



# A multiresolution data visualization tool for applications in neutron time-of-flight spectroscopy<sup>☆</sup>

I. Bustinduy<sup>a,\*</sup>, F.J. Bermejo<sup>a</sup>, T.G. Perring<sup>b</sup>, G. Bordel<sup>a</sup>

<sup>a</sup>*C.S.I.C - Department of Electricity and Electronics, University of Basque Country, P.O. Box 644, Bilbao 48080, Spain*

<sup>b</sup>*ISIS Facility, Rutherford Appleton Laboratory, Chilton, Didcot, Oxon OX11 0QX, UK*

Received 11 February 2005; accepted 16 March 2005

Available online 10 May 2005

## Abstract

Volume modelling and visualization pose important challenges for scientific data handling. Neutron spectroscopy is an intensity-limited technique where efficient data capture is a must. Such requirements are usually met using wide angular coverage detectors. Such devices may generate several gigabytes of information produced for an individual experiment, which needs to be handled within short lapses of time.

This paper describes a technique for the construction of a volume model using information coming from different levels of ‘binning’ resolutions to be applied to the raw (as measured) data. The technique which is adaptive in nature provides an efficient representation of the information, allowing one to explore in detail regions of the experimental 4D space where the sought data are concentrated.

Crown Copyright © 2005 Published by Elsevier B.V. All rights reserved.

*PACS:* 06.30.-k; 07.05.-t; 61.12.-q; 25.40.Fq

*Keywords:* Neutron scattering; Data treatment and visualization; Volume visualization; Spin waves; Condensed matter

## 1. Introduction

Since its very inception in the mid-fifties, the study of excitations in condensed-matter bodies

using neutron beams has vastly enlarged our knowledge on microscopic space–time correlations. Neutron scattering poses stringent limitations to such an endeavor because of three main reasons. First, the technique has been since its beginning an intensity-limited tool. This results from the nature of the process of neutron production (fission reactors or accelerator-based sources) which limits the achievable fluxes of particles impinging on the specimen of interest.

<sup>☆</sup>Special thanks to all Excitations group members in ISIS Facility.

\*Corresponding author. Tel./fax: +34 94 6015371.

*E-mail address:* [ibu@we.lc.ehu.es](mailto:ibu@we.lc.ehu.es) (I. Bustinduy).

<sup>1</sup>Work supported in part by grant *MAT2002* – 04540 – C05 – 03 (Spain).

Second, a neutron spectroscopy experiment on a crystal needs to explore a four-dimensional (4D) space constituted by the three cartesian components of the momentum-transfer vector  $\mathbf{Q}$  as well as the scalar energy-transfer. For most applications, data may be sparse, mostly being concentrated in some regions of space. Finally, the statistical quality of the collected data may vary substantially from one region to another in the reciprocal (momentum-energy) space, because of instrumental factors.

For reasons just outlined, an accurate representation of the measured data, including some measure of their reliability constitutes the first step in data treatment. To do so, an accurate volume model needs to be developed from observations together with available a priori knowledge on the specific problem to be addressed (eg. crystal symmetry etc.).

Traditionally, the scientific visualization processes comprise three, well-defined stages:

- (1) Model building (interpolation, binning, resampling, subsetting, etc.).
- (2) Mapping (drawing of contour plots, isosurfaces, etc.).
- (3) Rendering (conversion of a high-level object-based description into a graphical image for display).

Here we focus ourselves on the construction of a volume model<sup>2</sup> technique for data sets generated by a neutron chopper spectrometer. This technique [1,2], in which visualization techniques are still in their infancy, requires building of a volume model. Because of the need to detect and filter out spurious data, as well as to avoid data correlation, a new way of building a hierarchical grid is proposed here. It is important to notice that

<sup>2</sup>By *volume model* we refer to a process oriented to model discrete sets of massive data. These consist in  $n$ -dimensional arrays specifying spatial coordinates as well as the measured value for the dependent variable. In this sense *modelling* refers to the process of wrapping those sets of data with extra information regarding the characteristics of nearest neighbors. This will provide mapping tools with the necessary information to perform operations such as contour plots, isosurfaces or volume slices.

practically all visualization tools require some type of volume model for their application, even in more innovative scenarios as the *integrated visualization model* proposed by Robertson and de Ferrari [3] or the *Steering Model* proposed by Earnshaw and Jern [4], the construction of a volume model is required for the efficient analysis, visualization and transmission of the scientific data. Making the proposed model interesting in other areas where intense limited tools are used, also constitutes an aim of the work reported herein.

## 2. Data characteristics

An idealized scattering experiment of the kind we are interested in (direct geometry), consists in the selection of a neutron beam with well-defined energy and propagation direction (or wavelength or neutron velocity), that is made to exchange energy and momentum with the specimen of interest. Energy and momentum exchange are then measured in terms of the change in the speed of the incident neutrons as well as in the direction of the outgoing (scattered) beams. For such a purpose a technique known as time-of-flight is commonly used, where the change in velocity of the outgoing neutrons is measured by the time taken to reach a detector positioned at a given distance while the change in momentum is measured by the use of detectors positioned at different angles with respect to the incoming beam.

A given detector is thus placed at a given angle with respect to the incident neutron beam, and located at a distance  $L_2$  from the sample, while this sits at  $L_1$  from the neutron source. A beam of incident neutrons travelling with wave-vector  $\vec{K}_i (= 2\pi/\lambda)$  where  $\lambda$  stands for their De Broglie wavelength, will interact with the sample resulting in scattering at different angles, having final wave-vector  $\vec{K}_f$ .

The relevant quantities here are the momentum transferred to sample

$$\vec{Q} = \vec{K}_i - \vec{K}_f$$

as well as the energy exchange

$$\varepsilon = \frac{\hbar^2}{2m_N} (|\vec{K}_i|^2 - |\vec{K}_f|^2). \quad (1)$$

The change in neutron energy is thus determined by the corresponding change in the time taken to reach the detector  $t$ . As we are dealing with neutrons within the thermal and epithermal ranges (below some eV or so) non-relativistic mechanics holds and thus the neutron velocity is calculated from  $\vec{v} = L/T$ , where  $L = (L_1 + L_2)$  is the total flight-path. The neutron energy is then given by

$$E = \frac{1}{2} m_N \vec{v}^2. \quad (2)$$

As a result, by measuring the time-of-arrival of neutrons to the detectors one derives the final wave vector for each one of the detected neutrons.

A crystalline solid by virtue of its inherent anisotropy requires a three-dimensional (3D) variable to describe its structure. The latter is specified in full in terms of a decomposition into crystal planes that are indexed according to well-established procedures (Miller indices). Because of the nature of the experimental technique the position of an entity within the crystal is most conveniently described by setting ourselves within the reciprocal (Fourier) space. There, the crystal planes are specified in terms of three  $h, k, l$  indices and the change in momentum thus involves a 3D variable  $\mathbf{Q}_{hkl}$ . In other words, the information provided by a scattering experiment involves searches within a 4D space, energy-transfer being the additional dimension. This sets our basic problem as one dealing with searches over a 4D space that contains the relevant information within rather restricted regions. That is, the scattered intensities corresponding to static (positional) correlations give rise to strong intensity ‘spots’ within  $\mathbf{Q}_{hkl}$  leaving most of it empty. Conversely, the signature of atomic or spin motions will be comprised within narrow regions of  $[\mathbf{Q}_{hkl}, \Delta\mathbf{E}]$ .

For a severely intensity-limited technique exploration of the adequate portions of  $[\mathbf{Q}_{hkl}, \Delta\mathbf{E}]$  is thus a must. Information will come in the form of histograms of spectrum number and time-of-flight bin. These quantities are then converted into momentum-transfer and frequency (or energy-transfer).

This study is focused onto the most demanding case, that are single-crystal experiments performed on time-of-flight neutron spectrometers. Last generation machines employ Position Sensitive Detectors (PSD) that provide close to continuous coverage over a large solid angle array in the forward direction (about 160,000 detector pixels and an additional narrower strip of detectors in the horizontal plane). On such instruments one can control the resolution in both momentum and energy transfers by choosing different incident energies and instrument setups. Such a pixellated detector array gives the experimenter a possibility for high-speed visualization and data assessment. Development of analysis software thus plays a vital role in the exploitation of data. Time-of-flight and pixel position of the scattered neutrons are stored in an array of  $10^8$  pixels, from which the scattering function is constructed, in software, in a volume of reciprocal space of typically containing  $10^7$  volume elements.

The data are then represented as intensity (i.e. neutron counts) measured along some trajectories within wave vector-energy space. The origin of these trajectories define a two-dimensional (2D) grid and so a set of trajectories defines a data volume. The data format of interest for physicists corresponds to projections in wave vector-energy space ( $Q_h, Q_l, Q_k, \varepsilon$ ). These projections are called *viewing axes*  $U_1, U_2, U_3$ ,<sup>3</sup> and for the purpose of plotting they should be chosen to be orthogonal. Notice that the fourth vector-component, that is the energy axis is, by definition orthogonal to all wave vector components.

Our aim here is to provide the necessary technology to be able to apply the different volume visualization techniques in this scientific area. The nature of the problem, requires to re-bin raw data in a hierarchical grid over relevant viewing axes, retaining high flexibility and preserving high statistics areas. This will necessarily pass through the construction of an adaptive volume model that

<sup>3</sup>The experimental technique (direct geometry) requires to fix the value of  $K_i$ , which means that it is possible to express the wave vector-energy space in terms of just three vectors, because the  $(Q_h, Q_l, Q_k, \varepsilon)$  components are not independent. (In the case of indirect geometry the fixed parameter would be  $K_f$ , not affecting the projection, from 4D to 3D space.)

keeps track of local variations, choosing a large integration step on low statistics zones, and a small integration step otherwise.

### 3. Related work

Work previously reported mostly deals with 2D images. Within those, the Adaptive Intensity Binning of Sanders and Fabian [5], is perhaps a close precursor of the methods described below. Such authors consider a bin as a collection of pixels, meaning by a pixel one of the individual picture elements. The algorithm attempts to adaptively bin a single image based on the number of photons in each region. The number of pixels plotted in a graph is always constant, as also is the number of pixels that form the image. Its main interest is to facilitate human perception, plotting with the same color, pixels, that are contained in the same bin. This may present problems in cases where, for example, we have two bins  $a$  and  $b$ , where  $a$  is made by just one pixel (good statistics), and the bin  $b$ , is made by 15 pixels, (bad statistics), depending on the user-selected color scaling, high statistics areas are practically undistinguishable by the user.

Also, a Voronoi Tessellation scheme has been discussed by Cappellari and Copin [6]. There, Centroidal Voronoi Tessellation is employed to produce optimal 2D binnings. The procedure introduces some improvements, and in particular it satisfies the *Topological*, *Morphological* and *Uniformity Requirement* presented by the authors of the paper (Fig. 1).

The algorithm presented in this paper shares a viewpoint close to that presented by Sanders and Fabian. Our choice was determined by factors related to the size of our input data (8 million counts per experiment), as well as the ease in implementing 3D versions.

### 4. Multiresolution algorithm

In contrast to a common binning algorithm, where given a data space a grid is built just adding to the bin all the counts or data points

enclosed into the boundaries of the bin  $b = \{b_1, b_2, b_3, \dots, b_k\}$ , the multiresolution binning algorithm herein described, creates an adaptive volume model from discrete volume data.

By convention we will consider here the term *bin* or *pixel* as a collection of *counts* or *data points*, where the data points are simply the raw data we want to analyze. The pixel term will be used when dealing with images produced by equal-width or fixed binning algorithms.

We will construct a grid where the bin size is not fixed, but rather the boundaries of each bin will be determined by the statistics of the data points that encloses each bin. Each data set has its own peculiarities; each instrument has its own unique data collection characteristics, and a given sample will produce different intensity patterns depending on the incident neutron energy, sample orientation, temperature and pressure. It is therefore impossible to determine a priori the statistical function that will define the bin size, and to implement a predefined automatic decision about the general parameters for the representations. The way to determine these parameters is to allow the user to interact with the data to optimize them for each particular data set [7–9]. Each scientist knows better than anybody the particular behavior of their sample. In particular, the user establishes the boundaries of the grid, the maximum binning size, and the minimum binning size applied to each dimension, as well as the ratio error/intensity, *RATIO* that will act as a threshold, that will control the size of each bin.

The algorithm will have bin counts on different grid sizes (*LEVELS*) starting from the minimum bin-size grid (*LEVEL nL*), where dimension is determined by the user, up to the maximum bin size grid (*LEVEL 1*), in each *LEVEL*: the algorithm will calculate the arithmetic mean of intensity and the associated error value in each bin as follows:

$$S = \frac{1}{n} \sum_{i=1}^n S(x_i)$$

$$\text{ERR} = \frac{1}{n} \sqrt{\sum_{i=1}^n \text{ERR}(x_i)^2}$$

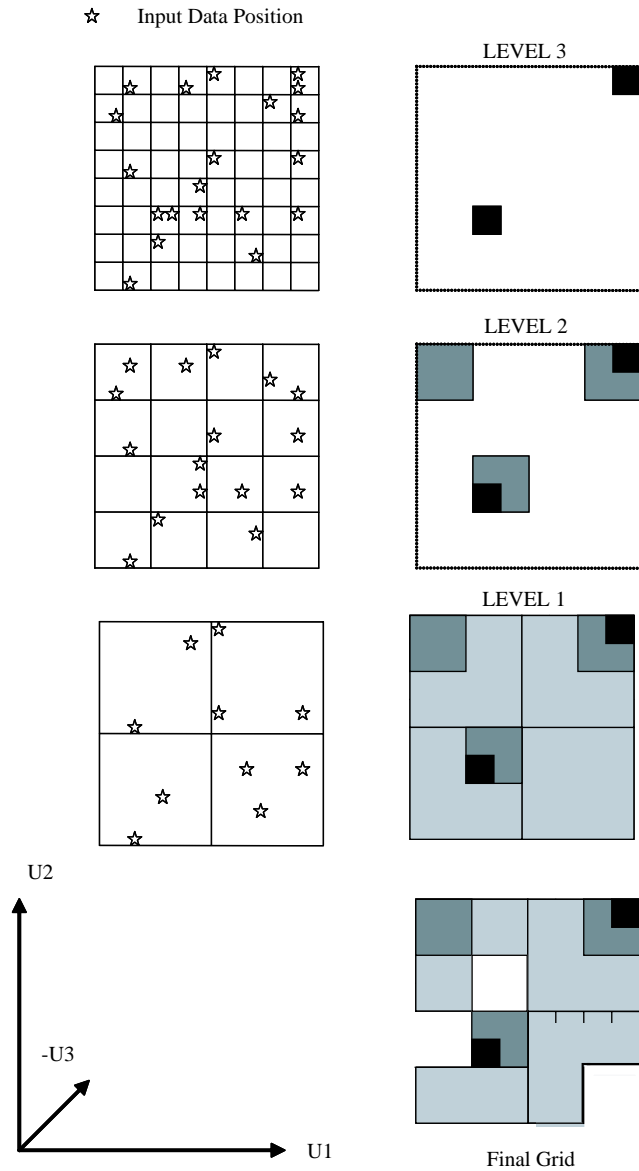


Fig. 1. To illustrate the multiresolution algorithm consider the 2D case, where the number of levels ( $nL$ ) is equal to 3, and the resolution pattern has chosen to be determined by the Level 2 binning size, although the principles can be extended trivially to higher dimensions.

where  $n$  stands for the data points  $x_i$ , enclosed by the boundaries of this bin. In our case the limits are taken as inclusive to the left and exclusive to the right along each axis ‘]’ to maintain compatibility with existing binning algorithms.

After that, a decision based on the error/signal ratio is made about accepting the bin as a valid one or not:

So **IF**  $ERR/S \leq RATIO$

**THEN** this bin will be selected, and the data points that form the bin, will be masked, removed

from the original data set. This way we impose the restriction that each data point will only contribute, in full to one and only one bin. This action is taken to avoid correlations between the errors (ERR) for different bins.

**ELSE** counts collected onto this rejected bin will be left to be binned presumably in successive LEVELS.

This process will continue up to the maximum binning size grid (LEVEL 1), doubling in each LEVEL the size of the previous LEVEL bin, (from  $(\Delta U1, \Delta U2, \Delta U3)$  at LEVEL  $j$  to  $(2\Delta U1, 2\Delta U2, 2\Delta U3)$  at LEVEL  $j - 1$ ) and removing those counts collected onto accepted bins.

Finally, the algorithm will mask the resulting image with a characteristic pattern of the instrument resolution, to avoid painting areas where there is no real input data.

So, the method can be seen as consisting of three steps. Given a DataSet of points  $\{x\}$ , the three steps are:

- (1) Construct the GRID for LEVEL = 1, and insert all data points  $x$  in DataSet into it.
- (2) For each level;
  - Extract the bins where ERR/S ratio is below the threshold to the FINAL GRID
  - Remove those contributing points from GRID
- (3) Mask the resulting image with the instrument resolution pattern.

#### 4.1. Implementation issues

In order to avoid possible problems related to the boundaries that arise from the specification of maximum and minimum binning size, the user is left to specify the grid boundaries and the maximum binning size, for each dimension. The same applies to the threshold value (RATIO), where the way in which the minimum binning size specified is given by the number of levels (nL). The algorithm will go up, doubling the size of the bins. Another issue concerns how to deal with data points having no signal ( $S = 0$ ). This means an instability in our condition and the way we have treated these points is to be above the threshold.

The relationship between the resolution in each axis is set from  $\Delta U_{\max} = 2^{(nL-1)} \cdot \Delta U_{\min}$ . This means that the number of levels this algorithm covers in each dimension will be the same. This restriction allows us to use consecutive convolution products between matrices, improving the execution time of the algorithm.

One of the main limitations is what to do with the remaining points that the algorithm has not collected, that is those which remain below the threshold even if we use the largest binning size. Different approximations have been taken such as to avoid plotting the remaining points or collecting them within the maximum binning size scheme, and plotting them using a different transparency level, with the aim of making them distinguishable with respect to the accepted bins. Our own experience tells that modification of the transparency attribute changes the perception of the color adopted by the bin. It thus seems better to show them with the same opacity level in the final image, since the contextual visual information [10,7,9] is more important than the imprecision we gain in the last level of integration, giving some level of uncertainty when the user visualizes the biggest bins.

The final implementation enables the user to fully interact with the algorithm by means of a Graphical User Interface, enabling to choose the boundaries, binning maximum and minimum sizes for each axis, error-intensity threshold and color axis scaling.

## 5. Results

### 5.1. 2D results

Although the algorithm has been largely used in different kinds of experiments, the following graphs correspond to experimental measurements on spin-dynamics of hexagonal cobalt [13] using the MAPS spectrometer located at the ISIS pulsed neutron source. Our choice is motivated by several reasons: it is a valid example of ‘low counting’ experiment, where the present technique is a must, and it also has a considerable number of counts to

be analyzed ( $\sim 8 \times 10^6$ ). These factors make this sample very suitable.

The most widely used method of volume visualization, consists in creating successively 2D plots, through cuts along the data volume. This provides a compromise between speed of volume viewing and accuracy of representation of details. It however presents some deficiencies, that will be explained below. At present, the available visualization software (named *Mslice* [11]), requires to take a *slice* of data as a function of any two variables from the four components of momentum and energy transfer ( $Q, \varepsilon$ ).

As a result, we obtain 2D color intensity plots that are shown in Fig. 2(a) and (c), where the bin size will be constant over the entire graph, the user will then be asked to decide whether to take a small, or large bin size, which of course, will be determined by statistics of the overall experiment, not adapting to the local peculiarities. At present *Mslice* allows a gaussian filter to smooth the obtained 2D color plot, helping the user to gain some insight of the dominant patterns (Fig. 2(d) and (b)). This represents the conventional solution adopted to represent ‘low statistics’ data, or to stand out a signal from the background. The chosen smoothing filter, a gaussian shape kernel, acts as a low pass filter, providing a gentler smoothing and preserving edges better than similarly sized common mean filters. In this particular case, the kernel is a rotationally symmetric gaussian low-pass filter of size  $3 \times 3$  pixels, with a standard deviation of approximately 0.5. The higher-order smoothing can be obtained by repeatedly convolving the same kernel.

In both previously mentioned figures, a pattern can be perfectly recognized, nevertheless, using a convolution filter we are introducing some data correlation, as the output will be a weighted average of each pixel’s neighborhood, with the average weighted toward the value of the central pixels. Such weighted average is only given in terms of intensity, since the error tolerance is not considered here. As a result we have that a pixel with a very low accuracy contributes the same way as it has a very high accuracy pixel, thus loosing accuracy in the displayed information. This could obviously mislead the researcher, since measured

data may not be clearly distinguishable in the obtained patterns.

Notice that especially in Fig. 2(b), most of the noise is still there, although it has decreased in magnitude somewhat. It has been smeared out over a larger spatial region. Increasing the standard deviation of the gaussian kernel, continues not only to reduce/blur the intensity of the noise, but also attenuates the high-frequency details (e.g. edges) significantly.

Finally, let us glance at Fig. 2(f) which is produced by the multiresolution algorithm. It not only displays information about the average intensity in each pixel but also yields the associated error bar. The intensity is represented by assigning color to the pixel, where the accuracy of such a measure will be given by the size of the pixel. In addition it represents the outline of the input data more accurately than the current implementation of the ‘fixed’ binning algorithm, where the boundaries of the detectors are not handled properly. In particular, we could wrongly infer from Fig. 2(d) that there is a high intensity peak centered in the  $[U1 = -0.35, U2 = 1, U3 = 3]$  position, where it is clear, see Fig. 2(e), that there is no input data.

We can also integrate along the third axis in some restricted range, in addition to specifying the integration ranges in the other two orthogonal directions. The result is then a 1D graph which traditionally gives accurate information, showing the intensity values, and their correspondent error-bars. Unfortunately, this plot only shows a very narrow information of the entire experiment, not letting users appreciate the nature of the experiment as a whole (Fig. 3).

## 5.2. 3D results

The present algorithm, was also implemented to construct a 3D grid, see Fig. 4. Volume visualization enables us to obtain a broader vision of the experiment, allowing to localize the area of interest, fast and accurately. It may also be possible to stick out properties of the material under study, that otherwise, using the traditional system; using 2D slices, would not be noticed. As a practical example, a crystal misalignment was

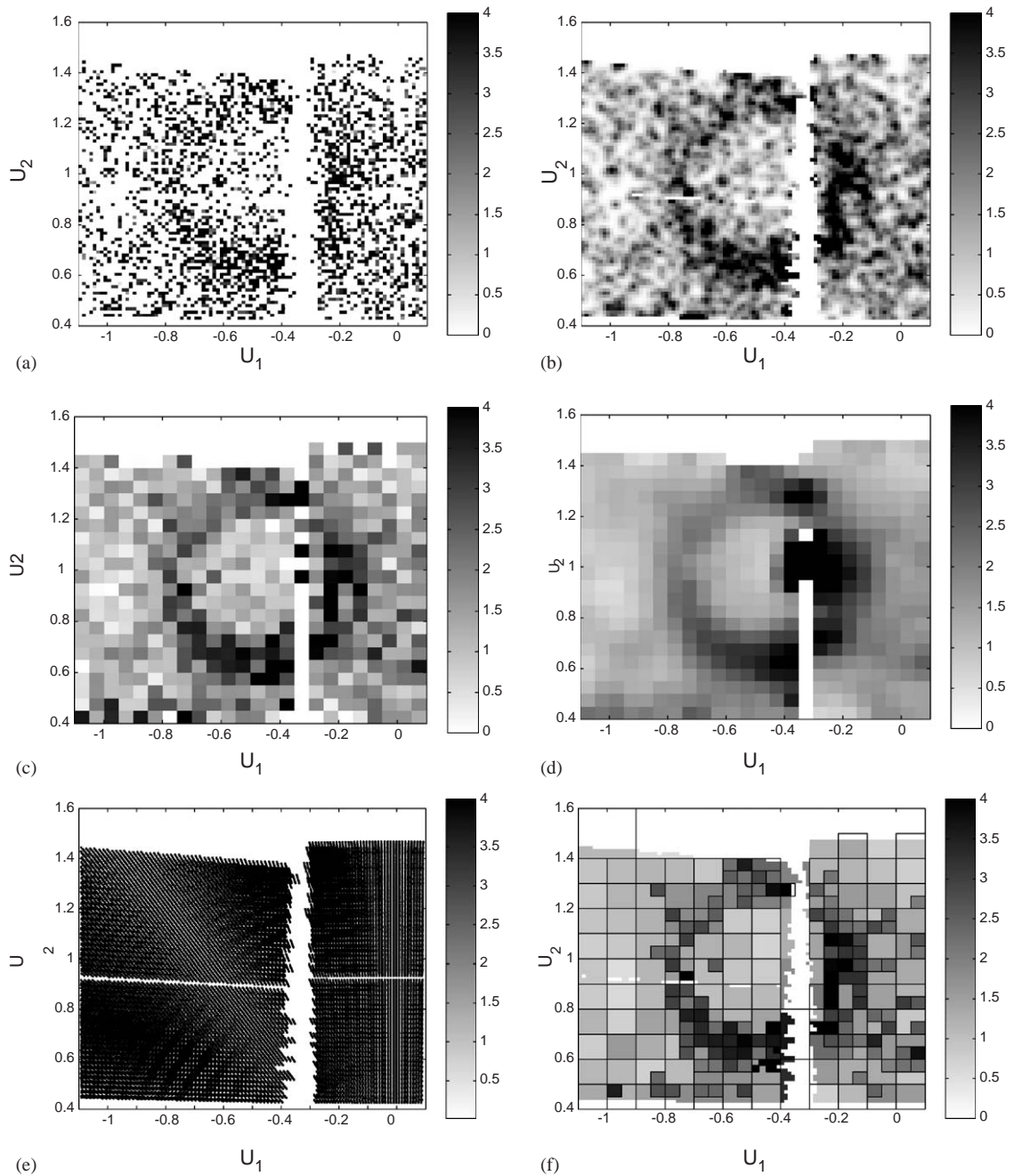


Fig. 2. Represents different color maps, integrated along the perpendicular axis  $U_3 = [0, 0, Q_j, 0]$ , from 2.9 to 3.1. Being  $U_1 = [Q_j, 0, 0, 0]$ , and  $U_2 = [0.5Q_j, -Q_j, 0, 0]$ . In all cases the color bar has chosen to go from 0 to 4 (abs. units). From (a) to (b) the figures have been produced using an equal-width binning algorithm, where bin size is  $[0.0125, 0.0125]$ . From (c) to (d) bin size is  $[0.05, 0.05]$ . Fig. (e) represents a scatter plot of raw input data. Fig. (f) has been produced using the present algorithm, with a threshold of 0.35, going from the maximum binning size:  $[0.2, 0.2]$  to the minimum binning size:  $[0.0125, 0.0125]$ . (a) Minimum binning size—'fixed' binning algorithm. (b) Smoothed minimum binning size—'fixed' binning algorithm. (c) Optimal binning size—'fixed' binning algorithm. (d) Optimal binning size—'fixed' binning algorithm after using a gaussian shape smoothing filter. (e) Input raw data. (f) 'Multiresolution binning' Algorithm.



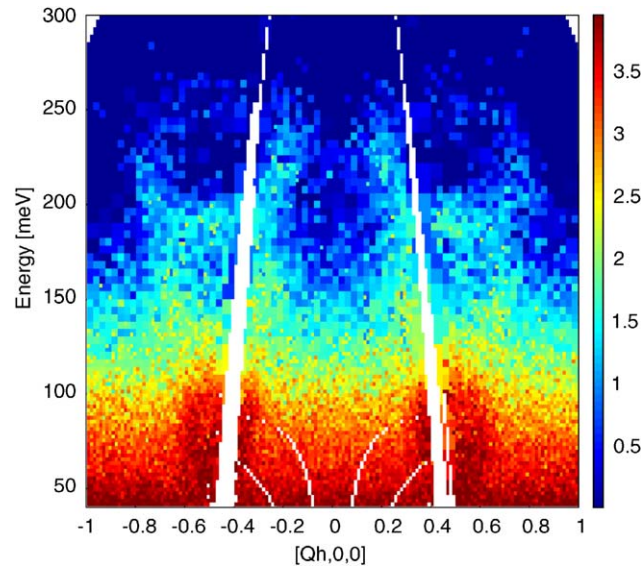


Fig. 3. High-energy spin waves in cobalt: electrons are also magnetized and hence much of the magnetism is delocalized. Low-energy excitations can be understood as spin wave modes, but at high energies they become damp due to interactions with single electron–hole excitations (Stoner modes), and interband resonances. The viewing axes have been chosen as  $U1 = [Q_h, 0, 0, 0]$  and  $U2 = \text{Energy}$ . Integrated along the perpendicular axis  $U3 = [0.5Q_h, -Q_h, 0, 0]$ , from 0.8 to 1.2. Image is mapped by means of different binning sizes, with a threshold of 0.35, going from the maximum binning size:  $[0.2, 32]$  to the minimum binning size:  $[0.0125, 2]$ . Color codes display the logarithm of the signal intensity.

discovered analyzing data from previous experiment [12], thanks to the 3D representation.

One of the most popular volume visualization techniques consist in displaying surfaces of constant intensity, better known as *isosurfaces*. Unfortunately, to get the most of this technique, it is required to have a minimum quality in the input data in terms of signal error, otherwise the obtained image may be confusing. For very low statistics data, it has been proven that the best way of visualizing volume data sets consists in using interactive slices and cuts of the volume under study.

Once the volume data set has been constructed we can apply the different volume visualization techniques, we consider to extract better information. In this case Fig. 4 shows constant slices cutting the volume data set to bring this information.

In the 2D version we could always modify the thickness of the plane, adding more points to each pixel. This will not affect the resolution in the

visualization of the plane. The net result of such an addition will lead to an intensity that will be smoothed in each pixel. In contrast, in the 3D version the resolution in one axis and the information contained in the rectangular cubic lattice are linked. This forces us to think in terms of rectangular cubic pixels, where the pixel size will need a careful thinking, in order to choose the best compromise between resolution and information content. The choosing of a very narrow step in one axis forces the perpendicular plane's bins to contain fewer points. Here again we rely on visual feedback from the end users to choose the correct dimensions.

## 6. Interactivity—CPU time

Time consumption is a fundamental issue to ensure rapid visual feedback. The formulation of queries by direct manipulation and the immediate display of the results has many advantages for

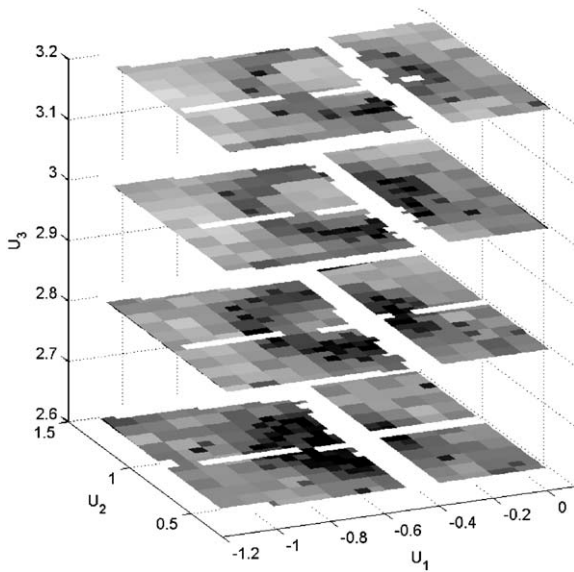


Fig. 4. Slices made from a volume data generated from cobalt experiment using the 3D implementation of the multiresolution algorithm. The used threshold value was again 0.35, and bin sizes go from maximum bin size of  $[0.2, 0.2, 0.4]$  to minimum binning size of:  $[0.025, 0.025, 0.05]$ , color bar goes from 0 to 4 (abs. units).

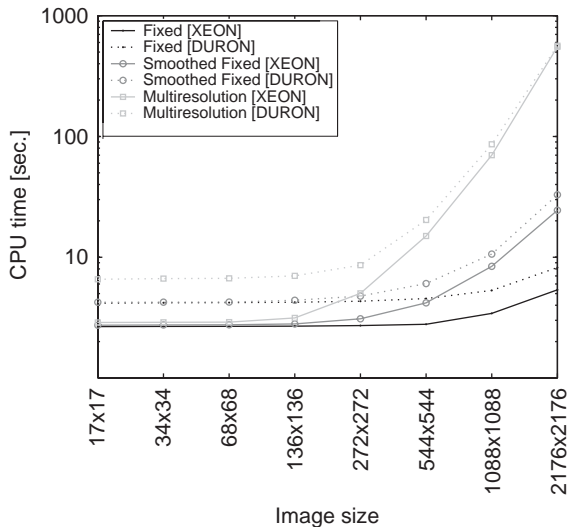


Fig. 5. Required times (in logarithmic scale) to construct the final image, the x-axis show the image size, in the *multiresolution* case, means the size of the grid constructed when  $LEVEL = 1$  (using the minimum bin size), which has been compared with the time required to construct the same size grid in both *Fixed* and *Smoothed Fixed* cases.

both new and expert users. This enables users to gain a deeper understanding of the data set, providing information about the nature of the problem that cannot be grasped with a static representation [7,14,15].

A few measurements have been carried out to test the computational cost of our algorithm. The chosen data set for this estimation had a total of 8.036.160 data points to be analyzed:

This data set was tested on two different machines:

- AMD DURON 1.3 GHz Clock Speed, 1,024,000 KB of RAM
- INTEL XEON 2.4 GHz Clock Speed, 2,096,092 KB of RAM

Fig. 5. compares the required CPU time to build a 2D grid. It represents three different runs: the traditional binning algorithm, the same algorithm after applying a gaussian filter, and the multiresolution algorithm. However, in practice we work within the first 5 levels, which means a resolution divergence of  $2^4$  times. The proposed version takes less than 9 s to run.

## 7. Conclusion

Experiments with different data sets demonstrate clearly the effectiveness of the multiresolution binning algorithm when compared with the current ‘fixed’ version.

The current visualization technique constructs a lattice where each pixel’s dimension is fixed for the entire slice or volume. This entails the decision between having good resolution (with the shortcoming of getting low statistics values) or trying to get high statistics, making a broader integration, to construct the lattice (and therefore losing resolution in high statistics areas of the reciprocal-energy transfer space).

The main problems that a fixed binning algorithm presents are that outliers may dominate the final presentation and also skewed data are not well handled. Correlation may exist when we try to smooth it, even after using the most modern adaptive kernel smoothing algorithms.

Both versions, 2D and 3D of the multiresolution algorithm, show information about the intensity and its correspondent error-levels simultaneously without distorting the resulting image. The size of the pixels is used in a convenient way to stand out the high–low statistics areas. By displaying high statistics areas with higher resolution and poor error/intensity ratio zones with lower resolution, we not only *filter* the image but also show the *accuracy* of the displayed intensity. This means that, in principle we can guarantee that the displayed information has a minimum level of accuracy.

We thus have built a volume model, specifically thought for the neutron scattering data, which will enable the application of the different volume visualization techniques. This opens a gate to the application of the latest techniques to this field. It is important to notice that practically all visualization tools require some type of volume model for their application, and therefore a correct implementation of a volume model will allow the implementation of the different visualization techniques that are already available for the rest of the scientific areas, such as medical image processing, earth sciences and so on [16,17].

Specifically the generated volume data sets were visualized in terms of separate slices, not having a true 3D representation of the data, limiting the possibilities of the user. The mind can glean information from animation that is virtually impossible to obtain by separately viewing still images [18]. The impact of iteration, which is another dimension of scientific visualization, allows the researcher, to cooperate with the computer by directing the computations, increasing the chances for inspiration, insight, and understanding.

## References

- [1] S.M. Hayden, H.A. Mook, P. Dai, T.G. Perring, F. Dogan, *Nature* 429 (6991) (2004) 531–534.
- [2] J.M. Tranquada, H. Woo, T.G. Perring, H. Goka, G.D. Gu, G. Xu, M. Fujita, K. Yamada, *Nature* 429 (6991) (2004) 534–538.
- [3] P.K. Robertson, L. de Ferrari, in: L. Roseblum, et al. (Eds.), *Scientific Visualization-Advances and Challenges*, Academic Press, London, 1994, pp. 287–305.
- [4] R.A. Earnshaw, M. Jern, in: L. Roseblum, et al. (Eds.), *Scientific Visualization-Advances and Challenges*, Academic Press, London, 1994, pp. 223–238.
- [5] J.S. Sanders, A.C. Fabian, *Mon. Not. R. Astron. Soc.* 325 (1) (2001) 178–186.
- [6] M. Cappellari, Y. Copin, *Mon. Not. R. Astron. Soc.* 342 (2) (2003) 345–354.
- [7] P.C. Wong, *IEEE Comput. Graphics Appl.* 19 (5) (1999) 20–21.
- [8] W. Hibbard, H. Levkowitz, J. Haswell, P. Rheingans, F. Schroeder, in: G.G. Grinstein, H. Levkowitz (Eds.), *Perceptual Issues in Visualization*, Springer, Berlin, 1995, pp. 23–32.
- [9] S.K. Card, J.D. Mackinlay, B. Shneiderman (Eds.), *Readings in Information Visualization*, San Francisco, USA, 1999.
- [10] L. Gregory J. Kittler, 11th International Conference on Image Analysis and Processing, Palermo, Italy, 2001.
- [11] R. Coldea, ISIS/Rutherford Appleton Laboratory, Mslice, <http://www.isis.rl.ac.uk/excitations/mslice/>, 2001.
- [12] R.A. Cowley, Spin fluctuations in  $\text{Rb}_2\text{MnF}_4$ , ISIS Experimental Report, Rutherford Appleton Laboratory.
- [13] T.G. Perring, High energy spin waves in hexagonal cobalt, ISIS Experimental Report, Rutherford Appleton Laboratory 2002.
- [14] J.M. Hellerstein, R. Avnur, A. Chou, C. Hidber, C. Olston, V. Raman, T. Roth, P.J. Haas, *Computer* 32 (8) (1999) 51–58.
- [15] B. Shneiderman, *IEEE Software* 11 (6) (1994) 70–77.
- [16] M. Chen, J.V. Tucker, *Comput. Graphics Forum* 19 (4) (2000) 281–293.
- [17] G.M. Nielson, in: M. Chen, et al. (Eds.), *Volume Graphics*, Springer, Berlin, 2000 pp. 29–48.
- [18] G.M. Nielson, *IEEE Comput.* 22 (8) (1989) 10–11.