

# Efficient Algorithms of Multidimensional $\gamma$ -Ray Spectra Compression Using Address Randomizing Transforms

---

## Miroslav Morháč

*Institute of Physics, Slovak Academy of Sciences  
Dúbravská cesta 9, Bratislava, 845 11, Slovak Republic  
E-mail: Miroslav.Morhac@savba.sk*

## Vladislav Matoušek<sup>1</sup>

*Institute of Physics, Slovak Academy of Sciences  
Dúbravská cesta 9, Bratislava, 845 11, Slovak Republic  
E-mail: Vladislav.Matousek@savba.sk*

## Ivan Turzo

*Institute of Physics, Slovak Academy of Sciences  
Dúbravská cesta 9, Bratislava, 845 11, Slovak Republic  
E-mail: Ivan.Turzo@savba.sk*

In the paper we present efficient algorithm to compress multidimensional symmetrical  $\gamma$ -ray events. The compression of data volume is achieved due to both the symmetry of the  $\gamma$ -ray spectra and compression capabilities of the employed randomizing transform. The algorithm of compression is very fast and can be implemented for on-line compression of events. Acquired compressed data can be later processed in an interactive way.

*XI International Workshop on Advanced Computing and Analysis Techniques in Physics Research  
Amsterdam, the Netherlands  
23-27 April, 2007*

---

<sup>1</sup> Speaker

## 1. Introduction

The measurements of data in nuclear physics experiments are oriented towards gathering a large amount of data. Nuclear physics is one of the scientific fields where the requirements to higher dimensions and better resolutions are continuously increasing. In spite of unceasing growth of today's computer memory capacities the requirements for storing multidimensional data very frequently go beyond the available storage media volumes. Multiparameter data taken from nuclear experiments can be collected either in the form of list of events as they come from the source or analyzed and stored as multidimensional histograms-spectra.

To store data in the form of events has its disadvantage in the enormous amount of the information. This results in long time needed to process all data. To handle the data interactively it is desirable to work with multidimensional histograms. However, then the storage requirements for several parameters are enormous. The three-dimensional spectrum (cube) of  $\gamma$ - $\gamma$ - $\gamma$  coincidences with the resolution of 14 bits (16384 channels) from the GAMMASPHERE can serve as a good example. If we assume that the content of each channel is represented by two bytes, its storage in a conventional way would require 8 TB memory. To fit the capacity of computer memory it is needed to compress the data to the size of available memory while preserving as much information as possible.

Successful data compression techniques must satisfy two conflicting criteria. During the coding phase of the compression, the data are transformed from their native format into a format that requires less storage. At the same time, this transformation must preserve the information so that the difference between the original and decoded data arrays is not significant.

Data from nuclear experiments are received as a train of events. We can analyze and compress each event separately. Thus, the multidimensional amplitude analysis can be performed together with compression, event by event in on-line acquisition mode. Data reduction is carried out during the acquisition. After the experiment, one can extract any slice from the transformed and compressed multidimensional data.

In a typical experiment with spectrometers like GAMMASPHERE [1], EUROBALL [2], each coincidence event consists of a set of  $n$  integers ( $e_1, e_2, \dots, e_n$ ), which are proportional to the energies of the coincident  $\gamma$ -rays. Such a coincidence specifies a point in an  $n$ -dimensional hypercube. For most high-spin experiments carried out with the GAMMASPHERE, the optimal fold has been found to be four [3]. Proposed spectrometers with higher efficiencies, such as Greta [4], can attain their maximum sensitivity at folds of seven or eight. Consequently, the employment of compression tools for the data from these spectrometers is becoming urgent.

## 2. Symmetrical $\gamma$ -ray spectra

The simplest way of reducing the huge amount of data is to employ special properties of data. For instance, in the multidimensional  $\gamma$ -ray spectra from the GAMMASPHERE one can utilize the property of symmetry of the data. It holds

- for two-dimensional spectra

$$E(\gamma_1, \gamma_2) = E(\gamma_2, \gamma_1),$$

- for three-dimensional spectra

$$\begin{aligned} E(\gamma_1, \gamma_2, \gamma_3) &= E(\gamma_1, \gamma_3, \gamma_2) = E(\gamma_2, \gamma_1, \gamma_3) = \\ &= E(\gamma_2, \gamma_3, \gamma_1) = E(\gamma_3, \gamma_1, \gamma_2) = E(\gamma_3, \gamma_2, \gamma_1). \end{aligned}$$

This fact allows one to decrease the size of memory for two-dimensional spectra by factor of 2 and for three-dimensional spectra by factor of 6. In general, in the  $n$ -dimensional case, by utilizing the property of symmetry, the size of memory can be decreased by factor of  $n!$  without loss of information.

To generalize the utilization of the property of symmetry for the reduction of the memory space let us start with two-dimensional data. Let us illustrate the data of this type for the resolution  $R = 4$  in Fig 1.

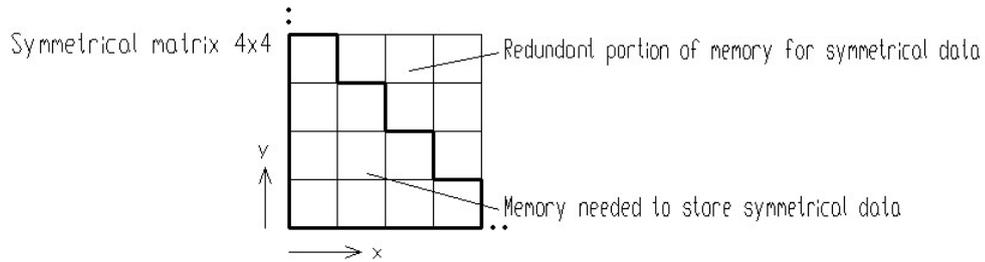


Fig. 1 Principle of storage of two-dimensional symmetrical data.

The size of reduced space according to Fig. 1 can be simply expressed

$$N_2 = \sum_{i=1}^R i = \frac{R^2 + R}{2}. \quad (1)$$

By composition of triangles of the sizes  $R, R - 1, \dots, 2, 1$ , i.e., by putting them on each other, according to Fig. 2 we get the geometrical shape called tetrahedron. Fig. 2 and 3 illustrate the example of such a composition for  $R = 4$ .

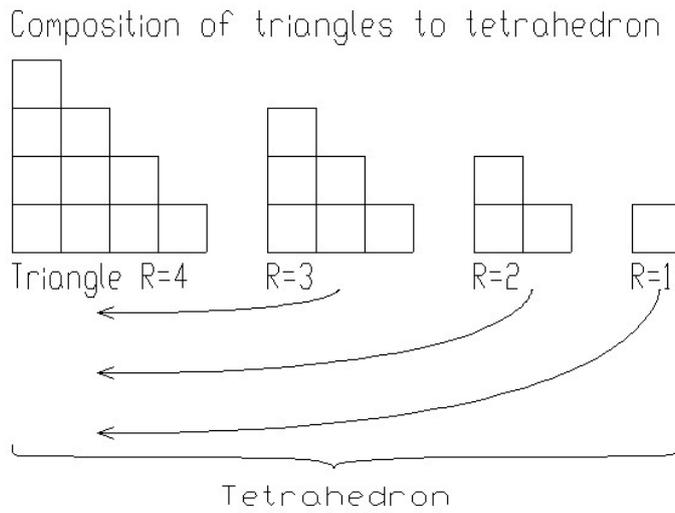


Fig. 2 Principle of composition of two-dimensional triangles to tetrahedron.

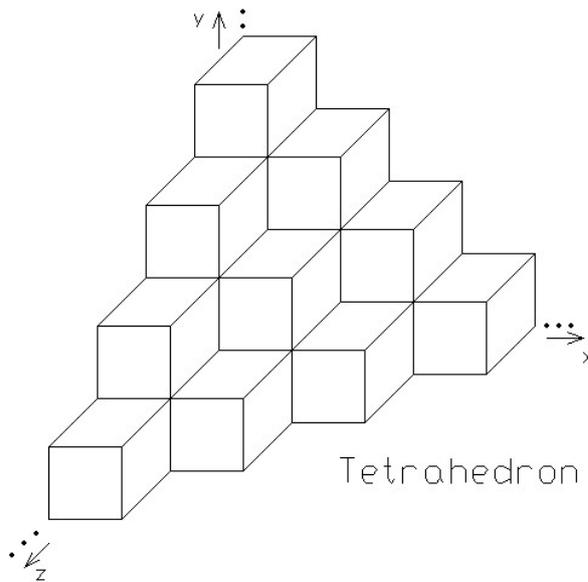


Fig. 3 An example of storage three-dimensional symmetrical data in the form of tetrahedron.

The size (volume) of the reduced space of the tetrahedron, utilizing (1), is

$$N_3 = \sum_{i=1}^R \frac{i^2 + i}{2} = \frac{R^3 + 3R^2 + 2R}{6}. \tag{2}$$

Let us go on with the generalization to four-dimensional data. Again by compositions of the tetrahedrons according to Fig. 4 we obtain hyperhedron of the 4-th order for  $R = 4$ .

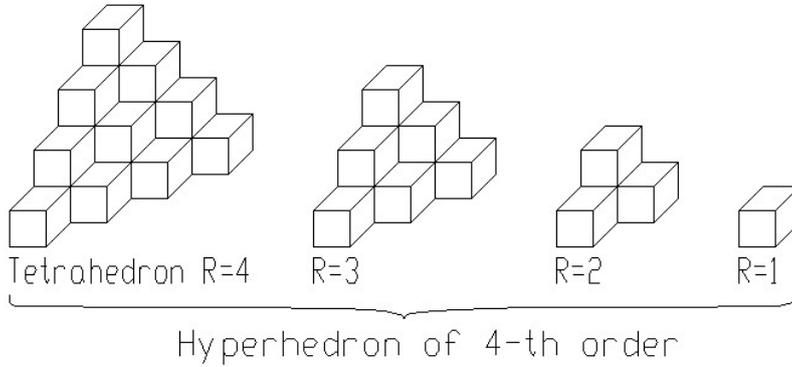


Fig. 4 Principle of composition of three-dimensional tetrahedrons to hyperhedron of 4-th order

The volume of the hyperhedron of 4-th order can be, again utilizing (1), (2), expressed as

$$N_4 = \sum_{i=1}^R \frac{i^3 + 3i^2 + 2i}{6} = \frac{R^4 + 6R^3 + 11R^2 + 6R}{24}. \quad (3)$$

The principle of compositions of hyperhedrons can be directly extended to higher dimensions. Then analogously for the volume of the hyperhedron of 5-th order one can write

$$N_5 = \sum_{i=1}^R \frac{i^4 + 6i^3 + 11i^2 + 6i}{24} = \frac{R^5 + 10R^4 + 35R^3 + 50R^2 + 24R}{120}. \quad (4)$$

Let us denote

$$S_k = \sum_{i=1}^R i^k.$$

Then utilizing the recursive formula

$$S_k = \frac{(R+1)^{k+1} - \left[ \binom{k+1}{2} S_{k-1} + \dots + \binom{k+1}{k} S_1 + \binom{k+1}{k+1} S_0 + 1 \right]}{k+1},$$

one can derive even the formulas for the volumes of the hyperhedrons of higher orders. For example for the volume of the hyperhedron of the 6-th order one gets

$$N_6 = \frac{R^6 + 15R^5 + 85R^4 + 225R^3 + 274R^2 + 120R}{720}.$$

Assuming that for each element of the data array we need 4 bytes (e.g. float format) after the reduction according (1), (2), (3) and (4) for the resolution  $R = 16384$  (14 bits ADC, e.g. data from the GAMMASPHERE) we get

$$\begin{aligned}
 4N_2 &= 5.369 \cdot 10^8 \text{ Bytes} \approx 536.9 \text{ MB} \\
 4N_3 &= 2.932 \cdot 10^{12} \text{ Bytes} \approx 2.93 \text{ TB} \\
 4N_4 &= 1.201 \cdot 10^{16} \text{ Bytes} \approx 12.01 \text{ PB} \\
 4N_5 &= 3.937 \cdot 10^{19} \text{ Bytes}
 \end{aligned} \tag{5}$$

Obviously, even if exploiting the property of symmetry of the data, these cannot be, at the present status of technical facilities, stored without employing additional compression method. In the well-known Radware package [5] the author combines both utilizing the property of the symmetry and binning. Apparently, the binning operation substantially decreases the resolution in the compressed spectrum. In the following part we describe the method of the compression of symmetrical data using randomizing transforms.

### 3. Pseudo-random transformation method

The advantage of conventional analyzer consists in the existence of direct correspondence between memory locations and descriptors. The conventional analyzers do not need to store descriptors and their cross-references, which is in the essence redundant information. However, for higher dimensions (3, 4, 5, ...), the size of multidimensional histograms stored in conventional analyzers is enormously large. On the other hand the number of different descriptors, i.e., the values read out from ADCs, which actually occurred during an experiment is much smaller. Therefore, the multidimensional space must be almost empty. Very large parts of locations of the multidimensional array are zeros or statistical fluctuations only. Hence this method does not provide statistically uniform distribution of the descriptors in the memory space and thus its more efficient utilization.

Alternative methods are based on maintaining the list of descriptors. Each descriptor describes fully the event, which has occurred. To preserve the correspondence between original space and compressed array the descriptor and counts are stored separately for each channel. The descriptors are passed through a pseudo-random transformation. Pseudo-random transformation suitable for nuclear spectroscopy should satisfy the following requirements:

- uniform (or quasi-uniform) distribution of descriptors over memory addresses for any shape of multidimensional spectra;
- clusters of descriptors in physical field, hypercube, must be spread over the whole range of possible addresses and adjacent descriptors must go to addresses far away from each other;
- transformation must be fast, so that it can be applied on-line for high-rate experiments.

There is an unlimited number of different methods to achieve digital transformation, but most of them do not obey the above-stated requirements. Methods are usually similar to the

generation of pseudo-random numbers (residues of modulo operations, Hamming's code technique, transformation through the division of polynomials [6-7], etc.). Through the transformation, each descriptor uniquely derives its storage address.

One of the approaches satisfying the above-stated criteria that gives pseudo-random distribution is the method based on the assignment of inverse number in the sense of modulo arithmetic to each value  $a$  (descriptor) read out from ADC

$$b = a^{-1} \pmod{M}, \quad (6)$$

where  $M$  is prime. This operation can be carried out through the use of look-up table of precomputed inverse numbers and therefore it is very fast. An example of inverse numbers in modulo arithmetic for  $M = 17$  is given in Table I. One can observe that the assignment between numbers and their inverses is uniquely unique. By definition for 0 we have taken inverse number to be 0.

$x$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$x^{-1} \pmod{M}$	0	1	9	6	13	7	3	5	15	2	12	14	10	4	11	8	16

Table I. Modulo inverse numbers for  $M = 17$ .

It would be ideal if the transformation produced the uniform distribution throughout the memory. In practice there exists a possibility of more descriptors being transformed to the same address. Because of that, we modified the method and the generated address  $b$  is used to determine position where to start searching in the bucket of the descriptors. A list of successive locations, where  $d$  is the depth of searching, are checked

- if in a location the descriptor coincides with read out descriptor, the counts in this location are incremented,
- if no descriptor coincides with read out descriptor and in a location within search depth is empty location, the descriptor is written to this location and its count is set to 1,
- if there is no empty location within search depth and no descriptor coincides with read out descriptor the event is discarded.

During the experiment, the events with higher counts (statistics) occur earlier and therefore there is a higher probability that free positions will be occupied by statistically relevant events.

Let us now illustrate the algorithm of calculation of the address of an event in the transformed area for three-dimensional events. Let us assume that from ADCs we have read out the values  $x, y, z$ . Then the algorithm is as follows:

1. arrange the coordinates so that  $x \leq y \leq z$  - symmetry removal,
2. calculate  $x_m = x \pmod{M}, y_m = y \pmod{M}, z_m = z \pmod{M}$ ,
3. calculate  $x_i = x_m^{-1} \pmod{M}, y_i = y_m^{-1} \pmod{M}, z_i = z_m^{-1} \pmod{M}$  - randomizing transform of all coordinates,

4. calculate address in the transformed area utilizing the property of symmetry (1), (2)

$$b = \frac{x_i^3 + 3x_i^2 + 2x_i}{6} + \frac{y_i^2 + y_i}{2} + z_i. \tag{7}$$

The extension of the algorithm for four-, and five-dimensional data utilizing relations (3), (4) is straightforward. For four-, and five-dimensional data we replace the relation (7) by

$$b = \frac{x_i^4 + 6x_i^3 + 11x_i^2 + 6x_i}{24} + \frac{y_i^3 + 3y_i^2 + 2y_i}{6} + \frac{z_i^2 + z_i}{2} + v_i \tag{8}$$

and

$$b = \frac{x_i^5 + 10x_i^4 + 35x_i^3 + 50x_i^2 + 24x_i}{120} + \frac{y_i^4 + 6y_i^3 + 11y_i^2 + 6y_i}{24} + \frac{z_i^3 + 3z_i^2 + 2z_i}{6} + \frac{v_i^2 + v_i}{2} + w_i, \tag{9}$$

respectively. We have denoted the forth and fifth variables by  $v_i, w_i$ . The algorithm of searching descriptor in the list of events using above presented compression method is outlined in Fig. 5.

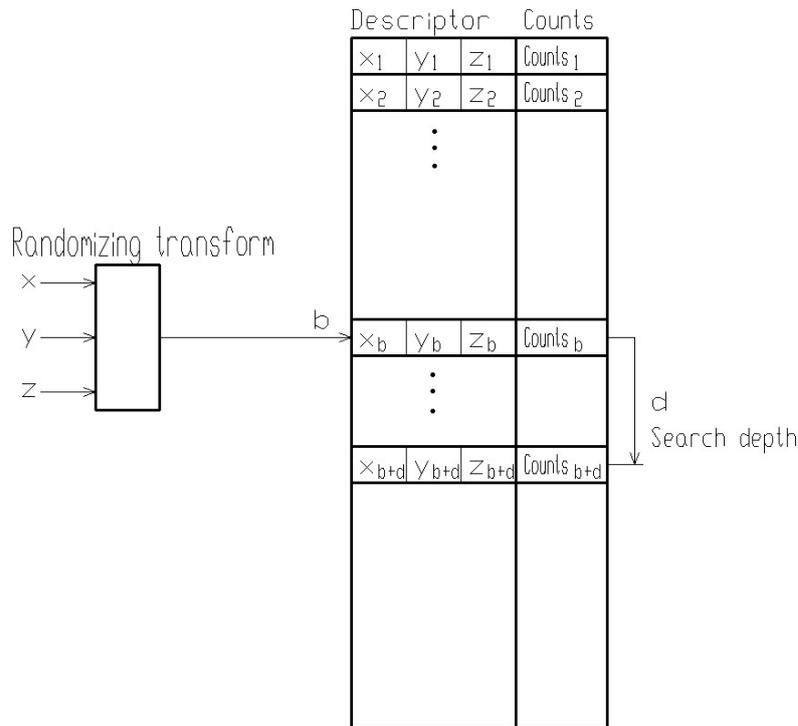


Fig. 5 Outline of the searching descriptor in the list of events using randomizing transform.

## 4. Discussion and results

### 4.1 Database of three-fold coincidence events

In this section, we propose the randomizing transformation compression method with calculation of the descriptor address based on inverse numbers in the sense of modulo arithmetic according to the algorithm described in the previous section. First, let us create the database of three-dimensional events from the list files of experimental data from the GAMMASPHERE. The descriptor of each event contains the addresses  $x, y, z$  and counts (short integers), i.e., each event takes 8 bytes. We utilize the property of symmetry of the multidimensional  $\gamma$ -ray spectra. Then according to (2) the chosen prime modulus  $M$  has to satisfy the condition

$$\frac{M^3 + 3M^2 + 2M}{6} \cdot 8 < \text{Memory}. \quad (10)$$

To fit the size of operating memory, we have chosen the prime modulus  $M = 601$ . It allows one to store 36 361 101 (290 888 808 bytes) different descriptors. The assignment between the numbers from ring  $\langle 1, 600 \rangle$  and their inverse numbers in the sense of modulo arithmetic is uniquely unique. It is illustrated in Fig. 6.

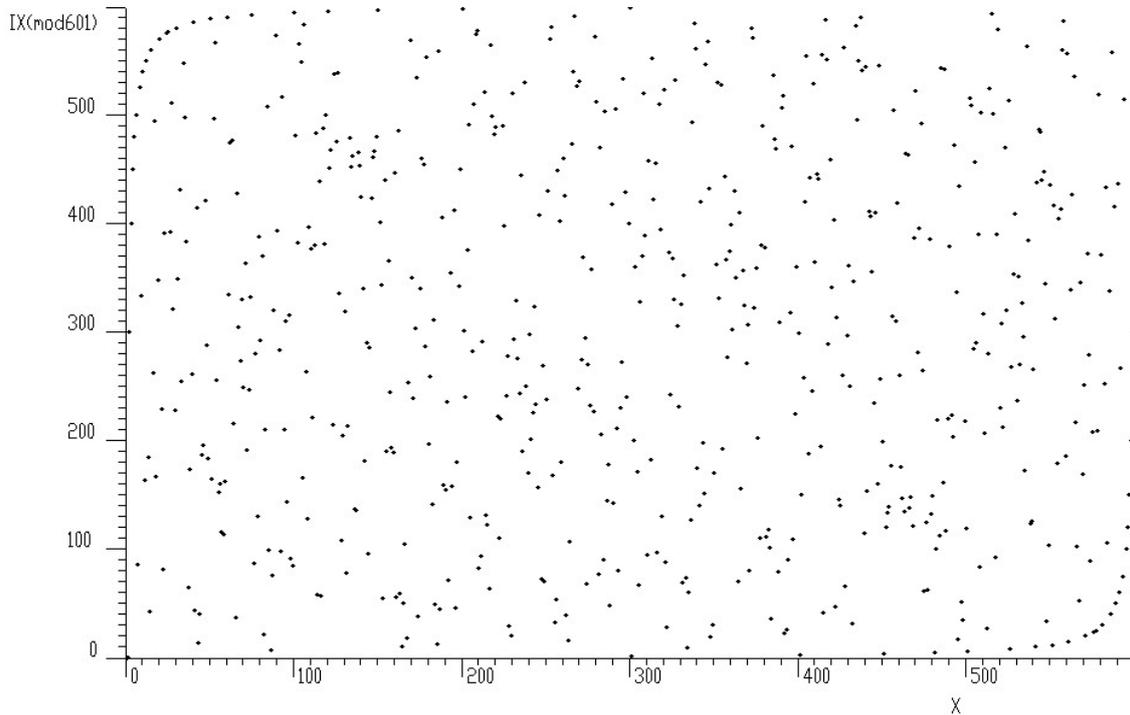


Fig. 6 Assignment between numbers from ring  $\langle 1, 600 \rangle$  and their modulo inverse numbers.

According to the requirements stated in the previous section, the descriptors should be spread over the whole range of possible addresses. In Fig. 7 we present the spectrum of distances between two adjacent modulo inverse numbers (mod 601).

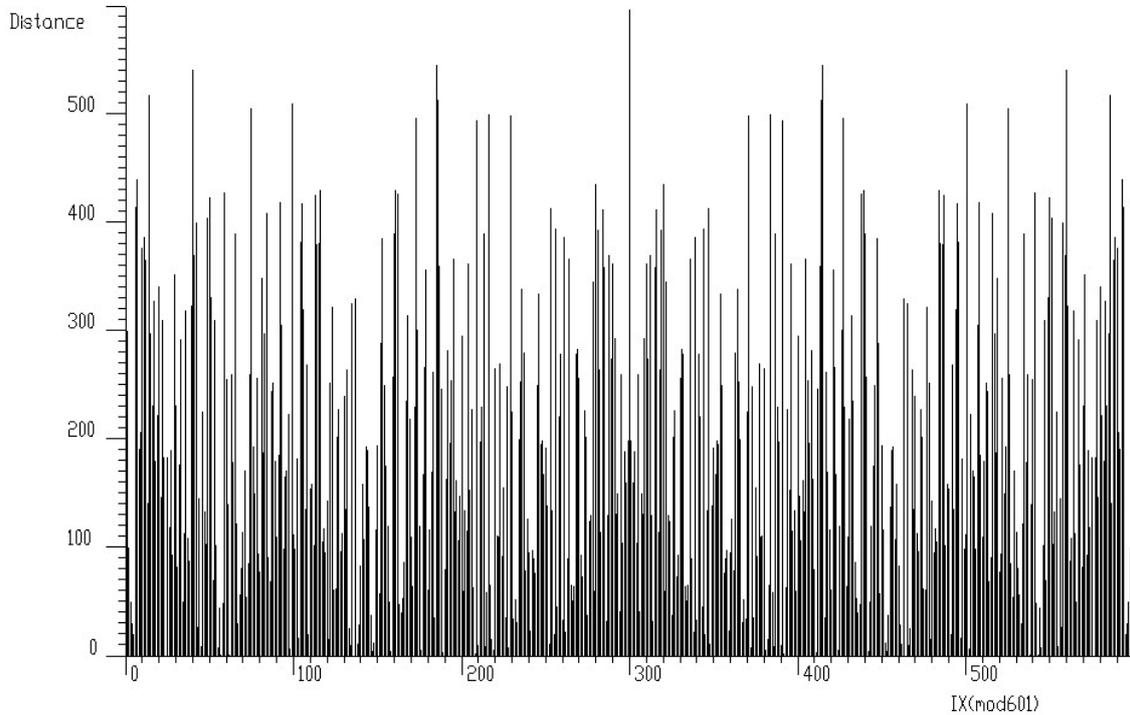


Fig. 7 Spectrum of distances between two adjacent modulo inverse numbers.

One can observe great scattering in these distances. This allows quasi-uniform distribution of descriptors in the transformed area.

The distribution of descriptors counts in the transformed domain is shown in Fig. 8. The whole linear array of descriptors (36 361 808 items) has been mapped to the 16384 channels spectrum. One can observe quasi-constant distribution, which witnesses about quasi-uniform distribution of descriptors over all memory addresses in the transform domain.

Now let us investigate and compare one-dimensional slices from non-compressed original and compressed three-dimensional arrays for the searching depth  $d = 1000$  events. The detail of both spectra in high counts region is given in Fig. 9 (thick line - original spectrum, thin line - decompressed spectrum). Some noise appears in the decompressed data, but for the first stage analysis of the experimental data this result is fully convenient.

The detail in low counts region of both data is given in Fig. 10 (thick line - original spectrum, thin line - decompressed spectrum). Again, the noise is generated in the decompressed spectrum, however the main features of the spectrum are preserved.

In Fig. 11 we have studied the influence of the searching depth on the decompressed spectra. We see that increasing the length of searching in the buffer of compressed events improves the preservation of the peak heights.

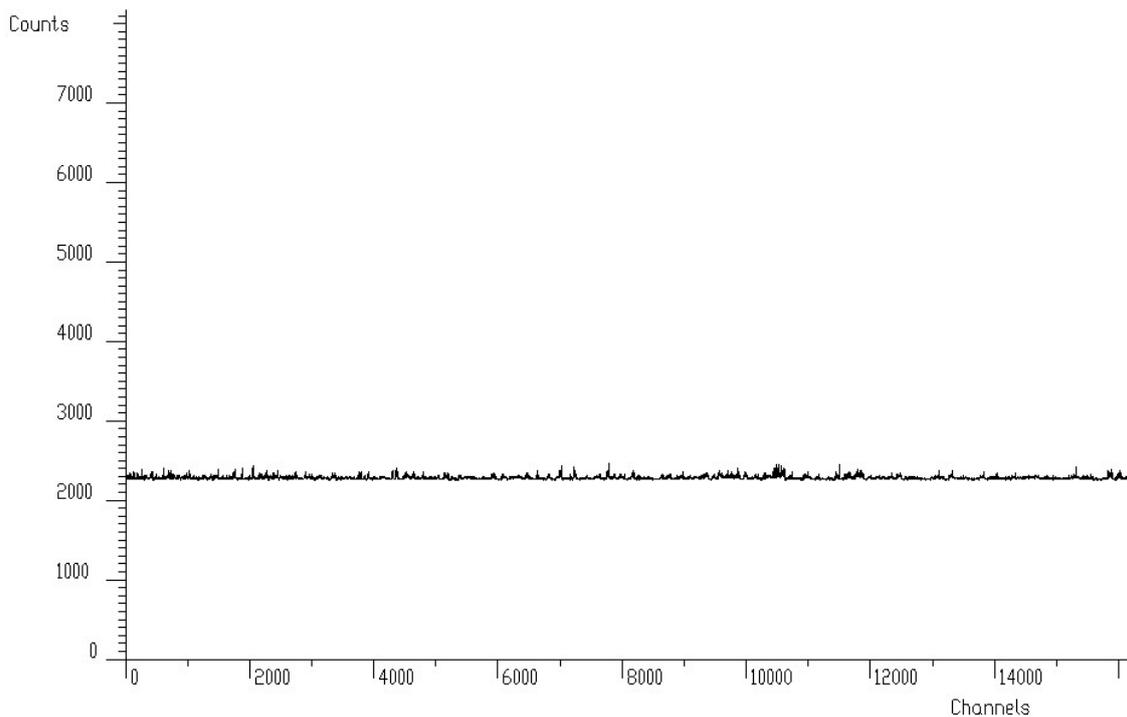


Fig. 8 Distribution of descriptor counts in the transformed domain.

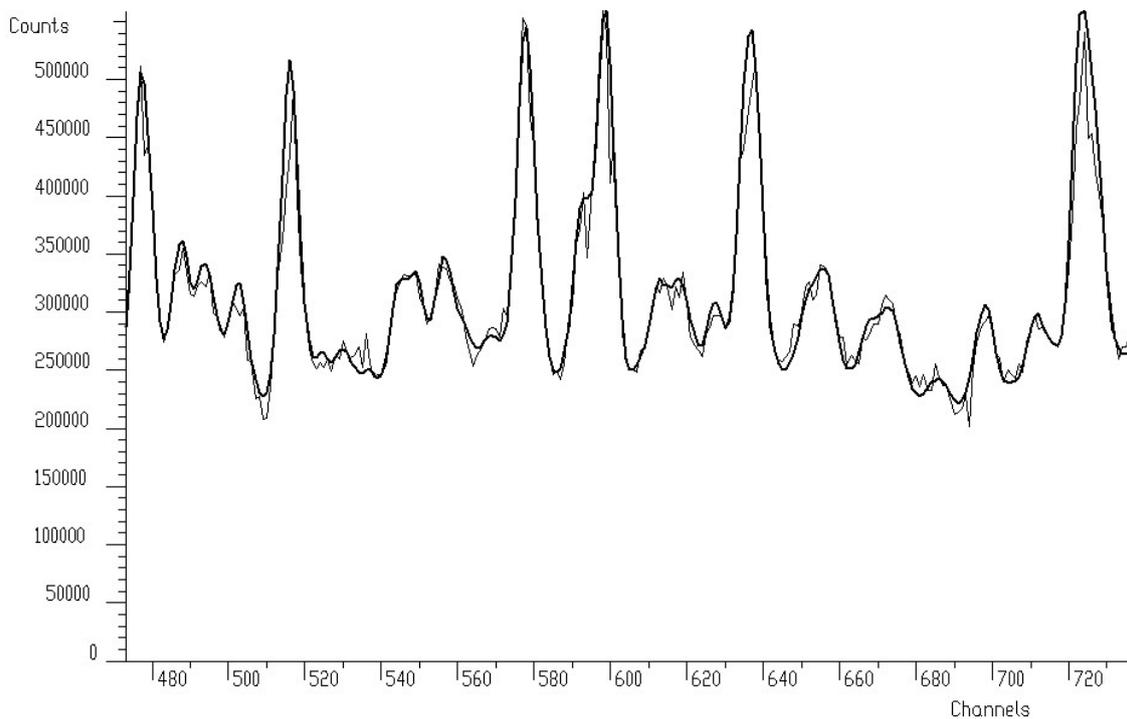


Fig. 9 High counts region of slice from the original data (thick line) and corresponding region of slice from the compressed data (thin line).

POS(ACTAT)036

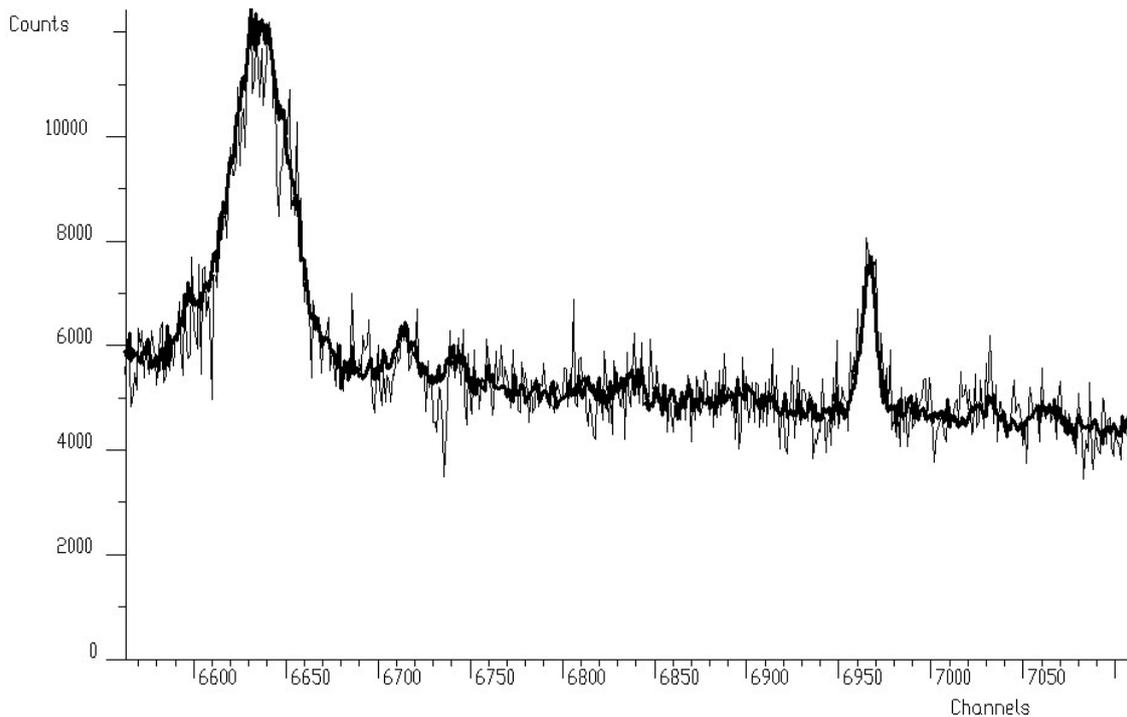


Fig. 10 Low counts region of slice from the original data (thick line) and corresponding region of slice from the compressed data (thin line).

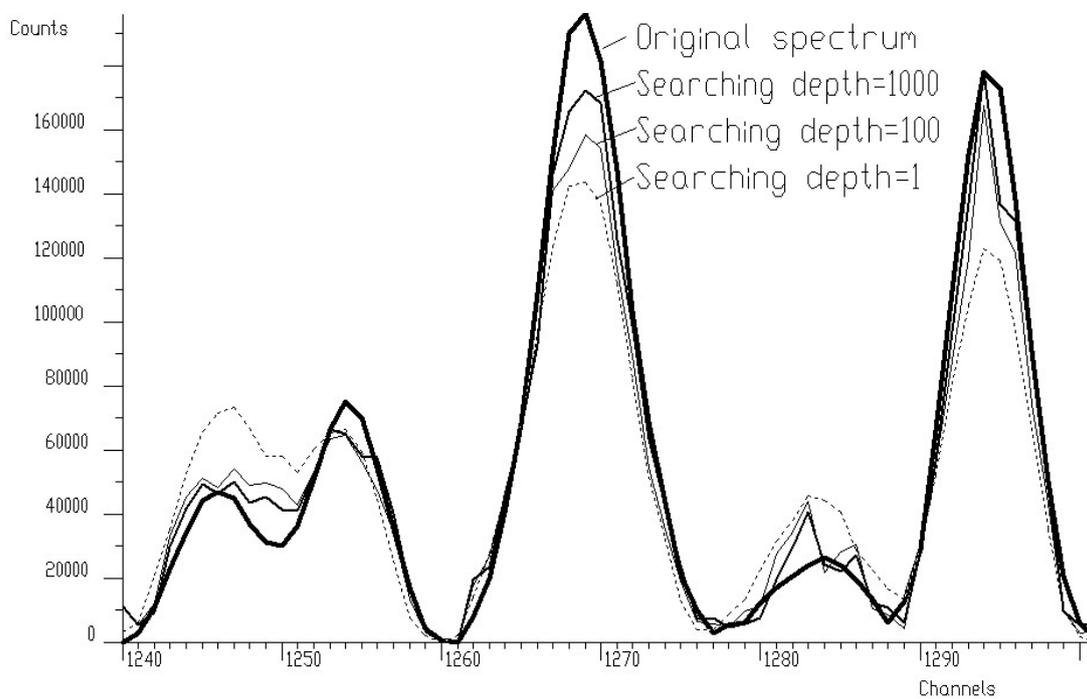


Fig. 11 The influence of searching depth on quality of the decompressed spectra.

Now let us proceed to two-dimensional slices from both original data and compressed array. The slices from the original and compressed data are shown in Fig. 12 and Fig. 13, respectively. Again, one can observe good fidelity between both data, though in the decompressed data some details are less distinct.

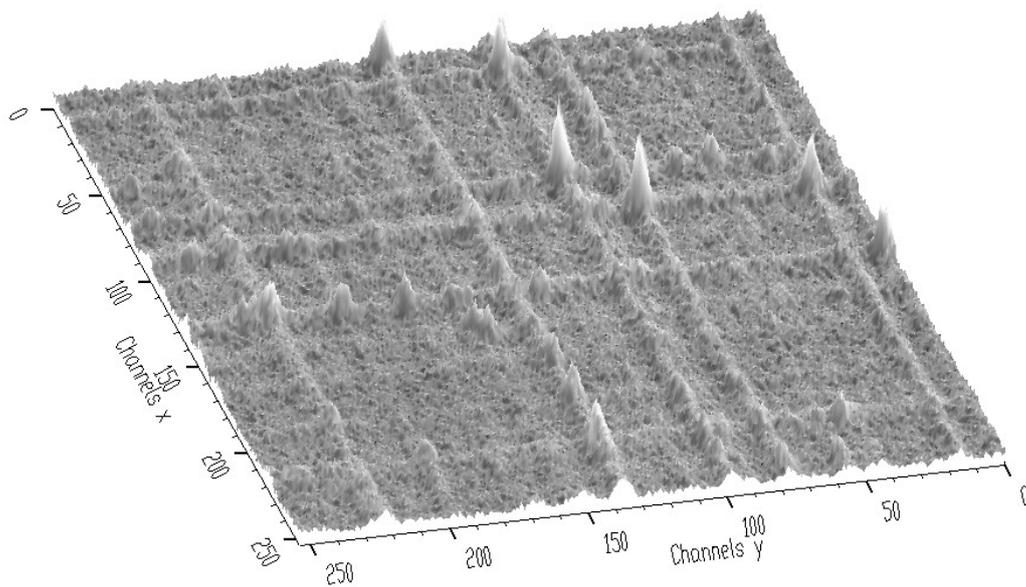


Fig. 12 Two-dimensional slice from the original three-dimensional data.

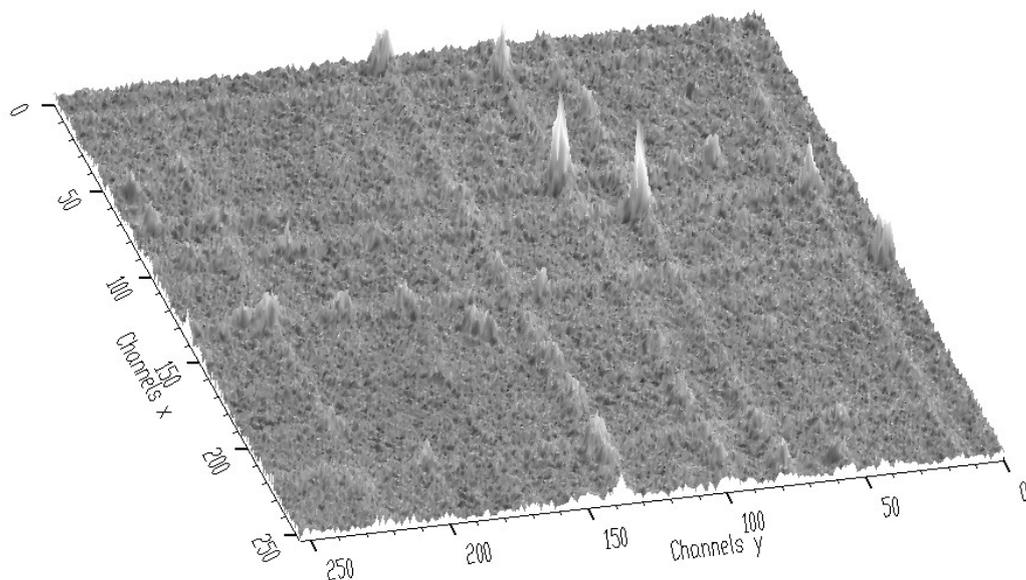


Fig. 13 Two-dimensional slice from the compressed three-dimensional data.

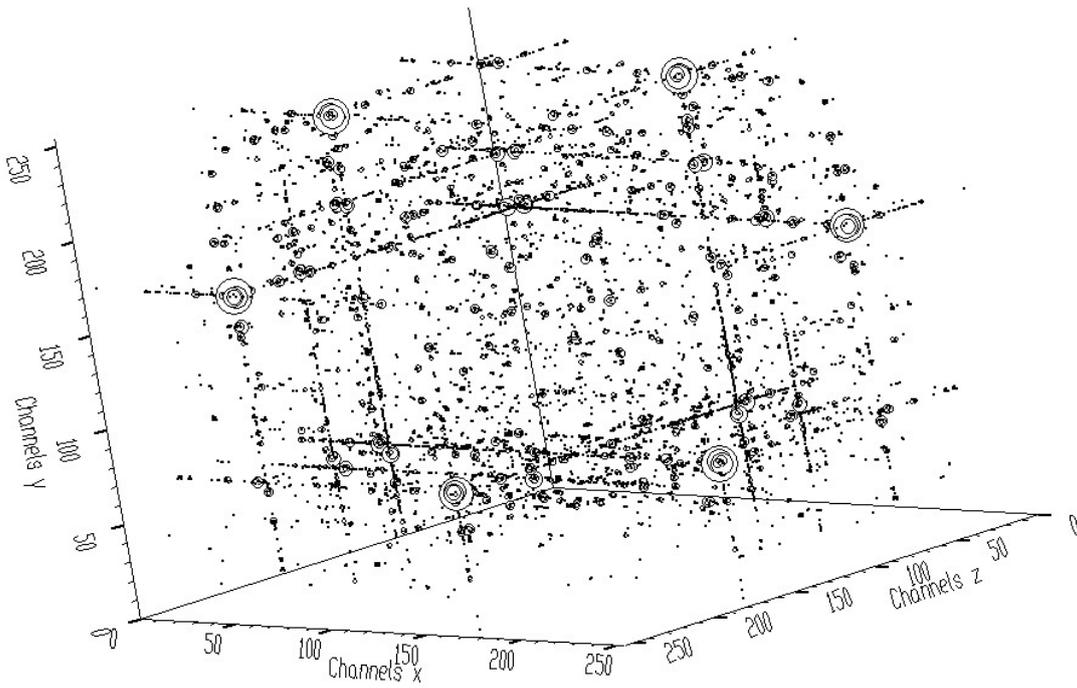


Fig. 14 Original three-dimensional data.

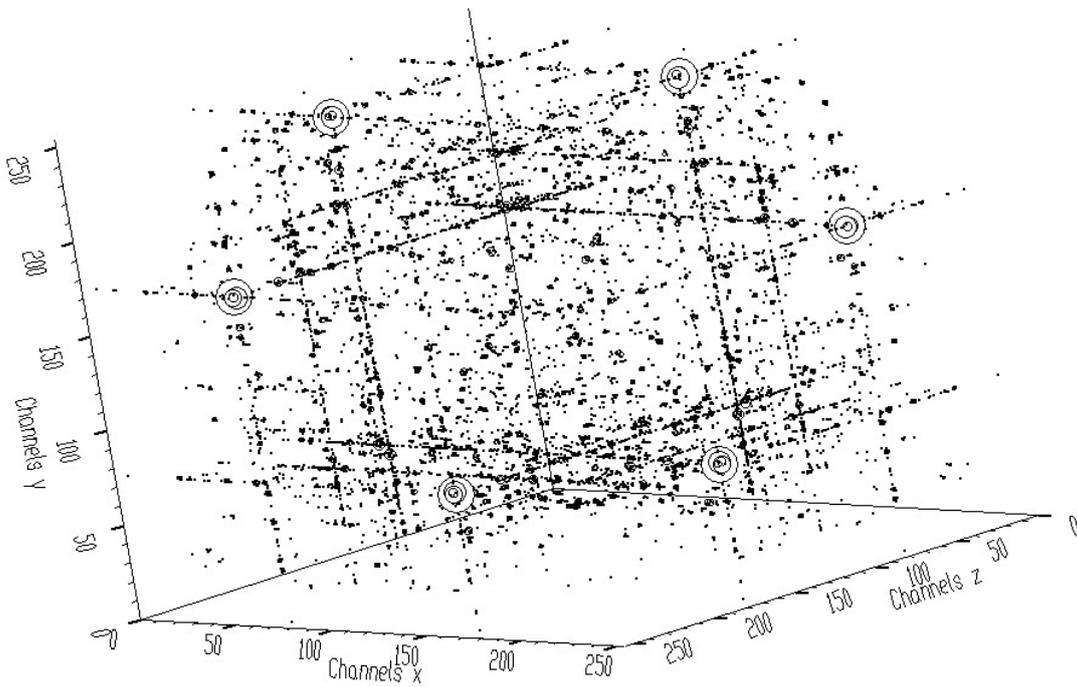


Fig. 15 Decompressed three-dimensional data.

Finally let us create three-dimensional slices from both the original and compressed events. The original data and decompressed data are shown in Fig. 14 and Fig. 15, respectively. The channels are shown as spheres. Three parameters determine the position of a sphere and its radius is proportional to counts it contains. The conclusions made in one-, and two-dimensional slices apply also for three-dimensional ones.

#### 4.2 Database of four-fold coincidence events

Next, we have created the database of four-dimensional events from the list files of experimental data from the GAMMASPHERE. Let us denote the fourth independent variable by letter  $v$ . Then the descriptor of each event contains the addresses  $x, y, z, v$  and counts (short integers), i.e., each event requires 10 bytes. Utilizing the property of symmetry of the multidimensional  $\gamma$ -ray spectra according to (3) the chosen prime modulus  $M$  has to satisfy the condition

$$\frac{M^4 + 6M^3 + 11M^2 + 6M}{24} \cdot 10 < \text{Memory}. \quad (11)$$

We have chosen the prime modulus 157. It allows to store 26 294 360 (262 943 600 bytes) different descriptors.

Analogously to three-dimensional data let us compare a part of one-dimensional slices from non-compressed original (thick line) and compressed four-dimensional (Fig. 16). One can observe very good correspondence between both data.

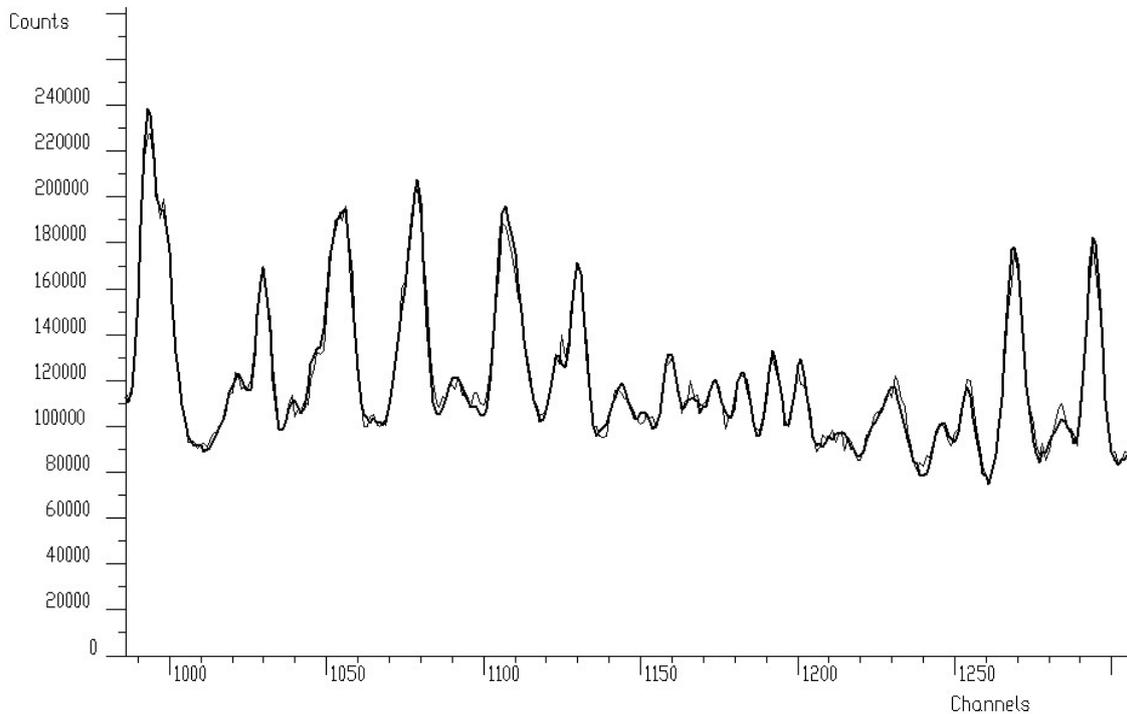


Fig. 16 A part of one-dimensional slices from original (thick line) and compressed (thin line) four-dimensional data.

Let us continue with two-dimensional slices from both original data and compressed array. The slices from original and compressed data are shown in Fig. 17 and Fig. 18, respectively. Though in the decompressed data in Fig. 18 one can observe some additional noise the overall shape of the spectrum is preserved.

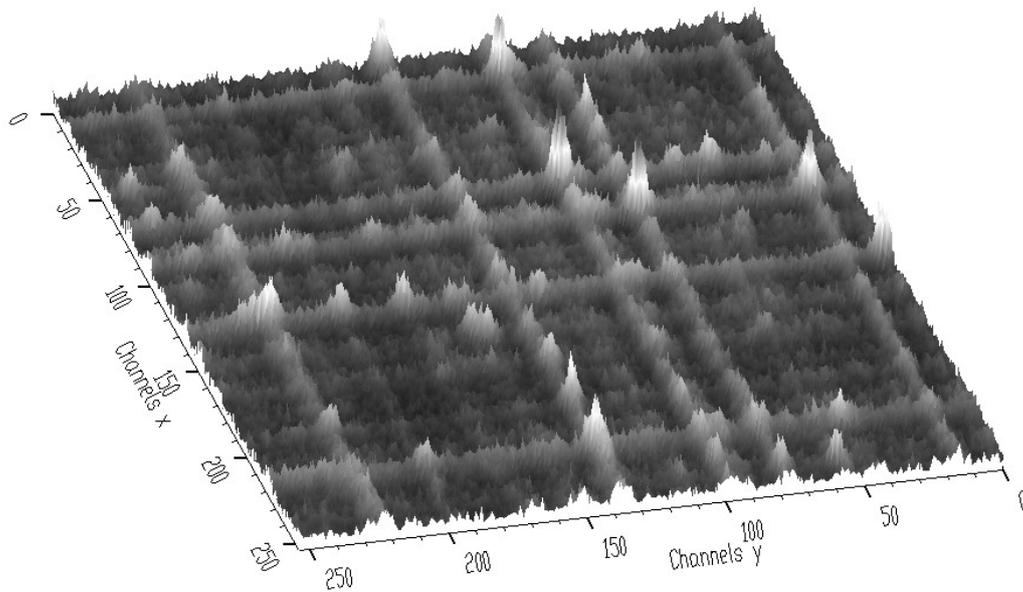


Fig. 17 Two-dimensional slice from the original four-dimensional data.

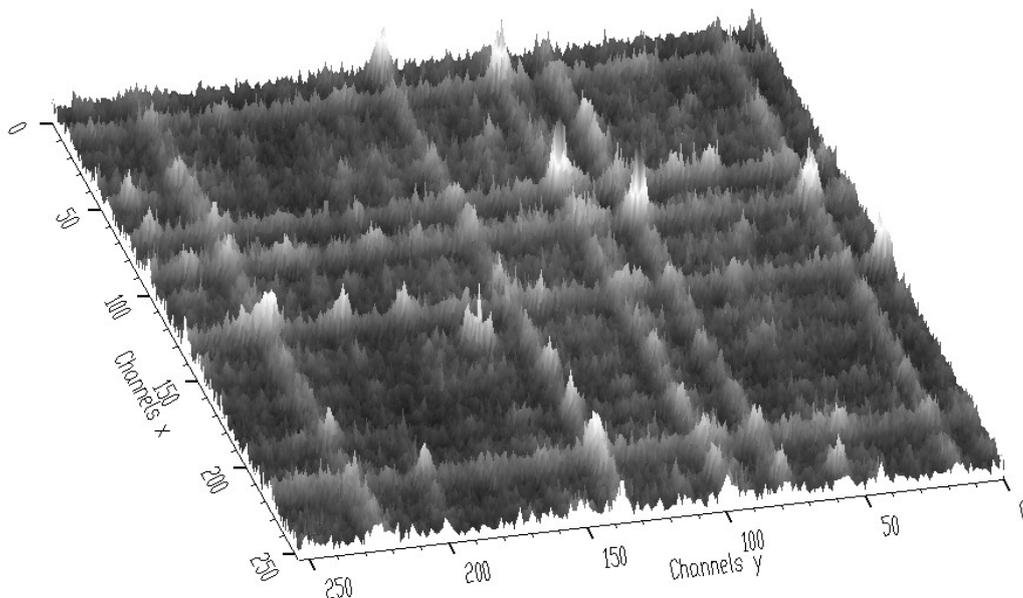


Fig. 18 Two-dimensional slice from the compressed four-dimensional data.

Next we have created three-dimensional slices from both original and compressed events. Original data and decompressed data, using the same display parameters, are shown in Fig. 19 and Fig. 20, respectively. Good correspondence between both data can be observed.

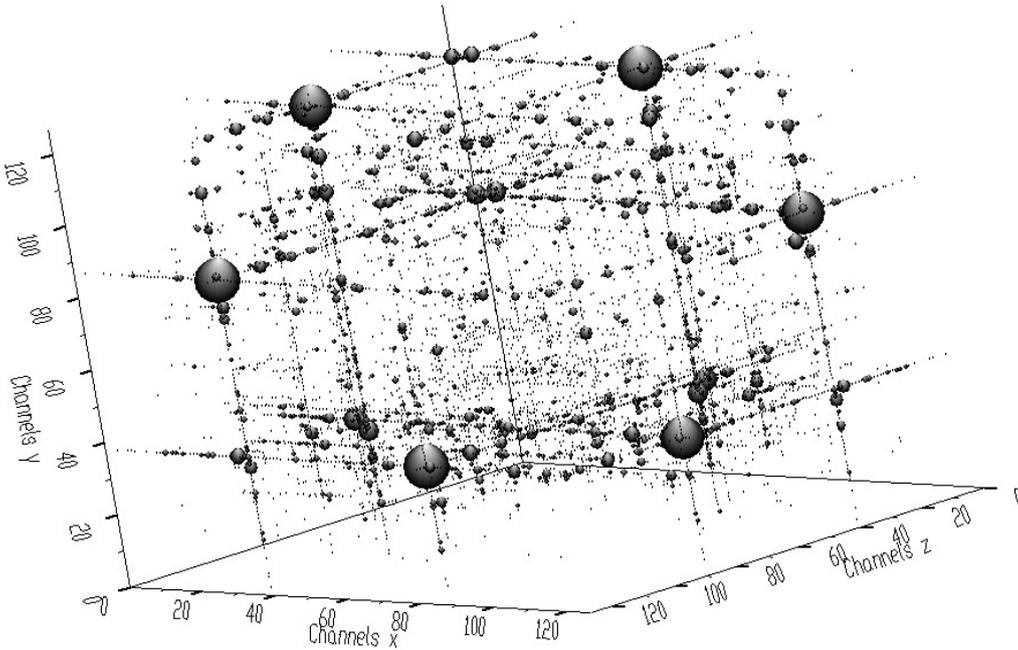


Fig. 19 Three-dimensional slice from the original four-dimensional data.

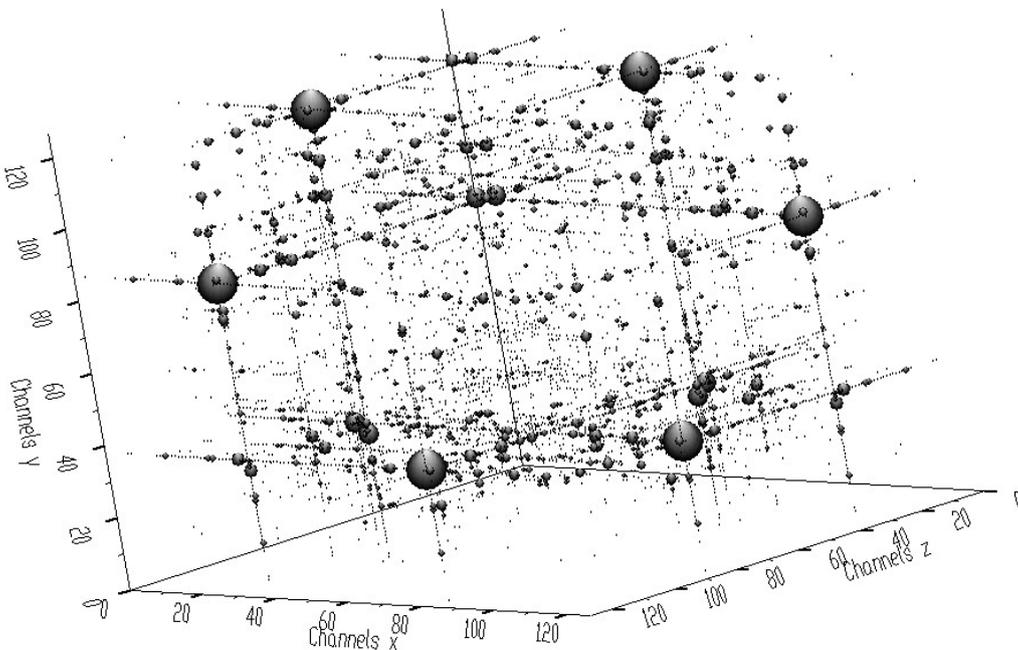


Fig. 20 Three-dimensional slice from the compressed four-dimensional data.

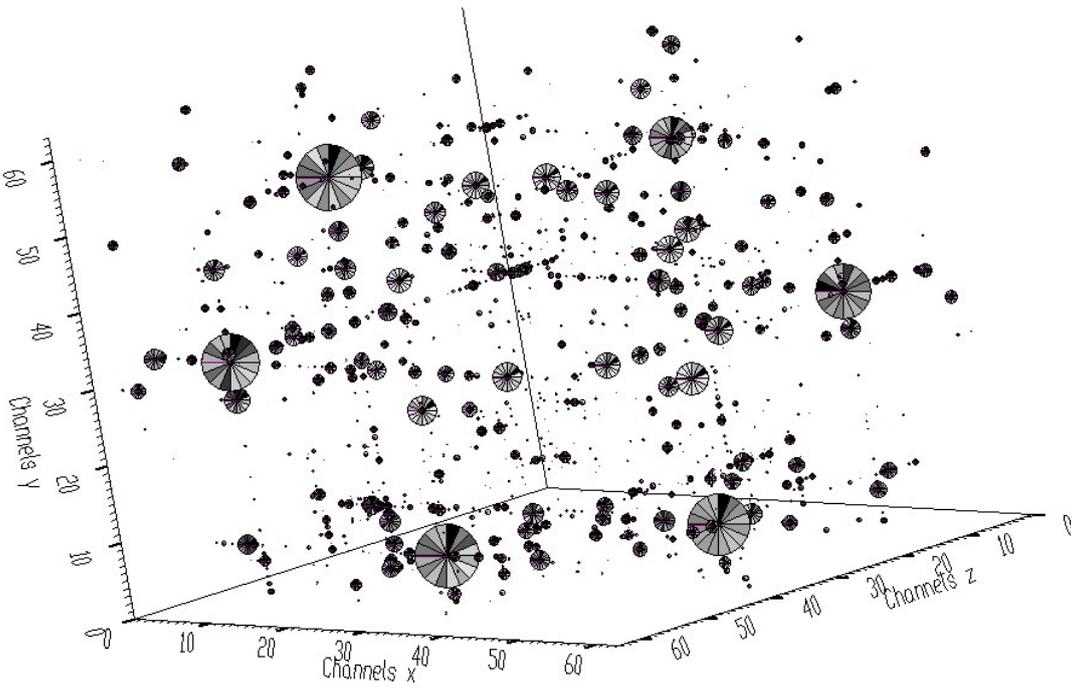


Fig. 21 Four-dimensional slice from the original four-dimensional data.

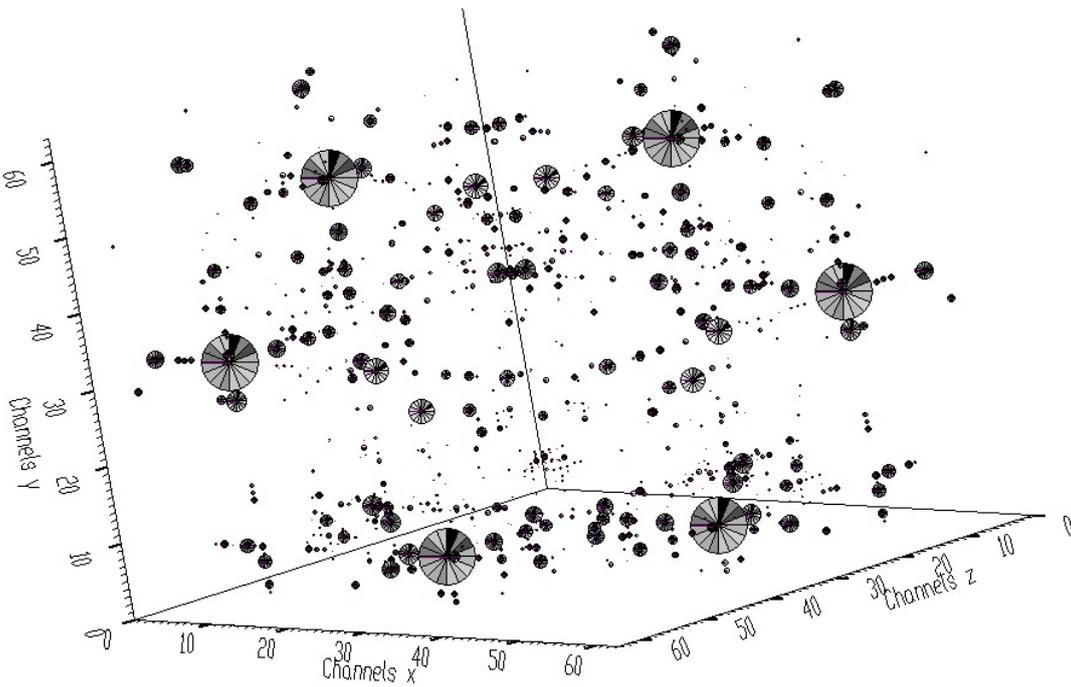


Fig. 22 Four-dimensional slice from the compressed four-dimensional data.

Finally we have created four-dimensional slices from both original and compressed events. The four-dimensional views in pies display mode (the sizes of balls are proportional to the volumes and the colors in the pies correspond to the content of channels in the fourth dimension) of the original data and decompressed data (64 channels in  $x, y, z$  dimensions and 16 channels in  $v$  dimension) are shown in Fig. 21 and Fig. 22, respectively. Big peaks correspond in both data, however in small peaks some differences can be observed.

### 4.3 Database of five-fold coincidence events

Now let us study the properties of compressed data in the database of five-dimensional events. The descriptor of each event contains the addresses  $x, y, z, v, w$  and counts (short integers), i.e., each event requires 12 bytes. Again, by employing the property of symmetry of the multidimensional  $\gamma$ -ray spectra according to (4) the chosen prime modulus  $M$  has to satisfy the condition

$$\frac{M^5 + 10M^4 + 35M^3 + 50M^2 + 24M}{120} \cdot 12 < \text{Memory} . \quad (12)$$

We have chosen the prime modulus 73. It allows to store 19 757 815 (237 093 780 bytes) different descriptors.

Analogously to three-, and four-dimensional data let us compare a part of one-dimensional slices from non-compressed original (thick line) and compressed five-dimensional (thin line) arrays (Fig. 23). Similarly, to the above-given examples one can observe very good correspondence between both data.

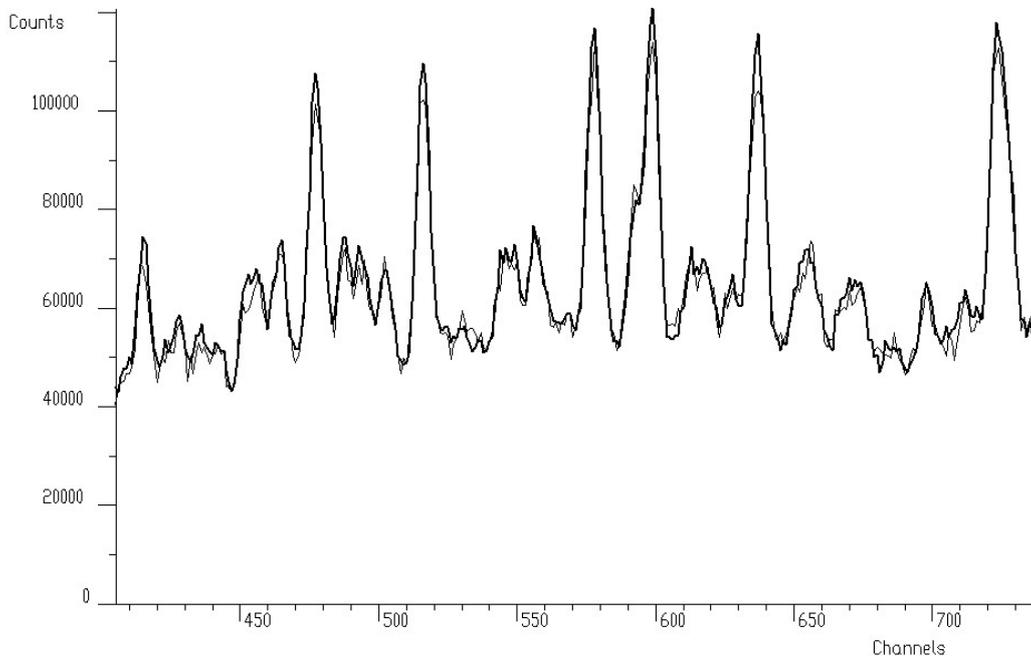


Fig. 23 A part of one-dimensional slices from original (thick line) and compressed (thin line) five-dimensional data.

Two-dimensional slices from both original data and compressed array are shown in Fig. 24 and Fig. 25, respectively. The overall shape of the spectrum is again preserved.

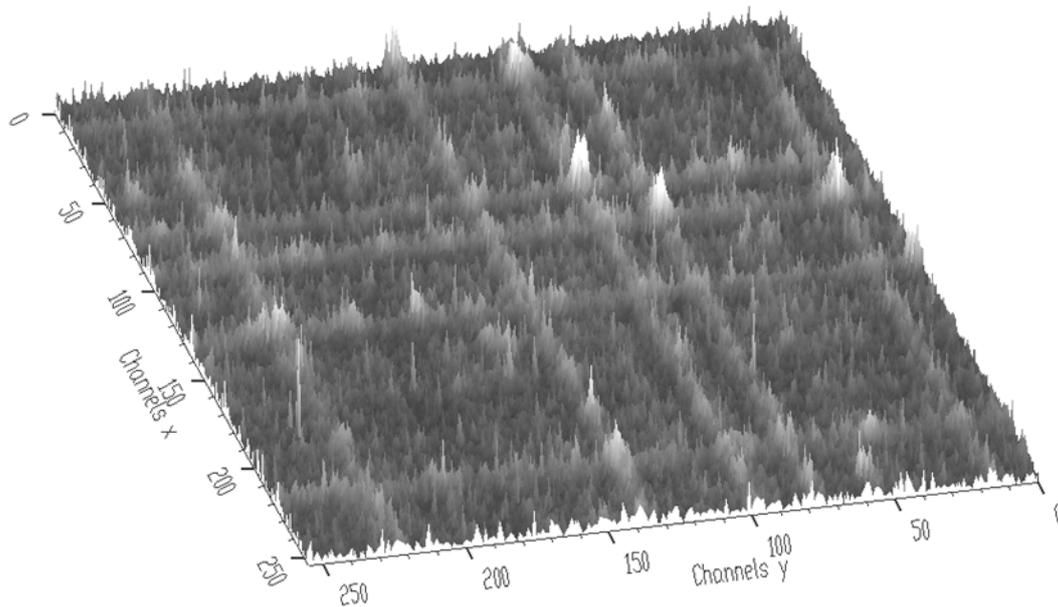


Fig. 24 Two-dimensional slice from the original five-dimensional data.

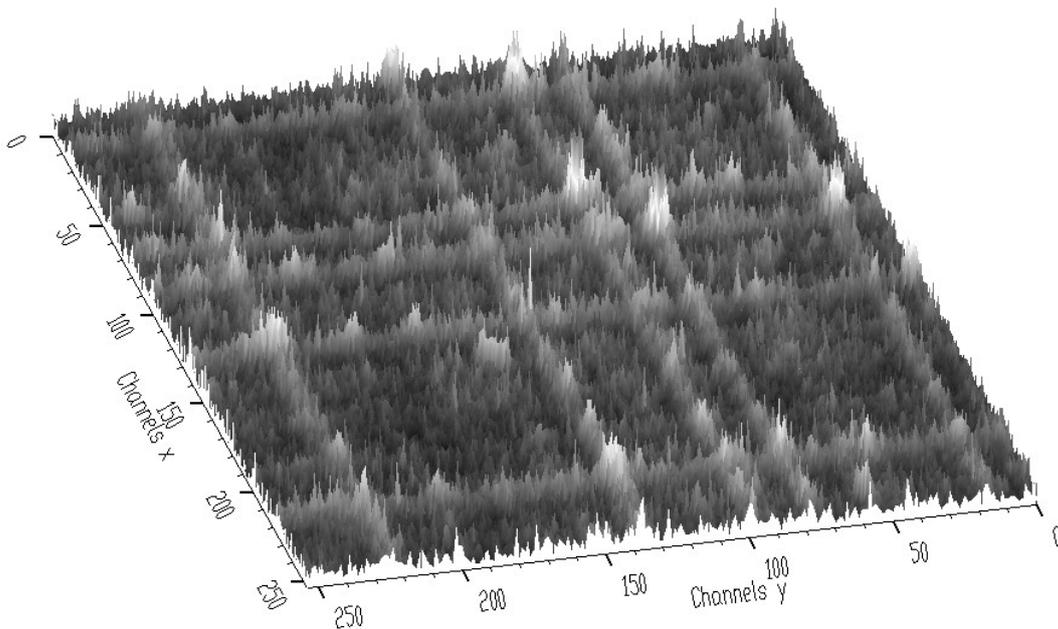


Fig. 25 Two-dimensional slice from the compressed five-dimensional data.

The results for higher order slices from the compressed five-dimensional events are similar to those above-given slices from three-, and four-dimensional compressed events.

## 5. Conclusions

The analysis of multidimensional  $\gamma$ -ray spectra is based on slices. Searching for required coincidences, that in fact exhibit the nuclear structure we are interested in, is long and tedious process. Obviously it requires many iteration steps. For instance the number of events from one GAMMASPHERE experiment is about 10G events. Only to read the event data of this size takes about 2 hours (using 3GHz PC). Consequently this format is unsuitable for interactive processing. Therefore the experimenters resort to histograms. The analysis of histograms based on slicing can be done very quickly in interactive way. However three and more dimensional histograms require also enormous space in memory. Therefore the compression of such data is unavoidable even at the price of loss of some information. After the search is finished, eventually, the final step can be done from original event data.

In the paper we have presented the methods of multidimensional coincident  $\gamma$ -ray spectra compression based on maintaining the list of event descriptors using pseudo-random address transformation method. The randomizing transform method is based on generation of inverse numbers in the sense of modulo arithmetic using prime moduli. The pseudo-random transformation proposed in this way satisfies the criterion of uniformity of the distribution of descriptors over the whole range of the allotted memory space. Adjacent descriptors go to addresses far away from each other.

To calculate inverse numbers in modulo arithmetic one can use Euclidian algorithm [8]. However for the ring of numbers  $\langle 0, M - 1 \rangle$ , where  $M$  is used modulus, one can precompute the inverse numbers and to create look-up table. Then the transformation is very fast and is suitable for on-line compression of events.

The compression algorithm proposed in the paper benefits also from the fact that the multidimensional  $\gamma$ -ray spectra are symmetrical. Then the compression is achieved on account of both symmetry of the multidimensional spectra and quasi-uniform distribution of the descriptors in the compressed space.

We have studied the randomizing methods based on division of polynomials [6-7] as well. The results achieved prove in the favor of the algorithm of pseudo-random transformation based on inverse numbers. Moreover, unlike the methods presented in [6-7] it allows to fit more precisely the space of pseudo-random numbers to the memory available.

The compression algorithm presented have been implemented in the data acquisition, processing and visualization system DaqProVis which is being developed at Institute of Physics, Slovak Academy of Sciences [9].

## Acknowledgements

The work is supported by the Grant Agency of Slovak Republic through contract GAV 2/7117/27.

## References

- [1] M. A. Deleplanque, R. M. Diamond (Eds.), *GAMMASPHERE a National Gamma-Ray Facility, a proposal*, 1988.
- [2] Simpson J., *The EUROBALL spectrometer*, *Z. Phys.* **A 358** (1997) 139.
- [3] M. Cromaz, et al., *Blue: a database for high-fold  $\gamma$ -ray coincidence data*, *Nucl. Instr. Meth.*, **A 462** (2001) 519.
- [4] M. A. Deleplanque, et al., *GRETA: utilizing new concepts in  $\gamma$ -ray detection*, *Nucl. Instr. Meth.* **A 430** (1999) 292.
- [5] D. C. Radford, *ESCL8R and LEVIT8R: Software for interactive graphical analysis of HPGe coincidence data sets*, *Nucl. Instr. Meth.* **A 361** (1995) 297.
- [6] V. Bonačić, B. Souček, K. Čuljat, *Pseudo-Random Digital Transformation*, *Nucl. Instr. Meth.* (1968) 213.
- [7] B. Souček, V. Bonačić, K. Čuljat, *Million Channel Pulse Height Analyser Through Pseudo-Random Digital Transformation*, *Nucl. Instr. Meth.* **66** (1968) 202.
- [8] Nussbaumer H.J., *Fast Fourier Transform and Convolution Algorithms*, Springer Verlag, Berlin 1981.
- [9] M. Morháč, V. Matoušek, I. Turzo, J. Kliman, *DaqProVis, a toolkit for acquisition, interactive analysis, processing and visualization of multidimensional data*, *Nucl. Instr. Meth.* **A559** (2006) 76.