E10-2001-205

M.V.Altaisky[1], G.A.Ososkov[2], A.G.Soloviev[3],
A.V.Stadnik, A.B.Shitov

WASP (WAVELET ANALYSIS OF SECONDARY PARTICLES ANGULAR DISTRIBUTIONS) PACKAGE. Version 1.2. LONG WRITE UP AND USER'S GUIDE

---

[1]E-mail: altaisky@mx.iki.rssi.ru
[2]E-mail: ososkov@cv.jinr.ru
[3]E-mail: solovjev@cv.jinr.ru

Contents

1 Introduction

During the last decade the number of wavelet applications increased drastically in various scientific fields, in high energy physics (HEP), in particular [1, 2]. Wavelet transform (WT) is demanded when one needs simultaneous knowledge of various signal frequencies, and also the time location of these frequencies.

Being a local integral transformation, WT overcomes two main shortcomings of the Fourier transform:

- Being nonlocal and having infinite support $(-\infty, \infty)$, Fourier transform requires a broad band of frequencies to decompose even a short signal;
- The distortion of a single harmonics in Fourier space affects, after reconstruction, the whole signal/image in coordinate space.

The main idea of the wavelet transform is to decompose the function under consideration with respect to a functional basis built by dilations and shifts of a single well localized function, called a *basic wavelet.*

For effective computer implementation of wavelet algorithms, the shifts and dilations should be in mutual concord. Fortunately, for a wide class of functions, used as basic wavelets, such algorithms exist. They have a pyramidal structure reducing by factor 2 the number of degrees of freedom being calculated at each step. These algorithms are usually referred to as fast wavelet transform (FWT).

Accomplishing of a wavelet analysis usually means the following:

1. make a proper choice of a wavelet type;
2. fulfill *a wavelet filtering* for denoising, removing pedestals and extracting some features of analyzed data. It is carried out by

   (a) transforming data to the wavelet domain;
   (b) applying desirable cuts on wavelet 2D-spectrum;
   (c) making inverse transform.

Wavelet applications to analyze 2D-patterns in the phase space of nuclear-nuclear events with the high multiplicity allows to reveal or emphasize some local properties of individual events inherent, say, for jets, as correlations or the presence of a dense group of particles.

In this paper after a long write up of one- and two-dimensional wavelet formalism and computational details we introduce a new version of the WASP (Wavelet Analysis of Secondary Particles angular distributions) package. The first version was described in [3] and successfully applied for data analysis [4].

Let us remind that WASP is a C++ program aimed to analyze angular distributions[1] of secondary particles generated in nuclear interactions. (It is designed for data analysis of the STAR and ALICE experiments.) It uses a wavelet analysis for this purpose and the vanishing momenta wavelets are chosen as a basis [5]. WASP version 1.2 allows one to perform both one- and two-dimensional wavelet analysis.

2 One-dimensional wavelet transform

2.1 Continuous WT

Formally, the wavelet transform of a function $f(x) \in L^2(\mathbb{R})$ [2] is a projection of f to the basic wavelet ψ dilated by factor a and shifted by b:

$$W_\psi(a,b)[f] = \int_{-\infty}^{\infty} \frac{1}{\sqrt{|a|}} \overline{\psi}\left(\frac{x-b}{a}\right) f(x)dx. \tag{1}$$

The reconstruction, or *inverse wavelet transform*, is

$$f(x) = C_\psi^{-1} \int\int \psi\left(\frac{x-b}{a}\right) W_\psi(a,b) \frac{da\,db}{a^2}, \tag{2}$$

where C_ψ is the normalization constant for a given basic wavelet ψ. It can be easily evaluated by substituting (1) into (2) and making Fourier transform

$$\phi(x) = \int_{-\infty}^{\infty} \exp(\imath\omega x)\hat{\phi}(\omega)\frac{d\omega}{2\pi}$$

---

[1] Other kinds of distributions can also be processed after minor corrections of WASP code.
[2] $L^2(\mathbb{R})$ is the square-integrable functional space.

of each term of (2), if the function ψ is chosen so that the integrals are finite:

$$C_\psi = \int\limits_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty, \tag{3}$$

where Fourier transform of $\phi(\omega)$ is denoted as $\hat{\phi}(\omega)$.

The *admissibility condition* (3), which provides a possibility of a function ψ to be used as a basic wavelet ψ, is rather loose. That is why there exist a lot of different basic wavelets invented by Daubechies [6], Meyer [7], Mallat and others [8].

2.1.1 Gaussian wavelets

Amongst all differentiable functions used as wavelets, the derivatives of the Gaussian

$$g_n(x) = (-1)^{n+1} \frac{d^n}{dx^n} e^{-x^2/2}, \quad n > 0, \tag{4}$$

also called the *vanishing momenta wavelets*, for their property $\int g_n(x)x^m dx = 0$, $m < n$ are of high importance. Roughly speaking, the $g_n(x)$ wavelet, when used for the analysis of a function $f(x)$, is most sensitive to the singularities of the n-th derivative $f^{(n)}(x)$. The normalizing coefficients for g_n family are $C_{g_n} = 2\pi(n-1)!$. The Gaussian itself (g_0) is not an admissible wavelet for it does not satisfy the condition (3).

It follows from the definition (4) that

$$\frac{d\,g_n(x)}{dx} = -g_{n+1}(x). \tag{5}$$

Other recurrent relations, valid for Hermitian polynomials also hold for g_n.

The localization properties for g_n family can be evaluated explicitly. In general case the continuous wavelet transform with basic wavelet ψ is centered at t^* and has the *window width* $2\Delta_\psi$:

$$t^* = \frac{1}{\|\psi\|_2^2} \int_{-\infty}^{\infty} x|\psi(x)|^2 dx, \tag{6}$$

$$\Delta_w = \frac{1}{\|\psi\|_2} \left\{ \int_{-\infty}^{\infty} (x-t)^2 |\psi(x)|^2 dx \right\}^{1/2}. \tag{7}$$

So, the wavelet coefficients $W_\psi(a,b)[f]$ are localized within the time window

$$[b + at^* - a\Delta_\psi, b + at^* + a\Delta_\psi]. \tag{8}$$

At the same time its Fourier spectrum is localized within the frequency window

$$\left[\frac{w^*}{a} - \frac{1}{a}\Delta_{\tilde{\psi}}, \frac{w^*}{a} + \frac{1}{a}\Delta_{\tilde{\psi}} \right].$$

3

Therefore, wavelet transform unfolds the signal locally in both time and frequency. It is worth noting, that the ratio of the central frequency ω^* to the window width is scale independent

$$\frac{\omega^*/a}{2\Delta_{\tilde{\psi}}/a} = \frac{\omega^*}{2\Delta_{\tilde{\psi}}}. \tag{9}$$

The detailed study of localization properties of wavelet transform can be found in [6]. As an example, let us present the window widths for two first wavelets from the vanishing momenta family (4). Using the definition (7):

$$\Delta_{g_n} = \frac{1}{\|g_n\|_2}\left\{\int_{-\infty}^{\infty} x^2|g_n(x)|^2 dx\right\}^{1/2}, \quad \|g_n\|_2 \equiv \sqrt{\int_{-\infty}^{\infty}|g_n(x)|^2 dx}$$

and Gaussian integrals

$$\int_{-\infty}^{\infty} x^{2n}\exp(-\alpha x^2)dx = (-1)^n\left(\frac{d}{d\alpha}\right)^n\sqrt{\frac{\pi}{\alpha}}$$

for two first wavelets we get

Wavelet	L_2-norm	Window width
g_1	$\|g_1\|_2 = \pi^{1/4}/\sqrt{2}$	$\Delta_{g_1} = \sqrt{3/2}$
g_2	$\|g_2\|_2 = \sqrt{3/4}\pi^{1/4}$	$\Delta_{g_2} = \sqrt{7/6}$

2.1.2 Examples

As the first example of how wavelets work, let us take a harmonic signal constructed by superposing the low-frequency one with the small fraction of the high-frequency one and then contaminating it by uniformly distributed random noise, see Fig. 1.
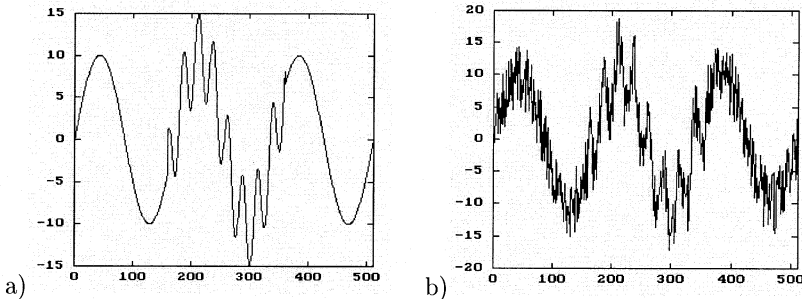

a)   b)

Fig. 1: Sample of a signal comprised of two harmonic components: a) without noise, b) contaminated by noise

4

Let us perform g_2-wavelet transform (widely known as "Mexican hat") of the contaminated signal. Fig. 2 (left) presents the wavelet spectrum. The shade-plot provides a powerful tool, which helps to display the structure of the signal. The set of wavelet coefficients can be presented as a projection of 3-dimensional surface $W = W(a, b)$ onto the a-b plane. Coefficients with higher values are shown in light colors, the lower ones in dark. Although the wavelet spectrum is very informative, it often brings too much visually redundant information. To make it more contrast, the gray-scaled image is often transformed to the so-called *wavelet skeleton*. Lines on the skeleton correspond to local maxima of the wavelet coefficients.
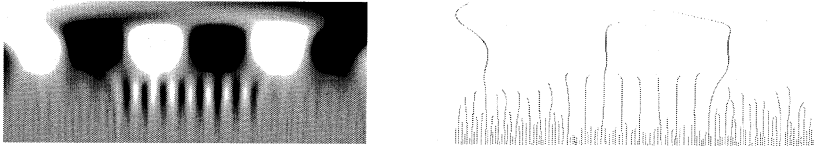

Fig. 2: Wavelet spectrum (left) and its skeleton (right) calculated for the signal shown in Fig. 1b. Horizontal axis corresponds to the time parameter (b), vertical one — to the scale parameter (a).

In the next figure the results of the final g_2-filtering of the same signal are shown. First, we accomplish g_2-filters with four selected scales only: 32, 64, 128 and 256. The result of the inversion is the signal in fig. 3a. When our signal was processed at scales 1, 2, 4, 8 and 16 its inversion gives result in Fig. 3b.
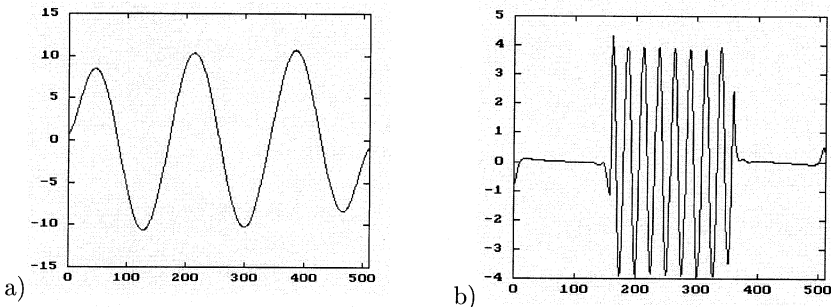

Fig. 3: Extracting data with g_2 filter from the signal of Fig. 1a: a) low frequency component, b) high frequency component.

It is seen that the filter allows to extract both components of the original signal. Thus selecting properly the scales of the wavelet transformation it is possible to highlight the components of desired scales. It should be pointed out here, that with the same signal the simple Fourier filter would fail. It could be done, in principle, as well by applying a Fourier filter, but as two-step procedure: to select the high

5

frequency short-living sin-wave you should, first, extract the low-frequency wave and than subtract it from the signal. The wavelet filter allows a direct extraction of the desired component.
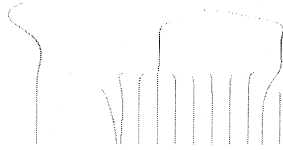

Fig. 4: Skeleton of the signal after denoising by g_2 wavelet. Comparing to Fig.2 (right) shows the disappearance of all short noise fractions smaller than $a = 2$.

Another important application of wavelets is signal denoising. The procedure of denoising consists of nullifying wavelet coefficients with small amplitude before inverse transform. The g_2 skeleton of the signal **after such denoising** is shown in Fig. 4. As one can see, its typical high-frequency part produced by noise is efficiently suppressed. Then the inverse transform would enhance the corrupted signal back to its original view in Fig. 1a.

2.1.3   Implementation in WASP

The direct evaluation of the convolution in the direct (1) and inverse (2) wavelet transform may be performed numerically but is expensive in memory and CPU time. At the same time, the self-similarity property of WT suggests the methods for constructing fast and effective algorithms. The straightforward way of WT implementation is to restrict the calculation on a discrete sub-lattice

$$a = a_0^m, \quad b = nb_0 a_0^m, \quad \text{where } m, n \in \mathbb{Z}, \quad b_0 \neq 0,$$

and set the dilation step greater than unity $a_0 > 1$. The bi-parametric family of wavelets $U(b, a)\psi(x)$ becomes then a discrete set

$$\psi_{mn}(x) = a_0^{-m/2} \psi(a_0^{-m} x - nb_0), \tag{10}$$

labeled by two integers $m, n \in \mathbb{Z}$.
   An inverse discrete transform

$$f(x) = \frac{2}{A + B} \sum_{m,n} \langle \psi_{mn}, f \rangle \psi_{mn}(x) + \text{error terms}, \tag{11}$$

turns to be numerically stable, if for some $A > 0, B < \infty$

$$A\|f\|^2 \leq \sum_{mn} |\langle \psi_{mn}, f \rangle|^2 \leq B\|f\|^2.$$

6

If it is the case, the basis ψ_{mn} is called a *frame*. In general, a frame is not orthogonal: only the case $A = B = 1$ gives orthogonal decomposition in (11).

However, if the analyzed signal consists of a discrete set of measured counts $\{h_k, \ k = \overline{1, N}\}$, we can realize two discretization schemes using the explicit integration formula of explicit integration of Gaussian wavelets over bins to speed up the computations.

For a discretized signal

$$f(x) = \frac{1}{N} \sum_{k=1}^{N} \delta(x - x_k) , \tag{12}$$

(1) is written as

$$W_\psi(a, b) f = \frac{1}{N} \frac{1}{\sqrt{|a|}} \sum_{k=1}^{N} \psi\left(\frac{x_k - b}{a}\right) , \tag{13}$$

and it looks like the Parsen–Rozenblatt estimates of the unknown probability density over a sample (see [9]). (13) is nothing but the "averaged sum" of the wavelets $\psi[(x_k - b)/a]$ compressed to the size a and "placed" at points x_k.

The formula (13) does not contain integral and this fact allows one to speed-up the wavelet transform algorithm for discretized signals like (12). Thus, (13) is just what is used in WASP for the wavelet analysis.

However for gaussian wavelets (GW) especially there is one more way to improve this algorithm [5]. Having a discretized signal $\{h_k, \ k = \overline{1, N}\}$, one can define

$$W_{g_n}(a, \, b) \, h = \frac{1}{|a|} \sum_{k=1}^{N} h_k \int_{x_{k-1}}^{x_k} g_n\left(\frac{x - b}{a}\right) dx,$$

where $x_k = x_0 + k \, \Delta x$.

Due to (5) we can replace the integral by the difference

$$W_{g_n}(a, \, b) \, h = \frac{a}{\sqrt{|a|}} \sum_{k=1}^{N} h_k \left[g_{n-1}\left(\frac{x_{k-1} - b}{a}\right) - g_{n-1}\left(\frac{x_k - b}{a}\right) \right]. \tag{14}$$

To speed up calculations it better to rearrange the summands:

$$W_{g_n}(a, \, b) \, h = \frac{a}{\sqrt{|a|}} \left[h_1 g_{n-1}\left(\frac{x_0 - b}{a}\right) + \sum_{k=1}^{N-1} (h_{k+1} - h_k) \, g_{n-1}\left(\frac{x_k - b}{a}\right) - \right.$$
$$\left. - h_N \, g_{n-1}\left(\frac{x_N - b}{a}\right) \right]. \tag{15}$$

That gives about a double gain in speed in comparison with (14).

However this expression is still rather slow to evaluate and could be used only for obtaining a few coefficients. For example, applying (15) to calculate $W_{g_n}(a, b_j)\, h$ while a is fixed, but b runs over M values $b_j = b_0 + j\,\Delta b$, $j = \overline{0,\, M-1}$ one needs to calculate $M(N+1)$ values of the wavelet g_{n-1} in points $\frac{x_k - b_j}{a}$. Nevertheless, if we restrict the choice of the shift steps, as

$$\Delta b = \frac{\Delta x}{n}, \quad n \in \mathbf{N} \tag{16}$$

or

$$\Delta b = n\Delta x, \quad n \in \mathbf{N} \tag{17}$$

it can speed up the calculation process. For instance, in the first case (16) to calculate $W_{g_n}(a,\, b)\, h$ one needs to obtain g_{n-1} in points $\frac{x_0 - b_0}{a} + \frac{\Delta b}{a}(kn - j)$. The GW argument is identically defined by the integer index $(kn - j)$. All possible combinations of k and j give $N + Mn + 1$ of different values. This number is growing much slower with the growth of N and M, than in the case of calculating by the formula (15). Thus we consider the values of g_{n-1} as the vector with $nN + M + 1$ components

$$g^* = \left\{ g_{n-1}\left(\frac{x_0 - b_0}{a} \right) + \frac{\Delta b}{a}(kn - j) \right\}.$$

The input discrete data can be also considered as the vector with $N + 1$ components

$$h^* = \{h_1,\ h_2 - h_1,\ h_3 - h_2,\ \ldots,\ h_N - h_{N-1},\ -h_N\}.$$

Then the raw of wavelet coefficients one can obtain as the scalar product of vectors h^* and g^{**}, where components of g^{**} are obtained from g^* just by selecting components with indexes $kn - j$.

In the second case (17) one has the vector $W_{g_n}(a,\, b_j)\, h$ with $nN + M + 1$ components and should replace everywhere Δb and $kn - j$ by Δx and $k - jn$.

The next substantial resource of increasing the speed of calculations is based on the practically compact support of the GW in space.

2.2 Discrete WT

The main idea of a discrete wavelet transform is to represent a given data as a decomposition using basis functions $\psi_{j,k} = 2^{-j/2}\psi(2^{-j}t - k)$, $\phi_{j,k} = 2^{-j/2}\phi(2^{-j}t - k)$. They are constructed as scaled and shifted two basic functions:

- a scaling function ϕ;
- a wavelet function (or "mother wavelet") ψ.

Both these functions should have a locality in time/space and frequency domains. The purpose is to get the following decomposition of a source data:

$$f(x_i) = \sum_{k=1}^{N} s_k \phi_{Lk}(x_i) + \sum_{j=1}^{L}\sum_{k=1}^{N} d_{jk}\psi_{jk}(x_i)\,,$$

8

where N is the number of samples in data sequence, and L is the decomposition level. As the result, we can describe our signal in terms of coefficients s_k and $d_{j,k}$, where s_k are called approximation coefficients, $d_{j,k}$ are details at the j-th level of decomposition. The way how to get this representation of the signal is shown on Fig. 5.
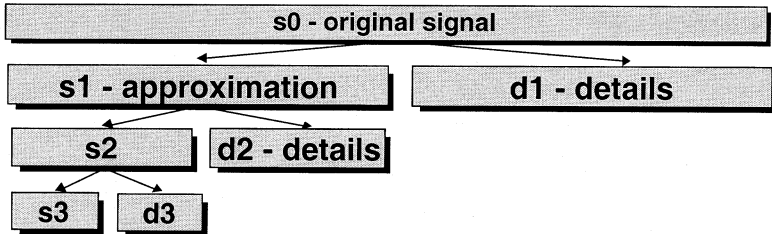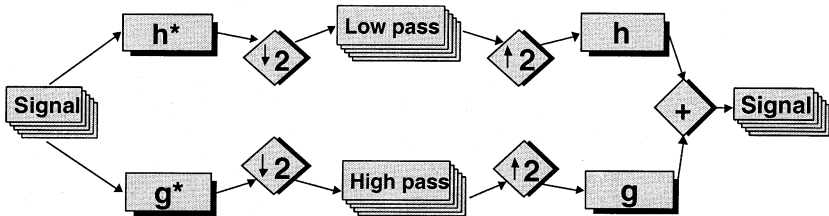

Fig. 5: Wavelet decomposition scheme

Fig. 6: Scheme of one decomposition step

Each decomposition step realized by using two filters: h_k — low pass filter and g_k — high pass filter. Filters h_k and g_k depend on functions ϕ and ψ as follows:

$$\phi(x) = 2^{1/2} \sum_{k=0}^{l-1} h_k \phi(2x - k) , \quad \psi(x) = 2^{1/2} \sum_{k=0}^{l-1} g_k \phi(2x - k) .$$

So, h_k and g_k are solutions of these scaling equations, where l means the length of filter. The Fig. 6 shows one decomposition step.

2.3 Data filtering

The filtering is carried out in three steps:

1. calculate the set of wavelet coefficients;
2. cut off the spectrum (e.g set to zero all coefficients which values are less then prescribed threshold);
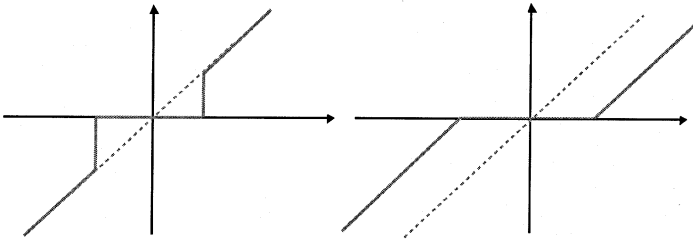3. perform the inverse wavelet transform of the obtained spectrum.

Fig. 7: Wavelet thresholding rules

One of the useful properties of WT is its ability to denoise a data by *thresholding*. There are two basic methods yielding a good result:

- "hard" thresholding rule. If λ is a threshold value, the rule looks like

$$\begin{cases} y \to 0, & y \leq \lambda; \\ y \to y, & y > \lambda. \end{cases}$$

- "soft" thresholding rule:

$$\text{sign}(y)(|y| - \lambda).0, \qquad y < \lambda.$$

Both "hard" and "soft" rules are represented on Fig. 7.

2.4 Lifting scheme

Lifting scheme is fundamentally other approach to building wavelet decompositions, proposed by W.Sweldens [10, 11]. Wavelet transform is said to be the way to decorrelate data. The wavelets generated by translations and dilations of one function are called first generation wavelets. This idea is extended by *lifting scheme*, which is a new tool for constructing of biorthogonal wavelets. The lifting scheme inherits the idea of FWT of applying low-pass and high-pass filters sequentially to the data array in place. The lifting scheme makes it in an optimal way, sometimes increasing the efficiency of calculations by factor 2, the inverse transform becomes just the reversing the order of operations with substitution + by −, and vice versa. Since lifting scheme does not rely on Fourier Transform (FT), it can be used even in cases where translations and dilations can't be used.
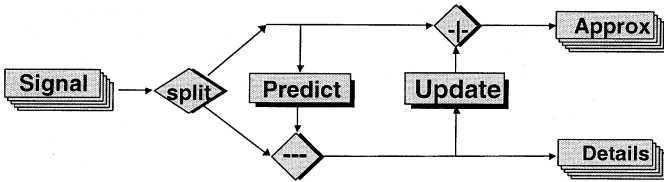
Fig. 8: One step of the lifting scheme.

Follow W.Sweldens [11], let us assume that we have a signal and we need to make a decomposition of this signal into a non-correlated parts. First we separate the signal into a two equal parts, by putting all even points into one array, and all odd into another. If the data in the source signal are correlated, than the first array contains some information about the second one. Let us denote these two arrays by λ and γ, respectively. Thus the first stage of the lifting scheme is two separate data into two classes. The second stage of the lifting scheme is to find a data-independent prediction operator, such that $\gamma = P(\lambda)$. Let us call it *prediction*. If the second array is functionally dependent, the prediction P is exact, if not, we can substitute the array γ, in place, by the differences, called *wavelets*: $\tilde{\lambda} = \lambda + U(\gamma)$. Let us call this *update rule*. The procedure is then recursed storing the lost details in place of removed data. On each stage of reconstruction the lost details are added to the result of prediction.

As it shown by I. Daubechies and W. Sweldens, the realization of the lifting scheme generates an orthogonal wavelets of the Daubechies family.

3   Two-dimensional wavelet transform

3.1   WT via fast Fourier transform (FFT)

The wavelet image of a function $f(\mathbf{x}), \mathbf{x} \in \mathcal{R}^2$ is a scalar product of a representation of the affine group acting on $\mathcal{R}^2$:

$$\mathbf{x}' = aR^\theta \mathbf{x} + \mathbf{b}, \quad U(\mathbf{a}, \mathbf{b}, \theta)\psi(\mathbf{x}) = \frac{1}{|a|}\psi\left(R^{-\theta}\frac{\mathbf{x} - \mathbf{b}}{a}\right)$$

where $R^\theta \mathbf{x} = (x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta)$. So, the definition of wavelet coefficient taken with the basic wavelet $\psi(\mathbf{x})$ is:

$$W(a, \mathbf{b}, \theta)[f] = \int \frac{1}{|a|}\overline{\psi}\left(R^{-\theta}\frac{\mathbf{x} - \mathbf{b}}{a}\right)f(\mathbf{x})d^2\mathbf{x}$$

For practical purposes it is faster to evaluate the convolution (3.1) in the Fourier space using FFT algorithm.

$$W(a, \mathbf{b}, \theta)[f] = a\int \exp(\imath \mathbf{b}\mathbf{k}).\overline{\hat{\psi}}\left(aR^{-\theta}\mathbf{k}\right)\hat{f}(\mathbf{k})d^2\mathbf{k}$$

It is possible to simplify the calculations it is possible to use rotationally symmetric basic wavelets $\psi = \psi(|\mathbf{x}|)$. In this case there will be no dependence on the angle variable θ.

One of the wavelets of most common use is the Mexican Hat wavelet

$$\psi(\mathbf{x}) = \left[2 - \mathbf{x}A\mathbf{x}\right].e^{-\frac{1}{2}\mathbf{x}A\mathbf{x}}, \quad \hat{\psi}(\mathbf{k}) = |det A^{-1}|(\mathbf{k}A^{-1}\mathbf{k})e^{-\frac{1}{2}\mathbf{k}A^{-1}\mathbf{k}}$$

For the rotationally symmetric case $(A = a\hat{I})$, the Mexican hat wavelet (3.1) is simplified to

$$\psi(\mathbf{x}) = (2 - \mathbf{x}^2)\exp(-\mathbf{x}^2/2), \quad \hat{\psi}(\mathbf{k}) = \mathbf{k}^2\exp(-\mathbf{k}^2/2) \tag{18}$$

The scheme of the practical implementation of the evaluation of

$$W(a, \mathbf{b})[f] = a \int \exp(i\mathbf{b}\mathbf{k})\overline{\hat{\psi}}(a\mathbf{k})\,\hat{f}(\mathbf{k})d^2\mathbf{k}$$

was as follows:

1. Evaluate Fourier transform of the original image $f(\mathbf{x}) \to \hat{f}(\mathbf{k})$ by means of FFT algorithm (Numerical Recipes library was used)
2. Multiply the $\hat{f}(\mathbf{k})$ at each point by $(a\mathbf{k})^2\exp(-(a\mathbf{k})^2/2)$.
3. Taking the inverse FFT of the result.

The normalization factor a of (3.1) was dropped in the numerical code.
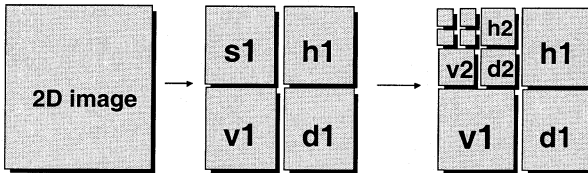
3.2   Discrete WT

Fig. 9:   Two dimensional wavelet decomposition.

One step of a wavelet transform of a signal with a dimension n higher than one is performed by transforming each dimension of the signal independently. Afterwords the n-dimensional subband that contains the low pass part in all dimensions is transformed further. The 2-dimensional case is presented on the Fig. 9. The areas denoted by letters:

- "v— vertical coefficients, the basis function — $\psi(y)\phi(x)$;
- "h— horizontal coefficients, the basis function — $\phi(y)\psi(x)$;
- "d— diagonal coefficients, the basis function — $\psi(y)\psi(x)$;
- "s— approximation coefficients, the basis function —$\phi(y)\phi(x)$.

12

4 WASP user's guide

WASP provides an user-friendly Graphical User Interface (GUI) which makes it quite easy to use. Both visualization and GUI are implemented based on some of the ROOT classes. So, WASP requires ROOT [12] to be installed first. However, WASP classes, that are not related to visualization or GUI, are ROOT-independent. In this user's guide we show how to work with WASP (version 1.2) using GUI.

4.1 Graphical User Interface

To get started, first WASP must be installed and compiled. Then just type `wasp` in your command line. WASP main window (see Fig. 10) appears. It has two panels. The raw event is visualized as a simple histogram on the left, the corresponding wavelet spectrum is shown on the right.
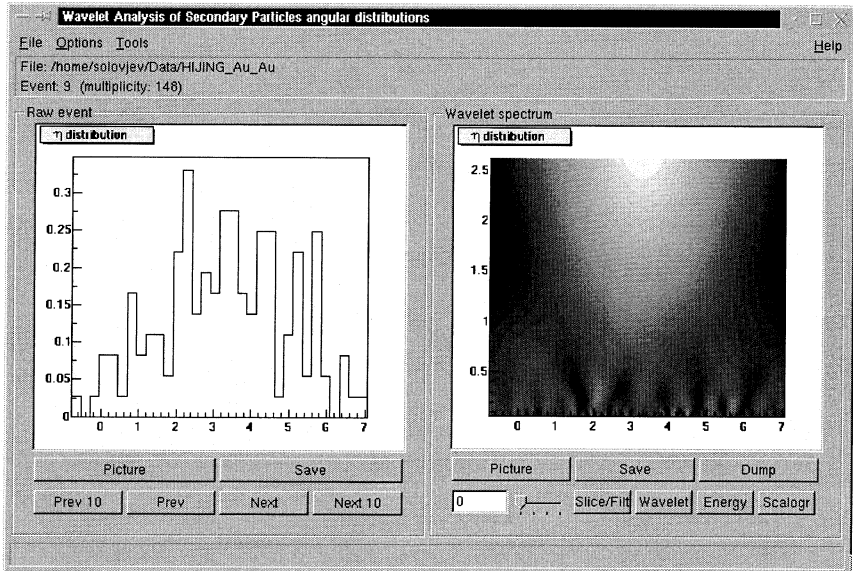

Fig. 10: Main Window

Click the **Open** item in the **File** pop-up menu to choose input data file. The **Reopen** item in this menu allows one to open the same file again (e.g., when the end-of-file is reached). Note, that the implementation of the WASP implies the following input file format. The file is read line-by-line. If the line contains at least two numbers, the first of them is interpreted as a polar angle in radians, the second one — as azimuthal angle in radians (moreover, it is transformed to be in $[0, 2\pi]$

if necessary), the rest of the line is ignored. Other lines are interpreted as event separators.

Use **Next**, **Prev**, **Next 10**, **Prev 10** buttons on the left-hand panel of the main window to get the needed event from file (these functions are duplicated as items of the **Tools** menu, moreover, there is an **All** item in it which allows one to collect events from a data file). One can use event selection feature tuning it by clicking the **Filter** item in the **Options** menu to skip events that do not satisfy the chosen settings. Event selection dialog is shown on Fig. 11.

One can choose the raw event representation by clicking **Picture** button on the main window left-hand panel and the wavelet spectrum representation — by clicking **Picture** button on the right-hand one. Instead of clicking **Picture** buttons one can choose corresponding items in the **Options** menu. Corresponding panels are shown on Figs. 12a and 12b.
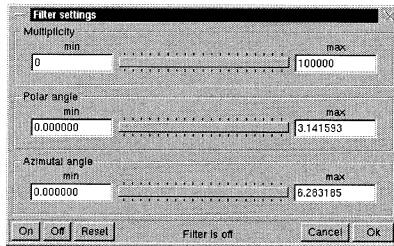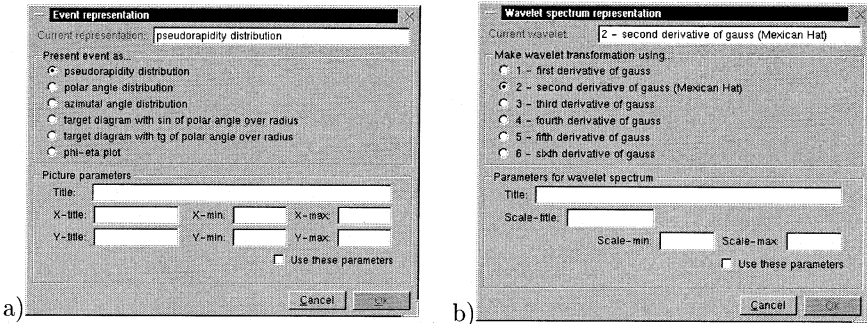

Fig. 11: Event Selection Dialog

Fig. 12: Event (a) and Wavelet Spectrum (b) Representation Dialog

The default event representation is pseudorapidity distribution. The histogram ranges are calculated automatically over each event. They can be fixed by hand using the panel shown on Fig. 12a instead, moreover, a histogram title, x-axis and y-axis labels can be chosen (Fig. 10 shows defaults). Note, that the x-range fixed for

14

one-dimensional histograms does not affect on the obtained wavelet spectrum, since it is built over the whole sample directly (without histograming). It is just a zoom feature. The y-range fixing option is not used for one-dimensional histograms. On the contrary, the two-dimensional wavelet analysis takes into account the histogram with its ranges, calculated automatically or fixed by hand.

The default one-dimensional wavelet used for transformation is the second derivative of the Gauss function ("Mexican Hat"). The x-range and the x-axis label are the same as for the raw event histogram. The default scale-range is from 0 to $1/3$ x-range. It can be fixed manually using the panel shown on Fig. 12b instead, moreover, a wavelet spectrum title and scale-axis labels can be chosen (Fig. 10 shows defaults).
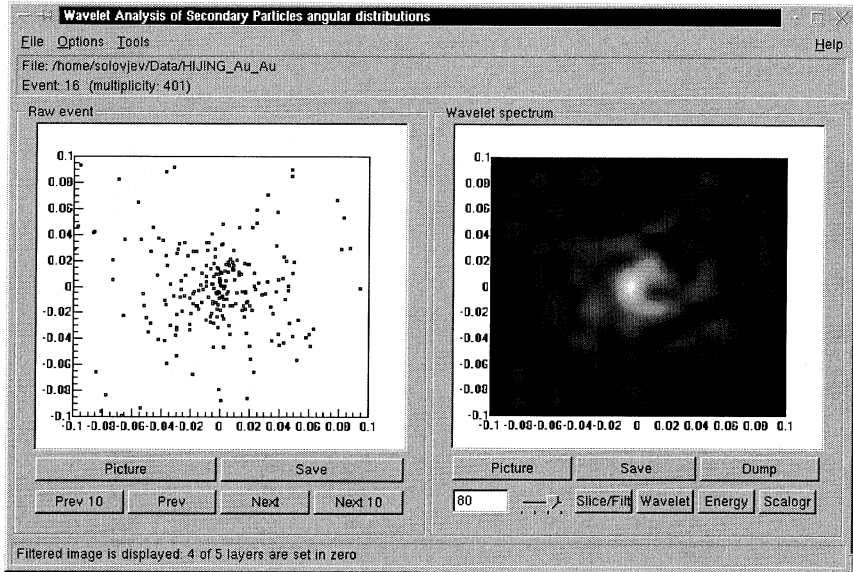

Fig. 13: Image Filtering. Five levels of decomposition are used here. Wavelet coefficients of the first four levels ("high frequencies") are nullified before reconstruction.

The **Slice/Filt** button has two functions. First, it allows one to view the wavelet transform of one-dimensional signal at a chosen scale parameter (the last one is set by a slider and controlled by a text entry). The corresponding view shown in [3] is omitted here. Second, it allows one to view the filtered image of a two-dimensional signal using discretized wavelet transform (see Fig. 13).

The **Energy** button allows one to draw wavelet energy spectrum: $|W_\psi(a,b)|^2$ (where a is a scale parameter, b is a number for one-dimensional signals and a two-dimensional vector for two-dimensional ones). The **Scalogram** button gives a scalogram: $\int |W_\psi(a,b)|^2 \, db$.

15

The **Wavelet** button allows one to draw wavelet spectrum with a better resolution or just redraw it (after **Slice/Filt** or **Energy** buttons are pressed, e.g.)
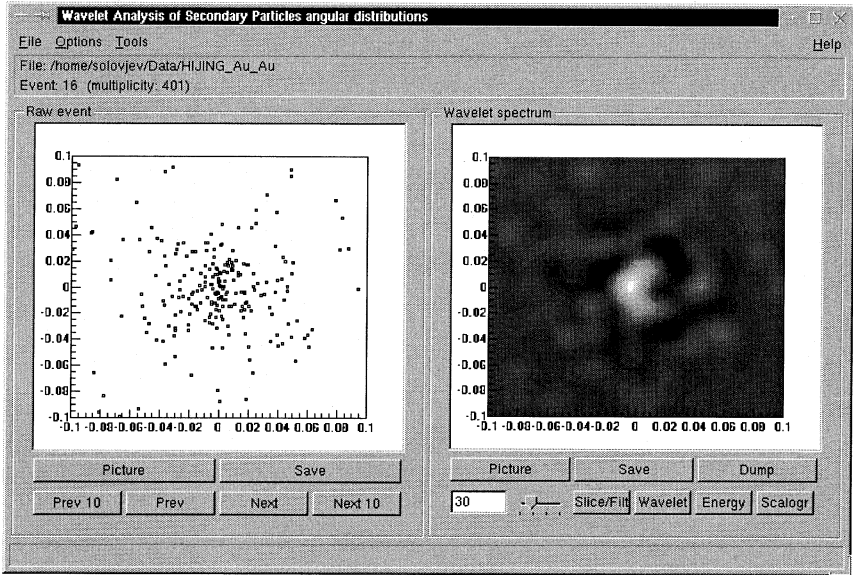

Fig. 14: Application of the 2D "Mexican Hat" wavelet. The view of the wavelet transform of a two-dimensional signal at a chosen scale parameter (the latter is set by slider).

The two-dimensional "Mexican Hat" function is used in WASP for the wavelet transformation of two-dimensional signals (see above). This transformation is performed after the **Wavelet** button is pushed (see Fig. 14).

Both **Save** buttons are used for saving corresponding images to EPS-files: file "en_r.eps" for raw event image and file "wsn_r.eps" for wavelet spectrum, where n is an event number in data file, r is an event representation number according to the panel on Fig. 12a.

The **Dump** button is used to save a wavelet spectrum matrix to the ASCII file (the x-axis grid x_i is written first, then the y-axis one y_i, and finally corresponding wavelet coefficients, line-by-line, from the minimum of y to the maximum). The file "wcmn_r.dat" is created when the **Dump** button is pressed (n and r have the same meaning as above).

Acknowledgments

References

[1] S.V. Afanasiev, M.V. Altaisky, and Yu.G. Zhestkov, "On applications of wavelet analysis to separation of secondary particles from nucleus-nucleus interactions", Nuovo Cimento A, 108:919 - 927, 1995.

[2] I.Dremin, et al., "Wavelet-patterns in nuclear-nucleus collisions at 158 GeV", http://ArXiv:hep-ph/0007060 v1 2000.

[3] A.Soloviev, "WASP (Wavelet Analysis of Secondary Particles angular distributions) package. Version 1.0. User's Guide". JINR Comm. E10-2001-105, Dubna, 2001.

[4] V.Uzhinskii, G.Ososkov, A.Polanski, A.Soloviev, A.Shitov, M.Chernyavski, "Wavelet analysis of secondary particles angular distributions in high energy nucleus-nucleus interactions. Analysis of pseudorapidity distributions". JINR Comm. P2-2001-119, Dubna, 2001.

[5] G.Ososkov, A.Shitov, "Gaussian wavelet features and their applications for analysis of discretized signals". CPC 126 (2000) 149–157.

[6] I.Daubechies, "Ten lectures on wavelets". SIAM, Philadelphia, 1992.

[7] Y. Meyer, "Wavelets: Algorithms and Applications". SIAM, Philadelphia, 1993.

[8] J.Morlet, et al., "Wave propagation and sampling theory". *Geophysics* **47**(1982)203-221.

[9] A.Soloviev, A.Sosnin, E.Zhidkov, "Improving the Accuracy of Estimation of Unknown Random Variable Probability Density over Empirical Data". JINR Comm., Dubna, 1999, P11-99-329.

[10] W.Sweldens, P.Schroder, "Building your own wavelets at home" Wavelets in Computer Graphics, pp. 15–87, ACM SIGGRAPH Course Notes, 1996.

[11] W.Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet construction". Proc. SPIE, 1995.

[12] R.Brun, N.Buncic, V.Fine, F.Rademakers, "ROOT User's Guide v0.6 (Draft, November 2000)". R.Brun, F.Rademakers, S.Panacek, D.Buskulik, J.Adamczewski, M.Hemberger. CERN, 2000.
"ROOT Classes Reference Manual", CERN, 1996.
http://root.cern.ch

WASP — программа на C++, предназначенная для анализа угловых распределений вторичных частиц, рождающихся в ядерных взаимодействиях. Программа основана на использовании вейвлет-преобразования. Данная работа представляет собой руководство пользователя, включающее описание используемых в программе математических методов и алгоритмов, графического интерфейса и класса задач ядерной физики, на который ориентирована программа.

Работа выполнена в Лаборатории информационных технологий и в Лаборатории ядерных реакций им. Г.Н.Флерова ОИЯИ.

WASP package is a C++ program aimed to analyze angular distributions of secondary particles generated in nuclear interactions. WASP package is based on wavelet transform algorithms. This work includes the user's guide, description of algorithms and mathematical methods, graphical user interface. We have also analyzed what problems of nuclear physics can be tackled with WASP.

The investigation has been performed at the Laboratory of Information Technologies and at the Flerov Laboratory of Nuclear Reactions, JINR.

Макет Т.Е.Попеко