69 and 1.19 bpe at 5 min resolution. This shows that a reduction in temporal resolution not only results in better compression ratios due to loss of information compared to the high resolution case (lower entropy per timogram entry) but also because the entropy of the differential timograms drops with count rate far more quickly than that of the raw timograms.

Fig. 7(a) shows average bpe for Huffman and Lempel–Ziv coding applied to original and differential timograms with

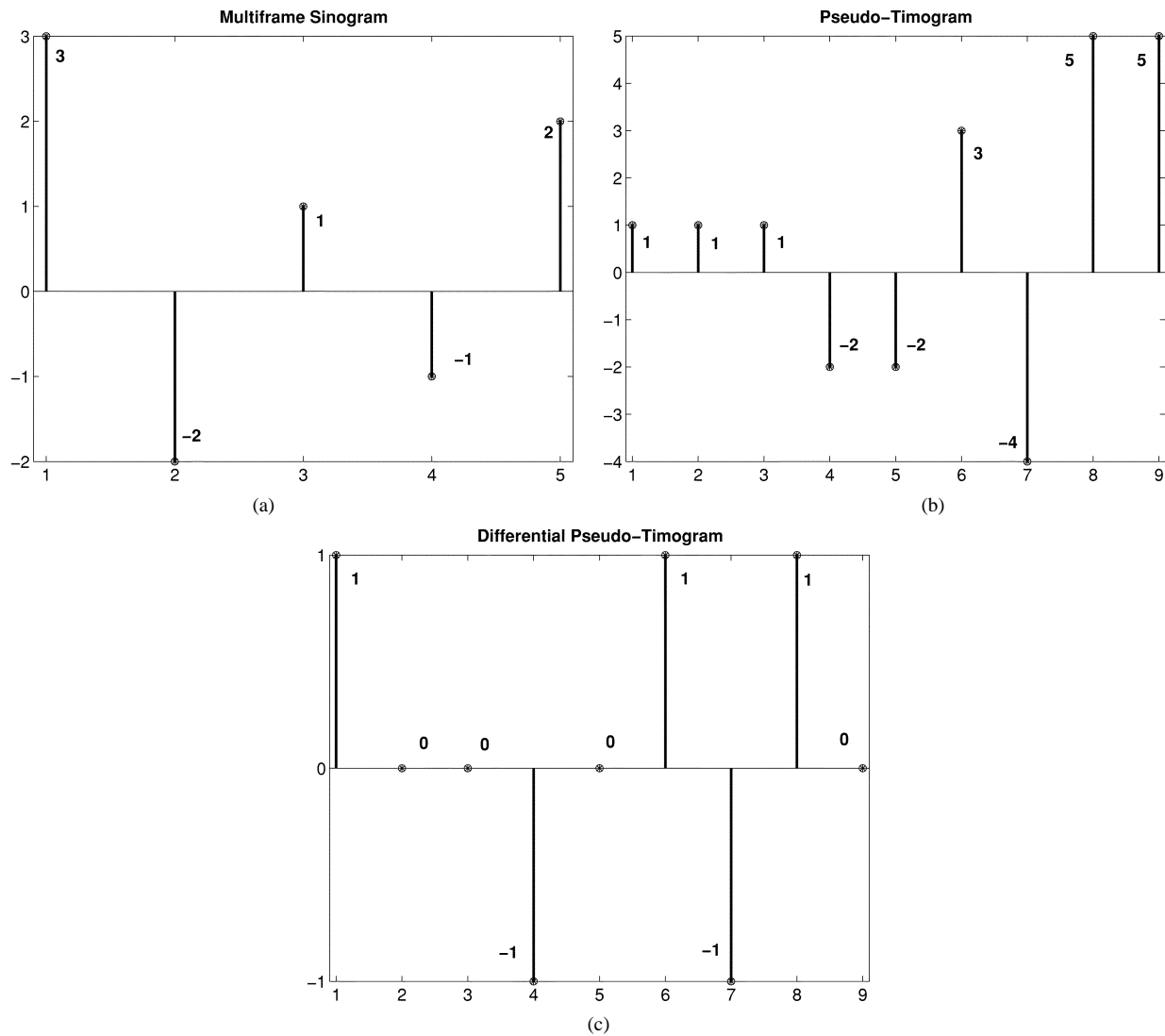


Fig. 8. Illustration of the procedure for handling negative sinogram entries due to randoms subtraction. (a) Multiframe sinogram entries for a single LOR; (b) the corresponding PT; and (c) the corresponding differential PT entries.

varying count rates. The advantage in storing interarrival 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. 7(b) compares the compression ratios of the same techniques against scan duration with a total of 10M counts in each case. As scan duration grows, the possible number of interarrival 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.

C. Multiframe Sinogram Compression

We have also applied our method to standard multiframe data collected from the ECAT HR+ scanner and multiframe data obtained by sorting list-mode data from Concorde P4 and R4 scanners in which we converted the multiframe data to a sinogram plus pseudo timogram format. The “time” index in the pseudo timogram is simply the frame number in the multiframe data. All frames were summed to form the aggregate sinogram.

An interesting problem that we encountered in applying our method to multiframe data was the presence of negative values in some sinogram entries due to randoms correction. We handled this situation by storing the negatives of frame numbers in timogram entries corresponding to negative sinogram values and storing *negatives* of differences between absolute values in differential timograms.

Fig. 8 illustrates the procedure for a single sinogram bin in a five-frame study which has nine events. In this example, entries over five frames are 3, -2 , 1, -1 , and 2, respectively, as shown at the top of Fig. 8. The difficulty in generating a timogram from these data is the presence of negative values at frames 2 and 4. The basic idea in converting this sequence into a pseudo timogram (PT) is to store the negatives of the frame numbers of those frames which contain negative entries. Therefore, entries corresponding to frames 2 and 4 are stored as -2 and -4 respectively as shown in Fig. 8. Finally, the conversion into “differential” pseudo-timograms (DPT) is through taking the differences between the **absolute** values of consecutive PT entries and using negatives of these differences if the corresponding PT entries are negative. In our example, the fourth entry in the DPT

is calculated by first taking the difference between the absolute values of the fourth ($abs(-2) = 2$) and third ($abs(1) = 1$) PT elements. The difference is 1 and since the fourth PT entry is negative, -1 is stored at the fourth entry of the DPT. The entire DPT is shown at the bottom of Fig. 8. Note that this procedure works because a negative entry in the DPT can only be caused by negative values in the corresponding sinogram entry.

In the deconversion process, when DPTs are converted back into PTs, the following rule is applied: If a DPT entry is zero, the corresponding PT entry is the same as the previous PT entry. If a DPT entry is positive, the corresponding PT entry is the cumulative absolute sum of DPT entries up to and including that point. Similarly, if the DPT entry is “negative”, the corresponding PT entry is the “negative” of the cumulative absolute sum of DPT values up to and including that point. Finally, multiframe sinograms are restored by forming a histogram of the PT entries. We found that this method could handle negative entries with minimal increases in the timogram size (increased by the absolute sum of negative sinogram entries), number of codewords (increased by a factor of two) and compressed filesize.

III. RESULTS

A. Simulation Studies

We applied the compression techniques described above to multiframe sinogram and sinogram/timogram data representations of dynamic data. 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.

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 prompt timogram entries together with separate prompt and delayed event 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 I-A shows the compressed data sizes using lossless coding of differential sinograms for multiframe dynamic studies with varying numbers of counts using the best coder for each case. Huffman coding was used for count levels above 10K counts and Lempel–Ziv coding for count levels under 10K. The reason for this is that we use 16-bit Huffman coding (each sinogram entry is taken as a symbol) and for counts under 10K, entropy per symbol goes below 1 bpe. To more closely approach the entropy limit we have to either use a longer word in the Huffman coder (i.e., 32 or 64 bits) or an algorithm which can better exploit long strings of similar entries such as Lempel–Ziv or run-length coding.

Compression factors ranged from 4 to 28 depending on the number of counts and temporal resolution. In Table I (B) and (C), we show the data sizes for compression of the same dynamic data using the sinogram/timogram format. We show results for 256 ms, 1 s, 30 s, and 5 min temporal resolution. For counts above 10M we used Lempel–Ziv coding because in those cases differential timograms have many zeros which causes the entropy to fall below 1 bpe and, hence,

TABLE I
COMPRESSED DATA SIZES FOR DIFFERENTIAL SINOGRAMS WITH 30 s AND 5 MIN TEMPORAL RESOLUTION (UPPER SINOGRAM/TIMOGRAMS WITH 256 MSEC, 1 s, 30 s AND 5 MIN TEMPORAL RESOLUTION (LOWER TWO TO SIMULATED SINGLE SLICE HR+ DATA AS A FUNCTION OF TOTAL NUMBER OF COUNTS IN A 90 MIN DYNAMIC STUDY

Total Counts	30 sec	5 min
1M	1.04 MB	220.57 KB
10M	2.46 MB	385.36 KB
100M	4.10 MB	556.01 KB
1000M	5.73 MB	732.81 KB
No compr.	29.8 MB	2.98 MB

(a)

Total Counts	Sinogram (prompts)+(delayed)
1M	15.10+16.55=31.65 KB
10M	20.33+25.08=45.41 KB
100M	24.81+33.68=58.49 KB
1000M	26.74+42.26=69.00 KB

(b)

Total Counts	Timogram 256 msec	Timogram 1 sec	Timogram 30 sec	Timogram 5min
1M	1.30 MB	1.03 MB	379.52 KB	101.78 KB
10M	8.39 MB	5.69 MB	1.08 MB	192.07 KB
100M	40.23 MB	20.14 MB	1.95 MB	284.95 KB
1000M	105.28 MB	46.90 MB	2.85 MB	359.70 KB

(c)

Huffman coding based on a 16-bit symbol is inefficient. For the same levels of temporal resolution and count levels, the timogram/sinogram format gives higher compression levels than the multiframe sinogram format data, with the savings using the former method ranging from a factor of 1.62 to 2.53.

B. Compression of Experimental List-Mode Data

Using the same methods, we also compressed a 90 min human C-11 raclopride data set collected from ECAT HR++ and a 90 min C-11 and F-18 dual-tracer phantom study, collected from ECAT HR+. Compression results are shown in Table II. We are able to achieve 256 ms resolution using the sinogram/timogram format at approximately the same compressed file size as is needed for 30 s resolution using the differentially compressed multiple frame sinogram format. We achieved similar compression ratios for the C-11 F-18 dual-tracer study. Although the timogram compression results are shown for 256 ms resolution, as the temporal resolution is reduced, the timogram file size reduces accordingly.

C. Compression of Multiframe Sinogram Data

As mentioned in Section II(C), the sinogram/timogram idea is also applicable to multiframe sinogram data with frame numbers replacing event arrival times. The results of multiframe sinogram data compression are summarized in Tables III and IV. Table III compares Huffman coding of the sinogram/PT format against Lempel–Ziv coding of the original multiframe sinograms. We see that unless there is a very high number of counts and relatively low number of frames, Huffman compression of sinogram/PTs provide substantial savings over Lempel–Ziv coding of multiframe sinograms for a wide range of scans.

Table IV compares Huffman coding of sinogram/PTs against Huffman coding of differenced sinograms for multiframe HR+ data. The 2-D studies contained 28 frames each and the 3-D studies contained 6 frames. For the 2-D data, the sinogram/PT format gives around 25% reduction in compressed file sizes compared to the differenced sinogram format. In the 3-D data

TABLE II

COMPRESSION RESULTS FOR EXPERIMENTAL LIST-MODE DATA.
LM: LIST-MODE FILESIZE, CS: COMPRESSED SINOGRAMS (30 S RES.),
PS: PROMPTS SINOGRAM, DS: DELAYED EVENTS SINOGRAM,
TIMO: TIMOGRAM (256 MS RES.), TOTAL:TS+RS+TIMO

Study	C-11 Raclopride	C-11, F-18 Dual Tracer
LM	1.1 GB	2.0 GB
PS	2.60 MB	2.59 MB
DS	1.48 MB	1.28 MB
Timo	262.64 MB	427.01 MB
Total	266.72 MB	430.88 MB
CS	242 MB	384.5 MB

TABLE III

COMPRESSED DATASIZES FOR MULTIFRAME DATASETS FROM ECAT
HR+ AND CONCORDE P4 AND R4 SCANNERS BY LEMPEL-ZIV
CODING OF ORIGINAL FILES (LZ) AND HUFFMAN CODING OF
THE SINOGRAM/PSEUDO-TIMOGRAM FORMAT (HCST)

Scanner	Counts	Frames	Original	LZ	HCST
P4	133 M	63	5.7 GB	114 MB	71 MB
P4	420 M	98	8.8 GB	283 MB	171 MB
P4	1240 M	19	1.7 GB	201 MB	221 MB
R4	441 M	15	340 MB	61 MB	71 MB
2D HR+	27 M	28	146 MB	17 MB	11 MB
2D HR+	24 M	28	146 MB	16 MB	11 MB
3D HR+	379 M	6	119 MB	53 MB	38 MB
3D HR+	557 M	6	119 MB	55 MB	41 MB

sets, the Huffman compressed differenced multiframe format is smaller than the timogram/sinogram format. Since this is counter to all other examples we show here, this may be due to the relatively small number of frames (6) in these 3-D studies, indicating that for small numbers of frames, the multiframe format may be preferable.

IV. CONCLUSION

We have seen that by applying Huffman coding to differential static sinograms, multiframe differential sinograms and differential sinogram/timogram formats, we can achieve substantial compression ratios compared to uncompressed 2 byte per element sinogram formats. 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

TABLE IV

RAW DATA AND COMPRESSED DATA SIZES FOR CLINICAL 2-D AND
3-D HR+ MULTIFRAME DATASETS USING HUFFMAN CODING OF
THE DIFFERENTIAL SINOGRAMS (HCDS) AND HUFFMAN CODING
OF THE SINOGRAM/PT (HCST) FORMAT

Study	RAW	HCDS	HCST
2D, 16.8M	146.3 MB	13.10 MB	9.75 MB
2D, 19.7M	146.3 MB	13.69 MB	10.45 MB
3D, 379M	118.9 MB	33.54 MB	37.78 MB
3D, 557M	118.9 MB	34.20 MB	41.60 MB

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.

REFERENCES

- [1] H. Barrett, T. White, and L. Parra, "List-mode likelihood," *J. Opt. Soc. Amer.*, 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 2-D PET," *IEEE Trans. Med. Imaging*, vol. 17, pp. 228–235, Apr. 1998.
- [3] B. Macq, M. Sibomana, A. Coppens, A. Bol, C. Michel, K. Baker, and B. Jones, "Lossless compression for 3-D PET," *IEEE Trans. Nucl. Sci.*, vol. 41, pp. 2765–2770, Dec. 1994.
- [4] K. Baker, W. F. Jones, L. Byars, M. Casey, and J. Reed, "Lossless data compression for fast dynamic frames in positron emission tomography," in *Proc. IEEE Nuclear Science Symp. Medical Imaging Conf.*, San Francisco, CA, 1993, pp. 1831–1834.
- [5] T. E. Nichols *et al.*, "Continuous time dynamic PET imaging using list mode data," in *Information Processing in Medical Imaging*, A. Cuba *et al.*, Eds. New York: Springer-Verlag, 1999, pp. 98–111.
- [6] T. E. Nichols, J. Qi, E. Asma, and R. M. Leahy, "Spatiotemporal reconstruction of list-mode PET data," *IEEE Trans. Med. Imaging*, vol. 21, pp. 396–405, Apr. 2002.
- [7] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 337–343, May 1977.
- [8] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. Institute of Radio Engineers*, vol. 40, no. 9, pp. 1098–1101, 1952.
- [9] S. W. Golomb, "Run-length encodings," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 399–401, Sept. 1966.
- [10] D. L. Snyder and M. I. Miller, *Random Point Processes in Time and Space*, 2nd ed. New York: Springer-Verlag, 1991.
- [11] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.