Lossy compression of TPC data and trajectory tracking efficiency for the ALICE experiment ⋆

A. Nicolaucig [a], M. Ivanov [b], M. Mattavelli [a,*]

[a] *Signal Processing Laboratory LTS3, Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland*

[b] *CERN, CH-1211 Geneva 23, Switzerland*

Abstract

In this paper a quasi-lossless algorithm for the on-line compression of the data generated by the Time Projection Chamber (TPC) detector of the ALICE experiment at CERN is described. The algorithm is based on a lossy source code modeling technique, i.e. it is based on a source model which is lossy if samples of the TPC signal are considered one by one; conversely, the source model is lossless or quasi-lossless if some physical quantities that are of main interest for the experiment are considered. These quantities are the area and the location of the center of mass of each TPC signal pulse, representing the pulse charge and the time localization of the pulse.

So as to evaluate the consequences of the error introduced by the lossy compression process, the results of the trajectory tracking algorithms that process data off-line after the experiment are analyzed, in particular, versus their sensibility to the noise introduced by the compression. Two different versions of these off-line algorithms are described, performing cluster finding and particle tracking. The results on how these algorithms are affected by the lossy compression are reported.

Entropy coding can be applied to the set of events defined by the source model to reduce the bit rate to the corresponding source entropy. Using TPC simulated data according to the expected ALICE TPC performance, the compression algorithm achieves a data reduction in the range of 34.2% down to 23.7% of the original data rate depending on the desired precision on the pulse center of mass.

The number of operations per input symbol required to implement the algorithm is relatively low, so that a real-time implementation of the compression process embedded in the TPC data acquisition chain using low-cost integrated electronics is a realistic option to effectively reduce the data storing cost of ALICE experiment.

Key words: ALICE, TPC, compression, lossy, cluster finding, tracking
PACS: 29.40.Gx
1991 MSC: 94A29, 94A45

Introduction

ALICE (A Large Ion Collider Experiment) is an experiment that will be held in 2005 at the LHC (Large Hadron Collider) at CERN [1,2]. The experiment will study collisions between heavy ions with energies around 5.5 TeV. The collisions will take place at the center of a set of several detectors, which are designed to track and identify the particles produced.

One of the main detectors of the ALICE experiment is the Time Projection Chamber (TPC). The TPC is a large horizontal cylinder, filled with gas, where a suitable axial electric field is present. When particles pass through, they ionize the gas atoms, and the resulting electrons drift in the electric field. By measuring the arrival of electrons at the end of the chamber, the TPC can reconstruct the path of the original charged particles. The electrons are collected by more than 570 000 sensitive pads where the signal is acquired in the form of pulses, each corresponding to the passage of one particle. This signal is amplified by a preamplifier/shaper and digitalized by a 10-bit A/D converter at a sampling frequency of 5.66 MHz. The digitalized signal is processed and formatted by an Application Specific Integrated Circuit (ASIC) called ALTRO (ALICE TPC Read-Out). At this stage, the overall throughput of the 570 000 channels is around 8.4 GByte/s.

Considering that the duration of the experiment is forecasted in a few months time lap, it is clear that the amount of data to be collected is expected to be extremely large. So as to keep the complexity and cost of the data storage equipment as low as possible, the goal is to reduce the volume of data using suitable data compression methods. The cost reduction of the data storage system can be considered roughly proportional to the data compression factor. Furthermore, it is advisable to implement the compression system in the front-end electronics at the output of the ALTRO circuit, so that the cost for the optical links, which carry data out of the chamber to the following stages of the acquisition chain, is also reduced [3,4]. Equivalently, once dimensions of optical links are decided, the duration of the experiment can be reduced thanks to compression: indeed, being the optical links the bottle-neck of the acquisition chain, they limit the storage bandwidth so that compression will allow to send the same quantity of information to the storage devices in a shorter period of time reducing the overall duration of the experiment.

Compression techniques can be classified into lossless and lossy depending on how the model of the information source defines, or better *models*, the set of

⋆ Authors would like to thank Karel Safarik and Luciano Musa for the information and the support for this work.

* Corresponding author. Telephone: +41 21 693 6984. Fax: +41 21 693 4663.
Email address: Marco.Mattavelli@epfl.ch (M. Mattavelli).

events that are then entropy coded. Using a lossless source model the data can be exactly reconstructed as they were before compression [5,6]. The use of a lossy source model, justified by the fact that it generally can provide significantly higher compression ratios compared to lossless models, has the drawback that an error in the reconstruction of data must be accepted. Lossy source models have become very popular in the last decade in the field of audio and video compression for their remarkable performance [7–10]. Lossy models have been carefully designed so that reconstruction errors are not perceived using psychovisual or psychoacoustic models or they remain comparable with the intrinsic signal noise.

Obviously, for physical data psychovisual or psychoacoustic tests are meaningless or even not applicable since the TPC signal is not to be observed by the human eye or ear. In [11], the compression noise introduced on the sample values by the described lossy or quasi-lossless techniques has been evaluated in terms of RMSE of the introduced error. In reality the RMSE, despite being a simple and well known distortion measure, is not a meaningful measure in this case. The fundamental information that has to be extracted from TPC data is not the sample values themselves but the physical quantities that enable the reconstruction of particle trajectories. In fact, the correct way to evaluate the importance of the distortion error introduced by the compression-decompression process is related to the high level information that is carried by the data. In particular, TPC data are collected with the objective to be processed by off-line algorithms to calculate particle energy and trajectory. Therefore, the most effective way to measure the consequences of the compression distortion error, is to observe how the performances of the algorithms for the extraction of energy and trajectories are affected by the compression-decompression process. A simple way to obtain these evaluations is to apply the cluster finding and tracking algorithms on both simulated data and their compressed-decompressed version and compare performances.

This paper is organized as follows: after a brief description of the nature of the TPC data that are compressed, the cluster finding and tracking algorithms are described. Then a rigorously speaking lossy source model, in which, however, some quantities of physical interest such as the energy (area of the electrical pulse) and the temporal position of each pulse (center of mass of the pulse) registered by the TPC pads are preserved without losses, is presented. The compression performances of the algorithm are reported, and the corresponding computational complexity is briefly discussed, aiming at evaluating a possible implementation of the system on low-cost electronic devices. Finally, the effects of the distortion error introduced by the compression-decompression process is analyzed by comparing cluster finding and tracking performances on the original data and on their compressed-decompressed version.

1 The ALICE TPC read-out data format

Before describing the compression algorithm, it is necessary to spend a few words on the format of data at the output of ALTRO circuit so as to understand how the compression algorithms are applied. Such data are indeed the input of the compression system [1,12].

In the so-called ALTRO data format, only the samples over a given threshold are considered, while the others are discarded. This means that, if we call *bunch* a group of adjacent over-threshold samples coming from one pad, the signal can be represented "bunch by bunch". More precisely, a bunch is described by three fields: temporal information (temporal position of the last sample in the bunch, one 10-bit word [1]), bunch length (i.e., number of samples [2] in the bunch, one 10-bit word), and sample amplitude values (10-bit words, i.e. a range between 0 and 1023).

2 Cluster finding and tracking algorithms

For off-line tracking in the ALICE TPC a classical approach was chosen. Before the tracking itself, two-dimensional *clusters* in the pad-row/time plane have to be found. A cluster is a group of non-zero samples which are temporally and spatially adjacent, so that they belong to adjacent time-bins or adjacent pads in the same pad row; in other words, one cluster is made of several bunches in adjacent pads in one pad row. Then the position of the corresponding space points, which are interpreted as the crossing point between the tracks and the centers of the pad-rows, is calculated. An additional quantity that has to be known after cluster finding is the charge deposited in the clusters. This information is necessary for particle identification by dE/dx measurements. In this paper two simple variants of the cluster finder are considered.

In the first cluster finder algorithm, "preclusters" are selected, i.e. groups of adjacent cells in pad-row/time plane whose signal is above the zero-suppression threshold. Then, for each precluster all local maxima are found. If there is only one local maximum the precluster is associated to one track. In this case the center of gravity of the precluster is stored as the reconstructed space point. If there are several local maxima the precluster is split into the corresponding

[1] It may be noted that, for each trigger selected collision, the acquisition process completes in 88 μs, which implies, at a sampling frequency of 5.66 MHz, a range for time information between 0 and 499, so that one 10-bit word suffices.

[2] Actually, the value transmitted by the ALTRO circuit is the number of samples *plus one* [1].

clusters in the following way: for each local maximum a group of adjacent cells with the signal greater than the signal at the nearest saddle point is selected. Then the centers of gravity of these groups of cells are taken as the reconstructed positions of the corresponding space points.

In the second approach a cluster is associated to each local maximum and the center of gravity of the closest 8 cells is taken. This approach is less sensitive to the fluctuation of the cluster shape and to the track overlaps.

The cluster unfolding could be improved, if tracking and cluster finding are performed simultaneously in the so called parallel tracking procedure. This approach is currently under study.

3 Lossy compression of TPC data

The objective of investigating lossy compression of TPC data is to explore the possibility of increasing the compression ratio beyond the limits achieved by the lossless compression techniques described in [11] while at the same time preserving in the compressed signal the quantities that are of main interest for the experiment. Such quantities are the energy and the temporal and spatial positions of clusters which are calculated by the above described off-line algorithms on the stored data. Therefore, the main idea is not to preserve the value of each single signal sample, as it is for the lossless compression techniques, but to code the information contained in the signal with the lowest number of bits so that the off-line processing is not affected or, at least, is only marginally affected. Ideally, an optimal acquisition system would directly evaluate such quantities on-line and directly code and store them. Unfortunately, the cluster finding and tracking algorithms are computationally too complex to be performed on-line. Furthermore, this approach would require data coming from one whole row (from 67 to 139 pads). Although the compression system receives data from up to 4 000 pads, the order in which they are received is neither geometrical nor fixed, since it depends on the physical connections of front-end cards to the read-out chambers and on the fullness of the output buffers of the ALTRO chips. This means that, in order to apply "row-oriented" algorithms, it would be necessary to buffer data from all 4 000 channels before performing cluster finding, energy and position computation and coding. This process would introduce delay and memory requirements to the acquisition system that cannot easily be satisfied.

Therefore, appropriate quantities have been defined which are related to the signal coming from each pad, i.e. referred to bunches, from which the energy, the temporal and the spatial position of the clusters can be successively computed. Such quantities are the area (or energy) of a bunch, defined as the

sum of its samples, and the Center of Mass (CoM), defined as its temporal position. In the post-processing phase the energy of the cluster can be calculated as the sum of the energies of the bunches it contains, and the positions as the weighted averages of the bunch positions. More complex is the case of overlapped clusters, in which contributions from different traces must be separated before processing.

It has to be noticed that the area and the CoM of the bunches are coded losslessly; this means that, even if the compression does not preserve the single values of the samples, it does preserve these quantities without any loss. To be more precise, only a quantization error for CoMs temporal position is introduced, but such error can be reduced below the error of the original signal.

3.1 Area of bunches

The area of a bunch is simply evaluated as the sum of the values of its samples. Direct coding on probability distribution is applied to the value obtained. In future work the quantization of the area could also be considered as a mean to further improve the compression ratio.

3.2 Center of mass of bunches

The position of the CoM of a bunch is evaluated as the average temporal position of its samples, i.e. $t_{CoM} = (\sum_i s_i t_i) / (\sum_i s_i)$, where s_i and t_i are the values and the temporal positions of the samples of the bunch, respectively. CoM positions are coded differentially, i.e. their values are substituted by the distances between CoMs of consecutive bunches. However, the distance information cannot be coded *as is*, i.e. without quantization; indeed, the number of possible values, although finite, is very large, so that the entropy of such model is very large. Moreover, direct coding of the exact CoM differential positions is also useless in practice because such precision gives an error which is far below the intrinsic quantization noise of the original signal.

Consequently, a quantization stage is appropriate before coding, in order to reduce the number of possible values that CoMs can assume, and consequently to reduce the entropy.

Obviously, quantization can be applied with different quantization steps; by increasing the resolution, i.e. decreasing the quantization interval, the quantization error decreases while the entropy increases. In order to set the quantization step, it makes sense to impose a quantization error comparable with

the error which is intrinsic to the data.

The intrinsic error of the reconstructed CoM for the ALICE TPC is mainly given by the electron diffusion along the drift length, the landau fluctuation of the ionization along the particle trajectory and the gas gain fluctuation; these contributions are summarized by the following:

$$\sigma^2_{CoM} = K_{df}\,\frac{\sigma^2_{df}\,L_{dr}}{N_{el}} + LF(N_{pr})\,\frac{\tan^2\theta\,L_{pd}}{12\,N_{pr}} + K^2 \qquad (1)$$

where $K_{df} = 2$ is due to the gas gain fluctuation, σ_{df} is the diffusion constant of the gas, L_{dr} the drift length, N_{el} is the number of electrons (proportional to the bunch area), N_{pr} the number of generated primary electrons, $LF(N_{pr})$ a factor depending on the number of primaries and taking into account the landau fluctuation of the charge deposit along the trajectory, $\tan\theta$ the tangent of the deep angle, L_{pd} the pad length; K^2 represents the contribution to the intrinsic error due to the threshold applied on the sample values and to the noise introduced by the front-end electronics and the ADC quantization, which in fact is negligible with respect to the first two addends.

As shown in equation 1, the error on CoMs due to the diffusion and angular effect depends on the deposited charge (through N_{pr} and N_{el}); for minimum ionizing particles (MIP) the typical resolution is about $\sigma = 1.1$ mm, while for high ionizing proton about $\sigma = 0.6$ mm. In the time direction, these correspond respectively to $0.20\,T_s$ and $0.11\,T_s$, T_s being the width of a time bin, i.e. the sampling period of the ADCs.

Consequently, a resolution equal to $T_s/4$ has been chosen; however, it has to be added that different quantization intervals, and in particular $T_s/32$ and T_s, have also been considered in the performance estimations.

3.3 Bunches from overlapping traces

The two parameters considered, i.e. area and CoM, provide a good description of the bunches in the case of *simple* bunches. However, such description is not sufficiently detailed for bunches which result from the superposition of two traces; in this case, two temporally close pulses are registered by a pad, with their tails overlapping, so that they are represented by a single bunch in the ALTRO data format. In this case it is necessary, before evaluating area and CoM of these bunches, to separate the contributions due to the different traces.

Rigorously speaking, this operation should be performed exactly in the same way as it is done by the above-mentioned off-line cluster finding and tracking

Fig. 1. Example of "cut" of a bunch originated by two temporally close traces. A 5-sample bunch is cut into two 3-sample ones.

algorithms. However, since these algorithms operate on one cluster data, which are, as mentioned above, not all at once available to the compression system, in the present study a simple technique has been used, since the focus here is on the possibility of higher compression rate; it is likely in fact that more specific splitting algorithms do not imply relevant changes in the entropy of the quantity to code. Specifically, each bunch having a relative minimum is "cut" in correspondence with the minimum (Fig. 1); two new bunches are then obtained, and the sample in the intermediate position, i.e. the one of the local minimum, is divided by two, assigning half of it to each bunch. In terms of compression performance, this approach, though simple, should yield results very close to those provided by the future, more sophisticated, schemes.

3.4 Reconstruction of the TPC signal

The compressed signal, represented, as already mentioned, by the area and the CoM of the bunches, has to be decompressed in form of a standard sampled signal to be transparent with the standard off-line processing stage, i.e. to be taken back to the ALTRO data format, in order to be processed by the off-line algorithms (which are, in fact, designed to process this kind of data). A procedure has been developed which, starting from the area and the CoM of the bunch, reconstructs samples; obviously some errors are introduced on their values if these are compared to the original signal. The precise reconstruction

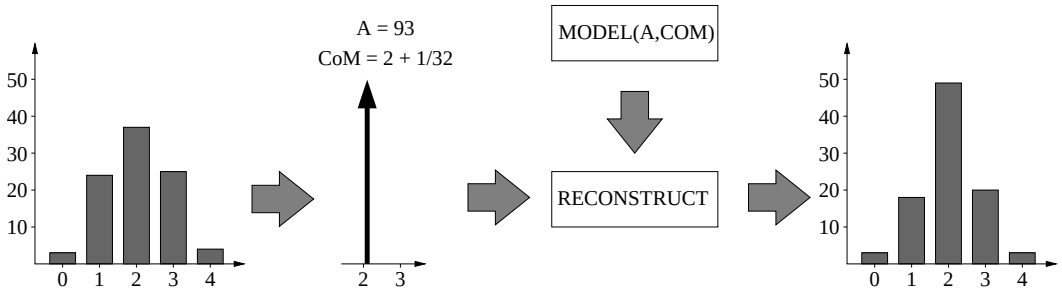

Fig. 2. Example of signal reconstruction for a bunch having area equal to 93 and decimal part of the CoM equal to 1/32.

procedure is described below.

The objective is to capture the average shape of pulses so as to minimize the sample-by-sample reconstruction error. All the bunches are classified according to the {Area, CoM} couple [3] and aligned at their CoM, and the mean value of the samples is computed for each time-bin. Thus, such operation defines a sort of "average bunch" (AB). Obviously, this AB has the same area and CoM as the bunches it represents because the computation of the mean value does not alter these parameters. Moreover, it may be noted that this operation minimizes the mean square error between the actual and the mean samples. What is obtained is a "library" of ABs, each characterized by a different {Area, CoM} couple.

The procedure then associates to each {Area, CoM} couple the corresponding AB, and re-builds the temporal sequence of samples using the samples of the ABs. In this way, the reconstructed signal, though having possibly different samples with respect to those of the original one, is made of bunches having same area and CoM of those of the original signal, and with minimum error between each original and reconstructed sample (Fig. 2).

4 Simulation results

In this section, simulation results related to the described compression algorithms are reported. The bit rate after compression is estimated by evaluating the entropy of the quantities to be coded. Therefore, the sensibility of the cluster finding and tracking algorithms to the error introduced by the lossy model is measured by comparing performances of these algorithms on some simulated data and on the same compressed-decompressed data.

[3] Actually, only the decimal part of the CoM is considered. The integral part, in fact, does not carry any information about the sample values, i.e. the shape of the bunch, but refers only to the global shift of the bunch itself.

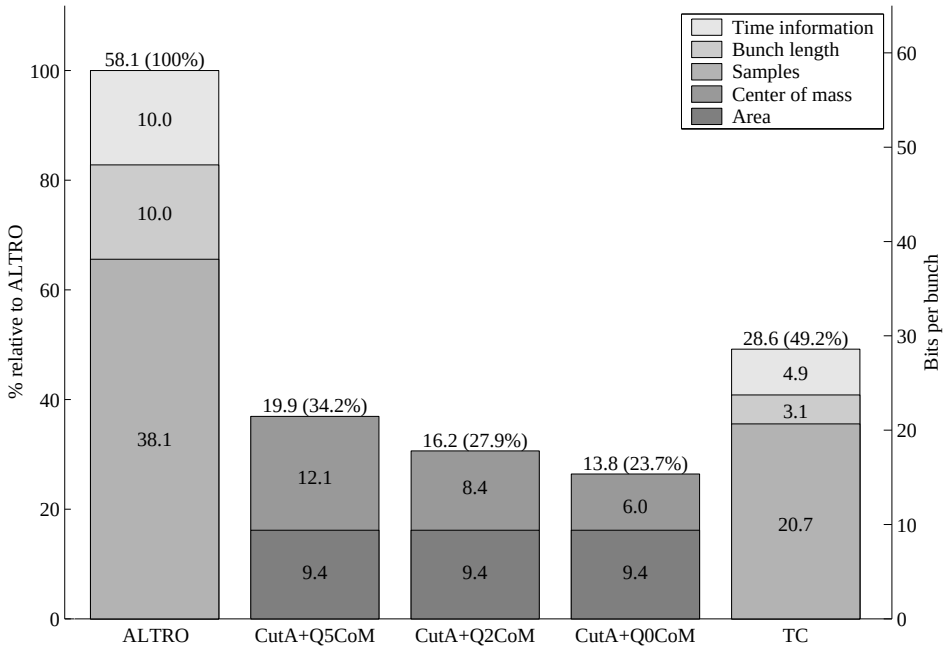

Fig. 3. Compression performance. ALTRO: original ALTRO data; CutA+Q5CoM: lossy coding with CoM quantization at $T_s/32$; CutA+Q2CoM: the same with CoM quantization at $T_s/4$; CutA+Q0CoM: the same, with T_s. For a comparison, the results of the best lossless compression technique (TC) described in [11] is also reported.
Numbers in the columns represent the number of bits per bunch dedicated to each field; numbers on top of the columns represent the overall number of bits per bunch and, in parenthesis, the size with respect to the original ALTRO data format.

4.1 Estimation of compression factors using entropy measures

In Fig. 3 the results for the lossy technique, with several quantization levels, are compared with the ALTRO data format and the best lossless technique presented in [11]. It may be noticed that with the quantization level equal to $T_s/4$ the volume of data is reduced to 27.9% of its original size; the performances for the resolutions $T_s/32$ and T_s are reported too and are 34.2% and 23.7% respectively.

4.2 Effects of the reconstruction error on cluster finding

As mentioned above, in order to measure the effect of the error due to the compression and decompression process to the calculations performed by the off-line cluster finding and tracking algorithms, the most effective way to proceed is to directly apply these algorithms both on a set of data generated by a Monte Carlo simulation and on its compressed-decompressed version. This

is done with CoM resolutions corresponding to $T_s/32$ and $T_s/4$ quantization intervals. Results are shown in Table 1. It might be noted that, independently from the version of the cluster finding algorithm, the loss in tracking efficiency is on the order of 2%, a slight loss if we consider the increase in number of observation that can be stored thanks to the improvement in compression ratio obtained by the lossy model with respect to a lossless one. The main source of the slight decrease in efficiency is the smoothing of the overlapped clusters.

5 Complexity of the compression algorithm

Finally to evaluate the feasibility of the real-time implementation of one of the proposed compression technique the number of operations per symbol and per second that have to be executed by the compression system has been estimated (see Table 2). Such operations mainly consist in evaluating the area and CoM values. The computations for the entropy coding step has also been taken into account; for this, the arithmetic coder presented in [13] has been adopted.

The number of operations per second is evaluated by assuming the worst case, in which one compression system will have to process up to 4 000 channels and 0.28 GSymbols per second. The frequency distribution tables will need 4 kBytes of memory and operations will be performed in 32 bit precision arithmetic.

The numbers obtained show that the described compression system can be easily implemented in real-time, either on DSPs, field programmable gate arrays, or ASICs. It may be worth noticing that even lower operation counts can be obtained by using more sophisticated arithmetic coders, such as those presented in [14,15], which do not need multiplications nor divisions, but only additions and shifts, or Huffman coders as in [16,17].

6 Conclusions

A lossy compression approach for the data generated by the TPC chamber in the ALICE experiment has been investigated. The main idea is to preserve the two quantities that, at a pad level, are most related to the particle energy and position which are of interest for the experiment; these are the area and the center of mass of bunches that are coded without loss by the proposed compression system. This approach achieves a reduction of the data rate to 27.9% accepting a quantization noise on the CoM position and errors on the

sample by sample values, that in principle do not affect the results on physical quantities of interest for the experiment.

The compression algorithm can be implemented using an arithmetic coder; the overall computational complexity turns out to be reasonable, so that a real-time implementation of the system on off-the-shelf electronic devices or on simple ASICs is feasible.

References

[1] ALICE Collaboration, ALICE Technical Design Report of the Time Projection Chamber, Tech. Rep. CERN/LHCC/2000–001, ALICE TDR 7, CERN, Geneva, Switzerland, http://alice.web.cern.ch/Alice/TDR/TPC/alice_tpc.pdf (January 2000).

[2] ALICE – A Large Ion Collider Experiment, http://alice.web.cern.ch/Alice.

[3] L. Musa, Electronics status, Presentation at the ALICE Week (28 May – 1 June 2001).

[4] P. V. Vyvre, Application of the data compression in the ALICE Readout and Data Acquisition System, ALICE DAQ Internal Note, CERN–EP (October 1999).

[5] J. Rissanen, G. Langdon, Universal modeling and coding, IEEE Transactions on Information Theory 27 (1) (1981) 12–23.

[6] A. N. Kolmogorov, Three approches to the quantitative definition of information, Problems of Information Transmission 1 (1) (1965) 1–11.

[7] S. Battista, F. Casalino, C. Lande, MPEG-4: a multimedia standard for the third millennium. Part 1, IEEE Multimedia 6 (4) (1999) 74–83.

[8] S. Battista, F. Casalino, C. Lande, MPEG-4: a multimedia standard for the third millennium. Part 2, IEEE Multimedia 7 (1) (2000) 76–84.

[9] L. Chiariglione, The development of an integrated audio-visual coding standard: MPEG, Proceedings of the IEEE 83 (2) (1995) 151–157.

[10] L. Chiariglione, Impact of MPEG standards on multimedia industry, Proceedings of the IEEE 86 (6) (1998) 1222–1227.

[11] A. Nicolaucig, M. Mattavelli, S. Carrato, Compression of TPC data in the ALICE experiment, Nuclear Instruments and Methods in Physics Research, A 487 (3) (2002) 542–556.

[12] L. Musa, R. E. Bosch, Alice TPC Readout chip, Internal note, CERN–EP/ED (November 2000).

[13] I. H. Witten, R. M. Neal, J. G. Cleary, Arithmetic coding for data compression, Communications of the ACM 30 (6) (1987) 520–540.

[14] J. Rissanen, K. M. Mohiuddin, A multiplication-free multialphabet arithmetic code, IEEE Transactions on Communications 37 (2) (1989) 93–98.

[15] A. Moffat, R. Neal, I. H. Witten, Arithmetic coding revisited, ACM Transactions on Information Systems 16 (3) (1998) 256–294.

[16] A. Moffat, A. Turpin, On the implementation of minimum redundancy prefix codes, IEEE Transactions on Communications 45 (10) (1997) 1200–1207.

[17] A. Moffat, J. Katajainen, In-place calculation of minimum-redundancy codes, in: 4th Workshop on Algorithms and Data Structures, Vol. 955, Springer-Verlag, Kingston, Canada, 1995, pp. 393–402, http://cs.mu.oz.au/~alistair/inplace.c.

List of Figures

Clustering algorithm version	Compression resolution	Tracking efficiency	ϕ resolution [mrad]	θ resolution [mrad]	$\Delta p/p$
v.1	No compression	91.29%	2.81	1.42	2.23%
	Compression 1/32	89.26%	2.71	1.46	2.19%
	Compression 1/4	89.18%	2.78	1.48	2.23%
v.2	No compression	89.07%	2.33	1.33	2.12%
	Compression 1/32	87.36%	2.42	1.57	2.15%
	Compression 1/4	86.98%	2.46	1.58	2.15%

Table 1
Effects of compression on cluster finding and tracking.

	sums	multiplications	divisions	jumps	memory accesses
Area and CoM (op/symb.)	1.9	0.27	0.21	0.48	0
Arith. coding (op/symb.)	69	8.2	0.85	11	1.7
Total (op/symb.)	71	8.5	1.1	12	1.7
Area and CoM (op/s)	516 M	75 M	58 M	133 M	0
Arith. coding (op/s)	19 G	2.2 G	0.23 G	3.1 G	0.47 G
Total (op/s)	20 G	2.3 G	0.29 G	3.2 G	0.47 G

Table 2
Evaluation of the complexity of the algorithm for lossy compression. The number of operations per second refers to the worst case processing of up to 4 000 channels and 0.28 GByte/s.